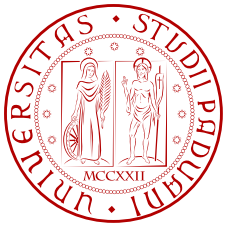

Università di Padova
Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione

Progettazione ed implementazione di una banca dati e suo accesso tramite interfaccia web

Studente
Filippo Campagnaro

Supervisore:
Chiar.^{mo} Prof. Michele Zorzi
Correlatore:
Dr. Beatrice Tomasi

Anno Accademico 2011/2012

Indice

Elenco delle figure

Elenco delle tabelle

Elenco dei codici

Abstract

The contribution of this thesis is twofold: first, we present the design and the implementation of an extensive database; then, we describe the web interface developed in order to make the access to the implemented database easy and user-friendly. In fact, the rationale behind this work, is:

- to organize large data sets of post-processing results;
- to make these results available and usable to scientific community.

Database and web interfaces make these two objectives possible. In particular the available data consist of environmental measures and estimates of underwater acoustic channel metrics, important for characterizing the channel conditions and the corresponding communication performance. Since numerous research groups all over the world are interested in accurately modeling these channel qualities, it is important to make available by the existing data sets. In order to do so, we design the web interface, such that these estimates could be accessed quickly, simply and intuitively. In particular, we organize the data in a relational database saved in a server, using MySQL RDBMS. Such database can be accessed by a dynamic website, developed using JavaScript and PHP languages. Moreover we used jQuery framework to improve the graphical interface. In order, to control the access to the data we also created a login page. Eventually we suggest possible modifications which improve the user interface and the speed of the data access.

Sommario

Il contributo di questa tesi può essere suddiviso in due parti: inizialmente si progetta e implementa un database; poi si descrive l'interfaccia web, che ha lo scopo di rendere l'accesso ai dati, contenuti nel database, facile ed intuitivo.

Infatti, gli obbiettivi del lavoro qui presentato sono:

- l'organizzazione di numerose elaborazioni ottenute da estesi insiemi di dati;
- l'estensione alla comunità scientifica dell'accesso a tali elaborazioni.

Questi due obbiettivi si possono ottenere usando banche dati e interfacce web. Tali strumenti permettono infatti di gestire e accedere alle informazioni desiderate tramite un'interfaccia grafica che consente di eseguire precise interrogazioni alla banca dati. In particolare, i dati resi disponibili sono sia misure ambientali, sia stime di metriche del canale acustico sottomarino, utili a caratterizzare la qualità del canale e le corrispondenti prestazioni di comunicazione.

Siccome molti gruppi di ricerca sono interessati a modellare accuratamente le qualità del canale in diverse condizioni ambientali, è importante e utile rendere disponibili i dati esistenti. A tale scopo, si è realizzato un database relazionale, utilizzando l'RDBMS gratuito MySQL. Inoltre, per accedere al database, è stato ideato un sito web dinamico utilizzando i linguaggi JavaScript e PHP, oltre al framework jQuery utile a migliorare l'interfaccia grafica. Allo scopo di controllare l'accesso ai dati, si è creata una pagina di login. Infine si propongono possibili modifiche per migliorare l'interfaccia utente e la velocità di accesso ai dati.

Acronimi

1NF	prima forma normale
2NF	seconda forma normale
3NF	terza forma normale
AJAX	Asynchronous JavaScript and XML
BER	Bit Error Rate
CIR	Channel Impulse Response
CSS	Cascading Style Sheets
csv	comma separated value
DBMS	DataBase Management System
DTD	Document Type Definition
E-R	Entità Relazioni
ECMA	European Computer Manufacturers Association
HTML	HiperText Markup Language
IBM	International Business Machines Corporation
IP	Internet Protocol
ISO	International Organization for Standardization
PHP	PHP: Hypertext Preprocessor
RDBMS	Relational DataBase Management System
SNR	Signal to Noise Ratio
SQL	Structured Query Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XHTML	eXtensible HTML
XML	eXtensible Markup Language

Capitolo 1

Introduzione

Questa tesi contiene una descrizione esaustiva delle diverse fasi di ideazione ed implementazione di una banca dati ed il suo accesso tramite interfaccia web. La creazione di un'unica banca dati ha lo scopo di migliorare l'usabilità dei dati disponibili, nonché di facilitare l'estrapolazione delle informazioni contenute negli insiemi di dati. Inoltre, l'implementazione di un'interfaccia web ha l'obiettivo di estendere l'accesso ai dati alle comunità scientifiche interessate a validare algoritmi e protocolli per le comunicazioni acustiche sottomarine. Infatti i dati in oggetto, sono misure di alcune metriche del canale acustico sottomarino, stimate a partire da registrazioni di segnali acustici trasmessi sott'acqua. Questo è dovuto al fatto che pochi riescono a condurre esperimenti in acqua con strumenti in grado di generare e registrare segnali acustici e allo stesso tempo misurare dati ambientali. Condividere le stime di metriche importanti per la comunicazione diventa determinante per lo sviluppo delle tecnologie per la comunicazione acustica sottomarina, in quanto i dati disponibili sono pochi.

Tali comunicazioni acustiche sottomarine sono impiegate per numerosi scopi, in diversi ambiti che vanno dal settore scientifico a quello militare. Ad esempio, per quanto riguarda le scienze, le reti sottomarine di dispositivi sensore possono essere usate per comprendere fenomeni oceanografici, che regolano i mutamenti climatici. Allo stesso modo, vengono usate per monitorare e studiare la vita e le comunicazioni acustiche di mammiferi marini, come balene e delfini. Un altro esempio di applicazione delle reti sottomarine, è in campo militare la sorveglianza delle zone costiere. Infatti, per motivi di sicurezza nazionale, è importante sapere in tempo reale se sottomarini di altre nazioni stanno navigando nelle acque territoriali. Analogamente, è importante poter monitorare in tempo reale un sito di estrazione del petrolio/gas in pozzi sottomarini, sempre più frequenti anche nel mar Mediterraneo. Proprio grazie al bisogno di abilitare tutte queste applicazioni, si è assistito negli ultimi vent'anni ad un crescente interesse da parte della comunità scientifica a sviluppare e migliorare le comunicazioni acustiche sottomarine. Tuttavia, le limitazioni dovute ad una scarsa larghezza di banda, lunghi tempi di propagazione e condizioni di canale altamente tempo e spazio-varianti, presentano numerose sfide a ingegneri e scienziati.

In particolare, il canale acustico sottomarino rappresenta uno degli aspetti chiave per lo sviluppo di tecniche di comunicazione efficaci ed è perciò oggetto di studi che hanno lo scopo di modellarne le principali caratteristiche. Di con-

seguenza diventa importante rendere accessibili le stime di metriche di canale in maniera veloce, semplice ed intuitiva.

Al fine di raccogliere e organizzare tali stime in una banca dati unica si è realizzato un database relazionale. Tale scelta è giustificata dalla sua appropriatezza per gli scopi prefissati, in quanto garantisce:

- indipendenza logica dalla struttura di memorizzazione fisica,
- minimizzazione della ridondanza,
- controllo su un'eventuale inconsistenza tra le informazioni inserite,
- un facile accesso tramite interrogazioni per la lettura, l'inserimento, l'eliminazione e la modifica dei dati scritte in linguaggio Structured Query Language (SQL).

Per testare il primo prototipo di database viene utilizzato il software Sqlite03, che permette in loco di creare piccole banche dati, senza dover installare un server. Dopo vari test sui prototipi, la banca dati viene implementata in MySQL, che permette di costruire database più efficienti con più funzioni e soprattutto in grado di memorizzare una più grande quantità di dati. Sqlite ha un limite massimo di circa 200 Megabyte . MySQL è un comodo Relational DataBase Management System (RDBMS) gratuito adatto a gestire ingenti quantità di dati e ben supportato da PHP: Hypertext Preprocessor (PHP), linguaggio client-server utile per rendere disponibile l'accesso alla banca dati tramite un' interfaccia web. Al fine di rendere semplice ed intuitivo l'accesso ai dati è stato realizzato un sito internet dinamico, conforme alle norme World Wide Web Consortium (W3C), in cui si è inserita sia una parte descrittiva dei data sets, sia una parte dedicata all'accesso guidato dei dati, ristretto solamente agli utenti registrati, muniti di login.

La tesi è strutturata in quattro capitoli. Nel capitolo 2, vengono descritte in dettaglio le fasi di progettazione e implementazione del database relazionale. Nel capitolo 3, viene illustrata l'ideazione del sito, ponendo particolare attenzione alla fase di registrazione, di login sicuro e all'interfaccia di accesso ai dati. Nel capitolo 4, si propongono infine dei miglioramenti al progetto.

Capitolo 2

Progettazione ed implementazione della banca dati unica

2.1 Breve descrizione dei dati

Nella prima parte di questa tesi viene illustrata la progettazione di una banca di dati contenente stime di metriche del canale acustico sottomarino e misure ambientali. In particolare ci riferiamo a dati raccolti durante tre esperimenti: SPACE08 svolto nel settembre 2008 presso l'isola Martha's Vineyard (MA, USA), SubNet09 avvenuto tra maggio e settembre 2009 all'isola di Pianosa (Italia) e KAM11 svolto all'isola KAUAI (Hawaii, USA) tra giugno e luglio 2011. Le informazioni sulle stime sono riportate in [2], mentre le caratteristiche delle trasmissioni in [4].

La struttura degli esperimenti è simile: segnali acustici vengono trasmessi tra due o più dispositivi subacquei posti ad una determinata distanza e profondità. Alcuni dispositivi (i trasmettitori) inviano pacchetti di dati ad una determinata potenza, frequenza e modulazione, mentre altri (i ricevitori) sono dotati di un array verticale di idrofoni e registrano i segnali ricevuti. Dai segnali registrati vengono stimate le prestazioni di comunicazione, ad esempio il Bit Error Rate (BER), e la corrispondente qualità del link come Signal to Noise Ratio (SNR) e Channel Impulse Response (CIR). BER e SNR sono numeri reali, mentre CIR è una matrice matlab contenente una serie temporale di risposte impulsive del canale.

Inoltre, vengono impiegati dei dispositivi che misurano le condizioni ambientali (temperatura e velocità del vento) in prossimità dei trasmittenti. Dato che tali parametri influenzano la propagazione acustica, risultano molto importanti per gli studi sul canale. Gli esperimenti differiscono tra loro per date del loro svolgimento, numero e posizionamento dei dispositivi utilizzati,[1] modulazione,

[1] Sei ricevitori e un trasmettitore in SPACE08, un ricevitore e tre trasmettitori in SubNet09, due ricevitori e un trasmettitore in KAM11.

banda e frequenza di trasmissione, differenti condizioni ambientali, e differente morfologia del territorio.[2]

Tramite interrogazioni (query) al database, l'utente finale può risalire alle seguenti informazioni per ogni misura acustica:

- l'esperimento di appartenenza,
- il dispositivo ricevente;
- l'istante di misurazione,
- la potenza di trasmissione,
- l'indice dell'idrofono da cui è stata ricevuta,
- la modulazione, la frequenza e la banda,
- la metrica stimata (SNR, BER o CIR),
- il valore stimato,
- il nome del file sorgente.

Analogamente, per i dati ambientali, si sono rese disponibili le seguenti informazioni:

- l'esperimento di appartenenza,
- il dispositivo che effettua la misura,
- l'istante di misurazione,
- il tipo di dato (temperatura o velocità del vento),
- il valore misurato,
- il nome del file sorgente.

Per salvare l'istante in cui è stato ricevuto ogni segnale acustico e l'istante in cui vene effettuata ogni misura ambientale, è stato scelto di utilizzare l'EPOCH.[3] E' essenziale salvare l'epoch sia delle misure acustiche che ambientali, così da rendere possibile l'analisi delle prestazioni di comunicazione in relazione alle condizioni esterne. Tale studio è molto utile per capire eventuali relazioni causa-effetto tra temperatura e/o velocità del vento e la qualità del canale corrispondente misurata. Inoltre, sono state rese accessibili altre informazioni descrittive, come ad esempio informazioni sul luogo di svolgimento e sulle date di inizio e fine esperimento. In questo modo, l'utente ha a disposizione una semplice e schematica descrizione temporale dei dati. Durante ciascun esperimento le trasmissioni acustiche avvengono con la stessa frequenza e banda di trasmissione, mentre lo schema di modulazione può cambiare.

I dispositivi che hanno fatto le misure vengono descritti dai seguenti parametri:

[2] Il vento viene misurato solamente durante SPACE08 e SubNet09, mentre la temperatura in SubNet09 e KAM11

[3] L'epoch, o Unix time, è il numero di secondi passati dal primo gennaio 1970. Ad esempio: 1331063441=(2012-03-06 19:50:41).

- sigla identificativa,
- latitudine della posizione,
- longitudine della posizione,
- profondità sul livello del mare,
- tipo di dispositivo.

Per tutta la durata degli esperimenti SPACE08 e SubNet09, ogni dispositivo è posizionato in un punto fisso, identificato con latitudine, longitudine e profondità sul livello del mare. Durante KAM11, invece, a metà esperimento vengono spostati gli idrofoni ad una profondità maggiore. Il posizionamento salvato nella banca dati è riferito alla prima metà dell'esperimento. Nel sito internet si trova l'indicazione della profondità a cui vengono spostati i dispositivi nella seconda fase. Le informazioni sui dispositivi sono ottenuti da [2] e [3]. La locazione esatta è un parametro fondamentale, in quanto è necessario, in fase di analisi, osservare le differenze delle prestazioni in funzione di profondità e distanza tra ricevitore e trasmettitore. Ogni ricevitore è formato da un array verticale di idrofoni tutti posti ad una differente profondità. Nella banca dati viene considerato dispositivo ogni singolo idrofono.

2.2 Il paradigma del database relazionale

Per rappresentare la realtà d'interesse,[4] descritta in 2.1, è stato progettato un database relazionale secondo il modello presentato da Edgar F. Codd[5] nel 1970 nell'articolo [5]. Codd per primo pensò di separare l'organizzazione logica dai metodi di memorizzazione fisica, rivoluzionando così il modo di migliorare le prestazioni: non più compito esclusivo dell'implementazione software ma anche attraverso l'organizzazione dei dati, introducendo concetti di indipendenza logica e fisica. Per organizzazione fisica dei dati si intende l'effettiva struttura dei file memorizzati, mentre per organizzazione logica si intende la modalità di gestione e organizzazione delle informazioni. Agli utenti finali comunque è permesso l'accesso solo a quest'ultima. Gli stessi programmatori che si occupano di implementare un database ignorano la parte fisica della banca dati, a carico invece del progettista del software database management system (DBMS), e si dedicano esclusivamente al modello logico. In questa tesi, si tratta della progettazione logica e del codice d'implementazione da far eseguire al DBMS, mentre l'aspetto di memorizzazione fisico nel calcolatore viene trascurato.

2.2.1 Modello Relazionale

Il modello relazionale prevede di rappresentare la realtà di informazione utilizzando semplici tabelle identificate da un nome, con un numero costante di colonne che definiscono le proprietà da salvare. Ogni riga rappresenta un elemento distinto della tabella. La realtà da informatizzare è rappresentata da un

[4] Con realtà di interesse si intende l'insieme logico dei dati che si vogliono informatizzare. Viene illustrata ad inizio progetto nella descrizione dei dati.

[5] Edgar F. Codd(1923-2003) era un informatico e matematico di IBM vincitore del premio Turing nel 1981

sottoinsieme del prodotto cartesiano degli elementi delle tabelle. Le definizioni utilizzate nel seguito sono tratte da [8] e [9].

Definizione 2.2.1 (Schema Relazionale)**.** *Uno schema relazionale, indicato con R(X), è costituito da:*

- *un simbolo, detto nome della relazione (o tabella) R,*
- *un insieme di attributi X={A1,A2,...,An} ad ognuno dei quali è associato un nome e un dominio.*

Definizione 2.2.2 (Schema di basi di dati)**.** *Uno schema di basi di dati R è un insieme di schemi di relazioni con nomi diversi, cioè R={R1(X1), R2(X2), ..., Rn(Xn)}.*

Definizione 2.2.3 (Istanza di relazione)**.** *Un'istanza di relazione su uno schema R(X) è un insieme r di tuple, cioè n-uple con proprietà commutativa.*

Definizione 2.2.4 (Istanza di basi di dati)**.** *Un'istanza di basi di dati su uno schema R={R1(X1), R2(X2), ..., Rn(Xn)} è un'insieme di relazioni r=(r1,r2,..,rn).*

Nel codice 2.1 è illustrata la sintassi utilizzata per descrivere le tabelle o relazioni.

Codice 2.1: Sintassi per creare tabelle secondo il modello relazionale.

```
Tab1={ attr1:dominio1, attr2:dominio2,attr3:dominio3}
```

dove gli attributi sottolineati indicano le chiavi (cioè le proprietà univoche) della relazione e alcuni attributi possono essere riferiti a relazioni esterne per permettere il collegamento tra tabelle.

Per quanto riguarda i domini la teoria relazionale non definisce esplicitamente quali tipi vengono supportati. La maggior parte dei DataBase Management System (DBMS) garantisce all'utente la possibilità di definire nuovi domini in aggiunta a quelli standard più utilizzati, quali numeri interi e reali, date, caratteri e stringhe.

2.2.2 SQL

Per l'accesso ai dati contenuti in un database relazionale e per l'implementazione delle tabelle tramite DBMS, si può utilizzare il linguaggio SQL [6] che sfrutta le regole dell'algebra relazionale per permettere associazioni tra tabelle. Infatti, la traduzione da schema relazionale a codice SQL è molto semplice: per creare una tabella si utilizza la sintassi 2.2.

Codice 2.2: Sintassi per creare tabelle tramite MySQL.

```
Create table Tab1 (attr1: dominio1 primary key ...,   attr2:
    dominio2 ...,
   <indici>
)
```

[6](SQL venne ideato da Donald Chamberlin e Raymond F. Boyce, informatici di International Business Machines Corporation (IBM), nel 1974.

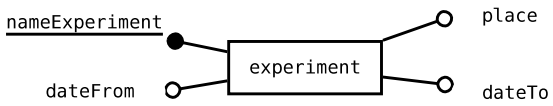

Figura 2.1: Modello E-R: rappresentazione di un'entità.

dove al posto dei tre punti possono essere inserite altre specifiche generiche degli attributi (ad esempio se l'attributo è opzionale o se è la chiave di collegamento con un'altra tabella), oppure essere specificate proprietà dell'attributo rispetto al suo dominio (ad esempio di un attributo stringa può essere specificata la lunghezza e di una data il formato). Infine al posto di "<indici>" vengono elencati gli attributi univoci che vengono memorizzati in un modo particolare (come le chiavi) per garantire un controllo: il controllo consiste nel verificare se un altro valore riferito allo stesso attributo è già presente nel database. Se si verifica tale condizione, il DBMS agisce secondo le specifiche inserite al momento della creazione dell'indice: ad esempio, può saltare l'inserimento, o eliminare l'istanza inserita in precedenza.

2.2.3 Diagramma Entità Relazioni

In fase di ideazione delle tabelle è utile servirsi del modello Entità Relazioni (E-R), ideato nel 1976 dall'informatico Peter Chen, per rappresentare in maniera più efficace il modello concettuale del database relazionale tramite un diagramma a blocchi semplice, intuitivo, ma completo.

Nel progetto illustrato in questa tesi il modello entità-relazione è stato utilizzato per schematizzare la realtà d'informazione prima di passare al modello relazionale. Le informazioni sulle regole del modello E-R sono ottenute da [9] e [11].

Definizione 2.2.5 (Entità)**.** *Un'entità rappresenta un concetto complesso e di rilievo che descrive una classe di oggetti con un'esistenza autonoma, cioè un insieme di soggetti ben definiti con caratteristiche in comune.*

Ogni entità, rappresentata in figura 2.1 da un rettangolo con dentro il suo nome, viene poi convertita in una tabella, dove gli attributi che identificano le proprietà dell'entità sono le colonne della tabella.

Definizione 2.2.6 (Istanza di un'entità)**.** *Una istanza di una entità è un oggetto della classe rappresentata, quindi una riga della tabella del modello relazionale.*

Definizione 2.2.7 (Relazione o Associazione)**.** *Una relazione o associazione, rappresentata in figura 2.2 da un rombo con all'interno un nome collegato a più entità, è un sottoinsieme del prodotto cartesiano di più entità. Può avere o meno attributi.*

A seconda della molteplicità e del numero di entità coinvolte, un'associazione viene convertita in due modi diversi:

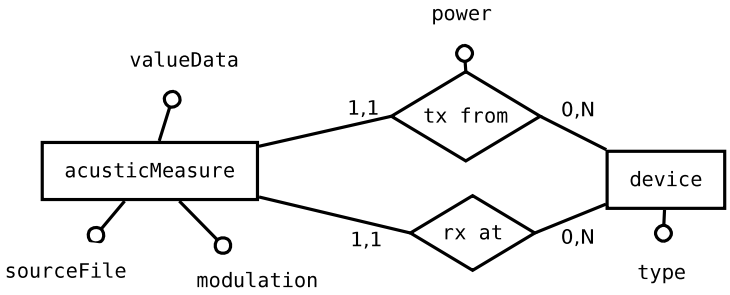

Figura 2.2: Modello E-R: rappresentazione di una relazione tra due entità.

- nel caso in cui le entità che partecipano alla relazione siano più di due oppure le loro istanze possano partecipare alla relazione con molteplicità n (presenti più volte nel prodotto cartesiano) si crea una tabella contenente gli attributi dell'associazione e le chiavi delle entità interessate;
- nel caso in cui solo due entità siano coinvolte e una della due abbia molteplicità uno (presente una sola volta nel prodotto cartesiano) allora nella tabella che rappresenta tale entità si inserisce la chiave dell'altra entità coinvolta.

Definizione 2.2.8 (Molteplicità)**.** *La molteplicità indica numero di volte che un'istanza di un'entità può partecipare ad un'associazione.*

La molteplicità è rappresentata nel diagramma E-R con una coppia di numeri tra parentesi "(a,b)" scritta sopra a tutti i segmenti che collegano le entità alle associazioni. "a" indica la molteplicità minima (in genere 0 o 1) cioè il numero minimo di volte che un'istanza dell'entità può partecipare al prodotto cartesiano, e "b" la molteplicità massima (generalmente 1 o n), cioè il numero massimo di volte che un'istanza dell'entità può partecipare alla relazione. In caso di relazioni tra solo due entità si parla di relazioni "(m,n)" dove "m" indica la molteplicità massima della prima entità e "n" la molteplicità massima della seconda entità.

Definizione 2.2.9 (Attributo)**.** *Un attributo rappresenta una proprietà di un'entità o una relazione ed è identificato da un nome, collegato direttamente all'entità o l'associazione a cui si riferisce tramite un segmento.*

Nella stessa entità e nella stessa relazione non possono esserci più attributi con lo stesso nome. Un attributo può essere obbligatorio, cioè ogni istanza ha sempre tale attributo inizializzato, oppure opzionale nel caso possa non avere alcun valore. Un attributo può essere :

1. semplice se ad ogni istanza può assumere un unico valore di un tipo base,
2. multiplo se ad ogni istanza può assumere più valori,
3. composto se è costituito da più attributi correlati,
4. calcolato se il valore che assume viene ottenuto da calcoli su altri attributi esistenti.

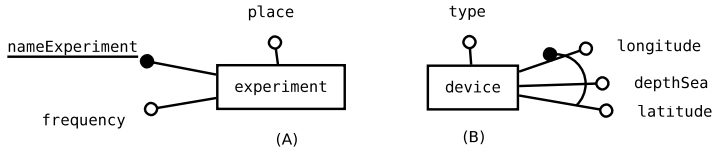

Figura 2.3: Modello E-R: rappresentazione delle chiavi: (A) entità con chiave semplice, (B) entità con chiave composta.

Definizione 2.2.10 (Chiave)**.** *Una chiave è un insieme di uno o più attributi univoci e minimali.*

Per univocità si intende che i valori degli attributi identificano univocamente le istanze dell'entità, ossia, non possono esistere due istanze diverse della stessa entità con lo stesso valore per la chiave. Ogni insieme di attributi che verifica questa proprietà è detto superchiave. Per minimalità si intende che rimuovendo un qualsiasi attributo dall'insieme si perde il requisito di univocità, quindi una chiave è una superchiave minimale. E' possibile che esistano più chiavi per un'entità; tali chiavi sono dette chiavi candidate. Tra queste occorre sceglierne una, detta chiave primaria, secondo dei criteri di decisione: la chiave primaria può essere composta solo da attributi obbligatori e deve essere più piccola e semplice possibile, per garantire efficienza in interrogazione.

Per questi motivi spesso nel passaggio in modello relazionale viene utilizzata la tecnica dei codici identificativi: si inserisce una chiave numerica intera auto incrementale che costituisce l'indice univoco e minimale più utilizzato. I codici identificativi vengono aggiunti già a livello di modello concettuale in tutti i casi in cui non si riesca a trovare una chiave che soddisfi i requisiti, oppure se un'entità non ha alcun insieme di attributi univoci. Utilizzando questi indici, solitamente di tipo intero, al costo di un ridotto incremento dello spazio utilizzato in memoria, si ottengono interrogazioni molto veloci, in quanto, per un calcolatore, il confronto tra numeri interi è un'operazione più veloce di un confronto tra stringhe. Un'entità senza alcun sottoinsieme di attributi univoco è un'entità debole e nel diagramma viene rappresentata come un'entità riquadrata. Una chiave viene rappresentata come un attributo; nel caso di chiave singola il nome è sottolineato (figura 2.3(A)), nel caso di chiave composta l'insieme degli attributi che la compongono viene collegato da un arco con un cerchietto nero disegnato ad un'estremità (figura 2.3(B)).

Definizione 2.2.11 (Specializzazione)**.** *Per gerarchia ISA*[7] *o specializzazione, si intende il caso in cui un'entità viene specializzata in una o più sotto-entità, che ereditano tutte le caratteristiche dell'entità genitore.*

La specializzazione può essere totale (tutti gli elementi dell'entità genitore sono specializzati in una sotto-entità) o parziale (alcune istanze dell'entità genitore possono non essere specializzate). La specializzazione totale è rappresenta in figura 2.4 da una freccia nera con la punta diretta verso l'entità genitore e la coda ramificata verso le sotto-entità, mentre la specializzazione parziale viene rappresentata da una freccia bianca. La traduzione di una gerarchia in modello relazionale può avvenire nei seguenti modi:

[7] Gerarchia ISA: "is a" dall'inglese significa "è un".

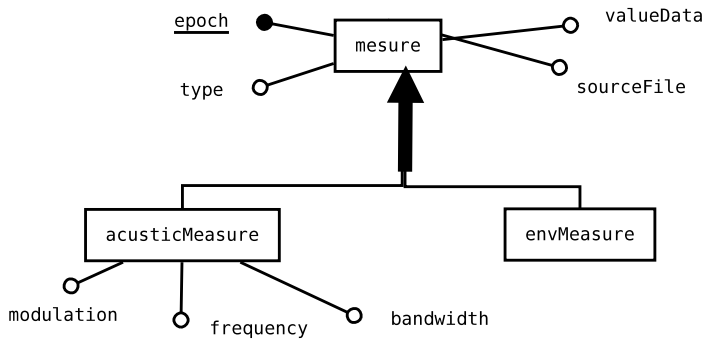

Figura 2.4: Modello E-R: rappresentazione della gerarchia ISA.

- inserendo un campo apposito detto flag nella tabella genitore (per indicare la specializzazione) e tutte le proprietà delle sottoentità come campi opzionali (inizializzate a seconda della specializzazione),
- traducendo direttamente le specializzazioni in tabelle distinte.

La scelta generalmente dipende dal numero di attributi delle specializzazioni rispetto il numero di attributi dell'entità genitore. Infine se le specializzazioni non hanno attributi, si inserisce nella tabella genitore solo il campo flag per indicare un'istanza a quale specializzazione appartiene.

Dopo aver steso il primo modello concettuale si passa alla sua normalizzazione secondo le norme BCNF ideate da Edgard F. Codd e Rymond F. Boyce nel 1974, per eliminare eventuali ridondanze e rischi di inconsistenza del database, cause di anomalie in inserimento, cancellazione e modifica. Se una relazione presenta più concetti tra loro indipendenti, allora viene decomposta in relazioni più piccole, una per ogni concetto. Ad esempio, è buona norma ottenere nel modello concettuale solo associazioni binarie (cioè composte da due entità), scomporre tutti gli attributi non semplici creando altre entità e avere chiavi primarie più semplici possibile. Tale processo è possibile solo se non comporta perdita di informazioni. Il processo di normalizzazione della base dei dati prevede 3 passaggi.

1. prima forma normale (1NF),
2. seconda forma normale (2NF),
3. terza forma normale (3NF).

La 1NF prevede di scomporre tutti gli attributi composti e multivalore in attributi semplici e assicurare una chiave primaria ad ogni tabella. Una base di dati è in 2NF se, oltre ad essere in 1NF, per ogni chiave primaria composta, si stabilisce che tutti gli altri attributi dipendono dall'intera chiave primaria. Una base di dati è in 3NF se è in 2NF e tutti gli attributi non-chiave dipendono dalla chiave soltanto, ossia non esistono attributi che dipendono da altri attributi non-chiave. Tale normalizzazione elimina la dipendenza transitiva degli attributi dalla chiave.

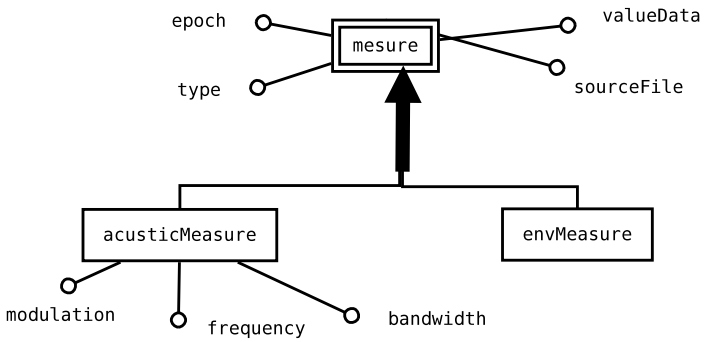

Figura 2.5: Modello E-R: gerarchia delle misure.

Tutti i database possono essere portate alla terza forma normale, anche se a volte il processo è difficile e può portare a rallentamenti in interrogazione per la presenza di troppe tabelle e relazioni. Lo scopo nella progettazione di un database relazionale è cercare di arrivare alla terza forma normale, garanzia di qualità del database. Le forme successive di normalizzazione non vengono quasi mai utilizzate, perché, oltre a non essere sempre applicabili, portano a grossi rallentamenti in fase di interrogazione e modifica del database. La normalizzazione può avvenire direttamente a livello relazionale oppure a livello di schema E-R. Nel progetto illustrato in questa tesi il processo di normalizzazione viene tentato durante il modello concettuale.

2.3 Progettazione della banca dati

2.3.1 Modello concettuale

Dall'analisi della realtà d'interesse descritta nel paragrafo 2.1 e seguendo il paradigma per ottenere il modello logico del database descritto nel paragrafo 2.2 si è passati alla costruzione del modello E-R.

Analizzando le misure, viene naturale costruire la gerarchia in figura 2.5 con un'entità genitore chiamata misura, con gli attributi in comune tra le misure acustiche e ambientali (cioè epoch, tipo, nome del file sorgente e valore della misura), e una specializzazione totale che porta alle sotto-entità misura ambientale e acustica, che in più ha gli attributi di modulazione, frequenza e banda.

Tuttavia, in questo schema ci sono degli errori. Innanzitutto, il valore assume tipologie differenti a seconda che la misura sia acustica o ambientale. Nelle misure ambientali si ottiene come valore sempre un numero decimale, invece nelle acustiche il valore ottenuto è una stringa, poiché, mentre SNR e BER sono numeri, CIR è una stringa che si riferisce al file contenente la risposta impulsiva del canale. Dato che non c'è una chiave univoca, la misura è un'entità debole, cosa da evitare per la normalizzazione. Durante un esperimento banda e frequenza non variano, quindi i due attributi vanno eliminati da misura acustica ed assegnati all'entità esperimento, in modo da occupare meno spazio in quanto non sarà necessario salvarle ad ogni inserimento di una misura acustica. Le mi-

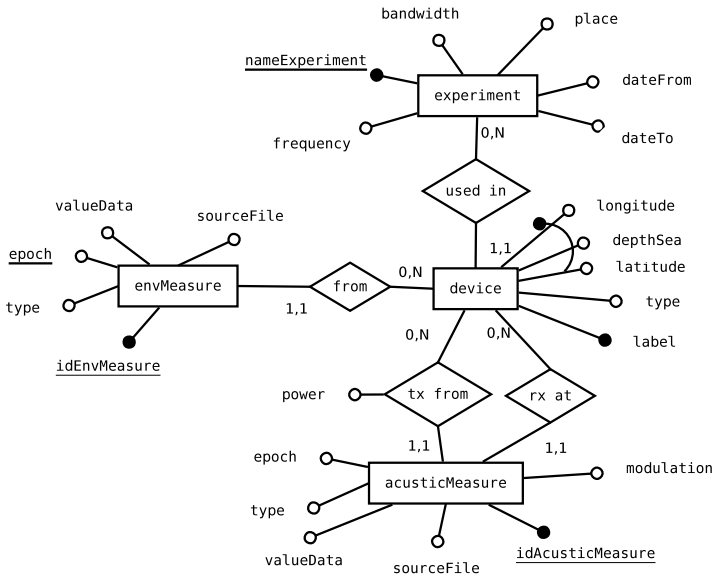

Figura 2.6: Modello concettuale E-R del database.

sure acustiche ed ambientali sono da considerare come due entità diverse date le loro intrinseche differenze. Infatti la traduzione della gerarchia ISA descritta in precedenza avrebbe comunque generato due tabelle distinte. Non avendo chiavi univoche, è opportuno aggiungere un codice identificativo a ciascuna istanza in modo da evitare la presenza di entità deboli.

Viene ora analizzato il problema dei dispositivi. Anche in questo caso si potrebbe pensare ad una gerarchia ISA in quanto esistono quattro diverse tipologie di dispositivi: idrofoni, trasmettitori, termistori (termometri) e anemometri. Come attributi in comune tutti i dispositivi hanno un posizionamento preciso (coordinate e profondità marina), vengono utilizzati in un determinato esperimento e hanno una sigla identificativa. L'unica caratteristica che distingue i trasmettitori dagli altri dispositivi è che gli viene associata una certa potenza. In questo modo la gerarchia è evitabile aggiungendo semplicemente l'attributo flag "tipo" all'entità dispositivo e l'attributo potenza alla relazione che lega ogni misura acustica al dispositivo che ha inviato il pacchetto.

Ogni misura è legata ad un esperimento dai dispositivi in esso utilizzati. A tale scopo è sufficiente creare una relazione (1,n) tra le entità "device" ed "experiment". In sintesi nella figura 2.6 è disegnato il modello concettuale E-R del database.

2.3.2 Dizionario dei dati

Ultimo passaggio prima della traduzione da modello concettuale a modello relazionale è costruire, a partire dal diagramma E-R, il dizionario dei dati, dove vengono riordinate le idee rappresentate nel modello concettuale e dove per la prima volta viene analizzato il dominio (cioè il tipo) degli attributi. Questo

processo consiste nel disegnare la tabella delle entità, la tabella delle relazioni e scrivere l'elenco dei vincoli.

Nella tabella delle entità sono presenti quattro colonne:

- nome dell'entità,
- breve descrizione,
- elenco degli attributi con il rispettivo dominio,
- elenco degli attributi che formano la chiave primaria dell'entità.

Anche nella tabella delle relazioni sono presenti quattro colonne:

- nome dell'associazione,
- breve descrizione (molteplicità, ecc),
- elenco delle entità coinvolte nell'associazione,
- elenco degli attributi con rispettivo dominio.

L'elenco dei vincoli invece serve a schematizzare tutte le caratteristiche del modello concettuale che non si riescono a rappresentare nelle due tabelle, come ad esempio molteplicità e opzionalità degli attributi, totalità di un'eventuale gerarchia.
Le tabelle delle entità e delle relazioni sono rappresentate nelle 2.1 e 2.2.

Vincoli:

- Tutti gli attributi delle entità e delle relazioni sono campi che vanno inizializzati obbligatoriamente.
- Tutte le relazioni sono 1,N ovvero un dispositivo viene usato solo in un esperimento mentre in un esperimento sono usati più dispositivi.
- L'epoch di una misura, una volta convertito in formato data-ora, deve essere compreso tra dateFrom e dateTo dell'esperimento.

2.4 Implementazione della banca dati

Una volta terminato il dizionario dei dati si costruiscono conseguentemente le tabelle del modello relazionale (listato nel codice A.1) seguendo i criteri descritti nel paragrafo 2.2. In particolare viene utilizzata la tecnica degli indici o codici identificativi.

Nel caso specifico, per passare dal modello concettuale illustrato nel paragrafo 2.3 al modello relazionale, è sufficiente tradurre le entità in tabelle, aggiungendo le chiavi esterne per permettere le associazioni. Infatti, avendo solamente relazioni (1,N) non occorre creare ulteriori tabelle per rappresentare le associazioni. Per permettere collegamenti veloci tra una tabella e l'altra, si utilizzano gli indici identificativi come chiavi primarie, aggiungendo nello specifico un campo denominato "idNomeTabella" di tipo intero per ogni tabella.

Occorre prestare attenzione alla relazione "tx from", che lega le misure acustiche ai dispositivi trasmettenti, perché è l'unica dotata di un attributo, in questo caso rappresentato dalla proprietà "power". Utilizzando la regola spiegata nel

Tabella 2.1: Tabella delle Entità.

NOME	DESCRIZIONE	ATTRIBUTI	CHIAVE
experiment	esperimento durante il quale sono state effettuale le misure	nameExperiment: String place: String dateFrom: Date dateTo: Date bandwidth: real frequency: real	nameExperiment
device	dispositivo utilizzato per trasmettere, ricevere o misurare dati	longitude: real latitude: real depthSea: real label: String type: String	(longitude, latitude, depthSea) label
envMeasure	misura ambientale effettuata da un dispositivo durante un esperimento	idEnvMeasure: integer epoch: integer valueData: real type: String sourceFile: String	idEnvMeasure
acMeasure	stima di una metrica del canale acustico sottomarino derivata della trasmissione da un trasmettitore ad un idrofono	idAcMeasure: integer epoch: integer valueData: real type: String sourceFile: String modulation: String	idAcMeasure

Tabella 2.2: Tabella delle Relazioni.

NOME	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
used in	dispositivi utilizzati in un esperimento	experiment (0,N) device (1,1)	
from	da quale dispositivo ottenuta una misura ambientale	EnvMeasure (1,1) device (0,N)	
tx from	da quale dispositivo trasmessi dati di misura acustica	device (0,N) acMeasure (1,1)	power: real
rx at	da quale idrofono ricevuti dati di misura acustica	device (0,N) acMeasure (1,1)	

paragrafo 2.2, nella conversione viene aggiunto il campo "power" come attributo opzionale nella tabella "device"; tale proprietà sarà inizializzata solamente nei dispositivi di tipo trasmettitore.[8]

Una volta scritto il modello relazionale il passo successivo è tradurre tale schema nel linguaggio SQL del DBMS utilizzato. In fase di sperimentazione sono stati costruiti vari prototipi scritti in Sqlite03, utilizzando un sottoinsieme di dati. Questo ha permesso di implementare il data base senza dover installare un server, necessario per utilizzare MySQL. E' proprio durante il test di questi prototipi che sono state notate le modifiche allo schema concettuale illustrate nella sezione 2.3, come ad esempio, spostare gli attributi frequenza e banda da misura ambientale all'entità esperimento per diminuire lo spazio occupato, la convenienza di considerare le misure come entità distinte e i dispositivi come un'unica tabella col campo "power" opzionale. Una volta trovato il modello definitivo il progetto è passato alla fase successiva, cioè all'implementazione del database relazionate tramite MySQL. A tale scopo è stato utilizzato un calcolatore con installato il server Apache, necessario ad utilizzare l'RDBMS con tutte le sue funzionalità, e al caricamento dei dati da informatizzare.

Nel progetto illustrato viene utilizzato il DBMS gratuito MySQL, che possiede un linguaggio SQL molto ricco, è ben integrabile in PHP per l'accesso da web e permette di assegnare agli attributi molte proprietà rispetto al loro dominio. Le informazioni su MySQL sono ottenute da [6], [12] e [10].

Nel listato A.2 viene esplicitata la traduzione da modello relazionale a codice nel linguaggio SQL specifico. Eseguito il codice SQL tramite il DBMS, viene creato un database vuoto, strutturato secondo lo schema relazionale analizzato. Vengono create le tabelle "experiment", "device", "envMeasure" e "acousticMeasure" con tante colonne quanti gli attributi specificati.

Nella fase seguente tali tabelle devono essere riempite con le informazioni relative agli esperimenti, ai dispositivi e alle misure che si vogliono rendere disponibili. Per inserire dati in un database MySQL ci sono tre modi:

1. inserire le informazioni manualmente riga per riga,
2. estrarre, tramite pagina PHP, i dati da file e caricarli nella banca dati,
3. utilizzare le funzioni di caricamento automatico da file rese disponibili dal DBMS.

Nel progetto in esame viene utilizzato l'ultimo metodo, che garantisce un controllo più sicuro nell'inserimento dei dati. I file contenenti le informazioni da caricare sono strutturati nel seguente modo: in ogni riga del file è presente un'unica istanza della tabella e ogni attributo è separato da una virgola. I file sono in formato comma separated value (csv) facilmente modificabili sia tramite editor di testo che tramite gestore di fogli elettronici. Al fine di caricare automaticamente i dati contenuti in tali file nel database si utilizza il codice SQL 2.3.

Codice 2.3: Sintassi per caricare i dati nel DBMS.

```
LOAD DATA LOCAL INFILE 'nomeFile.csv'
INTO TABLE nometabella
FIELDS TERMINATED BY 'carattereSeparatore' ;
```

[8] Il campo "power" sarà inizializzato in tutti e soli i "device" con l'attributo flag "type"="transmitter".

Che carica dal file “nomeFile.csv” nella tabella nometabella i campi separati da il “carattereSeparatore”. Nello specifico per caricare i dati nel database è stato eseguito il codice 2.4 nel DBMS:

Codice 2.4: Caricamento dei dati nel DBMS.

```
LOAD DATA LOCAL INFILE 'experiment.csv'
INTO TABLE experiment
FIELDS TERMINATED BY ',' ;
LOAD DATA LOCAL INFILE 'acoustic.csv'
INTO TABLE acousticMeasure
FIELDS TERMINATED BY ','
LOAD DATA LOCAL INFILE 'device.csv'
INTO TABLE device
FIELDS TERMINATED BY ';'
LOAD DATA LOCAL INFILE 'env.csv'
INTO TABLE envMeasure
FIELDS TERMINATED BY ';'
```

Dopo il caricamento dei dati nel database, la progettazione e l’implementazione della banca dati unica sono completate. Il passo successivo è realizzare un’interfaccia grafica web, descritta nel capitolo 3.

Capitolo 3

Interfaccia web

3.1 Motivazioni e struttura del sito web

Un sito web o sito Internet è un insieme di pagine web correlate, ovvero una struttura ipertestuale di documenti che risiede, tramite hosting, su un web server. È accessibile, all'utente che ne fa richiesta, tramite un web browser sul World Wide Web della rete Internet, digitando in esso il rispettivo Uniform Resource Locator (URL) o l'indirizzo Internet Protocol (IP). Per rendere i dati accessibili dalla rete internet, è stato progettato ed implementato un sito web contenente sia pagine statiche, per la descrizione dei dati, sia pagine dinamiche, per la registrazione, il login e l'interrogazione del database. Le pagine di descrizione, scritte in linguaggio HiperText Markup Language (HTML), contengono i collegamenti ai siti degli istituti che hanno collaborato durante gli esperimenti, la descrizione dei singoli esperimenti e una guida per l'utilizzo ottimale del sito. Nel paragrafo 3.2 viene illustrato in dettaglio il metodo di creazione di una pagina internet statica o semi-dinamica lato client, secondo le normative W3C, definite dal consorzio che regola gli standard di sviluppo per il web.

Inoltre, questo permette di controllare che l'accesso ai dati sia effettuato solamente dai membri della comunità scientifica, appartenenti ad un istituto di ricerca o Università.[1] Per questa fase, viene utilizzato il linguaggio di scripting lato server PHP, che, invece di venire interpretato dal browser come tutti i linguaggi lato client, viene eseguito direttamente nel server.

È stato ritenuto opportuno raccogliere informazioni sugli utenti registrati. Inoltre, per questo motivo è stato creato un semplice database relazionale da interrogare e aggiornare ad ogni registrazione e login. Nel paragrafo 3.3 viene illustrato il processo di registrazione e login, nonché l'implementazione del database per gestire gli utenti. Una volta avvenuta la registrazione e il login, ad ogni utente è permesso l'accesso guidato alla banca dati illustrata nel capitolo 2. Al fine di rendere disponibili online solamente alcune informazioni presenti nel database, è stata creata un'interfaccia grafica semplice ed intuitiva, implementata in linguaggio PHP, che regola l'accesso ai dati. Infatti, l'utente viene guidato attraverso molteplici tipologie di query. In particolare, si distingue l'accesso alle stime di metriche del canale acustico sottomarino dall'accesso

[1] È stata creata una pagina per il login, l'accesso ai dati controllato e una pagina per richiedere la registrazione.

alle misure ambientali. Una volta che le informazioni sono state richieste, queste vengono rielaborate e visualizzate in grafici, tabelle oppure vengono creati dei file testuali in formato csv. Questi output possono essere poi scaricabili in locale, garantendo una presentazione e un'impaginazione chiara ed efficace, facilmente sfruttabile per l'analisi dei dati. A tale scopo vengono utilizzati alcuni strumenti forniti da jQuery. Quest'ultima è una libreria di funzioni JavaScript completa e utile per i nostri scopi. JavaScript è un linguaggio di scripting lato client.

3.2 Implementazione di pagine web statiche descrittive

Sono state create delle semplici ma complete pagine web statiche per presentare gli esperimenti SPACE08, SubNet09 e KAM11, e una semplice guida al sito per illustrare il metodo di accesso ai dati.

Definizioni e sintassi su pagine statiche, HTML e CSS utilizzate in questo paragrafo sono ricavate da [13] e [17].

Definizione 3.2.1 (Pagina Web Statica)**.** *Una pagina web si dice statica se tutti i suoi contenuti sono inseriti e gestiti esclusivamente da codice HTML, senza l'uso di linguaggi di scripting.*

Dopo l'elaborazione da parte del web browser del codice HTML, le pagine vengono visualizzate nel modo univoco deciso dallo sviluppatore.

3.2.1 HTML

L'HTML è un linguaggio di pubblico dominio, sviluppato verso la fine degli anni ottanta da Tim Berners-Lee al CERN di Ginevra, la cui sintassi è regolata dal W3C. Non è un linguaggio di programmazione ma solamente un linguaggio di markup, ovvero descrive le modalità di impaginazione, presentazione e visualizzazione grafica (layout) del contenuto di una pagina web attraverso delle etichette, chiamate tag di formattazione.

Ogni sessione aperta da un tag ha la struttura 3.1:

Codice 3.1: Struttura di un tag HTML

```
<tagA ...> testo </tagA>
```

dove con "<tagA>" viene aperto il tag, di seguito possono essere elencate le specifiche e gli attributi di formattazione del tag e al posto di "testo" viene inserito il testo che sarà visualizzato all'interno della sessione.

Il contenuto fornito dai siti web, in seguito a una richiesta dell'utente, consiste in un documento HTML e nei file ad esso correlati. Un web browser scarica da uno o più web server questi documenti, li elabora, interpretando il codice, e visualizza la pagina richiesta sullo schermo del computer.

Nella prima riga di codice viene specificato la Document Type Definition (DTD) cioè il tipo di documento e la versione HTML utilizzata, per permettere al browser di identificare le regole di interpretazione e visualizzazione appropriate per il documento. Questa definizione deve pertanto precedere tutti i tag

relativi al documento stesso. Il codice 3.2 indica la DTD utilizzata, nello specifico eXtensible HTML (XHTML) 1.1, ed è presente all'inizio di ogni pagina HTML del sito descritto in questa tesi.

Codice 3.2: Prima riga di codice di tutte le pagine web: DTD.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://
    www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Dopo aver specificato il tipo di documento inizia la stesura della pagina HTML, che assume una struttura ad albero annidato con sezioni aperte da tag, contenenti a loro volta altre sotto-sezioni aperte da altre etichette.

Il primo tag che viene aperto dopo la DTD è "<html>" all'interno del quale va inserito il resto del codice del documento. Alla fine del documento il tag di apertura viene chiuso con "</html>". Il codice da inserire all'interno di "<html> </html>" è diviso in due blocchi, chiamati header e body.

Il primo, compreso tra i tag "<head> </head>", contiene l'intestazione del documento, che comprende i seguenti tipi di sessione:

- il titolo della pagina, inserito tra le etichette "<title> </title>", è un elemento obbligatorio in ogni documento HTML e viene visualizzato nella barra del titolo del browser;
- i metadata, inseriti dentro i tag "<meta> </meta>", sono indispensabili per assegnare le norme International Organization for Standardization (ISO) rispettate dal documento;[2]
- i collegamenti, inseriti dentro l'etichetta "<link..>", servono a specificare i file esterni da cui il documento trae informazioni. Di norma vi è sempre un collegamento ad un foglio di stile in formato Cascading Style Sheets (CSS), utilizzato per definire le caratteristiche grafiche del documento. A tale scopo viene utilizzato il codice 3.3:

  Codice 3.3: Intestazione per inserire una pagina CSS.

  ```
  <link rel="stylesheet" type="text/css" href="url">;
  ```

- gli script utilizzati nel caso di pagine dinamiche (per generare ed analizzare dinamicamente codice html), inseriti tra le etichette "<script> </script>", oppure in file separati collegati alla pagina tramite il codice "<script src=... type=...>".

Terminata l'intestazione inizia il secondo blocco, cioè il corpo della pagina, compreso tra i tag "<body> </body>". All'interno del body viene inserita la parte testuale del sito, le immagini e i collegamenti che costituiscono la parte visualizzata dal browser. È buona norma inserire tali informazioni dentro tag ramificati, ai quali, nell'elenco degli attributi, vengono specificati una stringa identificativa e una classe di appartenenza. Si noti che nel documento HTML non viene scritta alcuna informazione sulle proprietà grafiche delle sezioni, nonostante ve ne sia la possibilità. Tali caratteristiche vengono specificate sul foglio di stile, dove vengono assegnate proprietà alle sessioni utilizzando le tipologie di tag, gli identificativi e le classi assegnate ai blocchi di codice. Segue una breve descrizione dei blocchi più utilizzati nella parte statica del sito. Il tag " "

[2]Nel sito in oggetto viene utilizzata la normativa iso-8859-1

definisce una divisione o una sessione in un documento HTML e solitamente è utilizzato per raggruppare blocchi di elementi ai quali applicare le stesse proprietà. Il tag "<p>" definisce un paragrafo del documento, mentre il tag "" viene utilizzato per raggruppare elementi presenti nella stessa riga, solitamente non per modificare opzioni grafiche ma per permettere di accedere a tale blocco tramite linguaggi di scripting. Il tag "
" definisce una lista non ordinata (cioè una lista puntata) , mentre "
" definisce una lista numerata; ogni elemento di una lista viene inserito all'interno del tag "
..
". Il tag "<table>" serve per definire una tabella; ogni riga della tabella viene inserita tra le etichette "<tr>..</tr>" mentre ogni colonna tra "<td>..</td>". Per inserire collegamenti ipertestuali all'interno del documento viene utilizzato il tag "nome", mentre per inserire un'immagine si utilizza il tag "". Altri blocchi quali form, pulsanti, select e aree di testo sono illustrati nei paragrafi successivi, in quanto propri delle pagine web dinamiche. Una volta creati i blocchi di codice HTML contenenti le informazioni da visualizzare nella pagina, si procede con l'impaginazione tramite foglio di stile CSS.

3.2.2 CSS

Il CSS è un linguaggio informatico usato per definire le proprietà grafiche dei documenti, secondo le direttive emanate dal W3C dal 1996. L'introduzione del CSS si è resa necessaria per separare i contenuti dall'impaginazione e permettere una programmazione più chiara e facile da usare, per gli autori delle pagine HTML e per gli utenti.

I principali motivi di tale separazione sono i seguenti:

- maggior ordine del codice HTML, che non presenta lunghi elenchi di proprietà grafiche,
- possibilità di assegnare caratteristiche grafiche ad un intero insieme di tag o ad una classe di blocchi, permettendo modifiche delle proprietà di molte sezioni in un unico passaggio.

Le proprietà grafiche più usuali che si possono specificare all'interno di un foglio di stile sono: dimensione e posizione del blocco e dei margini, tipologia, colore, sfondo, dimensione e allineamento del carattere di scrittura all'interno del blocco. La sintassi utilizzata a tale scopo è listata nel codice A.7.

Come "nomeBlocco" può essere inserito nome di un tag (ad esempio "body" o "div"), ".nomeClasse" per assegnare le proprietà ad un'intera classe di sezioni, oppure "#nomeId" per accedere un blocco a cui è stato assegnato un identificativo. Dimensioni e spessori possono essere inseriti in termini di pixel, percentuale o centimetri. Le pagine del sito sono state impostate tramite codice HTML e foglio di stile CSS.

In particolare, ogni pagina è suddivisa in in quattro blocchi fondamentali.

1. primo blocco in alto al centro contenente il titolo del sito, scritto in stampatello,
2. secondo blocco, contenente un menù per la navigazione,
3. terzo blocco, posto al centro della pagina, contiene le vere informazioni del documento,

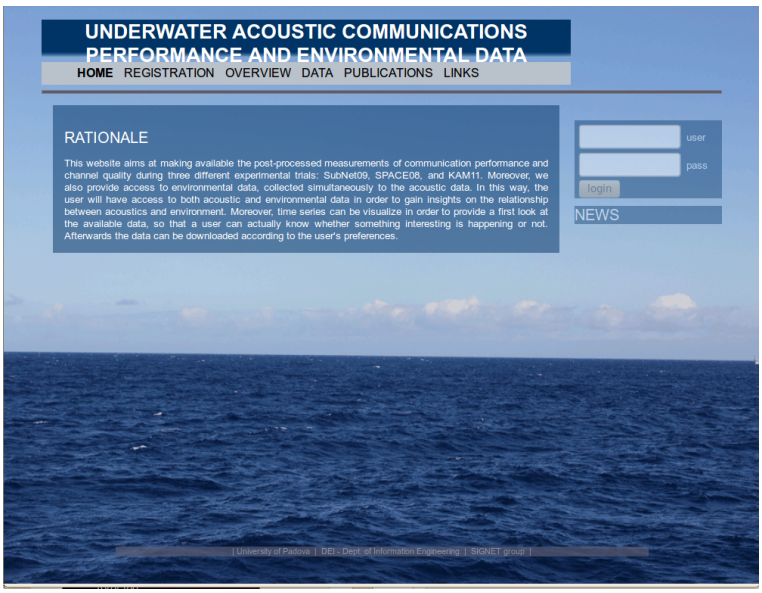

Figura 3.1: La Home del sito è una pagina web descrittiva.

4. quarto blocco a piè pagina è una barra contenente i collegamenti ai siti di vari gruppi di ricerca.

Finita la stesura dei codici HTML e CSS le pagine web statiche sono completate. Inoltre, in ogni pagina è disponibile una quinta sezione contenente un riquadro per il login e un riquadro per le news gestiti da codice JavaScript, e di conseguenza tutte le pagine del sito hanno una componente dinamica. Tale riquadro, tuttavia, è l'unica parte dinamica delle pagine descrittive, come si può vedere in figura 3.1.

3.3 Registrazione e login: pagine web dinamiche e basi di dati

Per permettere l'interazione tra utente e pagina web è necessario creare pagine dinamiche, il cui codice HTML viene in parte prodotto da rielaborazioni del server o da funzioni scritte in linguaggio script interpretate dal browser. Tali pagine sono strutturate similmente ai documenti HTML statici descritti nel paragrafo precedente, ad eccezione che l'intestazione contiene collegamenti a funzioni per gestire dinamicamente gli elementi del corpo della pagina. Inoltre, al fine di rendere l'accesso alla banca dati sicuro e controllato, è sorta la necessità di creare un secondo database per permettere la registrazione e il login agli utenti.

3.3.1 Database utente

Il semplice database per gestire gli utenti è stato realizzato in MySql, informatizzando la seguente realtà d'interesse. Ogni utente a cui è stato autorizzato

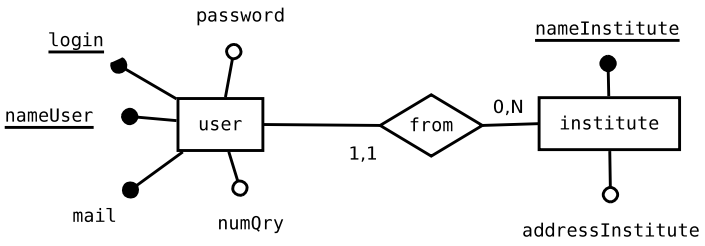

Figura 3.2: Modello concettuale per il login.

l'accesso viene registrato con un nominativo univoco,[3] la mail, lo userName o login (univoco), la password, il numero di interrogazioni alla banca dati e l'istituto di appartenenza. Di ogni istituto vengono salvati nome e indirizzo.

Dalla descrizione appena fatta, utilizzando il paradigma illustrato nel capitolo 2, è facile costruire il modello concettuale disegnato nella figura 3.2, formato da sole due entità, utente ed istituto, e una relazione per legare ogni utente all'istituto di appartenenza. L'analisi del dominio degli attributi è immediata; tutti i campi sono stringhe tranne il numero delle interrogazioni che è un valore intero. Vista la semplicità del modello concettuale, l'implementazione del modello relazionale e la stesura del codice MySql sono immediate. Per semplificare i collegamenti tra le due tabelle viene di nuovo utilizzata la tecnica degli indici identificativi. L'implementazione del database utente avviene tramite il modello relazionale e i codici SQL listati in A.3 e A.4. Creato il database per gestire gli utenti registrati è necessario inserire nel sito pagine web che permettano la registrazione a l'accesso alla banca dati.

3.3.2 Form e input

Per permettere la registrazione di un nuovo utente viene utilizzato uno specifico blocco HTML chiamato FORM. All'interno del tag "<form> </form>" è possibile inserire campi di input, compilabili dall'utente, che possono essere spediti ad un'altra pagina per eventuali rielaborazioni tramite l'evento "submit". Un form è strutturato secondo la sintassi 3.4:

Codice 3.4: Sintassi di un blocco form.

```
<form onsubmit="return funzioneJs(this)" action="pagina"
method="medodoInvio">
<input type=tipoInput name=nomeInput>
<input type="submit" value="invia">
</form>
```

Tra gli attributi di un form sono presenti "onsubmit", "action" e "method". Nella proprietà "onsubmit" viene indicata la funzione booleana in linguaggio JavaScript da eseguire al verificarsi dell'evento "submit", che avviene nel momento in cui un utente seleziona "submit" per l'invio dei dati, sempre presente all'interno di un blocco form. Nell'attributo "action" viene specificata la pagina alla quale

[3] Nel caso di più utenti omonimi verrà inserito un numero a fine nominativo.

inviare i dati se il risultato della funzione JavaScript è “true” (vero). Infine, in “method” viene indicato quale metodo di invio dei dati viene utilizzato tra i due disponibili: get e post.

Tramite il metodo get le informazioni vengono concatenate alla fine dell'URL della pagina di destinazione. Da notare che in questo caso, la lunghezza massima di tale stringa dipende dalle limitazioni imposte dal browser.

Invece, con il metodo post le informazioni vengono inviate dal browser tramite una transizione HTML non visibile dall'utente e senza limiti di dimensione. Durante la registrazione viene utilizzato il secondo metodo, perché oltre a non imporre limiti alla dimensione delle informazioni trasmissibili, è anche più sicuro, in quanto i dati inviati non vengono visualizzati direttamente nel browser.

Le informazioni inseribili in un form vengono contenuti nei tag di input, dotati di tre attributi fondamentali: “type”, “name”, “value”. In “type” viene specificato il tipo di input. I tipi più utilizzati sono:

- “submit” per l'invio dei dati,
- “text” per inserire informazioni testuali,
- “password”,
- “hidden” per inserire cambi invisibili o nascosti ,
- “button” per attivare funzioni,
- “checkbox” e “radio” per ottenere campi selezionabili.

I due tipi di input utilizzati nella pagina di registrazione sono campi di testo e password. In “name” viene inserito il nome per la trasmissione. In “value” viene inserito il valore fornito dall'utente. I dati inseriti nel form saranno inviati alla pagina di rielaborazione nel formato “name:value”, cioè nome dell'input : valore inserito. Per la registrazione (figura 3.3) sono necessarie le seguenti informazioni:

- nome,
- cognome,
- indirizzo email
- istituto di appartenenza del nuovo utente,
- una breve motivazione della richiesta dell'accesso alla banca dati,
- login (solo testo),
- password che deve essere riconfermata .

Una volta inserite le informazioni nei campi di input e selezionato di submit, prima della trasmissione si procede con la validazione del form tramite la funzione JavaScript specificata nell'attributo “onsubmit”.

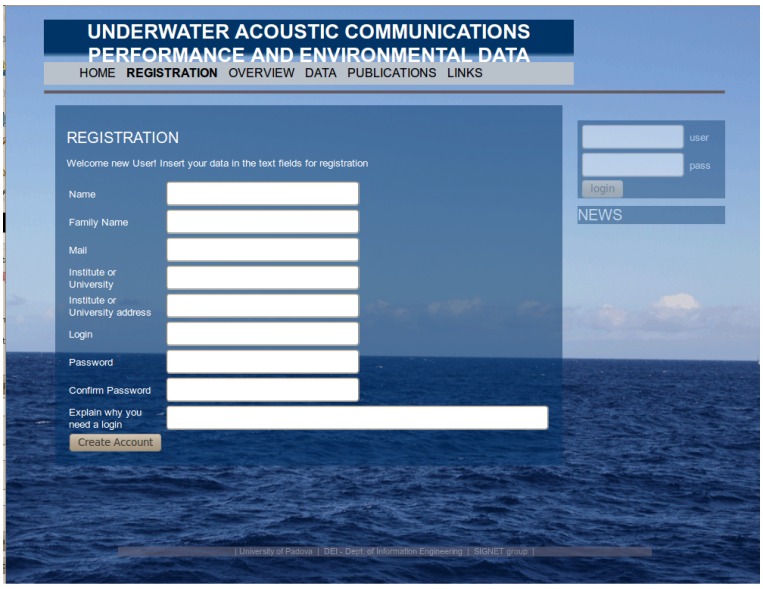

Figura 3.3: Pagina web per effettuare la domanda di registrazione.

3.3.3 Validazione in JavaScript

JavaScript è un linguaggio di scripting lato client, orientato ad oggetti, standardizzato da European Computer Manufacturers Association (ECMA) dal 1997 e comunemente utilizzato nelle pagine web. Regole sintattiche e definizioni di JavaScript utilizzate in questa tesi sono ottenute da [17], [15], [7] e [14]. Alla chiamata di una funzione in tale linguaggio, il codice script viene interpretato direttamente dal browser eseguendo sequenzialmente le istruzioni richieste. Il codice JavaScript può essere inserito in una pagina web in due modi diversi:

1. direttamente all'interno della head tra i tag "<script> </script>",
2. inserito all'interno di un file in formato .js, collegato al documento tramite il comando "<script src='nomeFile.js' type='javascript'>" nell'intestazione della pagina HTML.

Un file .js può contenere sia funzioni e variabili statiche che classi di oggetti. Nel progetto in dettaglio non è stata necessaria la creazione di alcuna classe di oggetti, in quanto gli unici oggetti utilizzati sono quelli già presenti di default nel documento.

In particolare vengono utilizzati "document" e "history". "Document" racchiude tutte le proprietà di un documento HTML, e "history" contiene informazioni sulla sequenza delle pagine web visitate da ogni utente. Dell'oggetto "document" vengono spesso utilizzati il metodo "getElementById('id')" e la proprietà "location". Il metodo "document.getElementById('id')" consente l'accesso e la modifica delle proprietà del blocco di codice HTML presente all'interno del tag identificato da 'id'. Tra le proprietà del blocco ottenuto, le più utilizzate nel progetto sono "innerHTML", che consente l'accesso e la modifica diretta del codice HTML di tale blocco, e "style" che consente di modificare le caratteristiche

di visualizzazione e visibilità di una sessione. Utilizzando "document.location" invece è possibile ottenere e modificare l'URL tramite la proprietà "href", per cambiare dinamicamente le pagine di navigazione.

Le variabili utilizzabili in JavaScript sono debolmente tipizzate, ovvero alla loro dichiarazione non è obbligatorio specificarne il dominio. Generalmente vengono tipizzate dinamicamente, ovvero sono definite semplicemente assegnando loro un valore. Le variabili dichiarate fuori da qualunque funzione sono in visibilità globale, accessibili dall'intera pagina web, mentre le variabili dichiarate dentro una funzione sono locali per quella funzione. In JavaScript il numero dei parametri passati quando si richiama una funzione non deve necessariamente essere uguale al numero degli argomenti presenti nella definizione della funzione.

La sintassi di una funzione è il codice 3.5:

Codice 3.5: Sintassi di una funzione JavaScript

```
function nomeFunz(...){
    istr;
    return val;
}
```

Dove al posto di "nomeFunz" viene specificato il nome della funzione, all'interno delle parentesi tonde vengono elencati i parametri formali, al posto di "istr" vengono scritte le istruzioni e al posto di "val" l'eventuale valore restituito dalla funzione. Ogni istruzione termina con il carattere delimitatore ";". La chiamata ad una funzione avviene semplicemente scrivendo il nome della funzione ed elencando tra parentesi tonde gli eventuali valori assegnati ai parametri formali. In particolare, al verificarsi di un evento che genera la chiamata di una funzione, può essere passato l'oggetto "this", contenente tutte le proprietà del blocco HTML dove si è verificato l'evento. Tutti i sottoblocchi sono attributi di tale oggetto, identificati dal nome assegnato loro nel codice HTML.

Per validare la pagina di registrazione sono necessarie istruzioni di controllo e di segnalazione errori. Il controllo degli ingressi avviene utilizzando il costrutto if-then-else secondo la sintassi "if(condizione) {istruzioni se vero} else {istruzioni se falso}". Per segnalare eventuali errori viene utilizzata la funzione "alert(stringa d'avviso);" che fa apparire all'utente una finestra d'avviso contenente la stringa specificata tra parentesi. La validazione avviene nel seguente modo. Al momento del verificarsi dell'evento "onsubmit" viene chiamata una funzione a cui vengono passate tutte le proprietà del form. Tale funzione controlla ogni campo di input e permette l'invio dei dati solo se tutte le informazioni sono state inserite nel modo corretto.

In particolare per validare il form di registrazione viene utilizzata la funzione booleana "sendReg(this);" (listata in A.5) che controlla che tutti i campi di input siano stati riempiti dall'utente: la mail sia nel formato corretto (nome@dominio), la lunghezza di login e password sia almeno di cinque caratteri e la seconda digitazione della password sia uguale alla prima. Se il controllo è andato a buon fine la funzione permette l'invio dei dati, altrimenti manda all'utente una segnalazione d'errore indicando l'input errato.

3.3.4 PHP: interrogazione al database utente e invio mail per registrazione

Una volta validata la pagina, le informazioni inviate tramite il form vengono utilizzate per organizzare una possibile registrazione, controllando che l'utente non sia già registrato e mandando una email con i dati inseriti al gestore del sito, che si farà carico di verificare l'identità degli utenti e di finalizzare la registrazione aggiornando il database.

Al fine di accedere al database e di comporre un'email è necessario utilizzare un linguaggio di programmazione lato server, il cui codice viene eseguito dal server prima di restituire le rielaborazioni all'utente. Nello specifico del progetto viene utilizzato il linguaggio PHP [4] ideato dal danese Rasmus Lerdorf nel 1994. Un programma scritto con questo linguaggio viene sempre elaborato sul server e mai reso disponibile all'utente, il quale può visualizzare solo il risultato del programma. Questo concetto è molto importante in quanto sta alla base della sicurezza e dell'affidabilità offerti dalla programmazione lato server. Regole sintattiche e definizioni di PHP sono ottenute da [17] e [12].

Per consentire le rielaborazioni sul server, nel parametro "action" del form viene indicata la pagina PHP alla quale inviare le informazioni inserite dall'utente. Una pagina con estensione .php contiene codice HTML, che viene normalmente interpretato dal browser, e codice PHP all'interno del tag "<?php ... ?>". Il linguaggio è debolmente tipizzato e supporta la programmazione ad oggetti, ma in questo progetto non è stato necessario implementare nuove classi, in quanto si sono utilizzati gli oggetti predefiniti in PHP.

In particolare, la registrazione è stata implementata utilizzando le funzioni predefinite nella libreria base del linguaggio. L'array "$_POST", contenente le informazioni inviate dal form tramite metodo post viene usato per accedere ai dati inviati dall'utente attraverso la regola sintattica "$_POST[nmDt]" dove al posto di "nmDt" viene inserito il nome del dato di cui si vuole ottenere informazioni. Tra le funzioni predefinite, ci sono quelle che gestiscono l'accesso ai database MySQL, rendono possibile l'invio di email, e reindirizzano l'URL. Inoltre, si è usato il costrutto if per fare verifiche e controlli sui dati.

Sintatticamente ogni variabile è dotata di un nome scritto dopo il carattere speciale "$", quindi compare nel formato "$nomeVar", a cui viene assegnato direttamente un valore. Per ogni tentativo di assegnazione viene fornita un'informazione booleana relativa all'esito dell'inserimento. Come in JavaScript ogni istruzione scritta in PHP termina con il carattere delimitatore ";".

Il processo di accesso al database MySQL avviene secondo il seguente protocollo:

1. accesso al DBMS,
2. selezione del database,
3. interrogazione delle tabelle,
4. chiusura della connessione.

L'accesso avviene tramite la funzione booleana "mysql_connect($db_host, $db_user, $db_pwd)" dove vengono specificati l'host (cioè il calcolatore della rete) dove è

[4] Preprocessore di ipertesti.

presente il DBMS, userName e password di sicurezza inseriti all'atto della creazione del database nell'RDBMS. Se l'accesso fallisce tale funzione restituisce "false", altrimenti se la connessione ha successo restituisce "true".

Una volta effettuato l'accesso al DBMS viene selezionato il database da interrogare con la funzione "mysql_select_db($nomeDatabase)", che, analogamente alla funzione precedente, restituisce un'informazione booleana riguardante il successo dell'accesso al database selezionata.

Successivamente, si interroga il database inserendo la query in SQL come parametro alla funzione "$result=mysql_query($queryString);", dove al posto di "$queryString" viene inserita l'interrogazione in linguaggio SQL e la variabile "$result" viene utilizzata per salvare la vista (view) della tabella risultato della query di selezione.

Una volta interrogato il database è possibile ottenere le informazioni richieste riga per riga con la funzione "$row = mysql_fetch_row($result)", che restituisce un array con i dati richiesti, i cui campi sono identificati con i nomi delle colonne selezionate. Per scandire ogni riga si è usato il costrutto "while($row = mysql_fetch_row ($result)){...}" che, oltre a ripetere un blocco di istruzioni, legge ad ogni ciclo una riga del risultato della query, finché tutte le entry non vengono lette. Analogamente, per scandire ogni cella di una riga viene utilizzato il costrutto "foreach($row as $cell)". Infine, la connessione viene chiusa tramite il metodo "mysql_free_result($result);".

Per inviare un messaggio di posta elettronica tramite PHP si può utilizzare la funzione mail, con la sintassi "mail(destinatario, oggetto, messaggio, mittente e altre informazioni);". Nel sito inoltre viene spesso utilizzata l'istruzione "echo codice HTML;" che permette la scrittura del codice HTML sul documento.

Con il linguaggio PHP viene implementata la pagina utilizzata per la registrazione, che è divisa in tre blocchi fondamentali:

- lettura delle informazioni ricevute dal form,
- interrogazione dal database,
- rifiuto della registrazione o invio mail di conferma.

Il primo blocco è il più semplice: basta iniziare delle variabili con il contenuto dell'array "$_POST".

Il secondo blocco è il più complesso, in quanto si accede per la prima volta ad un database MySQL dinamicamente, utilizzando le funzioni fornite da PHP e codice SQL. L'interrogazione avviene utilizzando una semplice query di selezione di alcune colonne di una tabella del database; tale query è strutturata secondo la sintassi "SELECT attr1,attr2 FROM table WHERE attr1=a OR attr2=b" cioè: seleziona gli attributi "attr1" e "attr2" dalla tabella "table" se "attr1" assume il valore "a" e "attr2" il valore "b". Al posto dell'elenco degli attributi dopo "select" può essere inserito il simbolo "*" che sta ad indicare la selezione di tutte le colonne della tabella. Per la registrazione, la query di selezione viene utilizzata per verificare se l'indirizzo email e il login inserite sono già state utilizzate da un altro utente. In tal caso la registrazione fallisce, altrimenti si procede con l'invio di una email al gestore del sito. Per verificare la validità della registrazione si osserva se il risultato dell'interrogazione non è vuoto, cioè se è stata selezionata almeno una riga. A tale scopo viene osservato il risultato del metodo "mysql_fetch_row (risultatoQuery)". Tale funzione restituisce la prima riga selezionata dalla query oppure nessun valore.

Nel caso non venga restituito alcun valore la registrazione va a buon fine, altrimenti fallisce in quanto login o mail inserite dall'utente risultano già utilizzate. A tale scopo viene chiamata una funzione JavaScript, dove tramite alert viene segnalato l'errore e tramite il metodo "history.go(-1)" si ritorna alla pagina precedente, per permettere un altro tentativo di registrazione. Se i campi inseriti dall'utente sono validi, avviene la conferma della domanda di registrazione inviando una email al gestore del sito con il metodo "mail(destinatario, oggetto, messaggio, mittente e altre informazioni);". Per il codice vedi il listato A.6.

3.3.5 Login: form e interrogazione database

Una volta che il gestore del sito ha finalizzato la registrazione è possibile effettuare l'accesso al database tramite login e password. A tale scopo in ogni pagina è presente un riquadro sulla destra contenente un form con due campi di input, uno testuale, per inserire il nome utente, e uno per la password. Inoltre, se un utente che non ha effettuato il login tenta di accedere alla banca dati, viene reindirizzato ad una pagina dedicata al login contenente un form identico a quello del riquadro appena descritto.

Una volta digitati username e password, tramite submit, i campi di input vengono inviati ad una pagina PHP dove avviene l'interrogazione al database per la verifica del login. Se nome utente o password sono errati si ritorna alla pagina di login per permettere all'utente di reinserire le sue credenziali, altrimenti se l'operazione di accesso è andata a buon fine, sul server viene aperta una sessione utente e inizia l'interrogazione guidata alla banca dati. La validazione del form di login avviene tramite una funzione JavaScript per controllare che i campi di input siano stati riempiti prima dell'invio, in modo da evitare l'accesso al database con un'interrogazione che provoca un esito negativo.

La correttezza delle credenziali d'accesso viene verificata tramite una pagina in PHP che, dopo aver letto i dati inviati dal form tramite metodo post, verifica nel database la presenza di un utente con login e password uguali a quelle inserite, utilizzando la query "SELECT * FROM user WHERE login=userName AND password=pass", dove "userName" e "pass" sono le credenziali inserite dall'utente. Se il risultato dell'interrogazione non è vuoto allora si procede con la creazione di una sessione che permettere l'accesso alla banca dati, altrimenti si ritorna al form di login.

3.3.6 Login: sessione utente

Definizione 3.3.1 (Sessione utente)**.** *La sessione è l'attività svolta tra un client e un server per trasferire dati in entrambi i sensi per tutta la durata del collegamento, cioè da quando l'utente si collega al server fino a quando interrompe la connessione internet, senza dover effettuare l'identificazione ad ogni nuova pagina visitata.*

Una sessione viene creata dal server al momento dell'esecuzione della funzione PHP "session_start();". Una volta aperta la sessione, vengono assegnati un nome univoco e un file di testo omonimo contenente tutte le proprietà, modificabili tramite l'array globale "$_SESSION[]". Alla creazione tutte le proprietà sono vuote, per permettere il riconoscimento dell'apertura di una sessione è necessario inizializzare manualmente almeno un attributo. Visto che la sessione

aperta in questo sito serve per permettere l'accesso a un utente specifico, viene inizializzata la proprietà "user" per salvare il nome dell'utente con l'istruzione "$_SESSION['user']=$usr;".

Così, durante la navigazione, in tutte le pagine tramite il coockie [5] di sessione, sarà possibile riconoscere l'utente che ha effettuato l'accesso. Per ottenere il coockie di sessione viene utilizzata l'istruzione "session_start();" come prima istruzione di ogni pagina di navigazione dopo il login, mentre per verificare se la proprietà user è stata inizializzata viene utilizzata la funzione booleana "isset($_SESSION['user'])". La sessione termina automaticamente alla chiusura del browser oppure con l'istruzione PHP "session_destroy();" utilizzata per il logout.

3.4 Accesso alla banca dati

Una volta effettuato il login e aperta la sessione di navigazione, l'utente può accedere alla banca dati. L'accesso ai dati è guidato e avviene selezionando le specifiche delle informazioni che si desidera ottenere. L'utente deve selezionare l'esperimento, il tipo di dato, i dispositivi coinvolti e l'arco temporale delle misure richieste.

3.4.1 Blocco select e option

Per selezionare da un menù a tendina l'opzione desiderata viene utilizzato un particolare blocco HTML, chiamato select, che utilizza la sintassi 3.6:

Codice 3.6: Sintassi di un blocco select in un documento HTML.

```
<select name='nomeSelect'>
<option value='0'>opzione1</option>
<option value='1'>opzione2</option>
...
</select>
```

All'interno dei tag "<option></option>" vengono specificati i nomi di ogni opzione selezionabile, con rispettivo valore identificativo inserito nell'attributo value. Al fine di inviare le opzioni selezionate ad una pagina PHP, dove avviene l'interrogazione della banca dati, è necessario inserire ogni select all'interno di un blocco form. La stringa inviata dal form è "nameSelect:value", dove al posto di value viene inserito il valore dato all'opzione selezionata. Le select presenti nel sito sono tutte create dinamicamente da funzioni PHP implementate appositamente, al fine di rendere selezionabili le proprietà del database.

3.4.2 Funzioni in PHP

Una funzione PHP utilizza la seguente sintassi: "function nomeFunzione($attributo1,..) istruzioni;" e, come in JavaScript, in fase di utilizzo non è necessario istanziare ogni attributo formale. Per creare le select contenenti opzioni ricavate dal database, è stata ideata la funzione "querySelect($name,$tab,$whr,$a,$b)", i cui campi servono a dichiarare il nome assegnato alla select ("$name") , il nome della tabella ("$tab") da cui leggere le colonne "$a" e "$b" e la clausola where ("$whr").

[5] Un coockie è una corta stringa di testo inviata dal server.

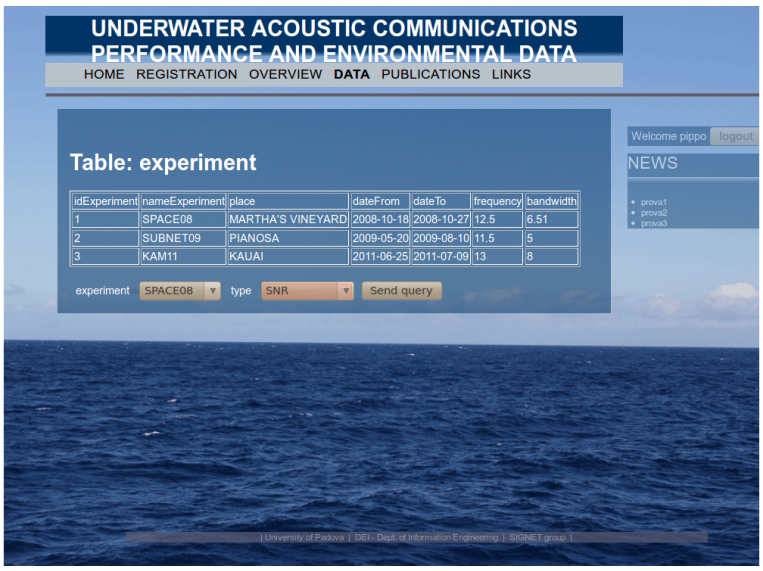

Figura 3.4: Pagina web contenente le select per l'interrogazione.

Tale funzione PHP, dopo aver interrogato il database con la query "SELECT $a,$b FROM $tab $whr", crea un blocco select con attributo "name=$name", contenente le opzioni presenti nella colonna "$b" il cui valore ("value") è assegnato dalla colonna "$a". A tale scopo viene utilizzata l'istruzione "echo 'codice HTML';" che permette la scrittura del codice HTML sul documento. Nella prima pagina (figura 3.4) utilizzata per l'interrogazione sono presenti:

- una tabella per illustrare le caratteristiche degli esperimenti (ottenuta interrogando la banca dati);
- un form contenente due blocchi select, uno creato con la funzione "querySelect" per selezionare l'esperimento, e uno creato senza la necessità di interrogare la banca dati contenente il tipo di dato desiderato.

Una volta selezionate le opzioni e validata la form tramite apposita funzione JavaScript, i due valori vengono inviati ad un'altra pagina PHP per completare la scelta delle specifiche. I valori ricevuti dalla form, oltre a venire rielaborati, vengono salvati in campi di input nascosti, chiamati hidden, al fine di poter essere inviati all'ultima pagina per l'interrogazione definitiva.

Se il dato richiesto è di tipo ambientale viene creata dinamicamente un'unica select, per selezionare da quale dispositivo sono state ricavate le misure. Se il dato richiesto è di tipo acustico, vengono create dinamicamente due select per poter selezionare sia il dispositivo che ha iniziato la trasmissione, sia il dispositivo che ha ricevuto il pacchetto di dati inviato. In entrambi i casi viene interrogata la tabella dei dispositivi ("device"): vengono selezionati per tipo ed esperimento, ordinati secondo la profondità e raggruppati per vertical array.

La funzione "querySelect" implementata in questa pagina ha un parametro in più della precedente, "$c", infatti le colonne da selezionare dalla tabella dispositivi sono tre: identificativo ("idDevice"), etichetta del dispositivo ("label")

e profondità ("depthSea"). Nelle opzioni vengono visualizzati etichette e profondità, mentre l'identificativo viene assegnato al valore delle opzioni. La query per selezionare i dispositivi è 3.7.

Codice 3.7: Query per selezionare i dispositivi.

```
"SELECT idDevice, label, depthSea FROM device
where type='$typeDevice'
AND idExperiment=$experiment
order by idDevice"
```

dove "$typeDevice" e "$experiment" sono variabili contenenti i valori passati dal form precedente.

3.4.3 JQuery e datepicker

Per inserire l'arco temporale durante il quale sono avvenute le misure che si vogliono analizzare, vengono utilizzati dei selettori di data ed ora, non presenti nella libreria HTML ma implementati tramite funzioni JavaScript e il plugin UI di jQuery. JQuery è una libreria di funzioni (o framework) JavaScript per le applicazioni web sviluppata da John Resig nel 2005, che ha come obiettivo la semplificazione della programmazione lato client delle pagine HTML. In particolare tale libreria rende disponibile un plugin ufficiale, chiamato jQueryUI (user interface), che provvede a fornire un'accattivante interfaccia grafica, includendo fogli di stile, finestre modali con resizing e moving, calendari, scrollbar e grafici. Libreria a plugin sono gratuitamente scaricabili dal sito ufficiale di jQuery. La sintassi di jQuery è ricavata da [17] e [16]. Per includere le librerie jQuery e jQueryUI in un sito web, nell'intestazione vanno inseriti i collegamenti agli script "jquery-ui-1.8.18.custom.min.js", " jquery-1.7.1.min.js" e al foglio di stile "jquery-ui-1.8.18.custom.css", presenti all'interno della libreria, con i tag listati in A.8. jQuery consente l'accesso ai blocchi HTML tramite una sintassi molto semplice, utilizzando le classi o gli identificativi assegnati ai tag. Attraverso la funzione "jQuery(..)", abbreviabile utilizzando il selettore "$(..)", si accede al codice HTML nei seguenti modi:

- con "$('.classe')" (o con "jQuery('.classe')") si accede a tutti i blocchi appartenenti a tale classe,
- con "$('#id')" si accede al blocco a cui è stato assegnato tale identificativo.

JQuery ha una sintassi simile al linguaggio CSS. Per accedere ad un campo di input appartenente ad una classe si utilizza il filtro di selezione ":" con la proposizione "$('.classe:input')". Per impostare gli attributi di un blocco del documento HTML basta utilizzare la sintassi presente nel listato 3.8:

Codice 3.8: Sintassi di jQuery.

```
$('blocco')({
   nomeAttr1:valore1,
   nomeAttr2:valore2,...
});
```

All'interno della form viene utilizzato il widget Datepicker (figura 3.5) che, una volta inizializzato, implementa all'interno di una casella testo un piccolo calendario a giorni selezionabili, per impostare le date di inizio e fine reperimento

dati desiderate. Tale widget consente inoltre di impostare la prima e l'ultima data selezionabili, in modo da permettere all'utente di trovare più facilmente il giorno cercato ed evitare inserimenti di date non significative per l'interrogazione. Nel progetto tali proprietà vengono inizializzate con le date di inizio e di fine esperimento, ottenibili dalla tabella experiment interrogando il database. I giorni limite vengono inseriti in Datepicker come oggetti Date, supportati dalla libreria base di JavaScript e istanziabili con le due stringhe ottenute dall'interrogazione. Lo script listato in 3.9 assegna un DatePicker ad ogni campo di testo appartenente alla classe "data" e inizializza gli attributi riguardante la prima e l'ultima data selezionabili.

Codice 3.9: Datepicker jQuery.

```
<script type='text/javascript'>
   $(function() {
      $('.data:input').datepicker( {
         minDate:new Date($date1),
         maxDate:new Date($date2),
      } );
   });
</script>
```

3.4.4 Timepicker in JavaScript

Per selezionare l'ora è stato creato un blocco dinamico apposito, controllato da una funzione scritta in JavaScript. È stata utilizzata una tabella con una riga e tre colonne per contenere ore, minuti e secondi. Tali valori numerici sono inseribili dentro un input di testo manualmente da tastiera o incrementando e decrementando i valori presenti di default, premendo su "+" e "-" rispettivamente. A titolo esemplificativo, in 3.10 viene mostrato solo il codice della colonna per l'inserimento dell'ora, in quanto il codice per l'inserimento di minuti e secondi è molto simile.

Codice 3.10: Timepicker jQuery.

```
<td>
<p class="incdec" onclick="refreshTime('hourFrom',1)">+</p>
at time 
<input id="hourFrom" class="time" type="text"
onkeypress="return onlyNumber(event,this,0);"
value="0" name="hourFrom">
<p class="incdec" onclick="refreshTime('hourFrom',-1)">-</p>
</td>
```

Variando i valori temporali utilizzando "+" o "-" si verifica l'evento "onclick", che causa la chiamata alla funzione JavaScript refreshTime('id',valore) per l'aggiornamento dell'ora, che, rispetto il segno di "valore", incrementa o decrementa i campi numerici ponendo attenzione a non superare il minimo e il massimo valore consentiti (da 0 a 23). Allo stesso modo, per quanto riguarda minuti e secondi, viene chiamata la funzione "refreshMinSec(id,valore)" che aggiorna i minuti o i secondi, ponendo attenzione a non superare il minimo e il massimo valore consentiti (da 0 a 59). Le funzioni refreshTime e refreshMinSec sono presenti nel listato A.9. Inoltre viene controllato che l'ingresso inserito da tastiera sui

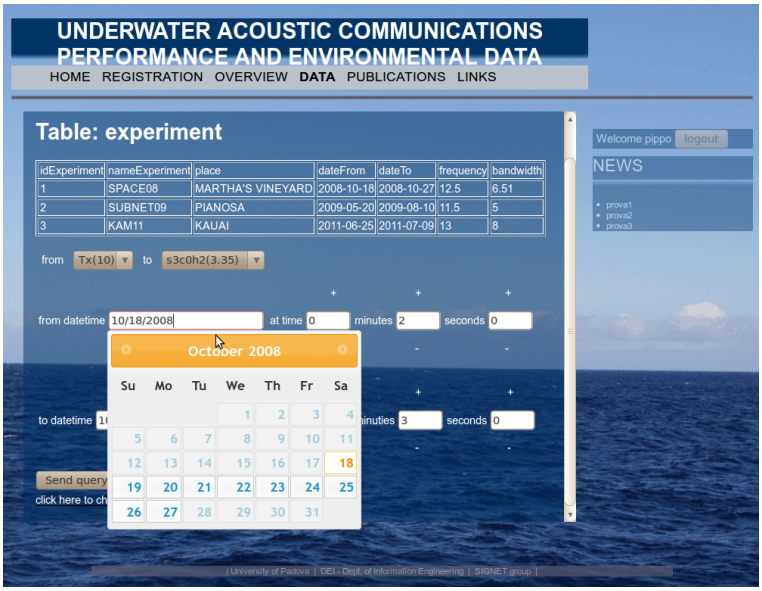

Figura 3.5: Pagina web contenente datepicker e timepicker per l'interrogazione.

campi di input sia valido, chiamando la funzione "onlyNumber(event,this,flag)" al verificarsi dell'evento di pressione di un tasto, chiamato "onkeypress". Il parametro "event" è un oggetto contenente tutte le informazioni sull'evento che ha richiamato la funzione, mentre "flag" è un valore numerico utilizzato per indicare se il campo è per l'ora, i minuti o i secondi. La funzione "onlyNumber" controlla che il tasto digitato sia corrispondente ad una cifra e che il numero inserito nel campo sia compreso tra i valori consentiti. Se una di queste condizioni non viene verificata l'evento viene ignorato, altrimenti l'inserimento ha successo. Questo viene mostrato in A.10. Una volta inserite le specifiche desiderate nei campi di input, prima dell'invio si procede con la validazione della form che, oltre a controllare che siano inizializzati tutti i campi richiesti, procede con la conversione dei parametri temporali da data e ora a epoch, ottenuto utilizzando la comoda funzione "getTime()" di un oggetto "Date" inizializzato con data e ora inseriti. Dopo la conversione le specifiche vengono inviati all'ultima pagina PHP dove vengono forniti all'utente i dati richiesti, e viene incrementato il campo indicante il numero di query richieste da tale utente.

3.4.5 Dati forniti all'utente: apertura file tramite PHP e grafici jqPlot

Per fornire le misure la banca dati viene interrogata con due tipi di query diverse, a seconda che i dati richiesti siano di tipo acustico o ambientale. Nella query per i dati ambientali si selezionano l'epoch e il valore della stima della misura dalla tabella delle misure ambientali, dove la misura è:

- effettuata da un determinato dispositivo,
- di una determinata tipologia

- effettuata tra due precisi istanti temporali.

Tale coppia di dati viene ordinata per successione temporale come mostrato in A.11. La query per i dati acustici è molto simile e viene listata in A.12. L'istruzione seleziona in maniera ordinata epoch e il valore della stima della misura acustica, dove la misura è:

- presente all'interno della tabella delle misure acustiche,
- riferita alla trasmissione di un pacchetto inviato da un dispositivo trasmettitore e ricevuto da un altro dispositivo ricevitore,
- appartenente ad una determinata tipologia (BER, CIR, SNR),
- effettuata tra due precisi istanti temporali.

Ovviamente tutte le caratteristiche delle misure, sia ambientali che acustiche, esplicitate nella clausola "where" sono ottenute leggendo le informazioni inviate dal form della pagina precedente. Una volta eseguita l'interrogazione, i dati ottenuti vengono forniti all'utente. A tale scopo vengono visualizzati sul browser una tabella (contenente due colonne: epoch e valore della misura) e un link dal quale scaricare la stessa tabella. Inoltre se il tipo di dato non è CIR, le misure vengono inserite in un grafico, dove l'asse x rappresenta l'epoch e l'asse y il valore numerico. Gli strumenti utilizzati per rendere disponibili all'utente tali funzionalità sono ancora PHP, tramite il quale viene interrogato il database e vengono creati dinamicamente una tabella e un file con i dati richiesti, JavaScript e jqPlot, particolare plugin di jQuery che permette di creare dei grafici su pagine web.

Per aprire i file in scrittura con PHP viene utilizzata l'istruzione "$file=fopen(nf, 'w+')", dove al posto di "nf" viene indicato il nome del file. Una volta aperto in scrittura è possibile scrivere nel file con il metodo "fputcsv($file, row)", dove al posto di "row" viene inserito direttamente l'array contenente una riga intera del risultato dell'interrogazione. Tale operazione (listato A.13) viene inserita in un ciclo while, al fine di essere eseguita per ogni riga ottenuta tramite la query.

Per costruire un grafico scientifico viene utilizzata la libreria jqPlot, gratuitamente scaricabile dal sito ufficiale. Per incorporare tale strumento in una pagina web, è necessario inserire nella header del documento HTML i seguenti collegamenti:

- a "jquery.jqplot.min.js",
- a jQuery e jQueryUI,
- a tutte le proprietà utilizzate (come le librerie per modificare le assi cartesiane),
- al foglio di stile di jqPlot.

A tale scopo, oltre ai blocchi comuni alla pagina precedente, sono stati inseriti i tag A.14. La comodità di utilizzare jqPlot rispetto altri strumenti per realizzare grafici sta nella presenza di molte variabili, utilizzate per la personalizzazione del grafico, e nella semplicità di collegamento ad una pagina web dinamica. In particolare è stata sfruttata la possibilità di personalizzare le assi del grafico. Assegnando il valore voluto ai parametri è possibile impostare, ad esempio:

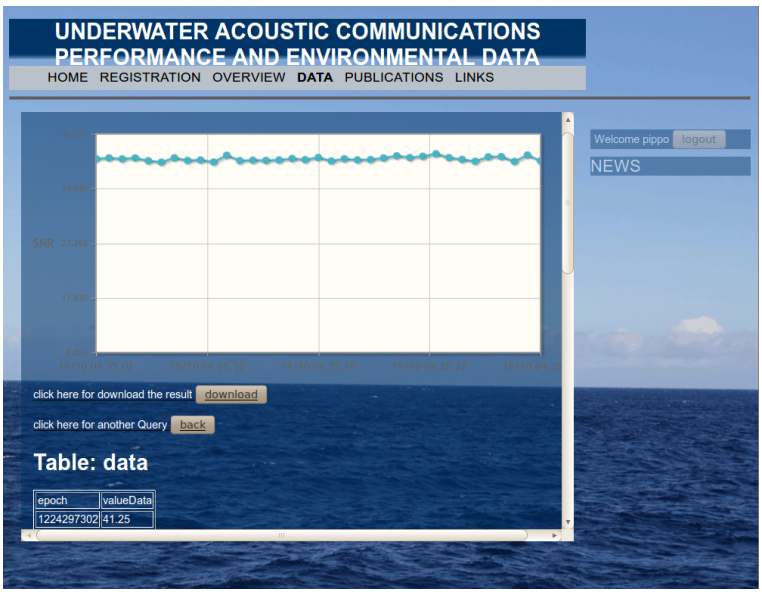

Figura 3.6: Pagina web per visualizzare i risultati dell'interrogazione.

- tipo ("renderer") e nome di un'asse ("label"),
- formato e intervallo delle etichette delle posizioni su un'asse ("tickInterval" e "tickOptions"),
- range di valori contenuti ("min", "max").

Sono state cambiate le proprietà dell'asse x per renderlo compatibile con i valori inseriti come epoch, ma visualizzati come data e ora (vedi listato A.15). All'istruzione per creare il grafico va passata una matrice di coppie di valori nel formato [[x1,y1],[x2,y2],..] che indicheranno le posizioni dei punti da rappresentare. Nel progetto illustrato le coppie nella matrice sono [epoch,value] e tale matrice viene creata dinamicamente, tramite PHP, con i dati ottenuti dall'interrogazione del database, vedi figura 3.6. Se il dato selezionato è di tipo CIR allora non viene visualizzato alcun grafico, ma solamente una tabella con dei collegamenti. La tabella è composta da due righe, una contenente l'epoch e l'altra il nome del file contenente i dati richiesti. Nell'ultima colonna vengono inseriti anche due collegamenti a file, chiamati ".mat" e ".png"; il primo permette di scaricare una matrice matlab e il secondo un'immagine, contenenti la risposta impulsiva tabulata e disegnata su un grafico. I file sono stati caricati nel server al momento dell'implementazione della banca dati.

3.4.6 Ulteriori funzionalità aggiuntive

Al fine di rendere la navigazione più comoda e completa, nel sito sono presenti alcune funzionalità aggiuntive.

Per registrare una statistica di accessi alla banca dati, all'invio di ogni query guidata viene incrementato il numero di interrogazioni eseguite dall'utente

connesso. Prima viene letto il numero di query richieste dall'utente fino a quel momento, poi tale valore viene incrementato e inserito nel database a sostituire il valore precedente. Per eseguire questa operazione vengono effettuati due accessi alla proprietà "numQry" della tabella utente, uno per la lettura e l'altro per l'aggiornamento (update) del parametro, tramite il codice SQL 3.11

Codice 3.11: Aggiornamento del numero di interrogazioni.

```
"select numQry from user
where login='".$_SESSION['user'].";
"update user set numQry=valoreAggiornato
where login='".$_SESSION['user'].".
```

Per rendere più sicura la navigazione, una volta effettuato il login, è possibile chiudere l'accesso alla sessione utente premendo sul pulsante "logout". Tale evento provoca l'accesso a codice PHP contenente le funzione "session_destroy();", che fa terminare l'accesso al server, e "header('location: home.html');" che reindirizza l'URL alla home del sito.

Un'altra utile funzionalità è la presenza del pulsante back nelle pagine PHP utilizzate per creare la query guidata. Premendo su tale campo avviene il reindirizzamento alla prima pagina dell'interrogazione, al fine di poter modificare i parametri scelti o creare una nuova query guidata.

Capitolo 4

Conclusioni e lavoro futuro

Per concludere, la realizzazione della banca dati e del sito web soddisfano le specifiche richieste per raccogliere e rendere disponibili alla comunità scientifica le misure ambientali e le stime delle metriche del canale acustico sottomarino. Attualmente il progetto è sotto forma di prototipo, utilizzabile per verificare le prestazioni del database e la comodità di navigazione del sito web, prima di implementare la versione definitiva in un server dell'Università.

In questo capitolo, vengono illustrate le possibili modifiche alla banca dati e al sito internet da apportare per migliorare l'accesso ai dati. Alcune modifiche implicano semplici aggiustamenti del codice, mentre altre presentano approcci tecnologi diversi. Nel primo paragrafo vengono proposti i miglioramenti alla banca dati e nel secondo utili modifiche al sito web.

4.1 Miglioramenti al database

In questo paragrafo vengono proposti ed illustrati dei cambiamenti alla banca dati, al fine di velocizzare l'accesso alle informazioni.

Visto che tutti i tipi di misure acustiche assumono valori numerici, con l'eccezione di CIR, sarebbe opportuno creare una tabella dedicata alle risposte impulsive, in modo da poter avere il valore delle altre misure salvato in un campo numerico e non come stringa. Questa modifica porterebbe alcuni notevoli vantaggi. In primo luogo, un valore numerico occupa in memoria meno spazio di una stringa e permette operazioni più veloci. In secondo luogo, gli ordinamenti tra valori numerici producono risultati diversi dagli ordinamenti di stringhe. Ad esempio, la verifica di disuguaglianza stretta, "$<$", tra i valori 750 e 1000 produce vero nel caso di ordinamento numerico, falso nel caso di ordinamento tra stringhe. Nell'attuale implementazione del database, per ottenere i valori delle misure acustiche come numeri è necessario eseguire l'operazione di casting da stringa a valore numerico. Con tale gestione, eseguire interrogazioni di selezione di misure acustiche con valori compresi tra due intervalli diventa inefficiente.

Nella fase delle interrogazioni guidate, l'operazione più lenta è il confronto del tipo di misura nella clausola "where". Infatti tale operazione è l'unico confronto tra stringhe nella query, mentre le altre sono tutte operazioni di confronto tra campi numerici. Per velocizzare tale processo si propone di creare un indice sul tipo di dato. Comunque, il confronto tra stringhe resta più lento del confronto

numerico. Una proposta per risolvere il problema è scorporare l'informazione type secondo il seguente processo:

1. eliminare la proprietà "type" dalle misure,
2. creare la tabella "type" contenente due campi univoci: "idType" indice intero utilizzato come chiave primaria e "label" contenente il tipo di misura,
3. associare la tabella "type" alle misure con una relazione (1,N).

Con questa modifica nelle tabelle delle misure il campo per indicare il tipo sarà un valore numerico intero, così l'operazione di confronto sarà velocizzata.

Al fine di velocizzare le query, una soluzione sarebbe cambiare totalmente la struttura del database, creando più tabelle per le misure, ad esempio una per esperimento o una per tipologia di misura, in modo da interrogare tabelle più piccole. Tuttavia, così facendo non si soddisfa la richiesta della creazione di una banca dati unica.

Un'altra soluzione per migliorare la gestione e la velocità di accesso ai dati è il cambio di tecnologia. Per l'implementazione del database relazionale è stato utilizzato MySQL, uno dei migliori software RDBMS gratuito attualmente a disposizione. Tra altri RDBMS attualmente disponibili, anche a pagamento, il migliore è Oracle, potente software utilizzato da molte aziende per manipolare efficientemente GigaByte di dati. Implementare lo stesso database in Oracle, quindi, garantirebbe maggiore efficienza e velocità.

4.2 Miglioramenti al sito

Nell'ultimo paragrafo della tesi viene fatto un piccolo elenco di funzionalità da aggiungere al sito internet, per rendere la navigazione più completa ed intuitiva.

Al fine di migliorare l'interfaccia utente nelle pagine dinamiche sarebbe opportuno ideare un timePicker più comodo e accattivante, ad esempio migliorando la selezione di "+" e "-" utilizzando dei pulsanti, oppure creando una struttura simile al datepicker di jQueryUI. Per rendere meno monotona la pagina si sarebbero potuti utilizzare anche i radio button invece di utilizzare solamente gli input select.

Sarebbe più comodo avere un'unica pagina per creare la query, inviando un unico form. Ciò permetterebbe la trasmissione delle stesse informazioni utilizzando meno banda internet. Per permettere tale proprietà bisogna utilizzare un'altra tecnica di sviluppo web, Asynchronous JavaScript and XML (AJAX).[1] Tale tecnica permette di realizzare applicazioni web interattive (Rich Internet Application) e asincrone. Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. Con la programmazione asincrona si supera il concetto di elaborazione dei dati solo se provenienti da moduli form, le informazioni aggiuntive vengono richieste al server e caricate in background senza interferire con il comportamento della pagina esistente. Di conseguenza, utilizzando questa tecnica, l'interfaccia web prevede la presenza di due select attive, experiment e type,

[1] AJAX è una tecnica di sviluppo web ideata da Jesse Garrett nel 2005. L'acronimo significa Asynchronous JavaScript and XML, cioè JavaScript asincrono e XML.

mentre gli altri ingressi vengono disattivati. Alla selezione di esperimento e tipo desiderati vengono attivati gli altri campi di input, per i dispositivi ed i valori temporali, inizializzati leggendo il database in modo asincrono, cioè senza dover cambiare pagina.

Una volta effettuato l'accesso se si naviga sulle pagine di descrizione, anche se la sessione utente è attiva, viene visualizzato il riquadro di login invece di segnalare che l'accesso è stato effettuato. Quando si ritorna alle pagine di interrogazione si risulta correttamente ancora connessi, in quanto non è stato effettuato alcun logout. Le pagine di descrizione sono scritte in HTML e con tale linguaggio non è permesso controllare se la sessione utente è attiva. Per permettere tale funzionalità occorre cambiare l'estensione in PHP e gestire la sessione, oppure cercare se tra le funzioni avanzate di jQuery è possibile verificare la connessione di un utente.

Utilità mancante al sito è una pagina per la personalizzazione delle caratteristiche dell'utente connesso, come ad esempio la possibilità di modificare la password. Le proprietà dell'utente sono minimali e le operazioni che può compiere sono specifiche per l'accesso della banca dati.

Se l'utenza fosse ampia potrebbe tornare utile la creazione di un forum o una mailing list di discussione in modo da ottenere facilmente un feedback dall'utente.

Appendice A

Annex

Codice A.1: Modello relazionale del database.

```
Experiment = {idExperiment:integer, nameExperiment:String, place
    :String, dateFrom:Date, dateTo:Date, frequency:real,
    bandwidth:real }

Device = {idDevice:integer, label:String, type:String, latitude
    :real, longitude:real, depthSea:real, power:real,
    idExperiment:integer = Experiment(idExperiment) }

EnvMeasure= {idEnvMeasure:integer, epoch:integer, type:String,
    valueData:real, sourceFile:String, idDevice:integer=
    Device(idDevice) }

AcousticMeasure {idAcousticMeasure:integer, epoch:integer, type:
    String, valueData: String, sourceFile:String, modulation:
    String, idTx:integer=Device(idDevice), idRx:integer=
    Device(idDevice) }
```

Codice A.2: Implementazione del database tramite RDBMS MySQL.

```
CREATE TABLE experiment (
   idExperiment INT NOT NULL AUTO_INCREMENT PRIMARY
KEY,
   nameExperiment VARCHAR(15) UNIQUE,
   place VARCHAR(50),
   dateFrom DATE NOT NULL,
   dateTo DATE NOT NULL,
   frequency REAL NOT NULL,
   bandwidth REAL NOT NULL
);
CREATE TABLE device (
   idDevice INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   label VARCHAR(50) NOT NULL,
   type VARCHAR(50) NOT NULL,
   latitude REAL NOT NULL,
   longitude REAL NOT NULL,
   depthSea REAL NOT NULL,
```

```
   power REAL,
   idExperiment INTEGER NOT NULL REFERENCES
experiment(idExperiment),
   UNIQUE (latitude, longitude, depthSea),
   UNIQUE (label)
);
CREATE TABLE envMeasure (
   idEnvMeasure INT NOT NULL AUTO_INCREMENT
PRIMARY KEY,
   epoch INT NOT NULL,
   type VARCHAR(50) NOT NULL,
   valueData REAL NOT NULL,
   sourceFile VARCHAR(50) NOT NULL,
   idDevice INT NOT NULL REFERENCES device(idDevice)
);
CREATE TABLE acousticMeasure (
   idAcousticMeasure INT NOT NULL AUTO_INCREMENT
PRIMARY KEY,
   epoch INT NOT NULL,
   type VARCHAR(50) NOT NULL,
   valueData VARCHAR(50) NOT NULL,
   sourceFile VARCHAR(50) NOT NULL,
   modulation VARCHAR(50) NOT NULL,
   idTx INT NOT NULL REFERENCES device(idDevice),
   idRx INT NOT NULL REFERENCES device(idDevice)
)
```

Codice A.3: Tabelle del database utente.

```
User { idUser : integer, nameUser: String, mail: String,
login: String, password: String, numQry: integer,
idInstitute: integer = Institute(idInstitute)}
Institute { idInstitute : integer, nameInstitute: String,
addressInstitute: String }
```

Codice A.4: Implementazione del database utente in SQL.

```
CREATE TABLE user (
   idUser INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   nameUser VARCHAR(45) UNIQUE NOT NULL,
   mail VARCHAR(50) NOT NULL,
   login VARCHAR(10) NOT NULL,
   password VARCHAR(10) NOT NULL,
   idInstitute INTEGER NOT NULL REFERENCES institute(
       idInstitute),
   numQry INTEGER NOT NULL default=0,
   UNIQUE (login)
   );
CREATE TABLE institute (
   idInstitute INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   nameInstitute VARCHAR(50) NOT NULL,
   addressInstitute VARCHAR(50) NOT NULL
   );
```

Codice A.5: Listato della funzione sendReg in JavaScript.

```
function sendReg(reg){
   var send=false;
   if (reg.userName.value=="")
      alert("Insert the name");
//... controllo analogo se tutti i campi sono stati inseriti
    ...
   else if (reg.mail.value.indexOf("@") == (-1) || reg.mail.
       value.indexOf(".") == (-1))
      alert("Error: wrong mail format, '@' and '.' is
          request");
   else if (reg.login.value.length<5)
      alert("The login length must be bigger than 5
          characters");
   else if (reg.password.value.length<5)
      alert("The password length must be bigger than 5
          characters");
   else if (reg.rePassword.value != reg.password.value)
      alert("password!=rePassword");
   else {
      send=true;
   reg.rePassword.value="";
   alert("sto inviando");
   }
   return send;
}
```

Codice A.6: Listato PHP per la registrazione con la divisione nei tre sottoblocchi.

```
<?php
//primo blocco: lettura POST
$name=$_POST['userName']." ".$_POST['userSurname'];
$login=$_POST['login'];
$password=$_POST['password'];
$istitute=$_POST['istitute'];
$address=$_POST['address'];
$motivation=$_POST['why'];
$mail="".$_POST['mail'];
....
//secondo blocco: interrogazione database
$table = 'user';
if (!mysql_connect($db_host, $db_user, $db_pwd))
    die("Can't connect to database");

if (!mysql_select_db($database))
    die("Can't select database");
   $result = mysql_query("SELECT login,mail FROM {$table}
       WHERE login='$login' OR mail='$mail'");
if (!$result) {
    die("Query to show fields from table failed");
}
....
//terzo blocco: mail o torna indietro
```

```
if($row = mysql_fetch_row($result)) {
$message= ($row[0]==$login) ? "login gi&agrave; in uso" : "
    mail gi&agrave;  presente";
mysql_free_result($result);
echo "<script type=\"text/javascript\"> alert('$message');
    history.go(-1)</script>";

}
else if(strlen($name)>1 && strlen($mail)>1 && strlen($login)
    >1){
mysql_free_result($result);
$message= "Domanda di registrazione effettuata,
aspetta la mail di conferma per registrarti.";
$message.= " Domanda di registrazione da ".$name." mail= ".
    $mail." login= ".$login." password= ".$password."dell'
    istituto= ".$istitute." indirizzo= ".$address." motivo= "
    .$motivation;
mail($to,$subject,$message,"From: ".$mail);
}
else{
mysql_free_result($result);
header("location:registration.html");
}
?>
```

Codice A.7: Sintassi CSS.

```
nomeBlocco{
   font-family: nomefont; //indica il carattere
         utilizzato, ed esempio sans-serif
   height (e width): 95%;//altezza e larghezza del blocco
   margin: 0; //per specificare i margini cioè gli spazi
       esterni ad un riquadro
   padding: 0; //per specificare spazio interno di un
       riquadro
   background: url(..) o colore; ///per specificare immagine
        o colore di sfondo
   background-size: 100% 100%; //dimensione dello sfondo
   color:white; //colore del carattere
   font-weight: bold; //spessore del carattere
   text-align:right; //allineamento del testo
   font-size:35pt;//dimensione del carattere
}
```

Codice A.8: Intestazione di collegamento a jQuery.

```
<script type="text/javascript" src="ui/js/jquery-1.7.1.min.
    js">
</script>
<script type="text/javascript" src="ui/js/jquery-ui-1.8.18.
    custom.min.js"></script>
<link rel="stylesheet" href="ui/css/ui-lightness/jquery-ui
    -1.8.18.custom.css" type="text/css" />
```

Codice A.9: Funzioni utilizzate per timepicker: refreshTime e refreshMinSec.

```
function refreshTime (id,p){
   if (document.getElementById(id).value=="")
      document.getElementById(id).value=0;
   t=(24+parseInt(p)+parseInt(document.getElementById(id).
       value))%24;
   document.getElementById(id).value=t;
}
function refreshMinSec (id,p){
   if (document.getElementById(id).value=="")
      document.getElementById(id).value=0;
   m=(60+parseInt(p)+parseInt(document.getElementById(id).
       value))%60;
   document.getElementById(id).value=m;
}
```

Codice A.10: Listato di onlyNumber.

```
function onlyNumber(evt,x,h) {
   var charCode = (evt.which) ? evt.which : event.keyCode ;
   if (charCode>31 && (charCode<48 || charCode>57)) {
      return false;
   }
   if (charCode!=8 && x.value.length>1)
      return false;
   if (h==0 && parseInt(x.value+ String.fromCharCode(
       charCode))>23 || parseInt(x.value+ String.fromCharCode
       (charCode))>59 || parseInt(x.value+charCode)<0)
      return false;
   return true;
}
```

Codice A.11: Query in SQL per misure ambientali e acustiche.

```
SELECT epoch, valueData FROM envMeasure
   where envMeasure.idDevice=dispositivo
   AND envMeasure.type=tipo
   AND epoch between epoch1 AND epoch2
   ORDER BY epoch
```

Codice A.12: Query in SQL per misure acustiche.

```
SELECT epoch, valueData FROM envMeasure
   where acousticMeasure.idTx=dispositivoTrasmittitore
   AND acousticMeasure.idRx=dispositivoRicevitore
   AND envMeasure.type=tipo
   AND epoch between epoch1 AND epoch2
   ORDER BY epoch
```

Codice A.13: Apertura di file tramite PHP.

```
$result = mysql_query(\textdollar{}queryStr);
$csv_filename = "Data-" .$_SESSION['user'].".csv";
$fp = fopen($csv_filename, 'w+');
if ($fp && $result)
while($row = mysql_fetch_row($result))
    fputcsv($fp, array_values($row));
```

Codice A.14: Intestazione per jqPlot.

```
<script type="text/javascript" src="jqplot/dist//plugins/
    jqplot.dateAxisRenderer.min.js"> </script>
<script type="text/javascript" src="jqplot/dist//plugins/
    jqplot.canvasTextRenderer.min.js"> </script>
<script type="text/javascript" src="jqplot/dist//plugins/
    jqplot.canvasAxisTickRenderer.min.js"> </script>
<script type="text/javascript" src="jqplot/dist//plugins/
    jqplot.categoryAxisRenderer.min.js"> </script>
<link rel="stylesheet" type="text/css" href="jqplot/dist/
    jquery.jqplot.min.css"/>
```

Codice A.15: Grafico tramite jqPlot.

```
<script type='text/javascript'>
   $(function() {
   data =matrice,
   $.jqplot('grafico',data.data,{
      axes:{
      xaxis: {
         min: data.min,
         max: data.max,
         renderer: $.jqplot.DateAxisRenderer,
         rendererOptions: { tickRenderer: $.
jqplot.CanvasAxisTickRenderer },
         tickInterval: '$int second',
         tickOptions: {
            formatString: '%d/%m %H.%M.%S' }
           },
      yaxis:{
         label:'$type',
         min:0
      }
      }
   });
   });
</script>
```

Bibliografia

[1] J.G. Proakis and E.M. Sozer, A. Rice, Milica Stojanovic , **Shallow Water Acoustic Networks** , *IEEE Communication Magazine*, November 2001.

[2] Beatrice Tomasi, **The Underwater Acoustic Channel and its Impact on Adaptive Communications Schemes and Networking Protocols**, *scuola di dottorato*, 2011/2012

[3] W.S. Hodgkiss, H.C. Song, G. Deane, **Kauai Acomms MURI 2011 (KAM11) Experiment Trip Report** , , 30 July 2011.

[4] Nevio Benvenuto, Michele Zorzi , **Principles of Communications Networks and Systems** , *Wiley*, Padova 2011.

[5] Edgar F. Codd, **A relational model of Data for Large Shared Data Banks**, *IBM Research Laboratory*, San Jose California 1970

[6] Philip J. Pratt, **Guida a SQL**, *Apogeo*, Milano 2001

[7] David Flanagan, **JavaScript The Definitive Guide**, *O'Reilly*, Fifth Edition, August 2006

[8] C. J. Date, **An Introduction To Database System**, *Addison Wesley*, San Jose California 2000

[9] Giuseppe Callegarin, **Basi di dati e sistemi informativi**, *CEDAM*, 1996

[10] Luke Welling, Laura Thomson, **MySQL Tutorial**, *Pearson Education Italia*, First edition 2004

[11] Mariastella Agostini, Nicola Ferro, **Elementi di basi di dati**, *Libreria Progetto*, Padova 2003

[12] Paul DuBois, **Professionale web MySQL**, *Pearson Addison Wesley*, 2004

[13] Laura Lemay, **HTML**, *McGraw-Hill*, 1997 Milano

[14] Christian Wenz, **JavaScript**, *Pearson Education*, Novembre 2007

[15] Castledine Earle, Sharkle Craing, **JavaScript**, *Apogeo*, MI

[16] Yank Kevin, Adams Cameron, **JQuery**, *Apogeo*, MI

[17] Guide ufficiali di w3c per HTML,JavaScript, jQuery, css, php, **http://www.w3schools.com/**, *w3c*, NY