



# Rapporti tecnici INGV

**Banca dati per la gestione delle reti  
strumentali in dotazione all'INGV -  
Sezione di Catania: dalla progettazione  
concettuale al software applicativo**

# 153



Istituto Nazionale di  
Geofisica e Vulcanologia

## **Direttore**

Enzo Boschi

## **Editorial Board**

Raffaele Azzaro (CT)

Sara Barsotti (PI)

Mario Castellano (NA)

Viviana Castelli (BO)

Rosa Anna Corsaro (CT)

Luigi Cucci (RM1)

Mauro Di Vito (NA)

Marcello Liotta (PA)

Simona Masina (BO)

Mario Mattia (CT)

Nicola Pagliuca (RM1)

Umberto Sciacca (RM1)

Salvatore Stramondo (CNT)

Andrea Tertulliani - Editor in Chief (RM1)

Aldo Winkler (RM2)

Gaetano Zonno (MI)

## **Segreteria di Redazione**

Francesca Di Stefano - coordinatore

Tel. +39 06 51860068

Fax +39 06 36915617

Rossella Celi

Tel. +39 06 51860055

Fax +39 06 36915617

[redazionecen@ingv.it](mailto:redazionecen@ingv.it)



# Rapporti tecnici INGV

## **BANCA DATI PER LA GESTIONE DELLE RETI STRUMENTALI IN DOTAZIONE ALL' INGV - SEZIONE DI CATANIA: DALLA PROGETTAZIONE CONCETTUALE AL SOFTWARE APPLICATIVO**

Marco Aliotta, Placido Montalto, Michele Prestifilippo, Andrea Cannata, Marcello D'Agostino,  
Ferruccio Ferrari, Orazio Torrisi

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania)

# 153



## **Indice**

Introduzione	5
1. Progettazione	5
1.1 Requisiti	5
1.2 Progettazione del database	6
1.3 Entità polimorfiche	6
1.4 Storicizzazione di un nodo	7
1.5 Specializzazione di un nodo	8
1.6 Raggruppamento dei nodi	9
1.7 Lo spazio dei nodi: il Grafo	11
2. Software di gestione	12
2.1 Visualizzazione delle stazioni sulla mappa	15
2.2 Gestione delle anagrafiche	16
3. Origine Dati	18
4. Conclusioni	20
Bibliografia	21



## Introduzione

Le reti di monitoraggio strumentali, in senso lato, servono a conoscere lo stato di determinate grandezze al variare del tempo. In particolare, una rete distribuita su di un territorio è inoltre in grado di fornire misure della stessa grandezza (temperatura ad esempio) in diverse aree geografiche. Al fine di perseguire tale scopo occorre distribuire nel territorio diverse tipologie di sensori. Questo comporta la realizzazione delle infrastrutture atte alla installazione dei sensori e la realizzazione di opportuni mezzi di trasmissione dei dati. La gestione di siffatte reti pone il problema di una corretta archiviazione delle informazioni ad esse connesse. Queste possono essere classificate secondo tre tipologie principali:

- 1) sensori
- 2) equipaggiamento per l'acquisizione e la trasmissione dei segnali
- 3) dato acquisito

Queste tre *classi* permettono di modellare la maggior parte delle entità impiegate nella realizzazione di una rete di monitoraggio costituita, appunto, da sensori, mezzi di trasmissione e di archiviazione dei dati.

Il presente lavoro si pone l'obiettivo di proporre un sistema software in grado di archiviare e gestire le suddette tipologie di informazioni. L'archiviazione di dati in forma strutturata viene perseguita attraverso la progettazione di un database basato sul modello relazionale o RDBMS (Relation Data Base Management System). Tale modello, partendo dal concetto di entità-relazione, prevede la definizione delle diverse entità, ovvero delle classi di oggetti che compongono il sistema da descrivere, degli attributi descrittivi e delle relazioni che intercorrono tra di essi. Mentre la progettazione concettuale richiede la ricerca e la modellazione di tutti gli oggetti che compongono il sistema in esame, la progettazione fisica viene eseguita traducendo il modello entità-relazione in una piattaforma software. Il presente lavoro parte dall'analisi dei requisiti, alla loro implementazione fino ad arrivare alla realizzazione del software per la gestione e la visualizzazione avanzata delle informazioni archiviate.

## 1. Progettazione

### 1.1 Requisiti

In generale, un sistema per l'archiviazione dei dati relativi ad una rete strumentale dovrebbe soddisfare le seguenti esigenze:

- rappresentare una grande varietà di dati eterogenei, mantenendo tali dati logicamente connessi tra di loro;
- mantenere una struttura scalabile al fine di rendere semplice l'aggiunta di altri tipi di dato in qualunque punto della struttura stessa, senza al tempo stesso comprometterne il funzionamento.

Il contributo sperimentale del presente lavoro riguarda la possibilità di rappresentare oggetti polimorfici mediante l'utilizzo di un modello entità relazione e la sua realizzazione fisica mediante RDBMS. Da un punto di vista strettamente concettuale tale sistema può essere rappresentato utilizzando classi di oggetti che di volta in volta vengono specializzati nella loro definizione. Ad esempio, una rete strumentale multi parametrica può essere vista come una struttura dove la misura acquisita transita attraverso nodi volti all'acquisizione, come ad esempio i sensori, alla trasmissione ed infine all'archiviazione. Il requisito fondamentale di una struttura di questo tipo è quello di poter

relazionare tra di loro le entità coinvolte nel trasferimento delle misure eseguite. Per far questo si è ricorsi al concetto di polimorfismo, ovvero un'entità generica, astratta, che si specializza in base all'oggetto che deve rappresentare. Una rete di sensori generalmente ospita diverse entità, tutte definibili come nodi della rete stessa, che differiscono in base alla loro funzionalità. Per esemplificare questo concetto basti pensare ad una rete formata da sensori, digitalizzatori, schede di acquisizione, router, server etc. Tutte queste entità, sebbene di natura diversa, sono tutti nodi della rete interconnessi (relazionati) tra di loro.

In quest'ottica la rete di sensori viene rappresentata mediante una struttura a grafo, avente per nodi gli oggetti presenti nella rete e come archi le relazioni che intercorrono tra essi. Un altro requisito fondamentale da considerare nella realizzazione del sistema è la possibilità di mantenere nel tempo uno storico di tutte le modifiche cui la rete è sottoposta. Questo garantisce di poter avere visione dell'evoluzione della struttura e di poter risalire allo stato della rete in un ben preciso istante temporale.

## 1.2 Progettazione del database

L'obiettivo primario dell'implementazione di una banca dati è l'organizzazione dei dati e le relative modalità d'archiviazione, al fine di rendere il materiale raccolto facilmente fruibile dall'utente. Il primo passo nella progettazione del sistema oggetto del lavoro prevede la scelta del tipo di RDBMS da implementare e l'analisi delle tecnologie da impiegare nella realizzazione del software di interfacciamento verso di esso.

Adottando come criterio selettivo la possibilità di utilizzare un tipo di RDBMS open source con licenza *GNU GPL* [GNU documentation] disponibile per diversi sistemi operativi (Windows, Linux, UNIX, Mac Os, etc...), la scelta è ricaduta sulla versione community di *MySQL 5.0* [MySQL 5.0 reference manual].

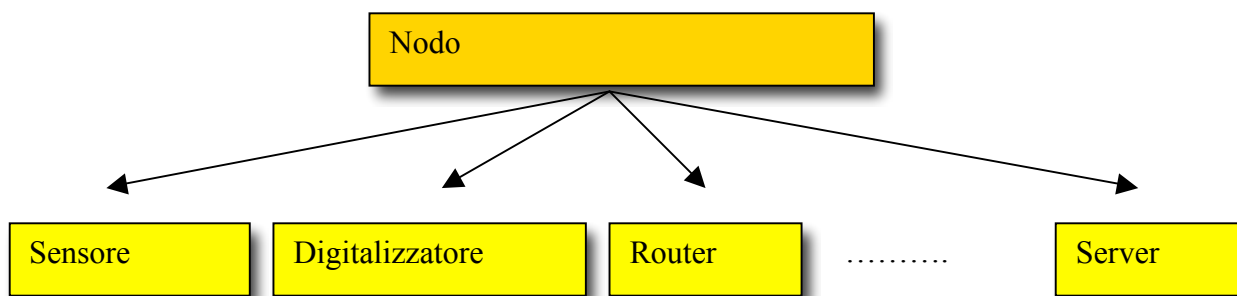
Nella progettazione logica l'entità descrive una classe di oggetti con caratteristiche comuni. Ogni entità è descritta mediante attributi. L'istanza di un'entità è l'oggetto. Ad esempio, definita la classe '*sensore*', un generico sensore può essere descritto mediante vari attributi quali, ad esempio, il numero di canali, la funzione di risposta armonica, il tipo di grandezza misurata etc. L'istanza della classe *sensore* rappresenterà nello spazio del database l'oggetto fisico in esso archiviato. Da un punto di vista operativo la classe è quella che viene fisicamente realizzata come tabella mentre l'oggetto, ovvero la sua istanza, rappresenta i dati in essa contenuti. Il legame tra le diverse entità è descritto dalle *relazioni*: ci sarà ad esempio un legame logico tra l'entità sensore e l'entità digitalizzatore.

## 1.3 Entità polimorfiche

Letteralmente, la parola polimorfismo indica la possibilità per uno stesso oggetto di assumere più forme. Riferendoci ad un sistema software ad oggetti, il polimorfismo indicherà l'attitudine di un oggetto a mostrare più implementazioni per una singola funzionalità.

Allo scopo di ottenere un modello altamente flessibile e generalizzato si è ricorsi a un'entità principale chiamata *Node* (Nodo), progettata in modo da risultare polimorfica (fig. 1). Un *Node* potrà così *impersonare* oggetti di natura diversa. In questo modo ci si potrà sempre riferire semplicemente all'entità *Node*, che assumerà, in maniera del tutto trasparente, "le sembianze" dell'oggetto realmente referenziato. Tutto ciò viene effettuato senza violare il vincolo di integrità referenziale.





**Figura 1.** Specializzazione del nodo polimorfo in entità differenti.

Per la rappresentazione delle reti strumentali mediante un modello relazionale vengono utilizzate 4 categorie principali di Nodi:

- Sensori
- Apparecchiature Informatiche (Equipaggiamenti)
- Software
- Prodotti Finali (come ad esempio il dato elaborato)

Ognuna di queste quattro categorie è realizzata mediante una tabella collegata a sua volta a una o più tabelle contenenti le informazioni relative alla tipologia associata.

In particolare, la tabella *Nodes* conterrà due campi, chiamati rispettivamente *IDSpecialized* e *NodeType*, che riferiranno l'identificativo dell'oggetto "figlio" della sua gerarchia di oggetti e il nome mnemonico della tipologia di oggetto che sta referenziando. La definizione delle categorie permette di associare diverse tipologie di oggetti a ciascuna macro-categoria individuata. Tale operazione viene realizzata mediante l'utilizzo di entità, ovvero tabelle del database, che definiscono le specifiche di una data categoria. Alcune delle entità utilizzate per la realizzazione del database sono riportate di seguito:

- Entità **Equipments**: rappresenta i vari tipi di strumentazione quali routers, pc, parabole e interfacce dati di varia natura;
- Entità **Sensors**: rappresenta i vari tipi di sensori, associati a loro volta ai canali di acquisizione dati;
- Entità **Software**: rappresenta le varie tipologie di software che interagiscono con il database quali, ad esempio, programmi diagnostici, software per l'elaborazione e l'archiviazione del dato;
- Entità **FinalProducts**: rappresenta l'output generato dalle elaborazioni effettuate sui segnali acquisiti.

Inoltre, un nodo avendo un'ubicazione fisica avrà ulteriori attributi, come latitudine e longitudine, utili alla sua rappresentazione su mappa mediante sistemi Geographic Information System (GIS).

#### **1.4 Storicizzazione di un nodo**

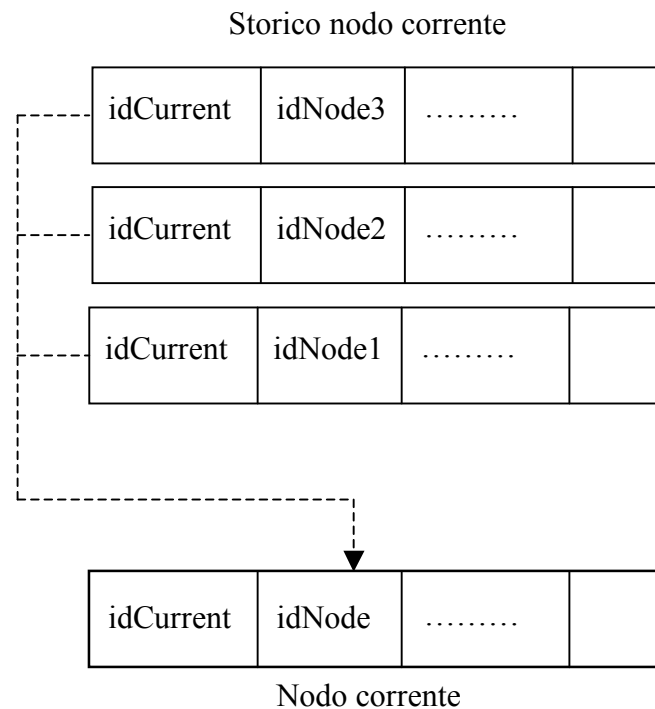
Il database supporta delle funzionalità di storicizzazione per i nodi. Ogni nodo ha un tempo di vita rappresentato da una data di inizio e una data di fine. Come prima accennato, l'esigenza di poter archiviare le modifiche effettuate sul nodo stesso ha reso necessario la progettazione di un sistema in grado di mantenere uno storico. Ad esempio, un qualsiasi tipo di equipaggiamento presente nella rete potrebbe essere aggiornato in base a nuove specifiche. In termini di database questo implica un aggiornamento degli attributi che descrivono il particolare nodo riferito. L'operazione di

storicizzazione del nodo non implica una semplice modifica, ma permette l'archiviazione dei nuovi valori associati ai diversi attributi al momento dell'aggiornamento.

Tale operazione consentirà di tenere traccia di tutte le modifiche apportate. In questo caso, se l'aggiornamento di uno o più attributi deve essere archiviato nello storico del nodo, verrà creata una nuova istanza del nodo etichettata come appartenente allo storico, mentre allo stesso tempo l'istanza modificata verrà etichettata come attiva o *'corrente'*. Per consentire quanto appena descritto, il grafo della rete contiene esclusivamente i nodi etichettati come attivi.

Le funzionalità discusse possono essere utilizzate anche per archiviare versioni di un determinato *Node* su specifica richiesta dell'utente. A tal fine viene utilizzato l'attributo *IDCurrent*, chiave del processo di storicizzazione. Un nodo il cui attributo *IDCurrent* risulta identico al proprio codice identificativo (*IDNode*, che è immutabile) rappresenterà, di fatto, un nodo "attivo". Se invece in un *Node* i campi *IDCurrent* e *IDNode* sono diversi tra loro, *IDCurrent* punterà all'istanza attualmente attiva, mentre l'*IDNode* risulterà semplicemente riferito ad una vecchia istanza del *Node* in oggetto e servirà solo per le funzioni di storicizzazione (fig. 2).

Mediante questa funzionalità il sistema è in grado di mantenere uno storico dei diversi oggetti e delle loro modifiche nel tempo.

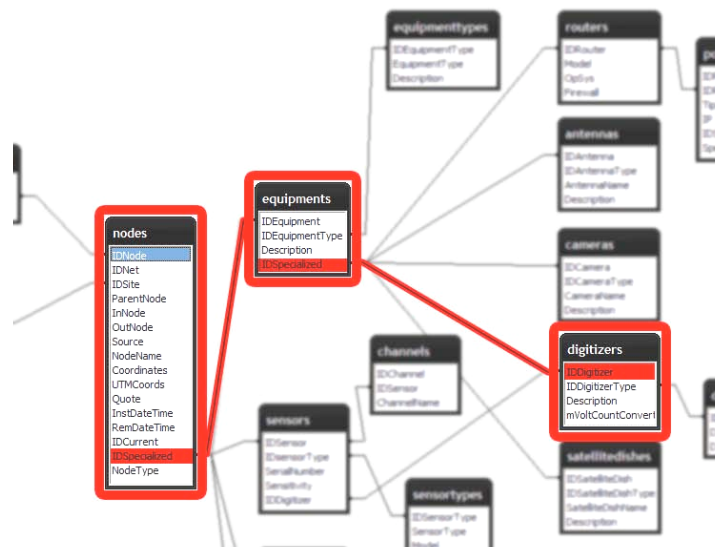


**Figura 2.** Storicizzazione di un nodo.

### 1.5 Specializzazione di un nodo

La totalità delle informazioni relative a una singola istanza di qualsiasi oggetto vengono reperite effettuando una *unione* o 'join' tra i *Nodes* attivi e le tabelle della sua discendenza, relativamente al ramo dove si trova l'oggetto in questione.

A titolo esplicativo viene riportato in fig. 3 l'implementazione del Nodo polimorfico relativo al tipo *Digitizer*.



**Figura 3.** Esempio di specializzazione di un nodo in entità digitalizzatore.

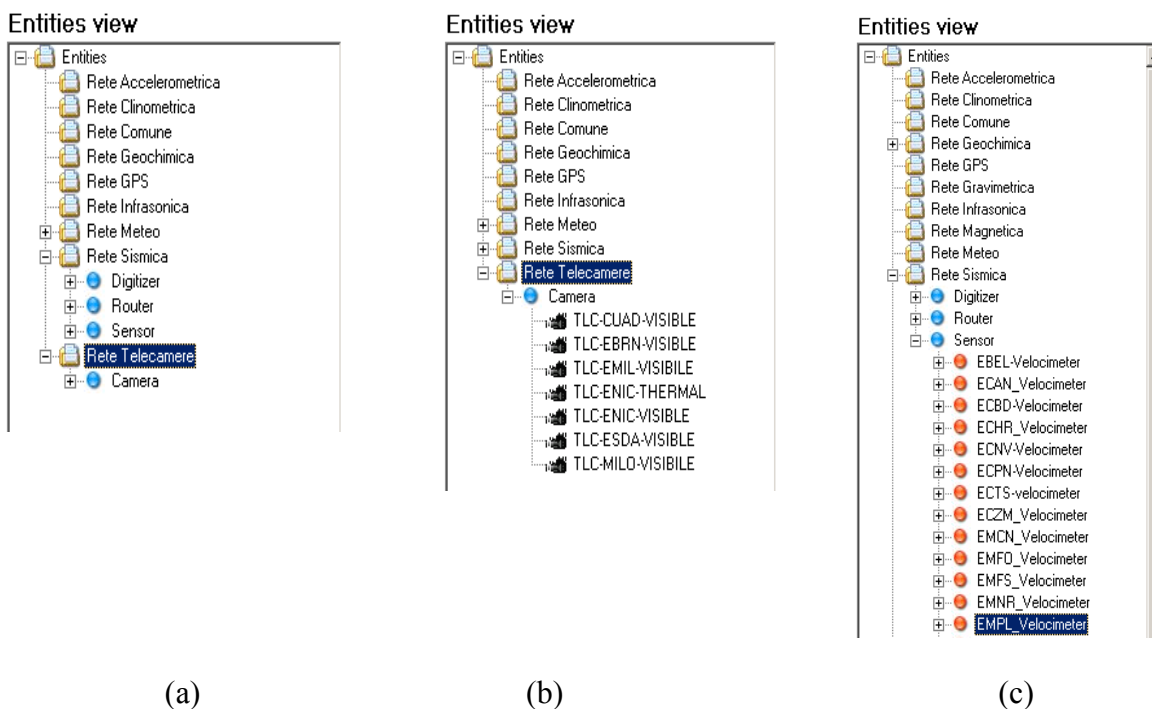
Nello specifico, la specializzazione del nodo nell'oggetto digitalizzatore è ottenuta con una relazione tra le tabelle (entità) nodes-equipments-digitizers. Queste interrogazioni vengono effettuate mediante istruzioni SQL del tipo *Right Join* e *Left Join* [Atzeni et al., 2002] che consentono di estrarre dati da più tabelle. Tali istruzioni, a differenza della *Inner Join* (istruzione usata più di frequente per unire più tabelle in un'interrogazione) ci restituirà un risultato anche nel caso di mancanza di informazione da un sottoinsieme delle tabelle coinvolte nell'interrogazione.

### 1.6 Raggruppamento dei nodi

Al fine di effettuare un raggruppamento tra i nodi vengono definite le seguenti entità:

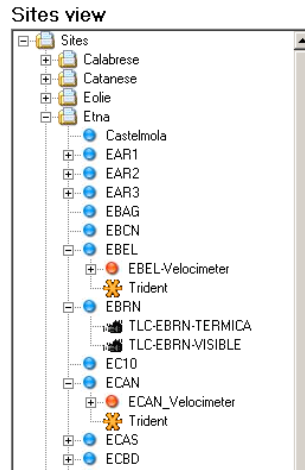
- **Groups:** è l'entità logica che individua i gruppi o "contenitori" di un insieme di apparecchiature, di software e/o di sensori che fanno capo alla stessa area geografico-funzionale. Ogni *Node* fa riferimento a un gruppo tramite il suo attributo *IDGroup*.
- **Adjacency\_List:** è l'entità logica che rappresenta la lista di adiacenza [Cormen et al., 2005] per la definizione del grafo dei nodi e descrive il collegamento logico/fisico che intercorre tra i nodi della rete;
- **Sites:** è l'entità logica che individua le località dove sono installati uno o più oggetti sensors/equipments. Ciascun sito raggruppa un insieme (anche vuoto) di *Nodes*. Tutte le stazioni esistenti potranno essere riunite in un certo numero di siti, in modo da facilitare una moltitudine di interrogazioni. Si noti che l'entità *Site* è l'unica, tra le entità principali, a non essere incapsulata dentro un *Node*. A ciascun oggetto *Site* è associata una localizzazione di massima tramite un campo (*IdSubNet*) che lo associa a una determinata zona geografica. Tali zone sono elencate nella tabella *SubNet*. L'oggetto *Site* qui descritto "contiene" in maniera logica uno o più *Nodes*, modellizzando così un oggetto complesso;
- **Nets:** è l'entità logica che definisce l'appartenenza ad una rete strumentale;
- **Subnets:** è l'entità logica che definisce una sottorete come sottoinsieme dell'entità precedente.

Mediante l'uso delle entità sopra definite è possibile fornire all'utente una rappresentazione visuale del risultato di una interrogazione relativa a siti e sensori mediante una vista ad *albero* dei dati. In figura 4 è riportata la visualizzazione dei nodi mediante interfaccia grafica (vedi paragrafi successivi). In questo esempio l'albero viene costruito raggruppando i nodi per tipo e per rete di appartenenza (entità Nets). La rete sismica contiene nodi di tipo digitalizzatore, sensore e router (fig 4a), mentre la rete delle telecamere contiene sensori video (*camera*) (fig 4b). Un ulteriore esploso dell'albero mostra la visualizzazione dei singoli nodi sensore afferenti alle diverse reti strumentali (fig 4c).



**Figura 4.** Visualizzazione ad albero: raggruppamento per entità.

In figura 5 viene riportato un altro tipo di visualizzazione, in cui i nodi vengono raggruppati per siti (entità *Sites*). In questo caso i siti EBEL ed ECAN contengono velocimetri e digitalizzatori (Trident), mentre il sito EBRN contiene una telecamera in banda visibile ed un'altra in banda termica.



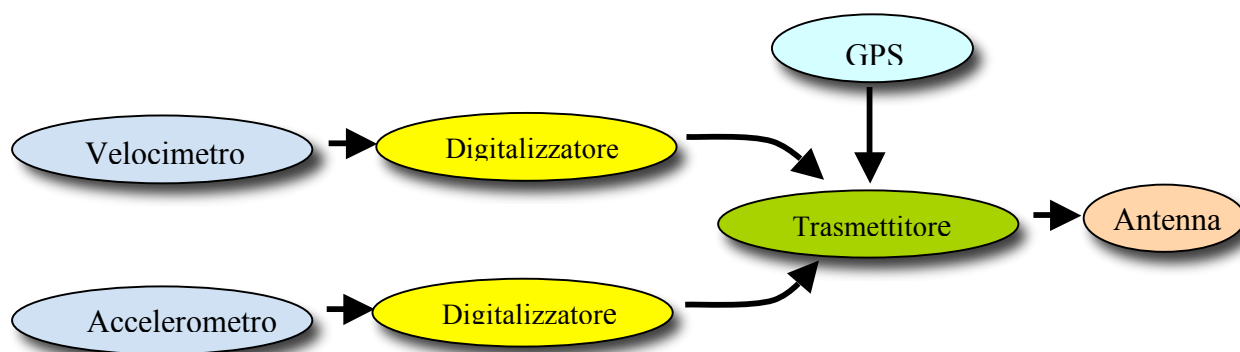
**Figura 5.** Visualizzazione ad albero: raggruppamento per siti.

### 1.7 Lo spazio dei nodi: il Grafo

Un'astrazione della popolazione dei nodi può essere efficacemente rappresentata mediante una struttura a grafo (fig. 6), ovvero tramite un insieme di nodi interconnessi da archi che, nel nostro caso, rappresentano i legami fisico/logici tra gli stessi. In questo modo è possibile applicare algoritmi per la visita del grafo e tener traccia del percorso effettivamente seguito da un segnale. Questo tipo di struttura rende possibile operazioni di diagnostica e visualizzazione.

A tale scopo viene mantenuta nel Database l'entità *Adjacency\_List*, che conterrà tutte le coppie di archi che mettono in relazione i *Nodes* presenti nel grafo. Ciascuna entità *Node* prevede inoltre un attributo *Source*, in modo che sia possibile conoscere la sorgente che ha generato l'informazione che transita in esso, sia che questa rappresenti un segnale o un dato elaborato.

Utilizzando come struttura dati la lista di adiacenza di cui sopra, sarà possibile effettuare delle visite del grafo e calcolare i cammini tra due coppie di nodi connessi tra loro. A tal scopo verrà utilizzato uno degli algoritmi noti in letteratura: in particolare ci si può avvalere dell'algoritmo di Dijkstra [Cormen et al., 2005], che permette di calcolare i cammini minimi in un grafo impostando come parametro di ingresso un nodo di partenza (sorgente). Nel nostro caso, per interrogare lo stato di funzionamento di un nodo o di un cammino che fa riferimento ad esso, basterà utilizzare l'algoritmo di Dijkstra tra l'entità *Node* e il relativo attributo *source*. L'esempio in figura 6 mostra un possibile grafo orientato contenente differenti tipi di nodo. In questo caso è possibile rappresentare il flusso dell'informazione che va dal nodo *Velocimetro 3D* fino al nodo *Antenna*.



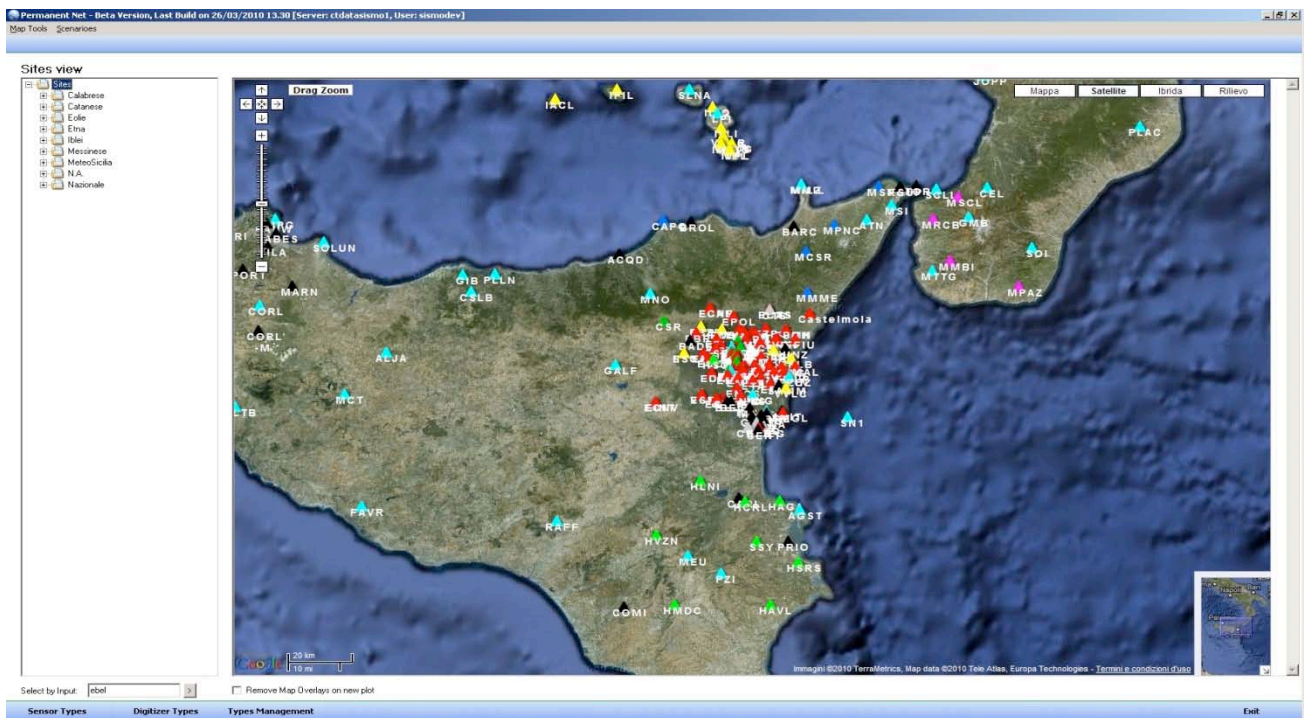
**Figura 6.** Rappresentazione dei nodi mediante un grafo orientato.

L'entità *Node* è concepita in modo da realizzare delle aggregazioni di *Nodes* in un'entità logica più grande, denominata *Macronode*. Un *Node* entra a far parte di un *Macronode* valorizzando il proprio attributo *Parent* con l'identificativo di un altro *Node* con cui esso vuole specificare una relazione di *appartenenza*. Ad esempio, un nodo che rappresenta un particolare tipo di dato può far parte di un macronodo *elaborazione*; un pc può far parte di un macro nodo che rappresenta un *cluster* e così via.

## 2. Software di gestione

Il software creato per la gestione della banca dati proposta, chiamato NetManager, (fig. 7) è stato realizzato mediante l'ambiente di sviluppo Embarcadero RAD Studio 2007 [Rad Studio Documentation]; esso dispone di un'interfaccia di navigazione mediante diagramma ad albero che permette di gestire tutte le entità archiviate. Le funzionalità implementate consentono di effettuare la modifica, l'inserimento e la cancellazione delle istanze delle entità presenti (siti, sensori, canali, equipaggiamenti, prodotti finali, software, etc.). Come descritto nei paragrafi precedenti, i vari nodi della rete possono essere georiferiti diventando così visualizzabili mediante sistemi GIS (Geographic Information System). Tale funzione viene integrata nel software NETMANAGER tramite l'utilizzo di un componente che costituisce un *wrapper* all'applicazione Google Maps (Montalto et al., 2009).

Per poter interrogare il database è necessario stabilire una connessione tra applicativo NetManager ed il server contenente la banca dati su cui è attivo l'RDBMS MySQL. La connessione avviene automaticamente all'avvio del programma a partire da parametri specificati su un opportuno file di configurazione XML. Una volta che la connessione è andata a buon fine, il pannello posto nella parte sinistra della finestra principale del programma (Figura 7) mostra una visualizzazione ad albero dei siti (chiamata "Sites View"), raggruppati in base all'area di appartenenza (Etna, Iblei, Eolie, ecc..). Sotto tale pannello, una 'search box' consente di individuare velocemente un elemento di tale struttura, digitandone parte del nome. E' possibile cambiare la modalità di visualizzazione della struttura ad albero in modo da rappresentare le entità raggruppandole per categoria di utilizzo (tasto destro del mouse sul pannello, e selezionare la voce di menu "Change View"). Eseguendo nuovamente questa azione si ritorna alla rappresentazione iniziale. Le viste ad albero possibili per la visualizzazione dei dati sono quelle discusse nel paragrafo 1.6 (fig. 4 e fig. 5).

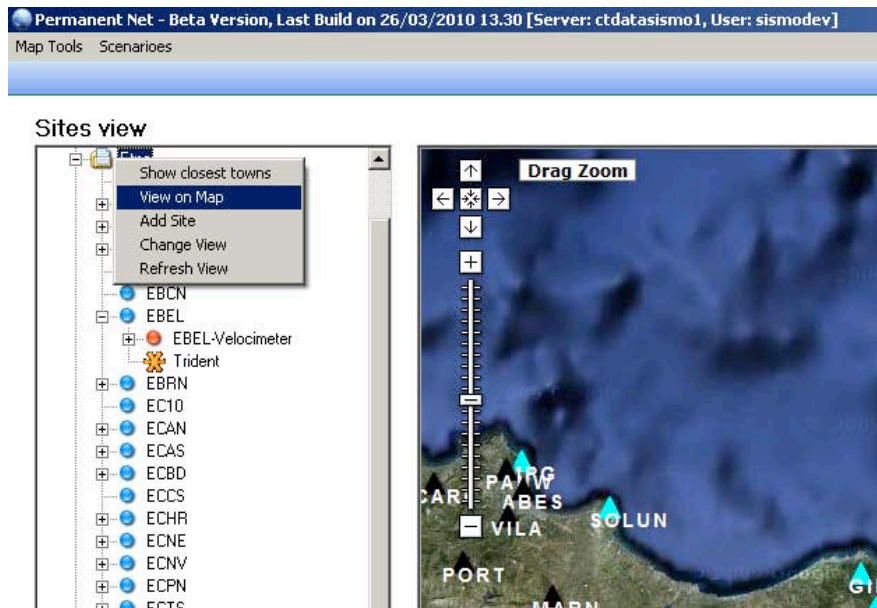


**Figura 7.** Schermata principale del software di gestione *NetManager*.

Il modulo Google Maps è stato concepito al fine di garantire un livello di interazione bidirezionale. Infatti esso aggiorna la propria vista in funzione delle scelte compiute dall'utente all'interno dell'interfaccia di controllo (Sites view e relativi sottomenu, dialogs di configurazione etc.). Allo stesso tempo l'interazione con la mappa di Google permette di far transitare le informazioni selezionate dalla mappa verso il programma di gestione. Questa modalità garantisce una interoperabilità tra il software gestionale della banca dati e il sistema GIS Google Maps.

Espandendo e comprimendo i rami della struttura ad albero è possibile navigare tra i vari elementi che la compongono. La pressione del tasto destro sui vari elementi produce ogni volta la comparsa di un diverso menu contestuale da cui è possibile scegliere (in base all'elemento selezionato) alcune tra le seguenti funzionalità:

- Voci disponibili su qualsiasi tipo di elemento (fig. 8):
- “Change View”: alterna la visualizzazione *Sites / Entities*;
- “Refresh View”: aggiorna la struttura ad albero mediante una nuova interrogazione al database;
- “View on Map”: visualizza sulla mappa l'elemento / il gruppo di elementi selezionato;



**Figura 8.** Voci del menu contestuale valide globalmente.

- Voci disponibili solo su elementi singoli (fig. 9):

“Set as Map Center”: imposta l’elemento come centro della mappa “evidenziandolo” con un rombo rosso;

“Show Closest Towns”: visualizza gli  $n$  toponimi (con  $n$  specificato all’interno del file di configurazione) più vicini all’elemento selezionato, ordinati in base alla distanza dall’elemento corrente, dal più vicino al più lontano. Da quel momento tale elenco rimane disponibile nella struttura ad albero, fino a che non viene eseguito un *Change View* o un *Refresh View*.

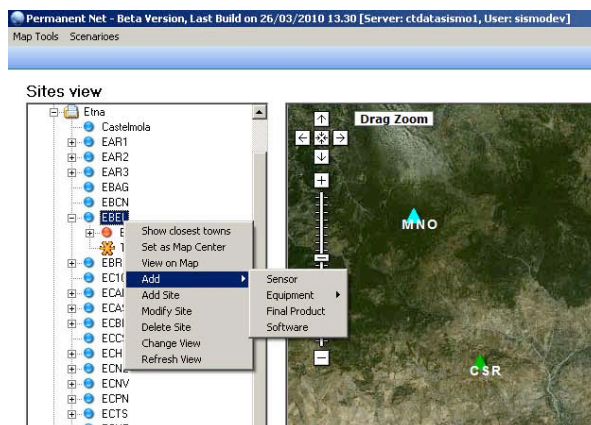
- Voci disponibili solo su elementi di tipo *Site* (fig. 9):

“Add Site”: permette di aggiungere un nuovo sito all’area correntemente selezionata;

“Modify Site”: permette di modificare i parametri generali del sito selezionato;

“Delete Site”: permette di eliminare il sito selezionato e tutti i Nodi in esso contenuti;

“Add”: mostra un sottomenu da cui è possibile aggiungere (e associare) al Sito uno dei 4 tipi di Nodi (*Sensor*, *Equipment*, *Final Product* o *Software*)



**Figura 9.** Voci del menu contestuali valide per singole entità (Sito, nodo, etc.).

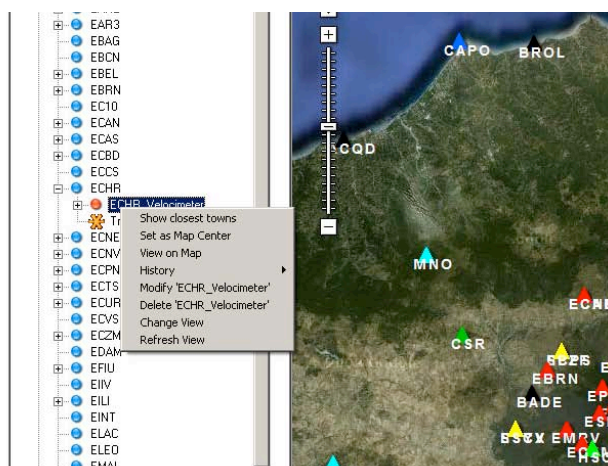


- Voci disponibili solo su elementi di tipo *Node* (fig. 10):

“Modify <NodeName>”: permette di modificare i parametri generali del nodo selezionato;

“Delete <NodeName>”: permette di eliminare il nodo selezionato;

“History”: mostra un sottomenu da cui è possibile aggiungere e/o visualizzare lo storico associato al Nodo



**Figura 10.** Voci del menu contestuali relative agli elementi di tipo *Node*.

## 2.1 Visualizzazione delle stazioni sulla mappa

Il software applicativo NETMANAGER, sfruttando le funzionalità messe a disposizione da *Google Maps* consente di localizzare e visualizzare le stazioni su mappa (fig. 11). Tali funzionalità sono state incapsulate in un componente software, integrato al programma, in modo da poter visualizzare le informazioni strutturate in un file XML o generate da una query al database. Tale componente è una generalizzazione di quello fornito insieme al software per la gestione della Rete Mobile (Montalto et al, 2009). L'idea che sta dietro all'implementazione del componente è quella di sfruttare la tecnologia ActiveX di Microsoft. Tale tecnologia permette agli sviluppatori di estendere le proprie applicazioni con le funzionalità proprie di Internet Explorer (o anche di altri browser, ad esempio *Mozilla Firefox*). In questo modo si può inserire all'interno delle proprie applicazioni un oggetto con le funzionalità di una pagina web, ottenendo così un'interazione bidirezionale con l'applicativo. La suddetta interazione è garantita dalla possibilità di utilizzare del codice javascript *inline* (javascript injection).

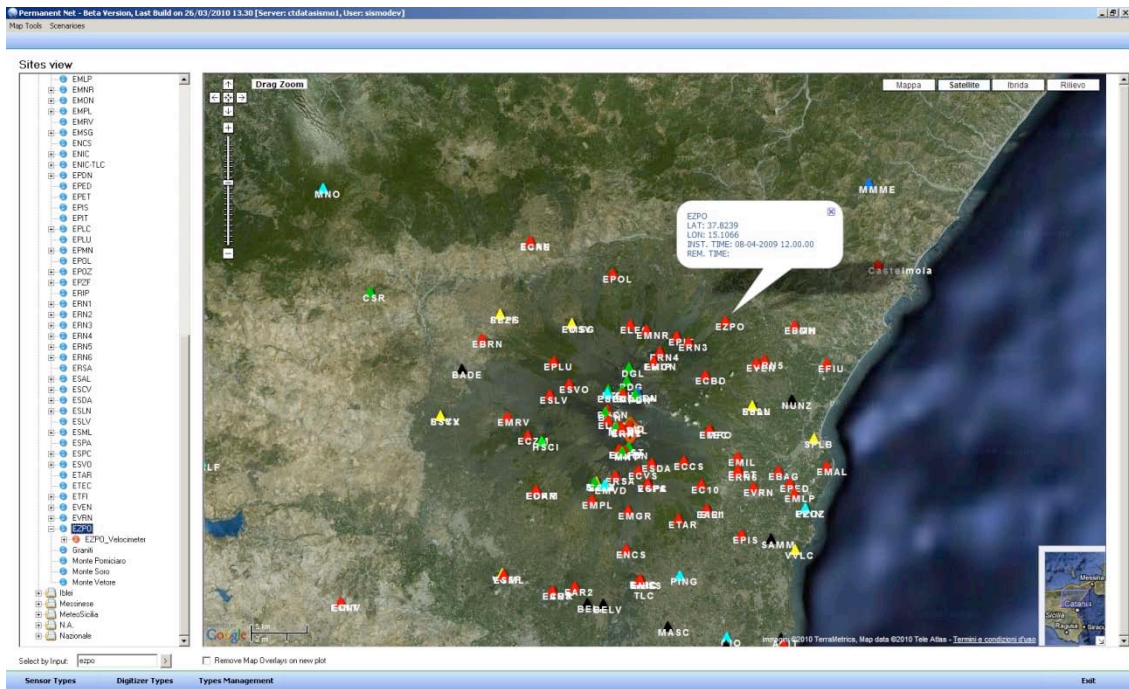


Figura 11. Visualizzazione delle stazioni elencate nell'albero di navigazione sulla mappa.

## 2.2 Gestione delle anagrafiche

L'applicativo NETMANAGER contiene tutte le funzionalità tipiche di un software gestionale: consente pertanto l'inserimento, la modifica e l'eliminazione di tutte le entità gestite. Tali operazioni vengono effettuate mediante le maschere di anagrafica. A titolo di esempio, in figura 12 è riportato la *form* per l'inserimento dei tipi di sensore:

Sensor Types Management

Sensor Type list

Type	Description	Model	HC...	Dynamic Ran.	Digitizer Is
accelerometer	accelerometro	N.A.	1		<input checked="" type="checkbox"/>
velocimeter	velocimetro	N.A.	1		<input checked="" type="checkbox"/>
geochemical	Sensore geochim.	N.A.	1		<input checked="" type="checkbox"/>
magnetic_scalar	Sensore magneti	N.A.	1		<input checked="" type="checkbox"/>
magnetic_vector	Sensore magneti	N.A.	1		<input checked="" type="checkbox"/>

Sensor Type Data

Sensor type:  Model:

Description:

N. of Components:   Digitizer is needed

New Save Undo Close

Figura 12. Anagrafica dei tipi sensore.

Tale *form* permette di inserire la tipologia di strumento (accelerometro, velocimetro, etc.), il modello e la descrizione. Analoghe funzionalità sono previste per tutte le altre entità descritte nei paragrafi precedenti. Più dettagliate invece risultano essere le anagrafiche relative alle entità sensore. In figura 13 è riportata la maschera di inserimento di un generico sensore. Come descritto, l'inserimento del sensore si attiva dal menu contestuale relativo ad un sito (fig. 9). Il *form* di inserimento del sensore (fig. 13a) permette di inserire i seguenti campi:

*Name*: nome del sensore;

*Net*: rete strumentale a cui associare il sensore;

*Sensor Type*: tipo di sensore selezionabile dai tipi caricati mediante l'anagrafica *carica tipo* riportata in figura 13;

*Digitizer*: (eventuale) digitalizzatore associato al sensore;

*Number of components*: numero di componenti del sensore. Ad esempio, un accelerometro triassiale avrà tre componenti. Nella tabella vanno specificati il nome di ciascun canale e, se esiste, la sorgente dati (vedi paragrafo successivo);

*Sensitivity*: sensibilità del sensore;

*Serial Number*: numero seriale del sensore;

*Coordinates*: ubicazione del sensore (esprimibile dall'utente in lat/long oppure in UTM. Il programma si occuperà di convertire da un tipo di coordinata all'altra, in modo da memorizzare entrambe le tipologie);

*Time Period*: data di installazione (ed eventuale dismissione, nel caso di nodo non corrente);

*Connections*: Questa scheda, mostrata più in dettaglio in figura 13b, permette di connettere il nodo sensore ad altri nodi.

(a)

(b)

**Figura 13.** a) Form per il caricamento dei sensori; b) dettaglio della connessione di un nodo con il relativo *parent* e *source*.

Una volta inserito il sensore ed i relativi canali è possibile visualizzare, sempre mediante l'albero di navigazione, l'associazione sensore-canale. Considerando ad esempio il sito EMSG (fig. 14), questo mostra l'installazione di un velocimetro a tre componenti e un digitalizzatore Trident (altre entità sono in loco presenti ma non ancora inserite nella banca dati). Come si vede dalla figura 14, la gestione di un sito è del tutto generica e permette l'inserimento di  $n$  entità. In questo caso vediamo installati nel sito ENIC (fig. 14) un velocimetro, un digitalizzatore ed un sensore geochimico.

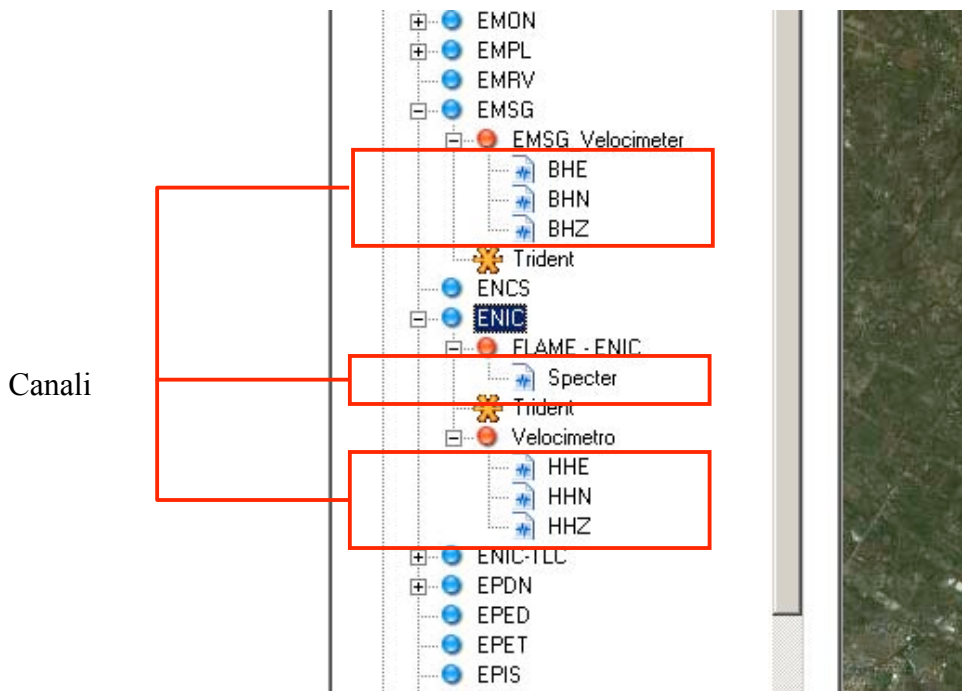


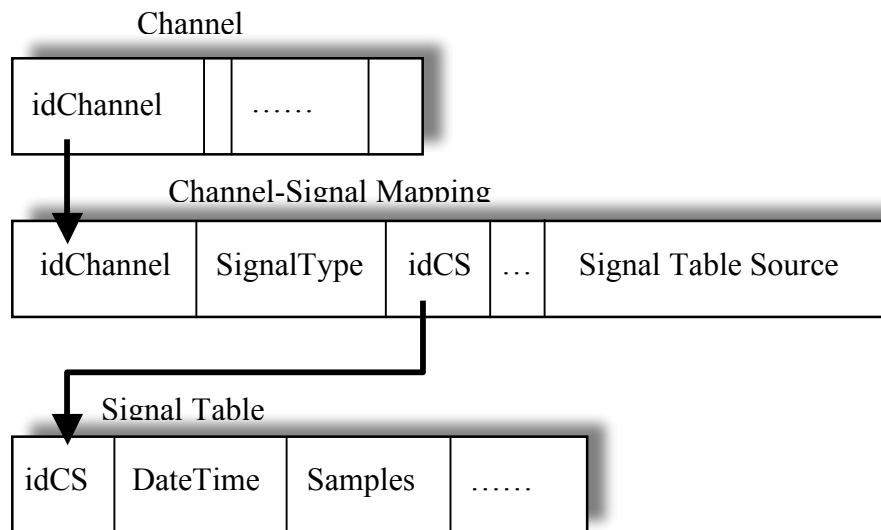
Figura 14. Visualizzazione ad albero *Sito-Sensore-Canale*.

### 3. Origine Dati

Il software NETMANAGER permette di associare ai singoli canali eventuali origini dati (fig. 15). Ciò è reso possibile mediante l'implementazione di un sistema di *plug-in* (ancora in versione preliminare), ovvero di moduli aggiuntivi del programma che possono essere di volta in volta progettati e integrati nell'applicativo.

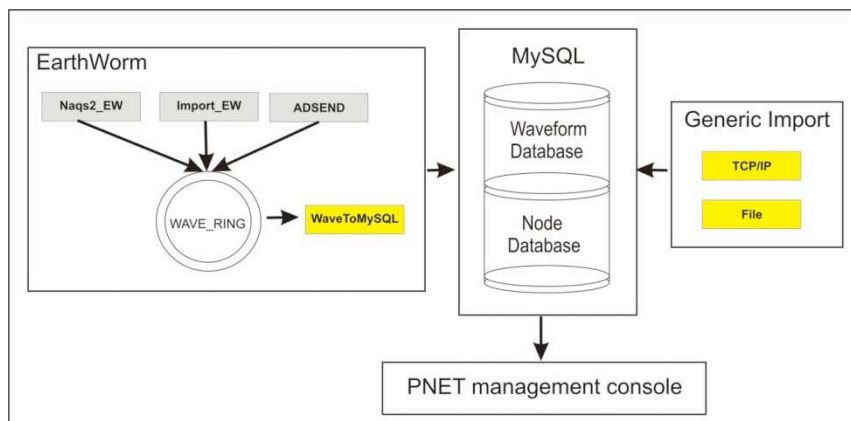
Sebbene la struttura della parte di banca dati relativa ai segnali non è argomento del presente lavoro, ci si riferirà genericamente al *waveform database*, ovvero una banca dati contenente i segnali acquisiti mediante i sensori della rete di monitoraggio le cui specifiche sono inserite nel sistema proposto.

Moduli differenti permettono la connessione tra NETMANAGER e diverse origini dati. Uno dei modi per implementare tali funzionalità è definire uno standard per l'archiviazione mediante RDBMS dei segnali relativi ai singoli canali. In questo modo ogni canale farà riferimento ad una o più tabelle contenenti i dati ed i segnali di pertinenza (fig. 16). Dal punto di vista del modello entità-relazione, la connessione tra canale e relativa tabella delle misure è realizzata mediante una terza entità chiamata *Channel-Signal Mapping*. Questa fornirà le informazioni sul tipo di misura associata al canale, il nome della tabella in cui vengono archiviati i dati, la frequenza di campionamento etc.



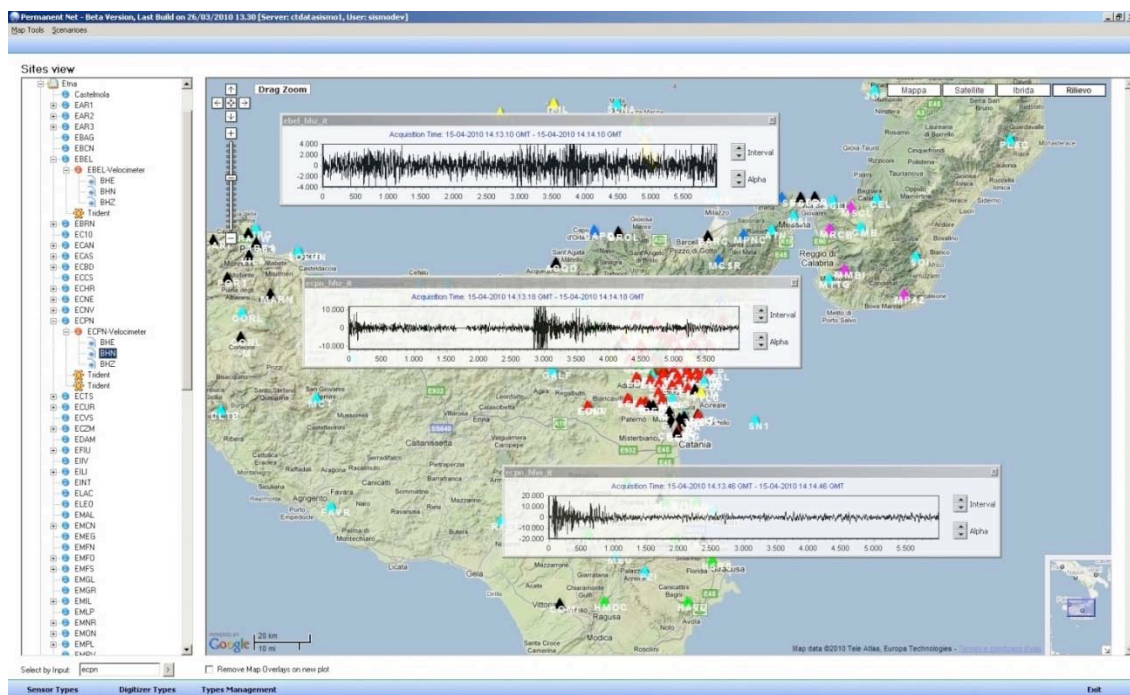
**Figura 15.** Connessione tra canale e segnale.

La funzionalità implementata al momento nel software NETMANAGER consente la visualizzazione delle forme d'onda dei segnali sismici acquisiti mediante il sistema Earthworm [Earthworm Documentation]. Ciò avviene mediante un apposito modulo che permette lo storage dei pacchetti contenuti nel WAVE RING direttamente nel Waveform Database (fig. 16). Ogni tabella dei segnali del waveform database viene collegata al database delle stazioni mediante l'entità *Channel-Signal Mapping* di cui sopra.



**Figura 16.** Origine dati.

Un'altra fonte dati è fornita dal modulo Generic Import (fig. 16), un software opportunamente implementato che consente di importare nel database segnali direttamente da file (SAC, Seed etc..) oppure via TCP/IP. I suddetti moduli sono tuttora in via di sviluppo ed è attualmente in test il collegamento tra la banca dati oggetto di questo lavoro (Node Database) e il database delle forme d'onda acquisite. Un esempio di visualizzazione *real time* delle forme d'onda relative a tre diversi canali è riportata in figura 17.



**Figura 17.** Visualizzazione dei segnali relativi ai diversi canali associati ai velocimetri (ECPN e EPDN).

## Conclusioni

Il sistema proposto permette l'archiviazione e la gestione di tutte le informazioni relative a una o più reti strumentali. L'architettura si basa sulla definizione di una entità astratta *Node* che può modellare i differenti oggetti presenti nel sistema. Questa implementazione consente di essere gestita come una struttura dati di tipo grafo, consentendo di modellare le interazioni che ci sono tra i diversi oggetti ad essa afferenti.

Oltre al progetto e alla realizzazione della banca dati mediante RDBMS MySQL, è stata sviluppata una interfaccia grafica per l'inserimento, la modifica e la visualizzazione delle informazioni. Queste funzionalità sono implementate all'interno del software NetManager il quale consente anche una rappresentazione visuale gerarchica dei diversi oggetti facenti parte del sistema. Il programma è provvisto di un modulo Google Maps per la visualizzazione spaziale delle informazioni georiferite presenti nel database. Altre funzionalità includono l'interfacciamento con diverse origini dati, come ad esempio il sistema Earthworm nel caso dei segnali sismici.

Il sistema sviluppato in questo lavoro, basandosi su un paradigma ad oggetti, permette la modellazione di qualsiasi tipo di entità pertinente ai nostri scopi. Ad esempio è possibile considerare come entità i software di elaborazione e considerare questi ultimi come nodi della rete. In questo modo è possibile seguire il flusso dei dati dalla loro acquisizione alla loro elaborazione.

Gli sviluppi futuri riguarderanno la stesura di diversi moduli aggiuntivi quali, ad esempio, organi di visualizzazione dello stato dei diversi nodi, visualizzazione dei dati acquisiti dai nodi sensore e sistemi per l'aggiornamento automatico del software.

## **Bibliografia**

Atzeni, P., Ceri S., Paraboschi S., Torlone R., (2002). Basi di dati. McGraw-Hill.

Cormen, T. H., Leiserson C. E., Rivest R.L., Stein C., (2005). Introduzione agli algoritmi e strutture dati. McGraw-Hill.

Earthworm Documentation (<http://folkworm.ceri.memphis.edu/ew-doc/>)

Free Pascal Documentation (<http://www.freepascal.org/docs.var>)

Rad Studio Documentation ([http://docs.embarcadero.com/products/rad\\_studio/](http://docs.embarcadero.com/products/rad_studio/))

Montalto P., Aliotta M., Zuccarello L., Rapisarda S., Cannata A., Messina A., 2009. Software applicativo per la gestione delle stazioni della rete sismica mobile in dotazione all'Istituto Nazionale di Geofisica e Vulcanologia – Sezione di Catania. Rapporti tecnici INGV, 114.

MySQL 5 Reference Manual (<http://dev.mysql.com/doc/#refman>).





**Coordinamento editoriale e impaginazione**

Centro Editoriale Nazionale | INGV

**Progetto grafico e redazionale**

Laboratorio Grafica e Immagini | INGV Roma

© 2010 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

**<http://www.ingv.it>**



**Istituto Nazionale di Geofisica e Vulcanologia**