Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Progettazione concettuale e logica di un data warehouse per dati genomici

M. R. Guarracino – S. Cuciniello

RT-ICAR-NA-2006-16 **09-2006**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: napoli@icar.cnr.it, URL: *www.na.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Progettazione concettuale e logica di un data warehouse per dati genomici

M.R.Guarracino[1] – S. Cuciniello[1]

Rapporto Tecnico N.: ***Data:***
RT-ICAR-NA-2006-16 **09-2 006**

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Napoli, Via P. Castellino 111, 80131 Napoli

Progettazione concettuale e logica di un data warehouse per dati genomici

M.R. Guarracino[1], Salvatore Cuciniello[1]

[1] Istituto di Calcolo e Reti ad Alte Prestazioni,
Consiglio Nazionale delle Ricerche, Italia

Sommario

Le tecniche di data warehousing sono utilizzate come sistema di supporto delle decisioni. Esse sono ampiamente utilizzate nelle realtà aziendali, dove si sono riscossi molti vantaggi e gli stessi benefici si possono ottenere nel campo della biomedicina se si realizza un *buon* data warehouse. In tale documento è spiegata l'implementazione di un data warehouse per dati genomici ed un suo possibile utilizzo. In lettaratura già esistono diverse prosposte e quella presentata nelle prossime pagine è confronata con quelle esistenti.

1 Introduzione

Una situazione comune in molteplici realtà aziendali prevede un sistema informativo con una grande mole di dati che spesso sono ridondanti, inconsistenti e disomogenei. Uno scenario tipico è quello di una grossa impresa con molte filiali, dove ognuna ha una propria banca dati in cui le informazioni contenute non riescono ad integrarsi facilmente con quelle memorizzate nelle banche dati delle restanti filiali. Dagli anni '80 il ruolo delle base di dati è cambiato, poiché non si ha solo l'esigenza di memorizzare dati, ma anche di poter effettuare analisi e valutazioni finalizzate alla pianificazione e al processo decisionale [12]. La grande quantità di dati delle aziende, però, rende difficile l'estrazione delle informazioni dai dati, poiché ciò significa eseguire complesse interrogazioni che portano ad un elevato utilizzo di risorse e tempo.

I biologi pubblicando le prime sequenze nucleotidiche si sono resi conto che hanno bisogno di un raccoglitore per memorizzare i dati e di strumenti efficaci

per estrarre in modo semplice e veloce le informazioni. Per archiviare i dati biologici sono state realizzate diverse banche di dati e quelle più conosciute sono: EMBL, GenBank e DDBJ.

Queste ultime sono ampiamente spiegate in [10], mentre in figura 1 è riportata solo il risultato di una *entry* per far capire al lettore come è difficile in questo caso poter estrarre delle informazioni in corrispondenza di un insieme di dati.

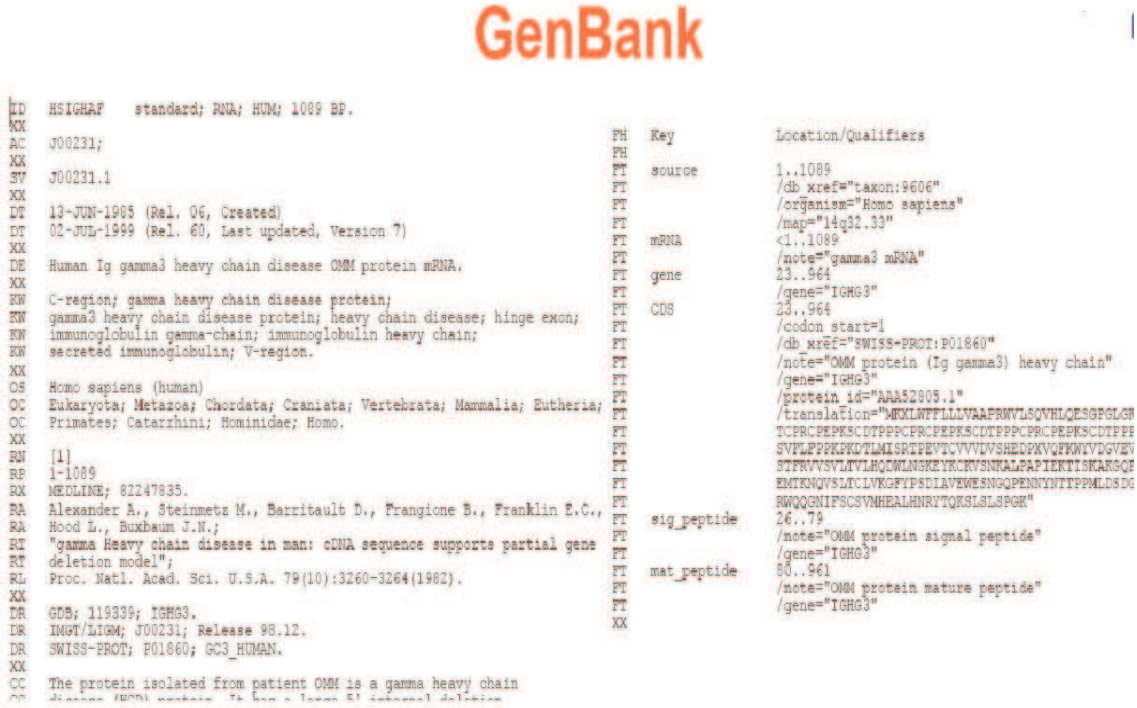

Figura 1: Esempio di entry della banca dati EMBL

Spesso avere a disposizione un *dato* non significa poter *modificare* la conoscenza dell'utente e questo può esserne considerato un *caso*.

Le diverse banche dati sono integrate tra loro tramite i cosiddetti *cross-referencing*: in corrispondenza di una *entry* di una banca dati esiste un riferimento ad un'entry di un'altra. Questa semplice integrazione prevede due grossi inconvenienti: la **ridondanza** e l'**inconsistenza**. Infatti può capitare che una banca di dati può avere dei dati diversi da quella che si può accedere tramite l'utilizzo del *cross-referencing*. SRS (Sequence Retrival System)[8] ed

Entrez [7] sono due esempi si sistemi che utilizzano il meccanismo del *cross-referencing* per consentire l'integrazione dei dati; anche se in realtà si può dire che tale metodo permette solamente di navigare tra le diverse banche dati e non una vera e propria integrazione.

Per permettere l'**integrazione** e l'**estrazione** delle informazioni per effettuare analisi si è deciso di progettare un data warehouse per dati genomici. Il ***data warehouse*** (letteralmente, magazzino di dati) è una base di dati che non sostituisce quelle esistenti , ma è costruita appositamente per integrare i dati provenienti dai diversi sistemi informativi esistenti. Il data warehousing è il sistema di supporto delle decisioni (Decision Support Systems - DSS) su cui si è maggiormente focalizzata l'attenzione negli ultimi anni. Esso permette di separare l'elaborazione di tipo analitico (OLAP- On-Line Analytical Processing) da quella legata alle transazioni (OLTP- On-Line Transactional Processing), costruendo un nuovo raccoglitore d'informazioni (il data warehouse) che integra i dati elementari provenienti da sorgenti di varia natura, li organizza in una forma appropriata e li rende quindi disponibili per l'analisi e la valutazione. Non sono state utilizzate le basi di dati tradizionali perché esse non si adattano bene a complesse procedure d'analisi. Il data warehouse, invece, è una base di dati per le analisi multidimensionali ed è definito dall'ideatore Inmon [11] come raccolta di dati integrata e permanente, ma focalizzata su un argomento e variabile nel tempo, che può fornire supporto alle decisioni. L'utilizzo dei data warehouse quale strumento di supporto delle decisioni non è stato effettivamente implementato per dati genomici, ma per analisi demografiche è diffuso sia in Italia, sia in tutto il mondo.
L'ISTAT raccoglie periodicamente i dati dei Comuni italiani per alimentare il proprio data warehouse [6]. I dati vengono elaborati e resi pubblici con un livello di aggregazione alle province, tramite il loro sito web. Altre esperienze riguardano il Comune di Modena [4] e la Regione Marche [5], in cui è possibile consultare i dati relativi all'intera regione. A livello internazionale esempi notevoli sono quelli del Department of Health and Human Services [2] degli Stati Uniti, che fornisce, tra gli altri, dati di carattere demografico, e del Population Division delle Nazioni Unite [3].

Fino ad oggi non è stato ancora realizzato un data warehouse vero e proprio per dati genomici, anche se in letteratura è spiegato la necessità di farlo e sono proposte diverse soluzioni.

Il documento è organizzato come segue. Nella sezione 2, è definto con precisione cosa è un data warehouse e le architetture proposte in letteratura. Nella sezione 3 è fatta una descrizione del contesto e nella sezione 4 è definito

lo schema concettuale e logico di un data mart. Nella sezione 5 sono fatte le conclusioni ed un possibile lavoro futuro.

2 Il Data Warehouse

Il data warehouse è un repository di dati utilizzato per ottenere informazioni di sintesi in un tempo ridotto ed è stato definito da Inmon (2002), nel seguente modo:

Data Warehouse 1 *Un Data Warehouse (DW) è una collezione di dati di supporto per il prodesso decisionale che presenta le seguenti caratteristiche:*

- *è orientata ai soggetti di interesse;*
- *è integrata e consistente;*
- *è rappresentativa dell'evoluazione temporale:*

Per *orientata ai soggetti* d'interesse s'intende che il data warehouse considera i dati d'interesse dell'azienda e non quelli concernenti i processi organizzativi.
Il data warehouse è *integrato* e *consistente* perché i dati provenienti da sorgenti informative eterogenee sono riconciliati eliminando tutte le disparità.
Il data warehouse è *rappresentativo dell'evoluzione dei dati* perché memorizza non solo informazioni recenti ma dati storici per eseguire confronti, previsioni ed individuare tendenze.

Il data warehousing, come è stato già specificato, è un insieme di metodologie, strumenti e dati tramite cui è possibile effettuare delle analisi. Le architetture che sono proposte in letteratura per il data warehousing sono ad uno, a due e a tre livelli.

2.1 L'architettura ad un livello

L'architettura ad un livello prevede che il data warehouse sia un database virtuale, ovvero costituito da viste che saranno costruite tramite uno strato d'elaborazione intermedio, chiamato *middleware*. Tale tipo d'architettura evita il problema della ridondanza dei dati, ma comporta che le transazioni analitiche e quelle transazionali siano inoltrate sulla stessa base di dati. Per eseguire l'elaborazione dei dati, dal punto di vista analitico, il middleware effettua interrogazioni sui dati operazionali. Questo va contro l'idea di base

di un data warehousing definita da Kimball [14], che prevede di separare le operazioni OLAP da quelle OLTP. Tale tipo d'architettura è utilizzato nel caso in cui non si hanno particolari esigenze d'analisi e rappresenta la sua formulazione più semplice.

Figura 2: Architettura ad un livello

2.2 L'architettura a due livelli

L'architettura a due livelli è così definita proprio per evidenziare che esistono due insiemi di dati: il livello sorgente (l'insieme dei dati operazionali dell'azienda o esterni all'azienda) e il livello del data warehouse. In questo caso, a differenza dell'architettura ad un livello, il data warehouse è fisicizzato. La ridondanza di dati permette di separare le operazioni d'analisi da quelle transazionali (requisito fondamentale del data warehousing). Come si può osservare dalla , oltre ai due livelli appena definiti esiste anche il livello d'alimentazione e quello d'analisi. Il livello d'alimentazione, più propriamente detto ETL (Extraction-Trasformation-Loading), prevede l'estrazione e la pulizia dei dati dal livello sorgente, la trasformazione ed il caricamento all'interno del data warehouse. Il livello d'analisi include l'insieme degli strumenti che permettono di effettuare operazioni di reportistica, OLAP e data mining. Inoltre al livello del warehouse, oltre ad esserci il data warehouse finora definito, sono presenti anche i cosiddetti data mart. Il cilindro etichettato in Figura 3 con il nome data warehouse è di solito chiamato data warehouse primario o data warehouse aziendale mentre i data mart sono

definiti data warehouse locali. Un data mart è definito in letteratura [15] nel seguente modo:

Data Mart. *Si intende un sottoinsieme o un'aggregazione dei dati presenti nel data warehouse primario, contiene l'insieme delle informazioni rilevanti per una particolare area di business, una particolare divisione dell'azienda, una particolare categoria di soggetti.*

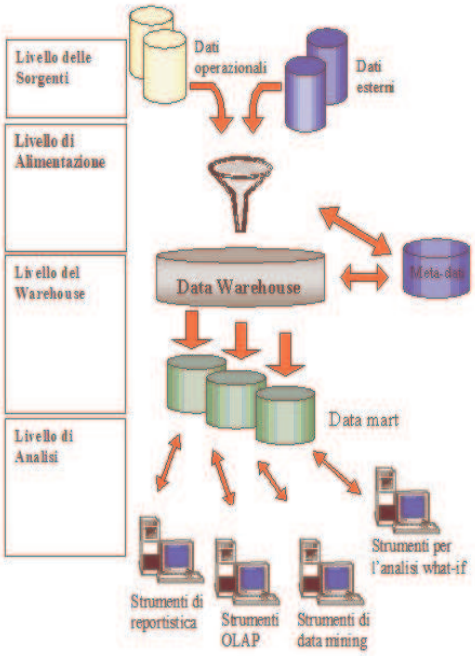

Figura 3: Architettura a due livelli

2.3 L'architettura a tre livelli

L'architettura a tre livelli, come illustrato nella Figura 4, mostra la presenza di dati riconciliati. In questo modo i dati prima di essere caricati all'interno del data warehouse sono integrati e trasformati. Nell'architettura a due livelli l'operazione d'integrazione, anche se non è implementata a livello fisico, è rappresentata a livello logico, poiché è necessario avere una versione integra dei dati prima di inserirli nel data warehouse.

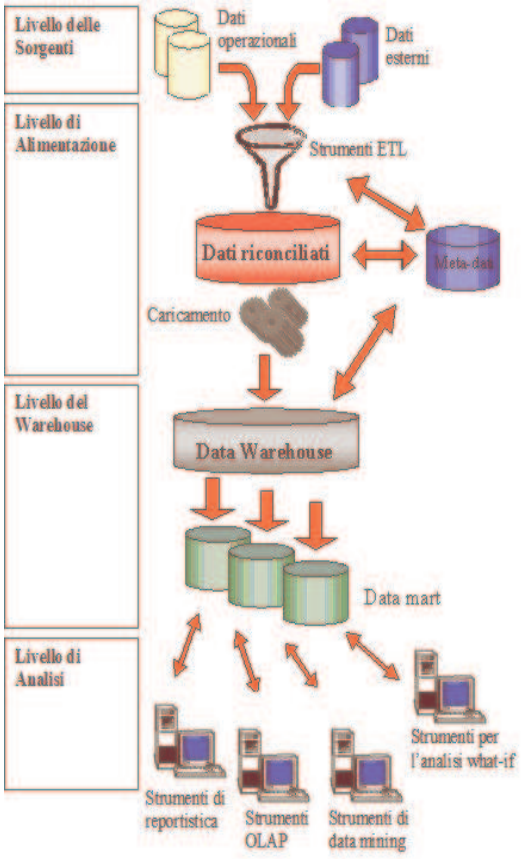

Figura 4: Architettura ad tre livelli

3 Descrizione del contesto

Per realizzare un *unico* sistema informativo biologico è necessario aggregare i dati provenienti da banche dati esistenti. Per raggiungere il pragmatico obiettivo di realizzare un data warehouse è necessario, come prima fase, l'analisi e la riconciliazione delle fonti dati.
Una descrizione delle banche dati esistenti è trattata in [10], ma a partire da essa non è possibile avere la presunzione di poter realizzare effettivamente un data warehouse. Quindi nel corso della trattazione è fatta una descrizione intuitiva, seguita da un'esperienza personale di realizzazione di un data warehouse per il Comune di Napoli [9].
Nelle pagine precedenti sono trattate le diverse architetture per un data warehousing e quella riguardante la Figura 4 mostra la presenza di un livello di dati riconciliati. In seguito i dati riconciliati saranno chiamti indistintamente *area di staging* e rappresentano un'area tecnica in cui i dati sono

integrati in modo da rendere più semplice le operazioni di caricamento del data warehouse. Si potrebbe evitare di materializzare tale area e la sua presenza virtuale sarebbe criptata all'interno delle procedure ETL (Extraction-Trasformation-Loading), tramite le quali si alimenta il data warehouse. Spesso, come in questo caso, si preferisce fisicizzare i cosiddetti dati riconciliati per rendere più semplice la comprensione e la scrittura delle procedure ETL. Le fasi principali per la progettazione del livello riconciliato sono rappresentate negli ovali in Figura 5.

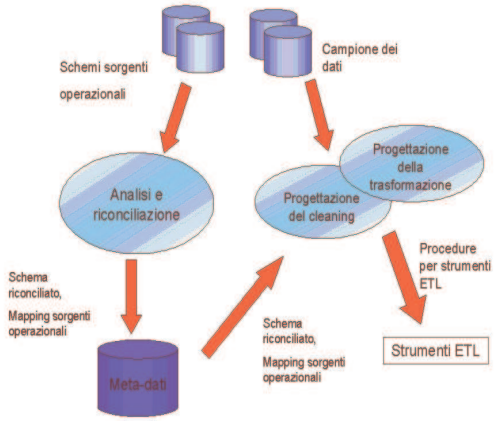

Figura 5: Le fasi per la progettazione del livello riconciliato

La fase di progettazione del cleaning e della trasformazione fanno parte del processo ETL e sono state definite in tale contesto per l'interdipendenza delle tre fasi.

La fase di analisi e riconciliazione prevede la ricognizione e la normalizzazione di ogni sorgente, e la fase d'integrazione tramite cui è dedinito uno schema globale a partire dagli schemi di ogni sorgente. Questa fase ha bisogno di uno studio dettagliato delle sorgenti esistenti, che non è possibile avere con il solo accesso alle banche dati e perciò, come è stato già specificato, è fatta una trattazione intuitiva.

La definizione dello schema globale non è immediata e bisogna eseguire i seguenti passi:

1. La **preintegrazione**, per individuare quali dati operazionali sono necessari al fine di soddisfare i fatti di interesse e quale tecnica utilizzare per definire uno schema globale. E'ovvio che non tutte le informazioni

sono necessarie e quindi non tutti i dati operazionali sono presi in considerazione. Inoltre in tale fase è necessario specificare una metodologia standard per integrare i diversi schemi: la tecnica binaria prevede di considerare due schemi alla volta, mentre la tecnica n-aria prevede di considerare più schemi alla volta ed integrarli contemporaneamente.

2. La **comparazione** degli schemi permette di definire le relazioni e le discrepanze tra gli schemi locali di interesse. I conflitti che si possono presentare sono: Conflitti di eterogeneità: si possono presentare perché la rappresentazione di ogni schema può utilizzare formalismi diversi; Conflitti sui nomi: si possono verificare problemi di omonimia (concetti diversi hanno lo stesso nome) e di sinonimia (stessi concetti sono espressi con nomi diversi); Conflitti semantici: diversi schemi rappresentano la stessa realtà ma in modo diverso perché ad un livello d'astrazione e dettaglio differente; Conflitti strutturali: scelte diverse di modellazione di stessi concetti.

3. L'**allineamento** degli schemi prevede di risolvere l'insieme dei conflitti presentati al passo precedente.

4. La **fusione** e la **ristrutturazione** degli schemi è la fase che permette di migliorare e/o correggere lo schema globale costruito secondo i passi precedenti. In pratica si deve controllare che lo schema globale costruito soddisfi i seguenti requisiti: Completo: definire nuove associazioni tra gli schemi locali non visibili nella fase di analisi; Minimo: evitare che stessi concetti si ripetono in porzioni differenti dello schema globale; Leggibilità: nel caso che la semplice integrazione non permette di poter esprimere i concetti d'interesse in modo chiaro, bisogna ristrutturare lo schema.

L'area di staging è un'area tecnica per rendere più semplice l'alimentazione del data warehouse vero e proprio, ma essa può anche essere vista come una base di dati integrata e consistente, che sono dei requisiti necessari in presenza di dati ridondanti e disomogenei. Infatti essa è alimentata tramite le cosidette procedure ETL. Un data warehouse non è realizzato semplicemente per aggregare i dati ma per rendere semplice e veloci le interrogazioni più frequenti dell'utente. In pratica bisogna costruire il data warehouse in corrispondenza dei requisiti dell'utente, ovvero dei cosiddetti *fatti di interesse*. Secondo l'architettura illustrata in figura 4 si può osservare che un data warehouse è un insieme di data mart e quindi in linea di principio ogni data mart corrisponde ad un fatto di interesse. Nello specifico non si conoscono

i fatti di interesse dei *business users* (i futuri utenti del data mart), perchè per definire essi è necessario una forte interazione tra *un* biologo ed *un* informatico. La fase che prevede la selezione degli intervistati, la preparazione dei questionari per le interviste e la determinazione dei fatti di interesse è chiamata *analisi dei requisiti utente*.

Un possibile fatto di interesse è il seguente: *identificare i geni la cui espressione genetica è cambiata con il manifestarsi di una malattia.* Tale fatto di interesse è proposto in [13] ed è importante, ad esempio, per capire se un individuo è malato o meno osservando *solo* i geni la cui espressione genetica non è cambiata per caso, ma per una particolare malattia. Conoscere i geni che sono alterati con una malattia è importante per i cosiddetti *problemi di classificazione*.
Un problema di classificazione per i dati genomici deve, ad esempio, poter decidere se un individuo è malato o meno in corrispondenza delle informazioni genetiche del soggetto. In letteratura esistono diversi algoritmi di classificazione, che si differiscono dal loro livello di accuratezza e dal tempo di esecuzione. Nel [16] è proposta una soluzione che prevede di considerare un sottoinsieme di esempi (esperimenti) per definire il classificatore . Una soluzione diversa potrebbe essere quella in cui si considererebbero un sottoinsieme di caratteristiche (geni) piuttosto che di esemplari, o di entrambi. I geni che sono necessari per definire il classificatore possono, quindi, essere definiti a partire dai dati memorizzati nel data warehouse.

4 Progettazione concettuale e logica

Lo scopo della tale fase cha va sotto il nome di *progettazione concettuale* è quello di rappresentare la realtà di interesse in un modo formale e completo ma indipendente dal DBMS utilizzato. L'obiettivo della progettazione concettuale è quello di produrre il cosiddetto schema concettuale. Il modello di dati concettuale più popolare è il modello Entità-Associazione (ER: Entity-Relationship), ma per la modellazione concettuale di un data warehouse non è molto utilizzato perché gli elementi di tale formalismo non riescono a rappresentare tutti i concetti d'interesse. Il modello concettuale per la progettazione di un data mart è il Dimensional Fact Model (DFM), che prevede un insieme di schemi di fatto in cui sono modellati: i fatti, le misure, le dimensioni e le gerarchie. Uno schema di fatto è del tipo di figura 6.

Nella Tabella 1 sono spiegati i termini presenti nella Figura 6 e che sono generalmente utilizzati per definire uno schema di fatto con il DFM.

Concetto	Definizione
fatto	Concetto di interesse per il processo decisionale; tipicamente modella un insieme di eventi che accadono nell'impresa
misura	Proprietà numerica di un fatto che ne descrive un aspetto quantitativo di interesse per l'analisi
dimensione	Proprietà con dominio finito di un fatto che ne descrive una coordinata di analisi.
evento primario	Occorrenza particolare di un fatto, individuata da un ennupla costituita da un valore per ciascuna dimensione. A ciascun evento primario è associato un valore per ciascuna misura.
attributo dimensionale	Dimensione ed eventuali attributi, sempre a valori discreti, che la descrivono.
gerarchia	Albero direzionato i cui nodi sono attributi dimensionali e i cui archi modellano associazioni molti-a-uno tra coppie di attributi dimensionali. Essa racchiude una dimensione, posta alla radice dell'albero, e tutti gli attributi dimensionali che la descrivono.

Tabella 1: Termini principali di un DFM

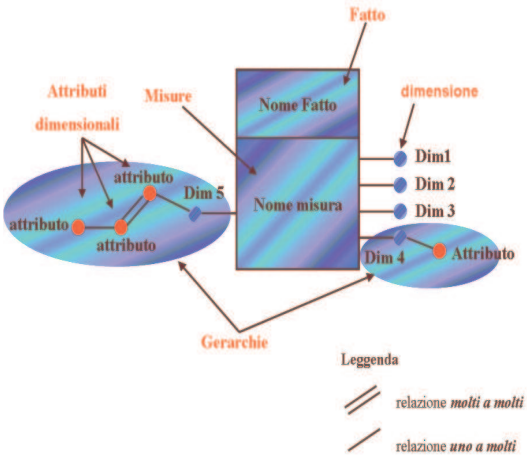

Figura 6: Schema di fatto

La progettazione logica è la fase che permette di definire, a partire dal modello concettuale, uno schema della base di dati nel modello di dati implementabile dal DBMS. Il prodotto ottenuto da questa fase è il cosiddetto schema logico. Nel caso specifico la progettazione logica include l'insieme dei passi che, a partire dallo schema dei fatti, permettono di determinare lo schema logico di ogni datamart. Il modello concettuale (il modello E/R o il DFM) definisce i concetti d'interesse senza dare alcuna informazione su come sono organizzati i dati. Nel modello logico, invece, è definita l'organizzazione dei dati pur senza soffermarsi sui dettagli implementativi.
Il modello logico utilizzato è il cosiddetto *modello multidimensionale*: i dati sono organizzati secondo delle strutture multidimensionali. Il modo più naturale per rappresentare i dati di un data warehouse è quello che prevede l'utilizzo di strutture multidimensionali. Un fatto può essere modellato tramite una matrice k-dimensionale, se le dimensioni (secondo quanto definito nella tabella 1) sono k.

Il cubo di figura 7 rappresenta il fatto ed i cubetti i cosiddetti eventi. All'interno di ogni cubetto sono memorizzati i valori delle misure, e per conoscerli basta fissare il valore di ogni dimensione e tracciare, per ognuna di esse, le perpendicolari agli assi; il punto d'intersezione è l'evento da considerare. Tale esempio permette di spiegare come tale rappresentazione renda semplice la comprensione dei fatti di interesse e come sia naturale il recupero delle informazioni. I modelli logici che rappresentano la struttura multidimensionale dei dati sono:
1. Il ROLAP (Relational On-Line Analytical Processing), che utilizza il modello relazionale per la rappresentazione dei dati multidimensionali;

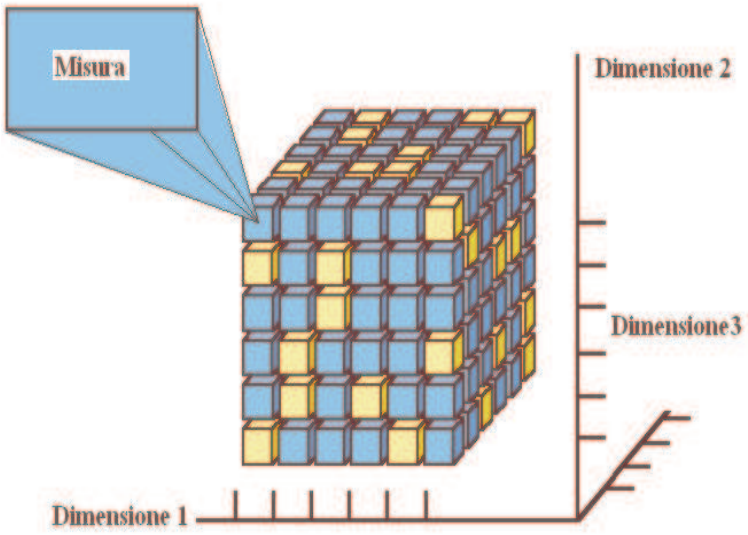

Figura 7: Cubo per la rappresentazione dei dati

2. Il MOLAP (Multidimensional On-Line Analytical Processing), che utilizza il modello mutidimensionale. La rappresentazione più naturale di un data warehouse è quella mutidimensionale ed inoltre tale tipo di organizzazione dei dati rende semplici e veloci le operazioni OLAP . Il principale limite di tale sistema è nella gestione della sparsità, dove con il termine di sparsità si intende che solo alcuni elementi della struttura dati utilizzata per rappresentare i dati contengono effettivamente informazioni. Ad esempio, nel caso in cui è utilizzata come struttura il cubo, in cui ogni evento è rappresentato da una cella, si avrà che solo alcune celle contengono delle informazioni, ovvero quelle che corrispondano ad eventi accaduti.

I modelli concettuali e logici descritti sono ispirati a quelli del [13]. In tale articolo è introdotto un nuovo modello multidimensionale, che è chiamato *Biostar*. Nel corso della trattazione si mostrerà che è possibile modellare gli stessi concetti non utilizzando tale modello multidimensionale, ma gli usuali costrutti che sono usati in un data warehouse aziendale.

Prima di fare questo si ricorda che quando si definisce uno schema concettuale di un data warehouse non si vogliono modellare le *relazioni* tra le entità, ma il cosiddetto *fatto* di interesse. Si consideri il modello concettuale di [13] relativo ai dati clinici di figura 8.

Si osservi, innanzitutto, che è stato utilizzato il modell E-R, che come è stato spiegato all'inizio di questo paragrafo è poco espressivo. Infatti, non

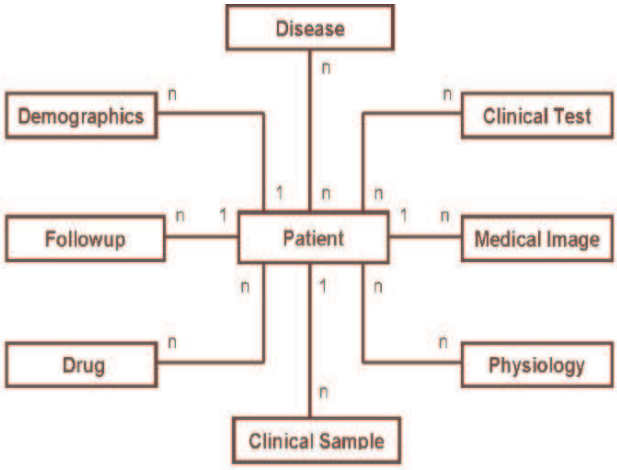

Figura 8: Modello concettuale per dati clinici

fornisce alcuna informazione circa a quali sono i fatti di interesse.

A partire da schemi concettuali che non esprimono i fatti di intersse è complicato dover tradurre essi in schemi logici, dove sono modellati i fatti, le dimensioni, le misure e così via.

Gli autori dell'articolo [13] propongono, così, un nuovo modello multidimensionale, il cosidetto ***BioStar***. La struttura di uno schema biostar è ilustarto in figura 9.

In figura 10 è riportato lo schema Biostar presente in [13] relativo ai dati clinici.

Gli autori di [13] presentano un nuovo modello perchè l'entità *Paziente* partecipa a **tutti** i fatti di interesse. In realtà non è necessario definire un nuovo modello, ma come è proposto in [15] basta semplicemente definire un data mart che contiene i fatti di interesse correlati. In questo caso il data mart può essere denominato *dati clinici* ed i fatti di interesse sono: *Diagnosi*, *RisultatiDiTest* e *FarmacoUsato*. E'definito un unico data mart perchè i diversi fatti hanno delle tabelle in comune (*Paziente*) e quindi per evitare di replicare i dati in diverse tabelle è definito un singolo data mart. In realtà un fatto di interesse non è modellato secondo quanto è illustrato in figura 10, poichè le tabelle dimensionali non contengono tutti gli attributi descrittivi di una dimensione. Le tabelle dimensionali dovrebbero contenere solo le gerarchie dimensionali rispetto a cui è poi possibile effettuare, eventualmente, le opportune aggregazioni. Le informazioni relative alle dimensioni sono memorizzate nell'area di staging, perchè si ricorda che il data warehouse è una *replica* e quindi si deve cercare di duplicare il minor numero di dati possibili. In questo caso specifico, quindi, si può pensare di memorizzare nella tabelle

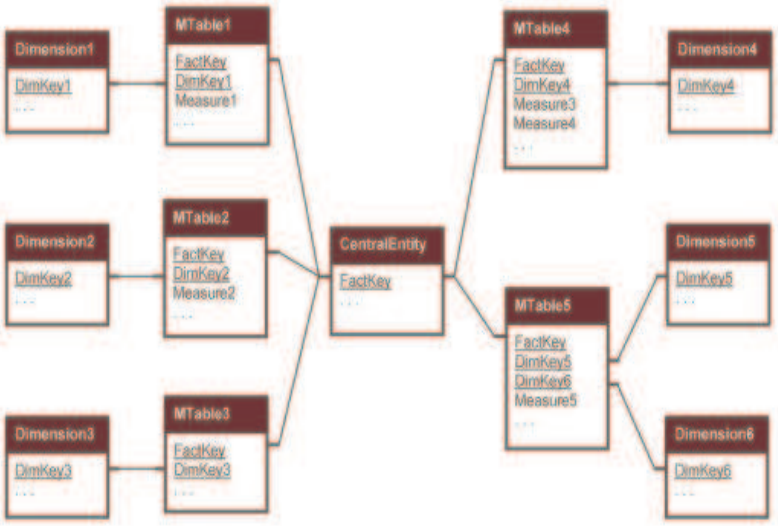

Figura 9: Modello Biostar

dimensionali solo le chiavi di ogni dimensione. Spesso non si richiede all'utente di fissare come valore della dimensione la sua chiave perchè, generalemente, rappresentano dei dati fittizi o perchè è poco intuitivo per l'utente. Nel caso in cui bisogna scegliere, ad esempio, la *MisurUnit* si può pensare che è fissato dall'utente il nome della misura e dalla tabella dell'area di staging è recuperato la chiave per poter accedere al valore della misura. Non è questo è il contesto per spiegare il recupero delle informazioni, ma sinteticamente si può dire che l'estrazione delle informazioni non sono operazioni sequenziali, ma si esegue il *join* tra la tabella dimensionale e la tabella dell'area di stag-

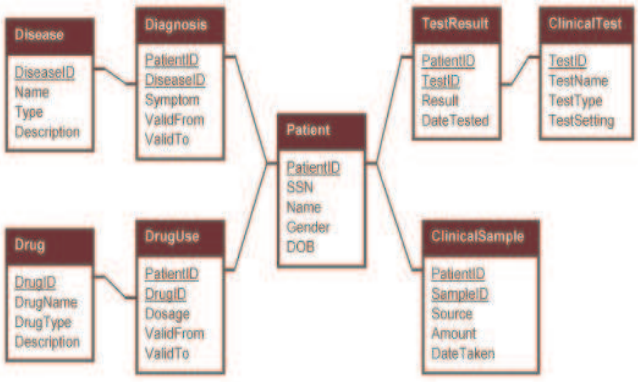

Figura 10: Modello Biostar dei dati clinici

ing, che contiene le informazioni necessarie. Prima di costruire i data mart che modellano i diversi fatti di interesse è necessario, allora, definire un'area di staging. Essa è rappresentata tramite il *Class Diagramm* in figura 11, siccome si suppone che l'area tecnica sia un data base relazionale ad oggetti. In questi ultimi anni si è molto diffusa la programmazione ad oggetti, che si ritrova sempre più spesso anche nelle base di dati. I più diffusi sistemi di gestione di basi di dati si avvalgono del modello relazionale ad oggetti. *Oracle*, il DBMS (*Database Management system*) utilizzato per realizzare l'intera soluzione data warehouse, permette la definizione e la gestione di un data base relazionale ad oggetti [1].

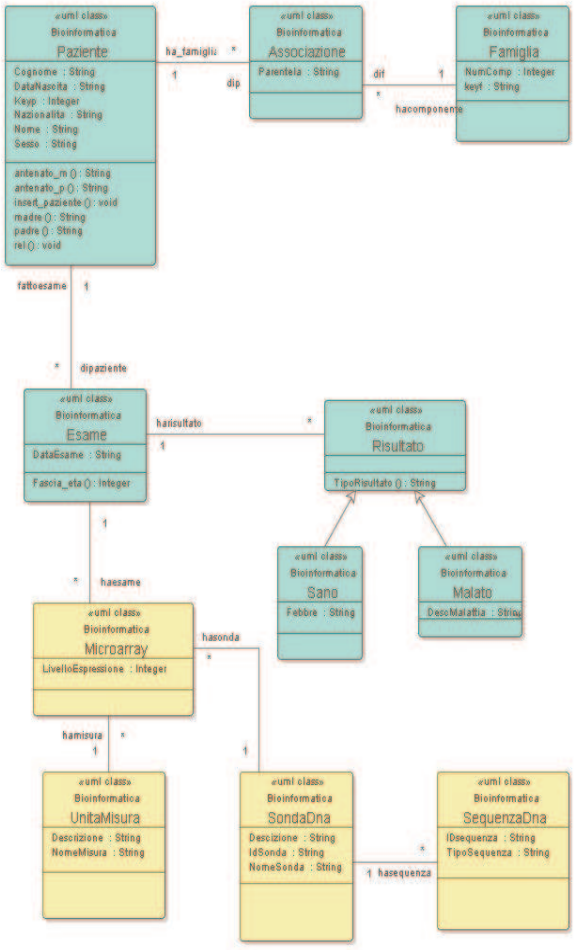

Figura 11: Class diagram dell'area di staging

A partire dall'area di staging è possibile alimentare il data warehouse vero

e proprio, che è propriamente relazionale. Il fatto *mrnaexpression* non avrà come modello logico quello di [13], che è stato riportato in figura 12.

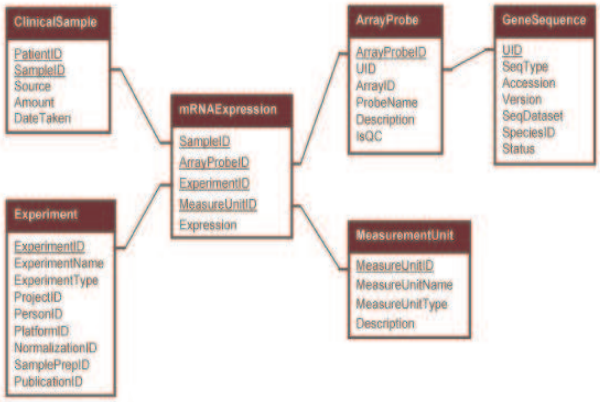

Figura 12: Modello logico del fatto mrna_expression proposto in[13]

Secondo quanto già spiegato le tabelle dimensionali non devono contenere gli attributi dimensionali, perchè sono memorizzate nell'area di staging. Le tabelle dimensionali nel caso specifico non sono proprio definite poichè le dimensioni sono degeneri, ovvero contengono un solo attributo. In questo caso si evita di creare una tabella con un unico attributo, si memorizza esso nella tabella di fatto. Il fatto di interesse può essere, così, modellato come è illustrato in figura 13.

Figura 13: Modello logico del fatto mrna_expression

La tabella *distr_student* è più chiara quando è spiegato il fatto di interesse proposto:

identificare i geni la cui espressione genetica è cambiata con la presenza di una malattia.

Dalla figura 13 si può capire che è possibile conoscere il valore della misura (livello di espressione) in corrispondenza delle seguenti dimensioni: id_sonda, sesso, fascia_eta, nazione, misura e malattia. Le dimensioni considerate in questo caso sono di più di quelle riportate in [13], ma è sembrato necessario considerare ulteriori dimensioni di analisi. Anche l'area di staging contiene qualche attributo in più in corrispondenza di alcune tabelle, come ad esempio *Paziente*. In quest'ultimo caso sono definiti come attributi: il nome e il cognome, che possono non essere inseriti in caso di necessità di privacy. Prima di spiegare nel dettaglio come è possibile ottenere l'informazione desiderata è necessario fare una breve digressione.

Come è stato più volte ripetuto si vuole conoscere quali geni sono alterati in corrispondenza di una particolare malattia e per fare ciò si considera il livello di espressione dei geni. Il livello di espressione stabilisce se un gene è attivo in una cellula e in quale misura. Tale tipo di informazione può essere conosciuta tramite la tecnica del **microarray**, che permette di immobilizzare sonde di DNA dei geni che si vogliono analizzare. La tecnica del microarray permette di conoscere il livello di espressione di diversi geni ed un apetto imporatante potrebbe essere quello di confrontare tessuti di pazienti malati e non, in questo modo si può capire come varia il livello di espressione con la presenza di una data malattia. E' abbastanza intuitivo immaginare che la malattia agisce solo su alcuni geni e delle analisi accurate potrebbero essere necessarie affinchè l'utente possa sapere quali geni sono alterati.
Nel [13] è proposta una possibile soluzione, che è riportata in questo contesto. Si ipotizza di poter conoscere quali geni sono alterati tramite uno studio statistico, ovvero tramite il concetto di *probabilità*. Infatti, l'utente può conoscere quali geni sono alterati non per caso secondo una certa probabilità. Per confrontare i geni dei pazienti malati e non è utilizzato il test ***t*** (detto anche t di Student).

Tale tipo di test è utile quando si vuole conoscere se due campioni differiscono per caso o non. In questo contesto la popolazione dei due campioni sono: i geni dei tessuti sani e quelli di una malattia fissata. Si ricorda che tale tipo di test può essere applicato fissando una mallatia in modo da considerare solo quei geni della malattia considerata, siccome malattie diverse possono influire in maniera distinta sui geni. Per ogni campione si calcola la media, la varianza ed il grado di libertà. Tali valori sono necessari per conoscere il valore ***t***, chè è così calcolato:

$$t = \frac{m_a - m_b}{s}\sqrt{\frac{n_a * n_b}{n_a + n_b}} \quad (1)$$

Il valore m_a e n_a rappresentano, rispettivamente, la media ed i gradi di libertà del campione relativo ad i geni non malati; mentre m_b e n_b sono relativi a quelli dei geni sani. I gradi di libertà è dato dal numero di elementi che costituiscono il campione.

Il valore ***t*** è calcolato in corrispondenza di ogni sonda di DNA e ognuno è confrontato con quello della distribuzione t-student, che è memorizzata nella tabella distr_student di figura 13. Il confronto è possibile fissando il cosidetto valore di **significatività**, ovvero la probabilità rispetto alla quale la differenza del livello di espressione non sia dovuta al caso. In questo modo se in corrispondenza del parametro di significatività fissato il valore di ***t*** per una sonda genomica è maggiore di quello della distribuzione della variabile t-student, allora significa che i geni della sonda di DNA considerata non si sono alterati per caso ma per la malattia fissata. In caso contrario, invece, le differenza è dovuta, secondo la probabilità fissata, al caso. Una possibile schermata che permette di scegliere una malattia e il livello di significatività è quella illustarata in figura 14. In base ad i valori scelti è fornito come output il numero di geni alterati ed i geni modificati.

Figura 14: Una possibile interfaccia

5 Conclusioni e lavoro futuro

In questo documento è stato spiegato l'utilità di un data warehouse per dati genomici. In particolare è stato definito un fatto di interesse e la proget-

tazione concettuale e logica del data mart del fatto. Inoltre è stato dimostrato che non è necessario implementare una soluzione della progettazione concettuale e logica diversa da quella per i data warehouse aziendali. La progettazione fisica del data warehouse è stata fatta su dati fittizi e quindi un lavoro futuro potrebbe essere quello di caricare dei dati reali. In questo modo è possibile, ad esempio, applicare gli algoritmi di classificazione citati anche in questo documeno.

Riferimenti bibliografici

[1] Oracle 9i, application developer's guide - object relational features.

[2] Sito del department of health and human services. http://datawarehouse.hrsa.gov/.

[3] Sito del popultation division. http://www.un.org/esa/popultation/unpop.htm.

[4] Sito della città di modena, dove è stato realizzato un data warehouse. http://sit.comune.modena.it/.

[5] Sito della regione marche, dove è stato realizzato un data warehouse. http://www.sistar.marche.it/dwh/comuni/index.htm.

[6] Sito dell'istat. http://demo.istat.it/index.html.

[7] Sito entrez. http://www.ncbi.nlm.nih.gov/Entrez/.

[8] Sito srs. http://srs.ebi.ac.uk.

[9] S. Cuciniello. Progetto e realizzazione di un data warehouse per analisi demografiche: il caso del comune di napoli. Master's thesis, Università degli Studi di Napoli Federico II, Facoltà di Scienze MM.FF.NN., 2004-2005.

[10] M. Attimonelli G. Pesole G. Valle, M. Helmer Citterich. *Introduzione alla Bioinformatica*. Zanichelli, 2005.

[11] W.H. Inmon. *Building the data warehouse*. John Wiley & Sons, 2002.

[12] J. Lechtengorger. *Data warehouse schema design*. DISDBIS 79, Akademische Verlagsgesellschaft Aka GmbH, 2001.

[13] Aidong Zhang Liangjiang Wang. Biostar models of clinical and genomic data for biomedical data warehouse design. *J. Boinformatics Research and Application*, 2005.

[14] M. Ross R. Kimball. *The data warehouse toolkit*. John Wiley & Sons, 2002.

[15] M. Ross R. Kimball. *Data Warehouse, teoria e pratica della progettazione*. McGraw-Hill, 2005.

[16] C. Cifarelli O. Seref P. M. Pardalos S. Cuciniello, M. R. Guarracino. Incremental classification with generalizeted eigenvalues. *Journal Classification*, 2006.