

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA
Corso di Laurea in Ingegneria e Scienze Informatiche

**Progettazione di un sistema
di Location Intelligence
per una multiutility**

**Relazione Finale in
Sistemi Informativi e Business Intelligence**

**Relatore:
Chiar.mo Prof.
Matteo Golfarelli**

**Presentata da:
Tomas Tassinari**

**Correlatori:
Dott. Nicola Bonoli
Dott. Paolo Rodeghiero**

**Sessione II
Anno Accademico 2015/2016**

Indice

Introduzione	1
1 Location Intelligence	5
1.1 Definizione	5
1.2 Differenze fra Location Intelligence e Analisi Geografica	6
1.2.1 Analisi Geografica	7
1.2.2 Location Intelligence	7
1.3 1854 - La nascita della Location Intelligence	8
1.4 Geographic information system	10
1.4.1 Geodesia e Mappatura Proiettiva	10
1.4.2 Rappresentazione geografica digitale	13
1.5 Architettura di un sistema di Location Intelligence .	17
1.5.1 Spatial Data Warehouse	18
1.6 Applicazioni di business	20
2 Tecnologie	23
2.1 Oracle Business Intelligence Enterprise Edition	24
2.2 MapViewer	28
2.3 Oracle MapBuilder	30
2.4 Oracle Spatial	32
2.4.1 Tipi di forme geometriche in Oracle	32
2.4.2 Tipo di dato SDO_Geometry	33
2.4.3 SQL/MM e OGC	36

2.4.4	Tabelle dei Metadati	38
2.4.5	R-Tree	40
3	Progetto	43
3.1	Obiettivi di Progetto	43
3.2	Requisiti dell'utente	45
3.3	Analisi Funzionale	46
3.3.1	Definizione KPI	47
3.3.2	Modello di analisi dei dati	49
3.3.3	Data profiling	52
3.4	Progettazione	55
3.4.1	Stack Architetturale	55
3.4.2	Data Loading	55
3.4.3	Architettura del Front-End	57
3.5	Sviluppo e test dell'applicazione	60
3.5.1	Overview del processo di ETL	60
3.5.2	Sviluppo del front-end	77
3.5.3	Fase di test	81
3.6	Deliverable progettuali e front-end	87
3.7	Sviluppi Futuri	90
	Conclusioni	93
	Bibliografia	95

Elenco delle figure

1.1	Mapa con la sovrapposizione dei casi di colera, in rosso, e delle pompe d'acqua della zona di Broad Street	9
1.2	Esempio di rappresentazione ellissoidale con linee che delimitano le aree con la stessa scala per il calcolo dell'area [14]	11
1.3	Proiezione di una sfera su un piano tangente (a), un cono tangente (b) e un cilindro tangente (c) [14]	12
1.4	Utilizzo di layer multipli per la rappresentazione della realtà geografica [11]	14
1.5	Latitudine, longitudine e assi di riferimento [16]	15
1.6	Esempio di architettura di un sistema di Location Intelligence	18
1.7	Strumento di analisi what-if relativo a strutture ospedaliere	20
1.8	Dashboard per l'analisi delle potenzialità di mercato del territorio	21
2.1	Posizionamento delle tecnologie in esame nello stack architetturale di un sistema di location intelligence	24
2.2	Rappresentazione del layer OBIEE rispetto al livello dei dati e quello di presentazione	25
2.3	Struttura dell'architettura di OBIEE 11g	26
2.4	Architettura del BI Server basata sul Common Enterprise Information Model	28

2.5	Architettura di Oracle MapViewer [26]	30
2.6	Tipologie di stili avanzati realizzabili con Oracle MapBuilder	31
2.7	Tipi di forme geometriche [27]	33
2.8	Esempio di albero di un R-Tree [37]	40
2.9	Esempio di creazione degli MBR in un R-Tree [37] . .	41
3.1	Fatto: Performance OdS	50
3.2	Fatto: Svuotamenti	51
3.3	Dimensione: Ordine di Servizio	52
3.4	Schema dei componenti utilizzati	56
3.5	Dettaglio dell'architettura relativa al front-end	58
3.6	Schema delle principali fasi del processo di ETL . . .	62
3.7	Esempio di interfaccia di MapBuilder dove viene definita la query per recuperare i dati da visualizzare . .	79
3.8	MapBuilder rendering rules	80
3.9	MapBuilder additional attributes	80
3.10	Elenco dei tematismi disponibili sull'applicativo . . .	88
3.11	Tema di dettaglio sull'analisi delle velocità medie con dettagli del punto di raccolta	89
3.12	Mashup di tematismi per l'individuazione di fenomeni complessi	90

Introduzione

Nel mondo della tecnologia consumer, abbiamo assistito a una diffusione massiccia di dispositivi mobili come smartphone, tablet, e dispositivi indossabili (wearable), tutti con l'obiettivo comune di permettere l'accesso alla rete e ai servizi da qualsiasi parte del globo e tutti entrati nella quotidianità di miliardi di persone. Anche grazie all'aumentare di questo tipo di dispositivi, è cresciuta esponenzialmente la produzione, e quindi la disponibilità, di dati geolocalizzati: quasi tutti questi strumenti sono dotati di location awareness, ovvero hanno la possibilità di conoscere in modo attivo o passivo la propria posizione sul globo terrestre. Ogni immagine scattata, ogni telefonata, ogni interazione con un social network porta con sé l'informazione di *dove* (e quando) è avvenuto uno specifico evento. Le grandi aziende tecnologiche internazionali hanno quindi studiato i modi migliori per utilizzare questo patrimonio informativo e per assicurarsi di mantenerlo in continua crescita, basti pensare a Google con il servizio di location history che traccia la posizione di uno smartphone o di un tablet ventiquattro ore su ventiquattro. Questi dati hanno permesso la creazione di una serie di prodotti e servizi di successo: Google Adwords per l'advertising, che fornisce messaggi pubblicitari basati sul luogo in cui ci si trova (così come fanno tutti i competitor, da Facebook a Twitter, fino ai servizi come TripAdvisor per strutture turistiche e di ristorazione); videogames di successo basati su realtà aumentata e posizione GPS; software come Google Street Map e

Panoramio che sfruttano immagini geolocalizzate per permettere a chiunque con un accesso alla rete di “visitare” luoghi lontani migliaia di chilometri, fino a strumenti come Google Maps o Waze a cui oggi molti si affidano per prevedere il traffico sui propri tragitti in auto.

Tutti noi produciamo dati georeferenziati e lo stesso succede in ogni fase di ogni processo di migliaia di aziende. La Location Intelligence è nata con l’obiettivo di analizzare e comprendere a pieno il patrimonio informativo presente nei dati geolocalizzati. Da quando esistono i DataWarehouse le aziende ricavano informazioni dai propri dati ma, oggi, è possibile fare di più: i sistemi di location intelligence permettono di correlare i dati con la loro componente geografica (*Spatial*) e, di conseguenza, con tutte le informazioni relative ad una specifica zona territoriale; non solo si aggiunge una nuova dimensione di analisi (il *dove*) ma, di fatto, anche una ulteriore possibilità di integrazione di dati precedentemente scorrelati; per portare solo alcuni esempi: informazioni sulla popolazione residente e la densità abitativa, sulla presenza di luoghi di interesse nelle vicinanze, sulle condizioni meteorologiche attese e passate, sulla conformazione del territorio e sulla prossimità rispetto ad altri oggetti di studio.

Un problema che si presenta tipicamente ad un’azienda che offre i propri servizi su un territorio, e che è in parte risolvibile con l’aiuto della location intelligence, è la necessità di controllare e monitorare sia le proprie performance, sia i cambiamenti che avvengono nell’ambiente in cui lavora. Da sempre le grandi aziende con grandi sedi centrali ma ampio raggio d’azione, distribuiscono sul territorio una serie di controllori o gestori locali che hanno il compito di monitorare i cambiamenti delle proprie zone di competenza, comprenderli, individuare soluzioni e mantenere un alto livello di performance locale. Il tutto viene però distribuito sul territorio e applicato singolarmente a ogni singola area, senza una visione di insieme. Questo perché si è sempre dovuto trovare un trade-off fra il controllo centrale, con

miglior capacità di ottimizzazione e coordinamento, e le specificità delle singole zone periferiche, ognuna con caratteristiche diverse e che è possibile comprendere solo grazie ad analisi locali.

In questo lavoro di tesi ci si pone l'obiettivo di produrre un sistema di location intelligence per una multiutility al fine di abilitare nuove analisi aggregate e storiche di indicatori riferiti alle performance aziendali sul territorio, così da permettere l'individuazione di criticità e prevedere eventuali trend negativi, tramite uno strumento centralizzato che arricchisca le informazioni tradizionalmente in possesso dell'azienda con nuovi dati geolocalizzati.

In questa relazione finale il lettore troverà nel primo capitolo un'overview sui concetti alla base della location intelligence e dei sistemi analitici di supporto alle decisioni; nel secondo capitolo verranno illustrate le principali tecnologie utilizzate per l'implementazione del sistema, mentre nel terzo capitolo si approfondiranno le fasi di progettazione e sviluppo della soluzione, descrivendone i punti salienti e più caratterizzanti. Infine, verranno illustrati i possibili successivi sviluppi di questo lavoro, arrivando alle conclusioni tratte dal completamento di questa esperienza.

Capitolo 1

Location Intelligence

1.1 Definizione

Il termine Location Intelligence viene usato per definire una serie di strumenti e tecniche che integrano la dimensione geografica all'interno delle piattaforme di Business Intelligence (BI) con l'obiettivo di aumentarne le capacità di analisi e riuscire a monitorare e interpretare meglio i fenomeni di business anche grazie alla loro collocazione sul territorio [1]. Si calcola che oltre l'80% dei dati di un'organizzazione comprende una componente geografica [2] e questo dato è indubbiamente destinato a salire drasticamente, soprattutto considerando quanto stiano diventando pervasivi i sistemi IoT e le tecnologie wearable nella vita quotidiana e nel mondo dell'industria. L'occhio umano processa i pattern grafici 60.000 volte più velocemente rispetto a dati raccolti in tabella [1], di conseguenza la rappresentazione su mappe geografiche porta un fondamentale miglioramento agli strumenti analitici; a questo scopo da decenni esistono tecnologie come i Geographical Information System (G.I.S.) creati appositamente per elaborare e visualizzare informazioni geografiche, il limite di questi prodotti si dimostra evidente nel momento in cui aumentano le necessità di analisi ed elaborazione di questa tipologia di dato: sono

infatti necessari strumenti e tecniche di BI per individuare fenomeni e pattern complessi. La location intelligence unisce le capacità di queste due tipologie di sistemi portando, come valore aggiunto, la capacità di individuare correlazioni fra fenomeni descritti da indicatori che condividono solo la prossimità geografica. Grazie a questi sistemi diventa possibile estrarre una ulteriore componente informativa dal dato geografico e utilizzarlo in enormi quantità tenendo conto degli andamenti rispetto alla dimensione temporale e aggregando rispetto alla gerarchia geografica [3].

Definizione di Business Intelligence Il termine Business Intelligence, utilizzato finora, non può essere definito né un prodotto né un sistema. È una architettura e *un insieme di strumenti di supporto alle decisioni e database, integrati*, per fornire agli utenti di business un accesso semplice ai dati della propria azienda ed esterni [4]. Con Business Intelligence si può altresì definire l'insieme delle tecniche di acquisizione e trasformazione dei dati in informazioni utili per chi vuole effettuare analisi di business [5]. Grazie all'analisi integrata di numerose fonti informative di grosse dimensioni, di tipo strutturato e non strutturato, e alla possibilità di considerare il modo in cui sono variate nel tempo grazie alle serie storiche, è possibile estrapolare contenuto informativo che le singole fonti o i singoli dati non possiedono ma che emergono solo nel momento in cui vengono messi in correlazione fra loro e studiati lungo l'asse temporale.

1.2 Differenze fra Location Intelligence e Analisi Geografica

Una delle più importanti disambiguazioni necessarie a comprendere a pieno l'ambito della location intelligence è la differenza fra

1.2 Differenze fra Location Intelligence e Analisi Geografica 7

quest'ultima e quella che, comunemente, viene definita *analisi geografica*.

1.2.1 Analisi Geografica

Capita spesso di trovare delle mappe geografiche nei report di diversi ambiti, si tratta nella maggior parte dei casi di layer su cui vengono applicati, nelle corrispondenti coordinate, dei marker a cui vengono associati valori di KPI o altri tipi di informazioni. In questo caso l'obiettivo raggiunto consiste nell'arricchire il contenuto informativo di quelli che, altrimenti, si limiterebbero a essere due valori numerici di latitudine e longitudine, difficilmente comprensibili e "contestualizzabili" da un usufruttore umano. Quindi abbiamo, nella sostanza, gli stessi identici dati che potremmo trovare in un report tabellare, ma con più contenuto informativo dato dall'aggiunta del layer della mappa. Tutti gli altri tipi di valutazioni possibili analizzando queste tipologie di report sono merito delle capacità deduttive dell'usufruttore umano. Ad esempio, il lettore potrebbe notare che una determinata regione presenta un accumulo di KPI con valori alti rispetto alla scala, identificando quella zona come critica rispetto all'analisi effettuata. Riassumendo, il valore aggiunto dell'analisi geografica, o "dashboarding geografico", sta nell'arricchimento informativo nell'ultima fase del processo, quando *i dati sono già stati calcolati e definiti*.

1.2.2 Location Intelligence

La caratteristica chiave che distingue la location intelligence e che rappresenta uno dei suoi più importanti valori aggiunti, consiste nell'integrazione di una nuova variabile nella fase di calcolo e analisi dei dati: *la prossimità geografica dei luoghi*. Grazie a questo nuovo fattore si rendono possibili una moltitudine di *calcoli* diversi, dando la

possibilità di ottenere risultati (dati, e di conseguenza informazioni) non ottenibili in altro modo. Non si tratta quindi di un arricchimento informativo di dati ma di produrne di nuovi e inediti. Questo è uno dei fattori che rendono questa branca della business intelligence un fondamentale strumento per l'analisi di tutti gli eventi che si verificano in un preciso luogo sulla superficie terrestre. Per continuare la comparazione con l'analisi geografica, se in quel caso è il cervello umano a individuare pattern derivanti dalla vicinanza degli eventi localizzati su una mappa, con la location intelligence è possibile automatizzare questo processo, in modo da ottenere la stessa *tipologia* di risposte, in automatico, minimizzando gli errori tipici di un intervento umano e scalando drasticamente la quantità di dati coinvolti nell'analisi. Nell'esempio storico del caso del Dr.Snow, descritto nella sezione 1.3, uno strumento di location intelligence avrebbe potuto evidenziare la fontana infetta suggerendo la deduzione del famoso scienziato.

1.3 1854 - La nascita della Location Intelligence

Il primo caso, rimasto nella storia, dell'utilizzo dei principi alla base della location intelligence, risale al 1854, a Londra, ed è attribuito al Dr. John Snow [6]. In quell'anno scoppiò un'epidemia di colera in una zona di Londra, precisamente a Broad Street [7]. Il Dr. Snow elaborò un metodo per individuarne e risolverne le cause scatenanti: Sovrappose infatti due mappe della zona colpita dalla malattia, in una erano indicati, e localizzati, i casi registrati di colera, nell'altra erano indicate le pompe d'acqua a disposizione della popolazione [8, 9]. Grazie a questa tecnica venne individuata un'alta concentrazione di casi nella zona circostante una specifica pompa,

chiudendola e bloccando l'emissione di acqua, si riuscì a limitare l'espandersi dell'epidemia.

Questa semplice ma significativa vicenda evidenzia come la componente geografica e di localizzazione nell'analisi degli eventi possa risultare fondamentale e imprescindibile per estrapolare alcune tipologie di informazioni. Se la stessa analisi effettuata con le mappe fosse stata tentata utilizzando degli elenchi di pompe d'acqua e un elenco delle persone colpite non si sarebbe, probabilmente, mai individuata la correlazione esistente, quantomeno non con la facilità e chiarezza che è resa evidente dalla figura 1.1.

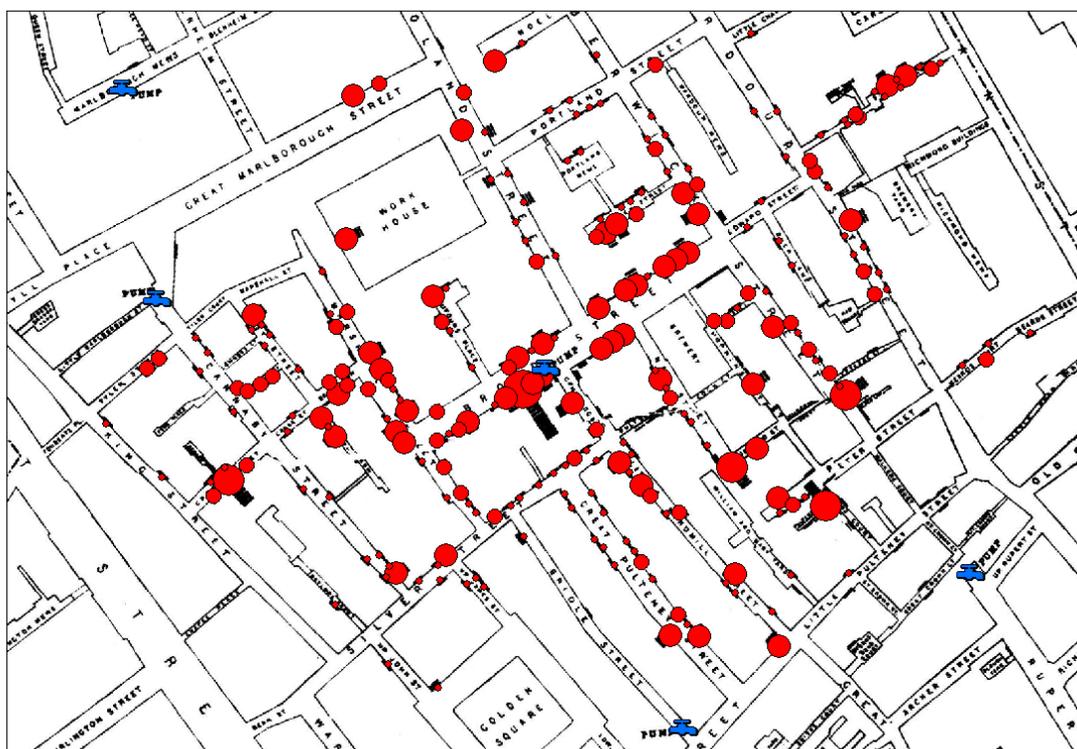


Figura 1.1: Mappa con la sovrapposizione dei casi di colera, in rosso, e delle pompe d'acqua della zona di Broad Street

1.4 Geographic information system

Un G.I.S. (Geographic Information System) è un sistema che permette di rappresentare, interrogare, visualizzare e analizzare entità spaziali o eventi che si verificano sul territorio tenendo traccia non solo del loro accadimento ma anche della posizione geografica [10]. Oltre alle funzionalità di un DBMS tradizionale, le capacità proprie di un G.I.S. sono infatti l'acquisizione, la gestione e l'integrazione di dati territoriali multisorgente [11] che lo rendono un'evoluzione di un database relazionale, in grado di interrogare, analizzare e rappresentare su mappa questa tipologia di informazioni. I principali obiettivi di un sistema G.I.S. sono l'ottimizzazione dell'efficienza di percorsi complessi, la minimizzazione della ridondanza di una base di dati, l'integrazione di informazioni di diverso tipo da fonti eterogenee e l'abilitazione ad analisi e interrogazioni complesse che utilizzano la georeferenziazione per generare nuovo patrimonio informativo [12].

1.4.1 Geodesia e Mappatura Proiettiva

La geodesia è la disciplina che si occupa della misura e della rappresentazione della Terra, del suo campo gravitazionale e dei fenomeni geodinamici [13]. Uno dei due metodi per rappresentare il globo terrestre consiste in un modello tridimensionale di forma ellissoidale, semplificando, il concetto alla base del classico mappamondo. Se, al contrario, vi è la necessità di rappresentare su un piano la superficie terrestre che è curva, è necessaria una conversione, questa viene definita *Mappatura Proiettiva*. Ogni qualvolta si esegue una conversione di questo tipo si ottengono, necessariamente, delle distorsioni; per questo motivo esistono molte proiezioni differenti che si caratterizzano per quali proprietà degli oggetti rappresentati si vogliono mantenere inalterate: forme, aree, angoli o distanze della superficie tridimensionale originale. In figura 1.2 si vede una rappresentazione

planare del globo dove vengono mantenute scale relative al calcolo dell'area.



Figura 1.2: Esempio di rappresentazione ellissoidale con linee che delimitano le aree con la stessa scala per il calcolo dell'area [14]

Ogni combinazione di un datum, una proiezione e una regione geografica presenta delle specifiche distorsioni ed è scelto per via delle informazioni che mantiene inalterate. Le principali misurazioni o proprietà può essere necessario non distorcere sono: la distanza fra due punti e il calcolo del percorso più breve fra due luoghi, le direzioni, le forme e i rapporti fra le aree.

Classificazione geometrica delle proiezioni

Le proiezioni sono generalmente suddivise in tre categorie: azimutale, cilintrica e conica, di cui si può avere un esempio in figura 1.3.

La proiezione azimutale ha un punto centrale senza distorsione, la quale aumenta in modo radiale dal centro verso l'esterno. La proiezione cilindrica ha una singola linea di assenza di distorsione che va ad aumentare nelle direzioni perpendicolari alla linea. In questo tipo

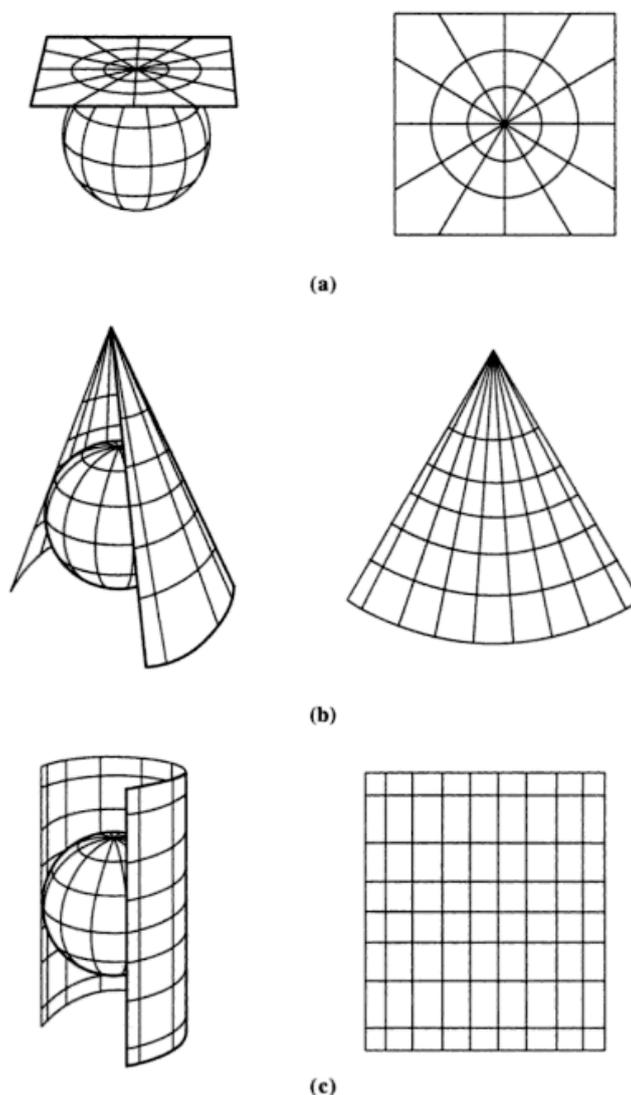


Figura 1.3: Proiezione di una sfera su un piano tangente (a), un cono tangente (b) e un cilindro tangente (c) [14]

di proiezione l'equatore ha quindi la sua esatta lunghezza. La proiezione conica ha una linea di assenza di distorsione in corrispondenza del cerchio del cono che "tocca" il globo. La distorsione aumenta nelle direzioni perpendicolari a questo cerchio. In questa proiezione la proporzione degli angoli dei meridiani è rispettata. Viene chiamato *parallelo standard* quello che coincide con la linea di assenza di di-

storsione. La maggior parte delle mappe proiettive appartiene a una di queste tre classi, a seconda che si voglia eliminare la distorsione degli angoli, delle aree o minimizzare entrambi quanto possibile.

1.4.2 Rappresentazione geografica digitale

Per riuscire a rappresentare zone della superficie terrestre su strumenti digitali sono stati definiti alcuni modelli grazie ai quali è stato possibile riprodurre, in modo accettabilmente fedele, tutto il patrimonio informativo contenuto nella realtà. Ci sono due principali metodi per la conversione di porzioni tridimensionali della superficie terrestre su un ambiente in due dimensioni [11], a seconda di quali tipi di informazioni geografiche si vogliono rappresentare, è opportuno utilizzare uno o l'altro metodo:

- *Raster*: lo spazio viene diviso in una griglia di celle a cui vengono associati una serie di valori numerici e attributi. Ogni singolo elemento ha dunque, almeno, la dimensione di una cella. Sono appropriati per caratteristiche continue come elevazione, temperatura, tipo di suolo e utilizzo dei terreni.
- *Vettoriale*: tutti gli elementi sono rappresentati da un insieme di punti individuati da coordinate, connessi da linee o archi. Gli attributi sono referenziati per mezzo di identificatori univoci in tabelle. Questa rappresentazione è ottimale per elementi con confini ben definiti, come i confini politici, oppure per la rappresentazione di tratte o percorsi.

Il problema che si vuole risolvere, però, è la descrizione fedele della realtà che presenta elementi di tipologie completamente diverse fra loro ma coesistenti in una stessa area. Per ottemperare a questa necessità, i G.I.S. utilizzano il concetto di *layer* o *theme* sovrapposti. Ogni layer contiene informazioni di *tipologia differente*, rappresentati

con i modelli più adatti; sovrapponendoli gli uni agli altri si ottiene un contenuto informativo che va oltre alla somma dei singoli layer, perchè permette di correlare i dati fra loro grazie alla condivisione della medesima posizione geografica. Un esempio di questo meccanismo è dato dalla figura 1.4 dove è rappresentata la sovrapposizione di layer di tipo raster e di tipo vettoriale per andare a comporre una completa rappresentazione digitale della realtà.

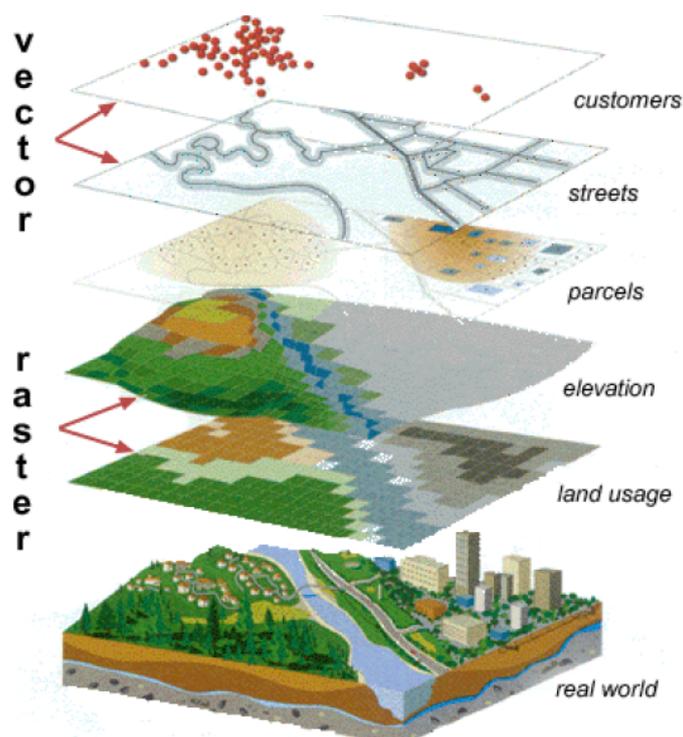


Figura 1.4: Utilizzo di layer multipli per la rappresentazione della realtà geografica [11]

Latitudine e Longitudine

Se la terra fosse sferica ogni punto della superficie sarebbe identificabile con un sistema bidimensionale di coordinate polari; un punto verrebbe localizzato da due angoli, o coordinate: la sua latitudine e

longitudine. Dato un asse polare, attorno al quale la terra ruota giornalmente, e un piano ortogonale che divide il globo a metà (l'equatore) ogni punto sulla superficie determina una latitudine, cioè il più piccolo angolo misurato dal centro della terra fra lui e il piano equatoriale, e una longitudine, l'angolo minore fra un asse arbitrario e la proiezione del punto sull'equatore determinato dalla latitudine [15] Viene quindi definita una griglia sferica di linee sulla superficie della terra, definita da cerchi posti su piani perpendicolari all'asse di rotazione: i paralleli, e da archi semicircolari che hanno l'asse come corda, detti meridiani. L'insieme dei paralleli e dei meridiani è infinito, ogni mappa ne rappresenta al più un sottoinsieme. La latitudine e la longitudine di un punto, solitamente definita in gradi decimali, definisce un incrocio fra un parallelo e un meridiano. La latitudine rappresenta quindi un angolo nord-sud del piano equatoriale e si misura convenzionalmente da 90° Sud a 90° Nord. La longitudine consiste invece nell'angolo ovest-est da un particolare meridiano convenuto come asse di riferimento e si misura da 180° Ovest a 180° Est.

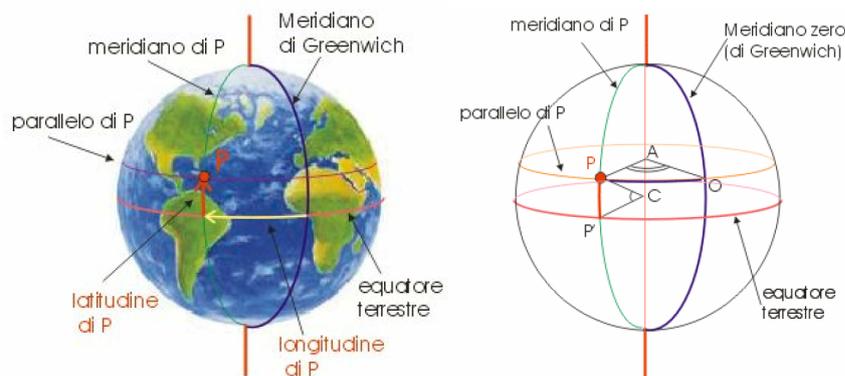


Figura 1.5: Latitudine, longitudine e assi di riferimento [16]

Datum

Per mappe precise non si può ignorare la forma ellissoidale della Terra. Un *Datum Geodetico* è un insieme di parametri che definiscono l'ellissoide di riferimento. Per ogni regione che si vuole rappresentare è fondamentale scegliere con cura il datum che rappresenta nel modo migliore le caratteristiche del territorio. La definizione dei dati contenuti in una mappa consiste nel misurare altezze e distanze da punti di riferimento in modo relativo a uno specifico datum, ciò significa che uno stesso punto reale viene indicato con coordinate diverse a seconda del datum scelto. Il confronto fra coordinate calcolate su datum diversi è facilmente eseguibile a patto che si conoscano i sistemi di riferimento di entrambe le misurazioni.

Datum significativi per l'Italia

- *Roma40*: basato sull'Ellissoide Internazionale di Hayford, orientato sulla verticale di Roma - Monte Mario. L'origine delle longitudini è sul meridiano di Monte Mario.
- *ED50 European Datum 1950*: basato sull'Ellissoide Internazionale di Hayford, orientamento medio europeo. Origine delle longitudini sul meridiano di Greenwich.
- *WGS84 World Geodetic System 1984*: è il più moderno e preferibile dei tre, è basato su uno specifico ellissoide geocentrico con schiacciamento ai poli pari a $1/298,257223563$. L'origine delle longitudini è sul meridiano di Greenwich.

WGS84 - World Geodetic System 1984

Il WGS84 [17] è il sistema di riferimento geodetico concepito per coprire tutto il globo terrestre, elaborato nel 1984 con le seguenti caratteristiche [18]:

- *centro*: centro di massa della Terra
- *asse Z*: passante per il Polo Nord, come definito dal BIH nel 1984
- *asse X*: passante per il meridiano di Greenwich, come definito dal BIH nel 1984
- *asse Y*: scelto in modo da dare una terna destrorsa, ovvero tale che un osservatore posto lungo l'asse Z veda l'asse X sovrapporsi a Y con moto antiorario, il che pone Y in Asia

Al sistema CTS WGS84 è associato l'ellissoide WGS84 mentre non ha una rappresentazione cartografica ufficiale, viene però comunemente utilizzata la rappresentazione UTM, che assume la denominazione UTM-WGS84 [19].

1.5 Architettura di un sistema di Location Intelligence

In figura 1.6 è mostrato un esempio architetturale di un sistema di location intelligence corrispondente a quello sviluppato in questo progetto di tesi. Il livello superiore rappresenta l'output del sistema, ovvero il front-end con strumenti analitici per gli end-user, mentre il livello inferiore consiste nelle fonti dati classiche da cui il sistema si alimenta. Nei livelli intermedi vi sono gli elementi tipici di un sistema di location intelligence, ovvero un Geo/Spatial DataWarehouse e un Location Intelligence Server, ovvero l'insieme dei software e delle componenti che permettono di effettuare elaborazione di dati con contenuto geolocalizzato.

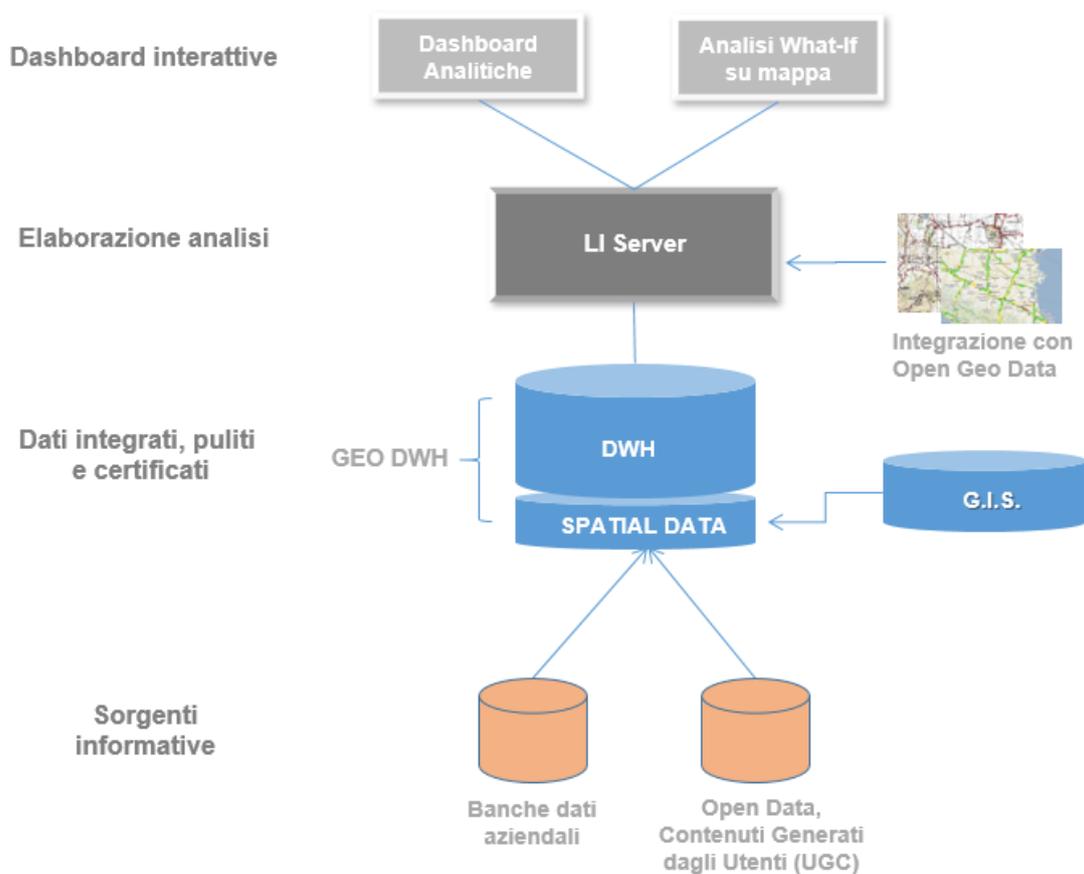


Figura 1.6: Esempio di architettura di un sistema di Location Intelligence

1.5.1 Spatial Data Warehouse

I Data Warehouse sono nati dall'esigenza di elaborare grosse moli di dati, ma, nel momento in cui la stragrande maggioranza di questi sono diventati di tipo geografico, si è avuta l'esigenza di adattare queste tecnologie all'elaborazione efficiente di questa nuova tipologia di elementi che prima venivano gestiti solo come dati alfanumerici [20]. Inizialmente, vi erano i sistemi G.I.S. che erano stati sviluppati per la gestione ottimale degli spatial-data. Questi strumenti soddisfacevano il bisogno di algoritmi e metodi di immagazzinamento e

visualizzazione dei dati geografici ma mancavano di tutte le caratteristiche proprie di un DataWarehouse: la capacità di gestire molti dati, di elaborarli in modo ottimale e di effettuare operazioni di tipo analitico (OLAP). La fusione di queste due tecnologie ha portato alla nascita degli *Spatial DataWarehouse*, che non possono essere considerati una semplice unione dei due elementi costitutivi ma bensì un prodotto di nuove tecniche per la modellazione multidimensionale, lo storage e l'interrogazione di dati di tipo geografico. Vi sono una serie di caratteristiche peculiari di questa nuova tecnologia che vanno a migliorare il concetto di DataWarehouse [21]:

- nuove tipologie di dato, ottimizzati per la memorizzazione di informazioni geografiche, non più semplicemente alfanumerici. Si veda la sezione 2.4 per approfondimenti relativi ai tipi `SDO_GEOMETRY` e `ST_GEOMETRY`;
- nuove strutture dati per l'indicizzazione, come gli R-Tree;
- nuovi algoritmi per le interrogazioni e la manipolazione di spatial-data come quelli sviluppati da Oracle [22] e trattati nella sezione 2.4;
- nuove tecniche di ETL che hanno preso il nome di *Spatial ETL* [23];
- nuove tecniche per l'elaborazione analitica dei dati, che hanno portato all'evoluzione dall'OnLine Analytics Process (*OLAP*) alla sua controparte con estensione *Spatial*, Spatial OnLine Analytics Process (*SOLAP*) [24]

1.6 Applicazioni di business

Il modo migliore per chiarire le potenzialità della location intelligence è sicuramente quello di analizzare le sue applicazioni in casi di business reali.

Nell'ambito della sanità, ad esempio, sono stati realizzati strumenti di analisi *Geo-What-If* per calcolare l'impatto della riorganizzazione dei dipartimenti ospedalieri (esempio in figura 1.7), tenendo conto dei tempi necessari alle persone per recarsi presso una struttura di ricovero e della densità e dislocazione della popolazione sul territorio. Nell'ambito della sicurezza pubblica si è riusciti nell'arduo compito di analizzare l'intero patrimonio storico dei crimini avvenuti nell'intero territorio nazionale per comprenderne le cause, le correlazioni con i luoghi dove sono avvenuti e imparare a prevenirli, utilizzando queste informazioni per l'organizzazione ottimale dei servizi di polizia e della dislocazione delle centrali sul territorio.

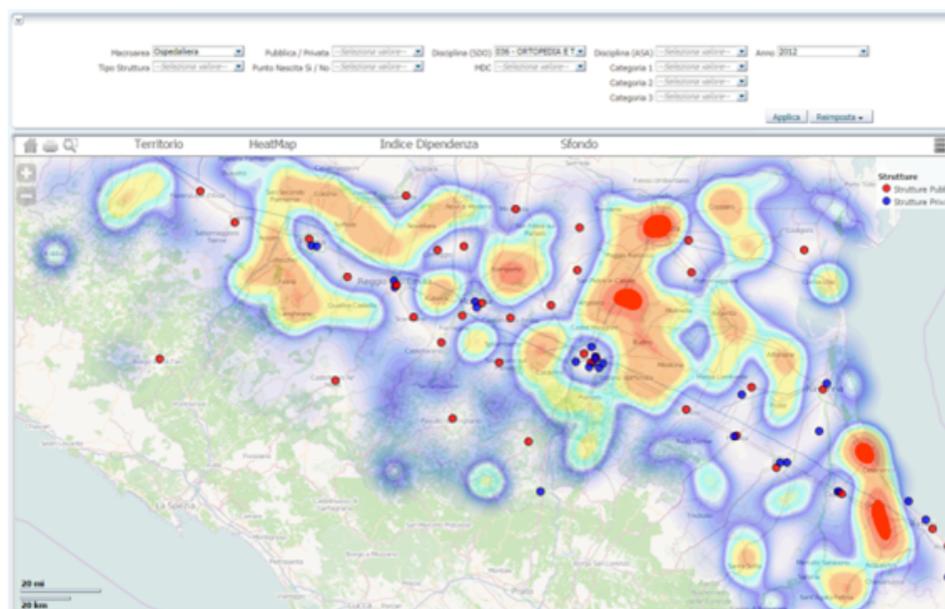


Figura 1.7: Strumento di analisi what-if relativo a strutture ospedaliere

Integrando sistemi di DataWarehouse, sensoristica IoT e strumenti di location intelligence si sono ottenuti importanti risultati nell'analisi della viabilità stradale per prevenire incidenti, ottimizzare la manutenzione delle infrastrutture e individuare strategie per la riduzione dell'inquinamento. Infine, come ultimo esempio, nell'ambito del marketing e del settore terziario si sono trovate applicazioni per l'indoor analysis dei negozi, permettendo di individuare correlazioni fra la collocazione delle strutture all'interno dei centri commerciali e la loro efficacia, fino al supporto alle decisioni riguardanti nuove aperture di punti vendita, basate sulle caratteristiche del territorio, della clientela potenziale e sulla presenza di sedi di aziende concorrenti (esempio in figura 1.8).

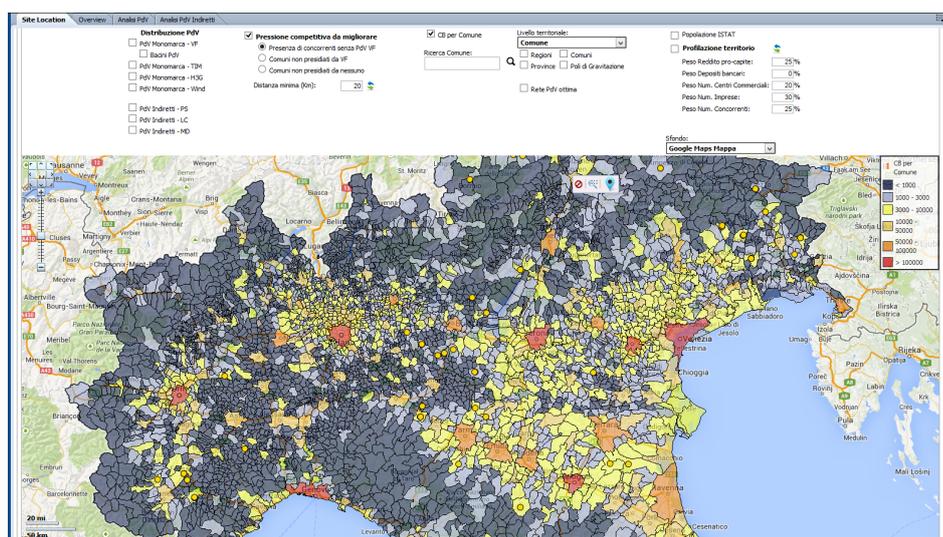


Figura 1.8: Dashboard per l'analisi delle potenzialità di mercato del territorio

Capitolo 2

Tecnologie

Sono di seguito approfondite alcune tecnologie caratterizzanti i sistemi di location intelligence ed utilizzate nell'implementazione del sistema. In figura 2.1 si fa riferimento allo stack mostrato precedentemente in figura 1.6 per meglio chiarire il collocamento delle tecnologie in esame fra i layer architetturali di un sistema di location intelligence. Oracle Business Intelligence Enterprise Edition (OBIEE) è uno strumento di front-end utilizzato per la reportistica e la visualizzazione di dashboard e strumenti di analisi su mappa. MapViewer è un componente che si colloca fra lo strato di visualizzazione grafica delle mappe e il DataWarehouse dove sono memorizzati i metadati che definiscono cosa deve essere mostrato all'utente. Oracle Spatial è il componente di Oracle Database che permette di elaborare dati di tipo geografico e infine MapBuilder è il tool visuale utilizzato per la definizione dei metadati che caratterizzano i tematismi visualizzati sulle mappe.

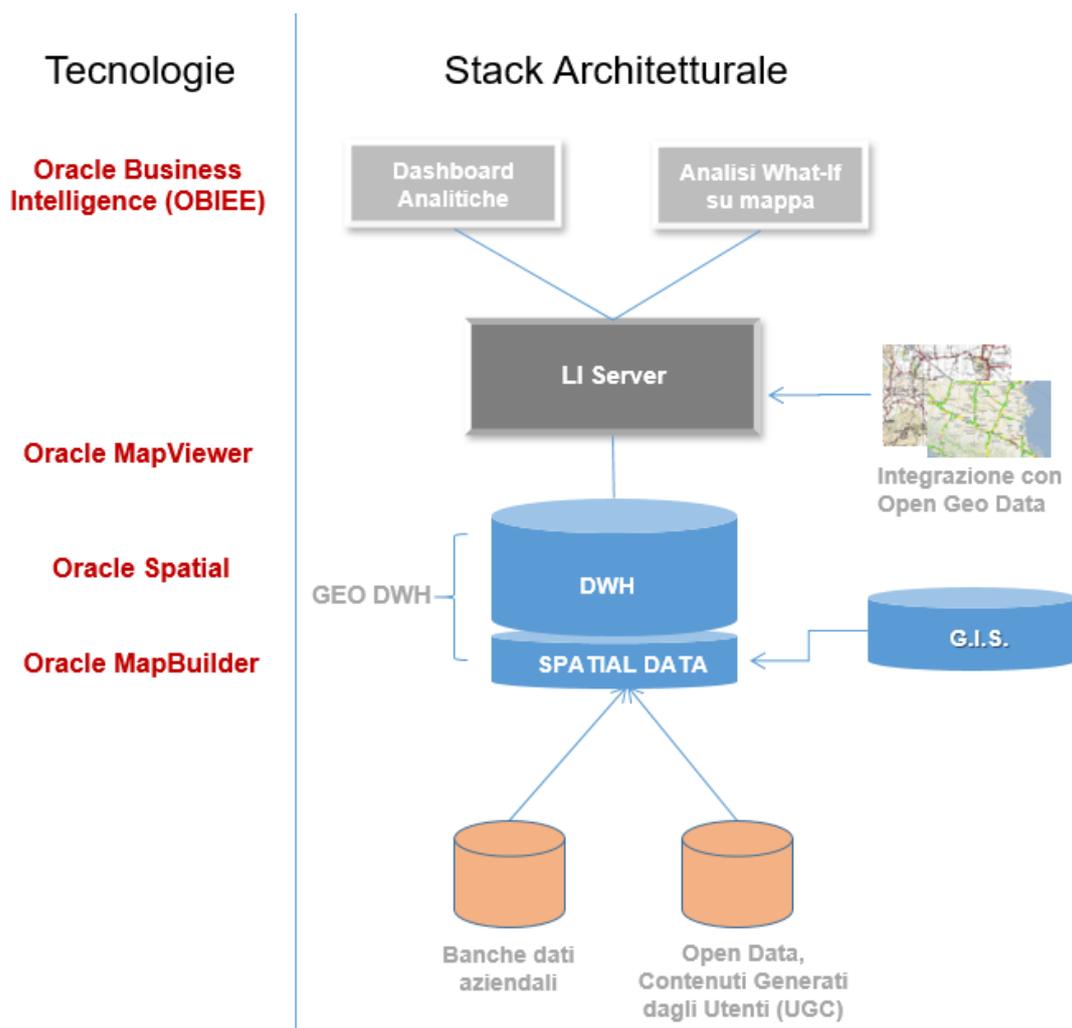


Figura 2.1: Posizionamento delle tecnologie in esame nello stack architeturale di un sistema di location intelligence

2.1 Oracle Business Intelligence Enterprise Edition

OBIEE (Oracle Business Intelligence Enterprise Edition) è un server di Business Intelligence sviluppato da Oracle. Include diversi tool di BI costruiti all'interno di un'architettura unica. Il server OBIEE fornisce un accesso centralizzato a tutte le informazioni di uno o più

DataWarehouse aziendali ponendosi in un layer di intermezzo fra le sorgenti dati e lo strato di presentazione, come illustrato in figura 2.2. L'obiettivo principale del prodotto consiste nel permettere la fruizione di grossi moli di dati per funzionalità di analisi e reporting in modo indipendente dal layer di immagazzinamento e gestione del dato.



Figura 2.2: Rappresentazione del layer OBIEE rispetto al livello dei dati e quello di presentazione

Architettura L'immagine 2.3 mostra i moduli che compongono l'architettura di un server OBIEE; di seguito si descrivono i principali illustrando le caratteristiche più importanti.

- Il sistema di Oracle BI viene denominato *Oracle BI Domain*. È costituito sia da componenti Java (deployati su WLS), sia da componenti System non-Java (eseguibili C/C++).

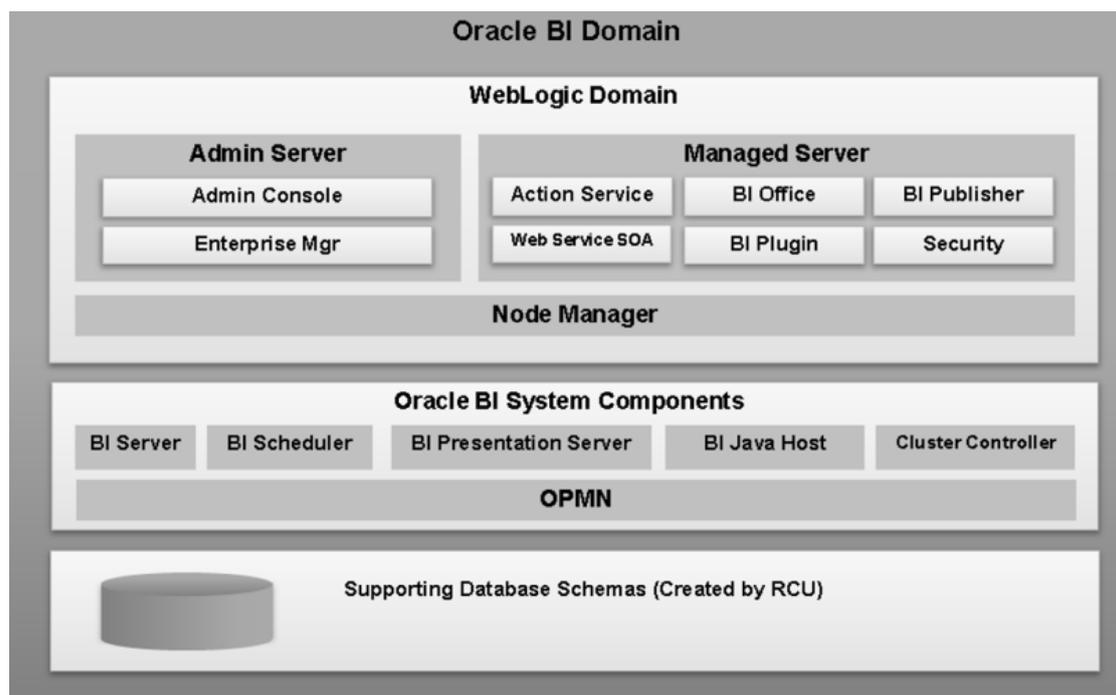


Figura 2.3: Struttura dell'architettura di OBIEE 11g

- Il *WebLogic Domain* è un insieme di istanze di WebLogic che gestiscono una o più applicazioni J2EE.
- Il *WebLogic Admin Server* è unico all'interno di un WebLogic Domain, è caratterizzato da una applicazione J2EE che svolge il ruolo di *Admin Console* per la gestione di funzionalità trasversali a tutto il dominio WebLogic. In aggiunta è presente *Oracle Enterprise Manager*, che fornisce funzionalità di gestione del sistema e di monitoraggio su tutto il dominio OBI.
- Il *Managed Server* contiene tutti i componenti delle applicazioni J2EE deployate.
- Gli *Oracle BI J2EE Components* sono i seguenti:
 - *Action Service* serve le richieste avanzate dall'Action Framework;

- *BI Office* supporta l'integrazione con MS Excel/PowerPoint tramite il plugin BI Office;
 - *BI Publisher* applicazione BI Publisher;
 - *Web Service SOA* serve le richieste che provengono ad OBIEE via WS;
 - *BI Plugin* è il principale componente dell'applicazione di BI Oracle;
 - *Security* serve le richieste di sicurezza di OBIEE;
- Il *Node Manager* è un demone che si occupa di garantire le funzionalità di start, stop, riavvio e monitoraggio remoto dei Managed Server. Equivale a OPMN ma per i componenti Java. È un processo unico per ogni WebLogic Domain.
 - Gli *Oracle BI System Components* sono gli stessi processi della versione 10g, a parte l'introduzione di *Oracle Process Manager Network* (OPMN) che si occupa dello start/stop/ping remoto dei System Components. *OPMN* si controlla da riga di comando o da Enterprise Manager.
 - Il BI Server è stato progettato per rappresentare una soluzione scalabile ed efficiente alle interrogazioni degli utenti. Utilizzando il *Common Enterprise Information Model* elimina la necessità per gli utenti di business di capire i meccanismi che regolano l'archiviazione fisica e l'interrogazione (a basso livello) dei dati (si veda figura 2.4).

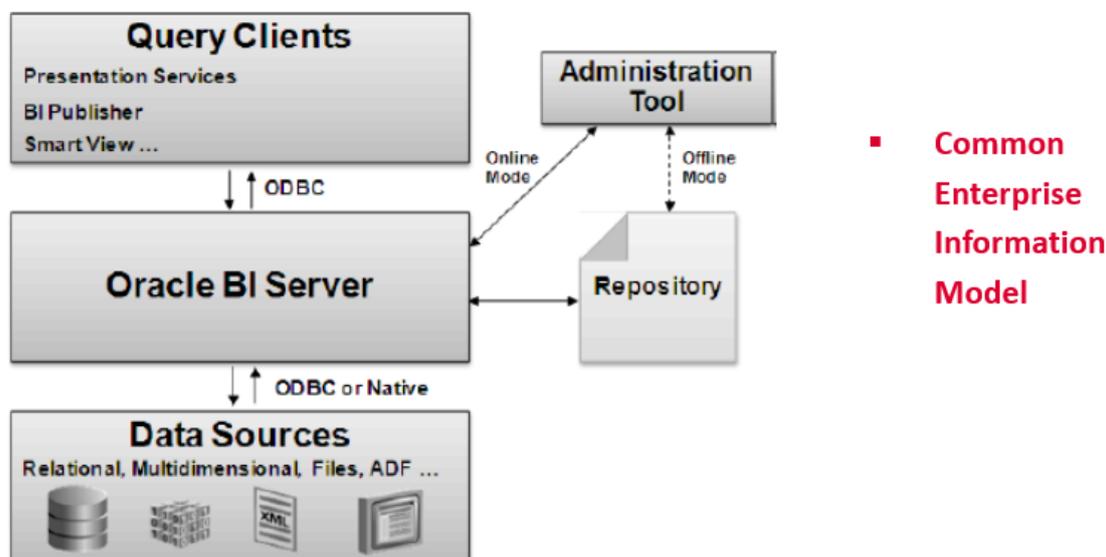


Figura 2.4: Architettura del BI Server basata sul Common Enterprise Information Model

2.2 MapViewer

Oracle MapViewer è uno strumento che si occupa del rendering di mappe interfacciandosi direttamente con i tipi di dato `SDO_GEOMETRY` gestiti da Oracle Spatial, oppure da Oracle Locator, nascondendo la complessità dovuta alla presenza di query spaziali e supportando notevoli livelli di personalizzazione pensati per un'utenza avanzata [25]. MapViewer necessita di un application server su cui essere distribuito: quello più naturale è Oracle WebLogic Server, tuttavia si professa indipendente dalla piattaforma, per cui è in grado di supportare anche altri application server come Apache Tomcat. MapViewer è un componente server-side scritto in Java incluso nel pacchetto Oracle Application Server, dedicato agli sviluppatori di applicazioni piuttosto che agli utenti finali, i cui principali componenti sono mostrati in figura 2.5.

- Map-rendering engine: si tratta di una servlet chiamata SDO-

VIS che restituisce mappe richieste dalle applicazioni client in un formato appropriato, sovrapponendo tra loro uno o più layer (o temi): questa operazione viene chiamata overlay (si veda la sezione 1.4.2). Ogni theme rappresenta un raggruppamento logico di oggetti spaziali (detti feature), ad esempio strade, residenze dei clienti, o fiumi; ogni feature viene rappresentata seguendo regole definite da specifici stili. L'applicazione client comunica con la servlet di MapViewer usando il protocollo HTTP adottando un modello request/response in cui i messaggi sono codificati in XML, mentre l'engine si collega al database Oracle mediante una connessione Java Database Connectivity (JDBC).

- Definizione delle mappe: sono i metadati che descrivono l'aspetto di ogni mappa e le regole di stile che vengono applicate ad ogni tematismo. I dati sono memorizzati nel database Oracle.
- Application Programming Interface (API): permettono di accedere alle funzionalità di MapViewer da diversi ambienti di sviluppo, in particolare sono presenti le interfacce per XML, Java e JSP, PL/SQL, e JavaScript (Ajax).
- Oracle Map Builder: è un tool grafico per gestire i metadati delle mappe.

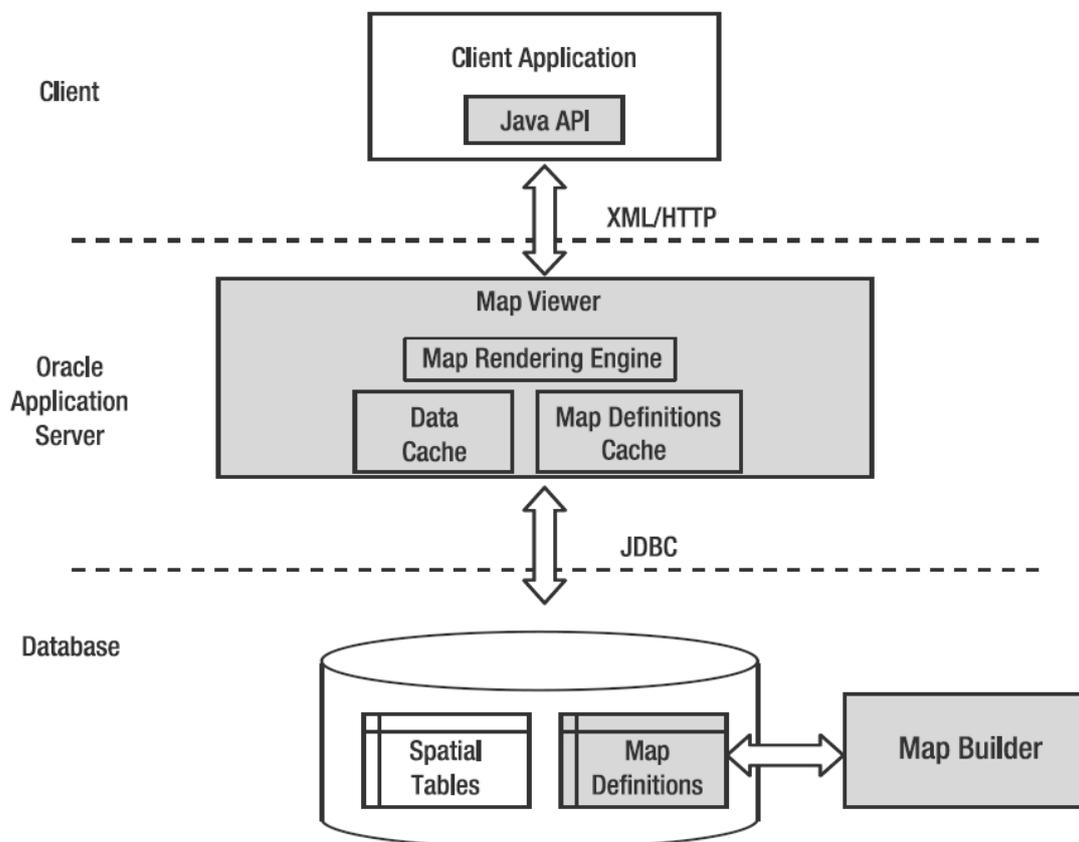


Figura 2.5: Architettura di Oracle MapViewer [26]

2.3 Oracle MapBuilder

Oracle MapBuilder è un'applicazione standalone, scritta in Java, che si interfaccia direttamente con il database Oracle per permettere la creazione, modifica ed eliminazione di tutti i metadati geografici necessari per la visualizzazione di dati spatial, tutto attraverso un'interfaccia grafica intuitiva che non richiede l'utilizzo diretto del codice XML altrimenti necessario.

I metadati possono definire temi composti da forme e colori semplici e statici o combinazioni di essi, chiamati stili avanzati, mostrati in figura 2.6; questi ultimi, oltre a essere una combinazione degli ele-

menti di base sopra citati, hanno l'importante capacità di variare la grafica visualizzata all'utente finale in base a valori definiti in colonne diverse da quella di tipo spatial, così da permettere la realizzazione di mappe dinamiche che riescono a rappresentare meglio determinati fenomeni.

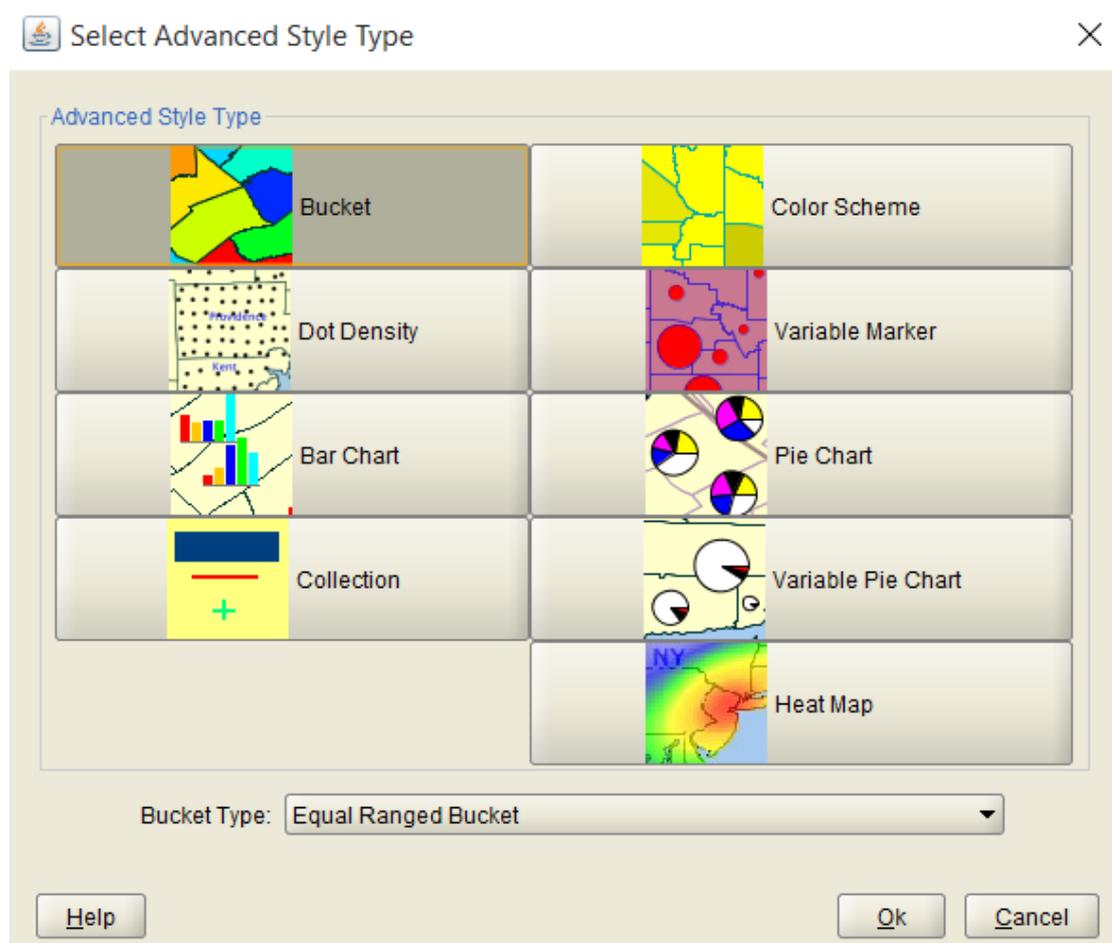


Figura 2.6: Tipologie di stili avanzati realizzabili con Oracle MapBuilder

2.4 Oracle Spatial

Oracle Spatial è un componente aggiuntivo per database Oracle che permette la gestione nativa di dati geografici, potendoli relazionare con tutti gli altri tipi di dato presenti sul database e fornendo la possibilità di interrogarli con SQL. Una delle caratteristiche peculiari di Oracle Spatial è il fatto di garantire la compatibilità con le funzioni standard SQL/MM e aggiungere un ulteriore set di strumenti proprietari denominati, genericamente, *SDO_Geometry*.

2.4.1 Tipi di forme geometriche in Oracle

Oracle permette la gestione di forme geometriche in 2 e 3 dimensioni, ogni tipologia di forma “base” può essere a sua volta compresa in una collezione che può essere omogenea (solo oggetti dello stesso tipo base) o eterogenea. Le 5 macro categorie di forme base sono le seguenti (si veda figura 2.7):

- *punto*: la forma geometrica più semplice, può rappresentare un punto nello spazio. La sua peculiarità consiste nel poter essere definito in modi diversi pur rappresentando lo stesso tipo di informazione, in pratica, si può definire un punto come “forma di tipo punto” oppure come una linea composta da una sola coordinata;
- *linea*: connette una serie di punti (o vertici); consiste in una serie di coordinate, ordinate, connesse fra loro da linee rette o da archi, o da una combinazione di questi;
- *poligono*: consiste in una linea che ha la particolarità di avere il primo e l’ultimo vertice uguali, di conseguenza si tratta di una superficie chiusa che è caratterizzata anche da un’area. Un poligono può avere al suo interno dei “buchi” che vengono

esclusi dal calcolo dell'area e dalla superficie compresa nella shape. In un piano tridimensionale, un poligono si definisce superficie poligonale;

- *solido*: un solido semplice è caratterizzato da una superficie esterna e zero o più superfici interne, lo spazio sotteso fra le superfici esterne e quelle interne definisce lo spazio del solido. Un solido, oltre a un perimetro e un'area, è caratterizzato anche da un volume;
- *composizione*: è un insieme di un numero qualsiasi di forme delle tipologie precedenti, è importante ricordare che in questo caso si avranno una serie di valori di perimetri, aree e lunghezze, ma si potranno eseguire calcoli e operazioni sull'intero set di forme in modo aggregato e unificato;

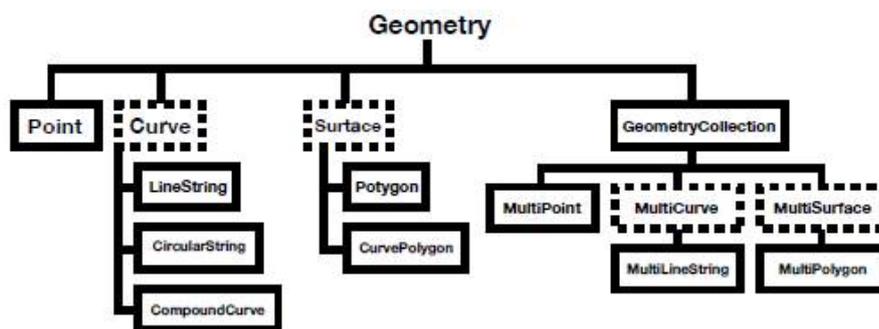


Figura 2.7: Tipi di forme geometriche [27]

2.4.2 Tipo di dato SDO_Geometry

Oracle Spatial definisce un nuovo tipo di dato proprietario, definito *SDO_Geometry* con cui gestisce le informazioni georeferenziate, con il supporto di opportuni metadati. Questo datatype consiste in un BLOB (Binary Large Object) strutturato nei seguenti campi interni:

- *SDO_GTYPE* definisce il tipo di shape;
- *SRID* definisce il sistema di coordinate di riferimento con cui è stata specificata la shape;
- *SDO_POINT* nel caso in cui la shape sia di tipo “punto”, le informazioni sono specificate in questo campo;
- *SDO_ORDINATE* array che contiene tutte le coordinate (coppie o triplette) della shape, ordinate;
- *SDO_ELEM_INFO* attributo che specifica a quale parte della shape fanno riferimento le coordinate salvate nel campo *SDO_ORDINATES*, come vengono connessi i punti fra loro (linee rette o archi) e se la shape è di tipo “punto” (vi sono infatti diversi modi di rappresentare una shape di tipo punto), linea o poligono;

Il frammento di codice 2.1 mostra un esempio di dichiarazione di un elemento *SDO_GEOMETRY*.

```

1 MDSYS.SDO_GEOMETRY(
2   2003, — poligono (3) di due dimensioni (2)
3   3785, — SRID, il sistema di riferimento
4   NULL, — Se fosse un punto (1001 nel primo campo) si potrebbero indicare qui
         le coordinate
5   MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3), — a partire dalla prima coordinata, si
         definisce un rettangolo
6   MDSYS.SDO_ORDINATE_ARRAY(1,1, 5,7) — Nel caso specifico del rettangolo, sono
         sufficienti due coordinate (per ognuna, X e Y del sistema cartesiano) che
         definiscono l'angolo in basso a sinistra e in alto a destra. L'ordine con cui
         sono specificati X e Y sono fissi
7 )

```

Listing 2.1: Esempio di dichiarazione di un dato *SDO_Geometry*

SDO_POINT L'attributo *SDO_POINT* è valorizzato solo nel caso si stia descrivendo un oggetto di tipo punto. Nella pratica rappresenta una modalità alternativa di specificare i dati relativi a una

shape, valida solo per i punti. Nel caso in cui si utilizzi questo attributo, i campi SDO_ORDINATE e SDO_ELEM_INFO saranno *NULL* e si potranno utilizzare le funzioni che gestiscono il sottotipo SDO_POINT_TYPE. Un esempio di creazione di una shape di questo tipo è visibile nell'esempio di codice 2.2.

```
1 INSERT INTO geometry_examples (name, description , geom) VALUES
2 (
3     --Punto bidimensionale di coordinate (-79,37) con SRID 8307
4     SDO_GEOMETRY
5     (
6         2001, --SDO_GTYPE formato: D00T. Impostato a 2001 per un punto bi-
7         dimensinonale
8         8307, --SDO_SRID (sistema di riferimento)
9         SDO_POINT_TYPE
10        (
11            -79  -- coordinata di Longitudine
12            37,  -- coordinata di Latitudine
13            NULL -- in questo caso abbiamo solo 2 dimensioni
14        ),
15        NULL,
16        NULL
17    );
```

Listing 2.2: Esempio di creazione di una shape di tipo *punto* utilizzando il campo SDO_POINT

SRID L'attributo SRID specifica il sistema di riferimento, o sistema di coordinate, utilizzato per la specifica forma geometrica. Questo parametro è fondamentale perchè una stessa shape sarà rappresentata in luoghi diversi della mappa geografica a seconda dello SRID specificato. È inoltre importante tenere in considerazione la scelta del sistema di riferimento in base ai calcoli e alle elaborazioni spatial che verranno applicate sui dati. Dal momento che i diversi sistemi di riferimento utilizzano unità di misura e precisioni molto diverse, spesso risulta utile mantenere i dati ad uno specifico SRID durante la parte di elaborazione e, solo nella fase finale di visualizzazione su front-end, convertire i dati risultanti nel sistema di riferimento utilizzato.

Gli SRID principali da tenere in considerazione sono:

- *8307 Longitude / Latitude (WGS 84)*: l'unità di misura è il grado, è usato nei sistemi GPS;
- *3785 Popular Visualisation CRS / Mercator*: l'unità di misura è in metri, è l'EPSG [28] più comunemente utilizzato, se si vogliono mostrare delle shape sovrapposte a dei tile di Google Maps, per esempio, è necessario utilizzare questo SRID o il 3857;
- *3857*: altro sistema di riferimento che utilizza la proiezione di Mercatore, utilizzata da servizi come Google Maps e OpenStreetMap;

Conversione fra SRID Per convertire un elemento geometrico da uno SRID all'altro, su Oracle sono fornite alcune funzioni da applicare ai dati [29]:

- *SDO_CS.TRANSFORM*: traduce un singolo elemento spatial in una nuova versione con le stesse coordinate tradotte nel sistema di riferimento (SRID) target;
- *SDO_CS.TRANSFORM_LAYER*: esegue la stessa operazione della precedente ma applicato a un intero *layer*, ovvero a un'intera colonna di tipo Geometry;
- *SDO_CS.VIEWPORT_TRANSFORM*: trasforma un rettangolo ottimizzato in un poligono valido per essere utilizzato con le funzioni di tipo *spatial*;

2.4.3 SQL/MM e OGC

SQL/MM è lo standard internazionale ISO/IEC per “testo, dati spatial e data mining” [26]. L'Open Gis Consortium (OGC [30]) ha

definito il “Simple Features Specification” [31] con lo scopo di standardizzare l’immagazzinamento, l’interrogazione e l’aggiornamento di elementi di tipo georeferenziato. SQL/MM “Part 3” [32] è un sottoinsieme dello standard che tratta nello specifico i dati di tipo Spatial *bidimensionali* e le relative funzioni per gestire, elaborare e modificare questo tipo di dato. Per quanto riguarda i dati tridimensionali, viene utilizzato il Geographic Markup Language (GML) di tipo 3 [33]. Questo standard specifica il tipo *ST_Geometry*, con i relativi sottotipi, che rappresentano l’implementazione usata per gestire dati geografici di qualsiasi tipologia (punti, linee, poligoni, ecc.). Oracle Spatial supporta completamente questo standard e ha reso il datatype *SDO_Geometry* intercambiabile con gli elementi di tipo *ST_Geometry*, garantendo una completa compatibilità e convertibilità da uno standard all’altro.

Oltre al tipi-dato, Oracle supporta anche tutti i set di funzioni dello standard OGC, quindi a seconda dell’utilizzo di *SDO_Geometry* o *ST_Geometry*, sarà possibile svolgere approssimativamente lo stesso tipo di operazioni e manipolazioni dei dati, eventualmente con alcune aggiunte fornite allo standard proprietario Oracle. Nell’esempio di codice 2.3 si può vedere come dichiarare elementi di tipo *ST_Geometry* [34] su Oracle Database.

```
1 CREATE TABLE geoms (  
2   id integer,  
3   geometry sde.st_geometry  
4 );  
5  
6 INSERT INTO GEOMS (id, geometry) VALUES (  
7   1901,  
8   sde.st_geometry ('point (1 2)', 4326)  
9 );  
10  
11 —Pe rinserire lo stesso punto utilizzando un metodo ottimizzato:  
12 INSERT INTO GEOMS (id, geometry) VALUES (  
13   1901,  
14   sde.st_geometry (1,2,null,null,4326)  
15 );  
16  
17 INSERT INTO GEOMS (id, geometry) VALUES (  
18   1902,
```

```
19 sde.st_geometry ('linestring (33 2, 34 3, 35 6)', 4326)
20 );
21
22 INSERT INTO GEOMS (id, geometry) VALUES (
23 1903,
24 sde.st_geometry ('polygon ((3 3, 4 6, 5 3, 3 3))', 4326)
25 );
```

Listing 2.3: Esempio di dichiarazione di un dato ST_Geometry in Oracle

2.4.4 Tabelle dei Metadati

Per la gestione dei dati spatial, Oracle utilizza uno schema (MD-SYS) che contiene i metadati per lo memorizzazione, la sintassi e la semantica dei tipi dato geometrici supportati, oltre agli indici, di tipo R-Tree, utilizzati per le query applicate a questo tipo di dato. Nello schema che contiene le tabelle con i dati si trovano alcune viste di metadati standard di fondamentale importanza:

- *USER_SDO_GEOM_METADATA*:
 - rappresenta l'effettiva memorizzazione su disco dei dati geometrici della tabella;
 - è necessaria per poter visualizzare dati spaziali con qualsiasi strumento;
 - lo SRID deve essere corrispondente a quello della tabella;
 - è di *fondamentale* importanza popolare adeguatamente il campo *DIMINFO* di questa tabella di metadati per ogni tabella contenente un campo Geometry. Questa colonna contiene informazioni sugli valori estremi delle coordinate e la tolleranza. Senza questa operazione si incorre nel rischio di avere risultati inconsistenti applicando funzioni di

tipo spatial sui dati, oltre a un ingente degrado delle prestazioni. Il codice 2.4 mostra un esempio di inserimento di metadati per una tabella;

- *USER_SDO_INDEX_METADATA*: contiene i dati degli indici R-Tree delle tabelle;
- *SDO_INDEX_TABLE*: spatial Index Table che contiene i dati di un indice R-Tree. Le tabelle degli indici non vanno trattate come normali tabelle di Oracle ma gestite solo tramite le opportune funzioni;

Nell'esempio di codice 2.4 viene mostrato, a titolo esemplificativo, l'inserimento dei metadati relativi a una tabella che include un campo di tipo geometrico.

```

1 INSERT INTO USER_SDO_GEOM_METADATA VALUES
2 (
3   'CUSTOMERS' — NOME TABELLA
4   , 'LOCATION'  — NOME DELLA COLONNA SPATIAL
5   , SDO_DIM_ARRAY — Attributo DIMINFO che contiene i punti estremi delle
6     coordinate e il valore di tolleranza
7     (
8       SDO_DIM_ELEMENT
9       (
10        'LONGITUDE', — valore DIMENSION NAME per la prima dimensione
11        -180,       — SDO_LB, lowerbound della dimensione
12        180,       — SDO_UB, uppdebound della dimensione
13        0.5        — Tolleranza di 0.5 metri
14      ),
15      SDO_DIM_ELEMENT
16      (
17        'LATITUDE', — valore DIMENSION NAME per la prima dimensione
18        -90,       — SDO_LB, lowerbound della dimensione
19        90,       — SDO_UB, uppdebound della dimensione
20        0.5       — Tolleranza di 0.5 metri
21      )
22    ),
23    8307 — SRID che specifica il sistema di coordinate
24 );

```

Listing 2.4: Esempio di inserimento dei metadati di una tabella con un campo Geometry

È fondamentale ricordare che quando si vuole stimare l'occupazione di spazio su DB di tabelle Spatial, non è sufficiente calcolare

l'occupazione in blocchi della tabella stessa ma bisogna tenere in considerazione lo schema MDSYS e una serie di oggetti utilizzati per la gestione dei metadati, salvati sullo schema dove sono generate le tabelle con i dati. Una possibile soluzione consiste nell'interrogare la vista di sistema *DBA_ALL_OBJECTS*.

2.4.5 R-Tree

Per costruire gli indici sulle colonne di tipo *spatial*, al posto dei classifici B^+ -Tree o indici binari utilizzati normalmente sui dati tabellari, vengono utilizzati gli R-Tree [35]. Gli R-Tree sono estensioni dei B^+ -Tree a spazi multi-dimensionali [36], ad esempio le coordinate spaziali (X,Y) (un esempio in figura 2.8).

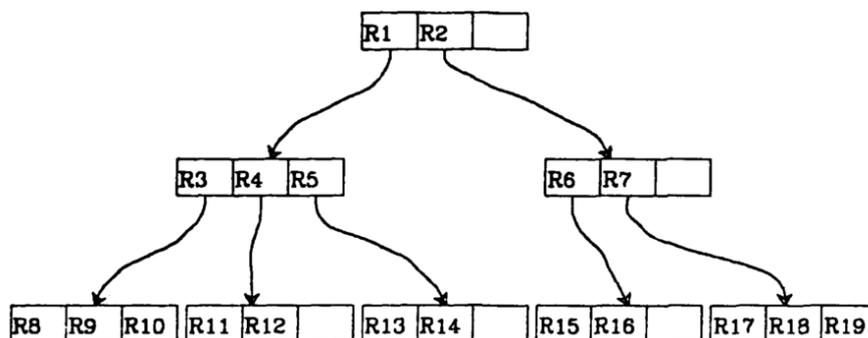


Figura 2.8: Esempio di albero di un R-Tree [37]

Mentre un B^+ -Tree organizza gli oggetti in insiemi di intervalli mono-dimensionali non sovrapposti in modo ricorsivo dalle foglie alla radice, la struttura dati degli R-Tree divide lo spazio in MBR (minium bounding rectangles, dà il nome agli R-Tree), innestati gerarchicamente e, quando possibile, sovrapposti (un esempio in figura 2.9).

Ogni nodo dell'R-Tree ha un numero variabile di entry (fino a un massimale predeterminato), ogni entry che non sia un nodo foglia contiene due entità, una identifica il nodo figlio, l'altra l'MBR che

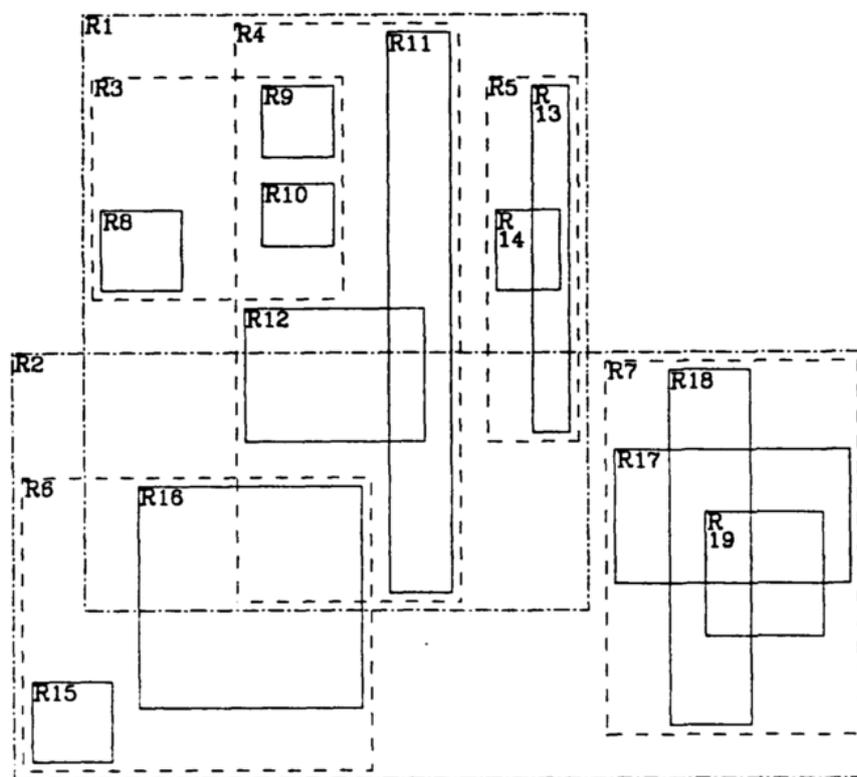


Figura 2.9: Esempio di creazione degli MBR in un R-Tree [37]

contiene tutte le entry del nodo figlio. L'algoritmo di inserimento e cancellazione di entry dagli MBR assicura che elementi "vicini" siano posizionati nello stesso posto/nodo foglia, un eventuale nuovo elemento sarà posizionato nel nodo foglia che richiede il minor numero di estensioni delle dimensioni dell'MBR.

Capitolo 3

Progetto

In questo capitolo verranno approfondite le fasi di progettazione e sviluppo del sistema. Inizialmente verrà fornita una breve descrizione del dominio di business in cui si configura il progetto, per poi trattare la fase di analisi dei requisiti e di progettazione della soluzione. Infine, si descriveranno le modalità utilizzate per la fase di test e si illustreranno i possibili sviluppi futuri.

3.1 Obiettivi di Progetto

Il committente ha evidenziato la necessità di adottare soluzioni analitiche nell'ambito dei servizi ambientali che, attraverso la rappresentazione cartografica del dato, siano in grado di produrre opportune dashboard attraverso cui verificare l'andamento dei servizi, analizzare e simulare le possibili anomalie, proporre efficientamenti o miglioramenti della qualità dei servizi ed essere maggiormente competitivi nelle proposte economiche per le gare di affidamento. Il settore specifico a cui si applicano le analisi è quello dell'organizzazione della raccolta dei rifiuti solidi urbani.

Il dominio di business comprende alcuni concetti fondamentali che è utile chiarire: un *percorso pianificato* o *route* è il tragitto stra-

dale predefinito per uno specifico turno di raccolta dei rifiuti per un determinato giorno della settimana; esso collega una serie di *punti di raccolta* (PdR) in un ordine prestabilito. Un PdR è un'area delimitata al cui interno sono posizionati i contenitori dei rifiuti, sia monomateriale, sia di tipo indifferenziato. L'ordine con cui sono collegati i punti di raccolta e la loro suddivisione in route diverse sono stati definiti in base a modelli di ottimizzazione. Ogni route è specifico di una singola classe di materiale (ad esempio carta, plastica, organico, rifiuti indifferenziati) per cui è implicito che ogni punto di raccolta presente in un percorso pianificato contenga uno o più contenitori di quella specifica tipologia. Un *ordine di servizio* (OdS) rappresenta l'esecuzione dello svuotamento di tutti i contenitori elencati dal percorso pianificato, nell'ordine indicato.

Nello specifico, si è condivisa l'esigenza di introdurre sistemi di Location Intelligence che abilitino le seguenti funzionalità:

- Arricchire con la componente Geografica le analisi delle performance del servizio
 - heatmap che segnalano le aree con la maggior incidenza degli eventi caratteristici del servizio (mancati svuotamenti, velocità media, lunghezza e durata degli OdS, ecc.);
 - analisi degli eventi caratteristici effettuata secondo specifici parametri per consentire l'individuazione di soluzioni per efficientare i servizi e ridurre i costi individuando le anomalie (ad es. lunghezze/tempi effettivi maggiori di quelli pianificati), le relative cause e le zone in cui si verificano;
- Consentire l'integrazione di sorgenti dati esterne (ISTAT, Nielsen, ecc.) per estendere le possibilità di analisi:

- individuazione delle aree nelle quali l'impatto di eventuali anomalie al servizio è maggiore per la presenza di elementi caratteristici del territorio (ad es. caratteristiche geografiche, demografiche, ecc.);
- analisi dell'andamento dei servizi erogati nel tempo e nello spazio per calcolarne le evoluzioni più efficaci con strumenti come l'analisi what-if e gli algoritmi predittivi;

Si è deciso di realizzare un progetto su una ristretta parte dei dati a disposizione che contenesse alcune proposte di funzionalità proprie di strumenti di location intelligence così da aiutare il committente a definire meglio le proprie esigenze.

Lo scope del progetto può quindi riassumersi nei seguenti punti:

- Trasmettere le potenzialità di uno strumento di location intelligence attraverso l'utilizzo di dati di proprietà del committente e quindi mappato sul suo dominio.
- Dimostrare la facilità d'utilizzo.
- Evidenziare un bisogno e una mancanza all'interno dell'attuale infrastruttura tecnologica aziendale, dimostrando la fattibilità di analisi non realizzabili con gli strumenti preesistenti.
- Realizzare un prodotto da utilizzare come embrione nello sviluppo e ideazione di una soluzione completa.

3.2 Requisiti dell'utente

Un requisito è definito come una *capacità* richiesta da un utente (stakeholder) per risolvere un problema o raggiungere un obiettivo [38, 39]. Per definire i requisiti di un progetto è fondamentale riuscire ad aiutare il cliente a distinguere fra quelli che sono i suoi desideri e

i suoi reali bisogni [40]. Nella fase di raccolta dei requisiti è necessario quindi definire questa disambiguazione e comprendere quali sono le relative aspettative del committente sul risultato del progetto. È esattamente durante questo processo che si è individuata la necessità di fornire uno strumento al cliente che potesse permettergli di comprendere meglio quali fossero le potenzialità della location intelligence, in modo tale da arrivare a una successiva iterazione progettuale e definire in modo più mirato i requisiti finali della soluzione.

Per quanto riguarda le *funzionalità* da realizzare nell'ambito del progetto, si è deciso di concentrarsi su due principali requisiti di business: l'analisi delle anomalie sui percorsi effettuati per lo svuotamento dei rifiuti e delle problematiche riguardanti lo svuotamento dei contenitori dislocati sul territorio. Si vogliono quindi fornire una serie di KPI atti a valutare la situazione attuale di una specifica area in carico all'azienda committente, potendo inoltre confrontare le anomalie con alcune caratteristiche tipiche dei territori in cui avvengono come, ad esempio, la popolazione e la presenza di specifici punti di interesse come scuole e ospedali. I dati analizzati faranno riferimento a un periodo temporale di sei mesi scelti dal committente, con la possibilità di filtrare le analisi in base al tempo e ad alcune caratteristiche proprie dell'organizzazione logistica dell'azienda.

3.3 Analisi Funzionale

Il documento di analisi funzionale, con l'accezione utilizzata in questa specifica metodologia di lavoro, è l'output di una serie di riunioni con il cliente che hanno l'obiettivo di definire le funzionalità che verranno implementate, senza definire il *come* verrà implementata la soluzione. L'input di questa fase è l'output della fase di analisi dei requisiti, si vuole infatti verificare e assicurare che il committente abbia compreso cosa verrà fatto all'interno del progetto [41], per de-

finirne al meglio lo scope e cautelare entrambe le parti durante la fase di consegna del prodotto. Durante la fase di user acceptance test si utilizza infatti il documento di analisi funzionale per certificare che tutte le funzionalità definite inizialmente siano state effettivamente implementate e risultino funzionanti. Questa parte del processo è inoltre estremamente utile per dimensionare le aspettative del committente che deve avere perfettamente chiaro cosa è realizzabile e cosa non lo è, così da scongiurare l'eventualità di scoprire, solo in fase di rilascio, che la soluzione non porta il beneficio sperato.

3.3.1 Definizione KPI

Basandosi sull'analisi dei requisiti, si sono definiti i seguenti KPI

- *Criticità lunghezza percorsi*: l'indicatore è calcolato ottenendo la differenza fra la lunghezza del percorso consuntivato e quella del percorso pianificato, proporzionata su quest'ultimo valore, in forma percentuale.

$$\frac{Km_{Ods} - Km_{Pianificato}}{Km_{Pianificato}} * 100 \quad (3.1)$$

- *Criticità discostamento percorsi*: questo valore è definito calcolando due valori di area distinti:
 1. valore dell'area compresa fra la linea tracciata dal percorso effettivo e quella del percorso pianificato;
 2. valore dell'area del quadrato avente come angoli estremi l'inizio e la fine del tratto analizzato, calcolando la diagonale con la distanza euclidea [42];

Il primo valore risulta una misurazione, in Km^2 , del discostamento, punto a punto, fra i due percorsi, sfruttando la logica

del calcolo integrale. Il secondo viene usato come valore di upperbound per ottenere un valore proporzionale, non è infatti interessante per l'utente finale capire l'esatta area calcolata nel punto 1, bensì una misura *confrontabile* per capire quali tratti si discostano maggiormente dai percorsi previsti. Il valore così ottenuto risulta una percentuale dell'area differenziale rispetto all'upperbound.

$$\frac{\left| \int_S^E \phi_{OdS} dx - \int_S^E \phi_{Pianificato} dx \right|}{\left(\frac{\sqrt{(Start_x - End_x)^2 + (Start_y - End_y)^2}}{2} \right)^2} * 100 \quad (3.2)$$

- *Velocità media*: il calcolo della velocità è ottenuto sfruttando i timestamp generati dalle rilevazioni GPS. Dati due punti S e E collegati da un percorso ϕ costruito dalla concatenazione di tutte le rilevazioni GPS comprese fra S e E, il calcolo della velocità è semplicemente così ottenuto:

$$\frac{\phi[S, E]}{|T_e - T_s|} \quad (3.3)$$

- *% di svuotamenti non effettuati*: questo KPI indica il numero di ordini di servizio durante i quali *non* è stato svuotato un singolo contenitore previsto nella pianificazione, rispetto al totale degli OdS nei quali è presente il contenitore. Un valore uguale a 0 significa che il contenitore è *sempre* stato svuotato ogni volta che era stato inserito in pianificazione, o che non è *mai* stato eseguito un ordine di servizio di una pianificazione che lo comprende. Un valore uguale a 100 sta ad indicare che il contenitore non è *mai* stato svuotato quando avrebbe dovuto esserlo.

$$\frac{\text{Svuotamenti contenitore } \alpha \text{ non effettuati}}{\text{Totale svuotamenti previsti del contenitore } \alpha} * 100 \quad (3.4)$$

3.3.2 Modello di analisi dei dati

In questa sezione si presenta il modello di analisi dei dati formalizzato tramite il costrutto del dimensional fact model (DFM [43]). In figura 3.1 è illustrata la struttura del fatto relativo alle performance dei percorsi consuntivati, definiti a partire dalle rilevazioni GPS dei mezzi di raccolta difiuti. La fact table presenta i valori di dettaglio riferiti al tratto compreso fra ogni coppia di punti di raccolta (PdR) consecutivi di uno specifico percorso, registrando, per ognuno, i relativi KPI che indicano quanto il percorso si è discostato da quello previsto, la differenza di lunghezza dal pianificato e la velocità media di quello specifico tratto. È utile evidenziare la convergenza presente sull'attributo Route, in quanto ogni segmento è specifico di uno specifico pianificato (Route), esattamente come ogni Ordine di Servizio (OdS) è l'istanziamento di un Route. Per questo motivo, selezionando un percorso pianificato, si implica una selezione dei segmenti e degli OdS relativi. Con il modello così strutturato, non vi è relazione diretta fra i punti di raccolta e i percorsi pianificati, bensì viene definito l'attributo segmento che definisce delle coppie, assegnate a uno specifico percorso. Di conseguenza, un punto di raccolta, può, correttamente, essere assegnato a 0, 1 o N percorsi pianificati. Non vi è ordine sequenziale fra i segmenti non risultando utile ai calcoli previsti, fornendo i risultati rappresentati su mappa si otterrà un implicito ordinamento comprensibile all'utente finale.

La figura 3.2 mostra il fatto relativo agli svuotamenti dei punti dei contenitori dei punti di raccolta. Per via delle semplificazioni al modello introdotte per la fase di proof of concept, i valori relativi a ogni PdR sono l'aggregazione di quelli di tutti i contenitori in esso

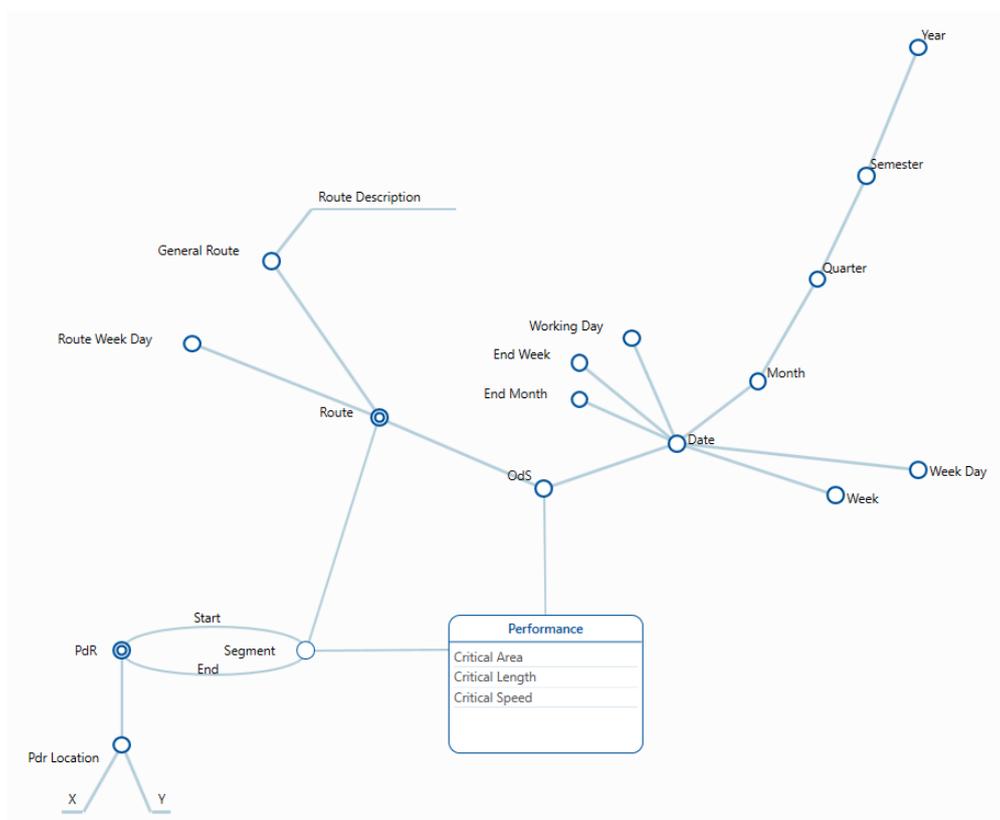


Figura 3.1: Fatto: Performance OdS

compresi. Si noti che, con questo modello, la relazione fra i PdR e i percorsi pianificati viene definita sulla base delle righe della fact table.

In figura 3.3 si vuole mostrare un focus sulla dimensione degli *ordini di servizio* che comprende tre dimensioni conformi a entrambi i fatti di questo modello:

- *OdS: Ordine di Servizio*: rappresenta l'identificativo di un turno di lavoro, l'esecuzione effettiva di un percorso pianificato. È caratterizzato da una data e da un percorso pianificato di riferimento. Nel caso in cui il giorno della settimana dell'OdS non corrispondesse a *Route Week Day* si identificherà un turno di lavoro extra eseguito fuori pianificazione.

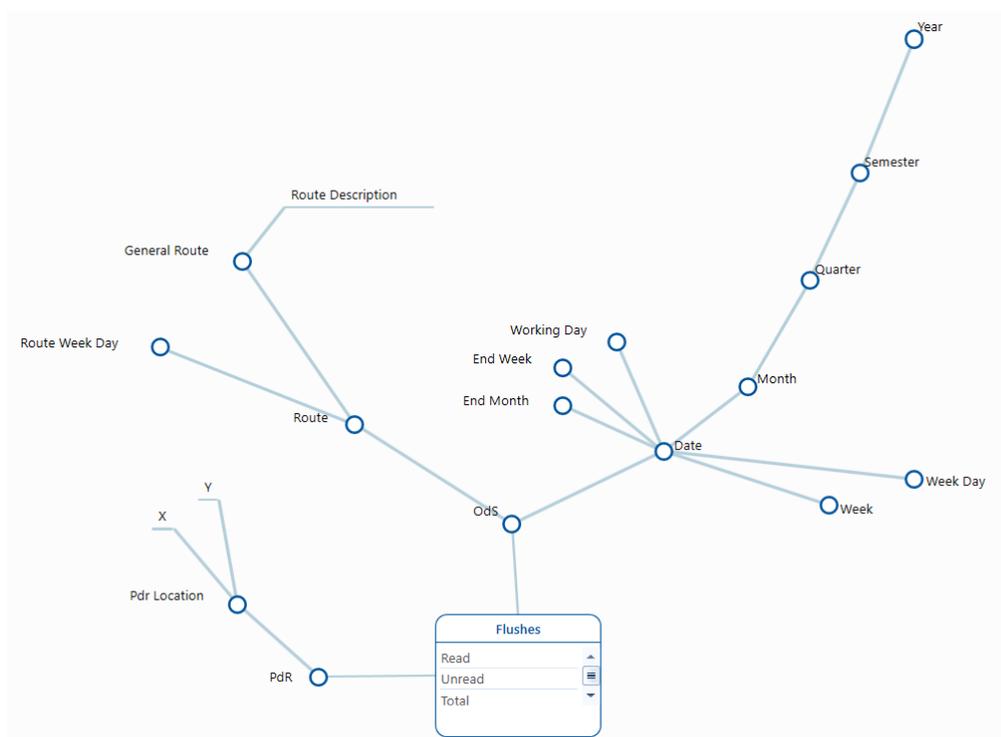


Figura 3.2: Fatto: Svuotamenti

- *Route: Percorso pianificato:* Questo attributo identifica un percorso pianificato. È composto dalla combinazione fra l'identificativo di una Route (che determina la descrizione nominale e, nel caso di modello completo, la tipologia di rifiuto raccolto e altre informazioni di anagrafica escluse dall'ambito del PoC) e il giorno della settimana in cui viene pianificato. È infatti importante evidenziare che un percorso consiste, nella realtà, di una sequenza ordinata di Punti di Raccolta ma questa sequenza varia, a parità di *General Route* a seconda del giorno della settimana in cui è programmato
- *Date: Dimensione temporale:* la dimensione temporale, con *Date* come livello 0, è collegata con rapporto 1:1 con l'attributo *OdS*. Per ottimizzare l'utilizzo di OBIEE, nell'implementazio-

ne dello star schema verrà portato direttamente in fact table la FK relativa al giorno, a causa delle peculiarità specifiche dello strumento che funziona in modo ottimale quando vi è una gerarchia di tipo temporale a livello 0

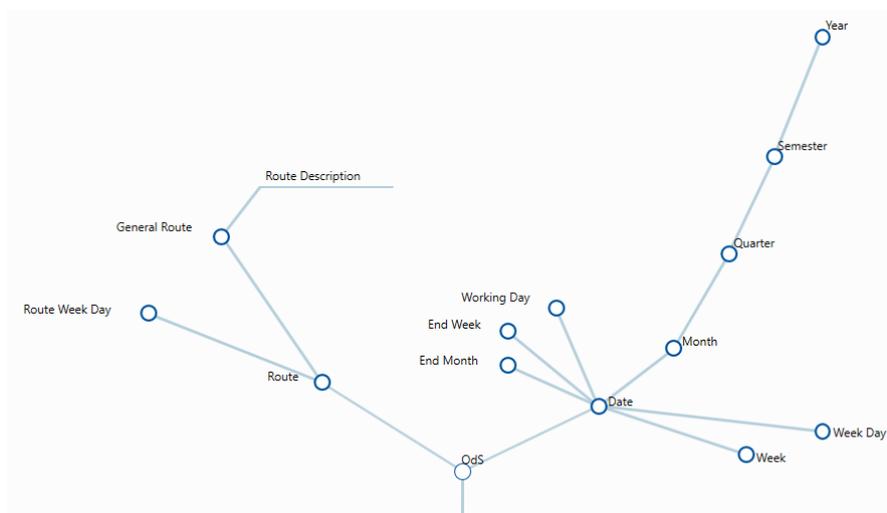


Figura 3.3: Dimensione: Ordine di Servizio

3.3.3 Data profiling

Il lavoro di data profiling è estremamente importante per poter definire precisamente quali saranno le funzionalità del prodotto, è infatti necessario comprendere quali dati si hanno a disposizione e di conseguenza quale patrimonio informativo se ne può ricavare. A seconda del risultato di questa fase si potrà quindi decidere quali funzionalità effettivamente implementare o, eventualmente, definire la richiesta di ulteriori dati al cliente specificando l'esigenza da soddisfare e per quale funzionalità sono richiesti.

Il committente ha fornito i dati prodotti dai dispositivi GPS montati su ogni mezzo di trasporto utilizzato per la raccolta dei rifiuti: è stata registrata la posizione geografica di ogni veicolo in attività, a

intervalli regolari, insieme alla data e ora della rilevazione (con dettaglio al secondo). È stata inoltre prodotto il report di dettaglio delle marcature RFID che permettono di tenere traccia degli svuotamenti realmente effettuati per ogni singolo contenitore. Si sono utilizzate le anagrafiche relative ai percorsi pianificati, agli ordini di servizio, ai punti di raccolta e ai contenitori. Nei file forniti erano inoltre espresse le correlazioni fra i dati sensoristici dei mezzi e il corrispondente turno di lavoro.

Best practices Durante l'evolversi del progetto si sono individuate alcune best practice che è necessario seguire per ottenere ingenti risparmi di tempo e minimizzare problematiche e imprevisti, si vuole quindi proporre una breve elencazione dei punti di attenzione relativi a questa fase di progetto:

- Parlare col cliente, il prima possibile. Una breve chiacchierata con una figura lato cliente che abbia conoscenza delle sorgenti dati fornite e contemporaneamente del business, può in brevissimo tempo fornire una chiave di lettura dei tracciati, soprattutto nei casi, come quello trattato in questa tesi, in cui non vi è un documento che li definisce. Tentare di comprenderli autonomamente molto spesso risulta una perdita di tempo estremamente maggiore con alta probabilità di interpretazioni errate. È importante trasmettere al cliente il fatto che una breve perdita di tempo da parte di entrambe le parti viene ripagata da una data di consegna più vicina, costi minori e soprattutto una qualità maggiore del prodotto finito.
- Identificare, prima di definire il metodo di caricamento, quale sottoparte dei dati è necessario importare in tabelle. Se si ha a che fare con file excel di qualche Mb, la cosa migliore è fare un'importazione completa e selezionare e pulire i dati

sul database. Al contrario, caricare i dati nella loro interezza causa un consumo di tempo drasticamente più alto rispetto al metodo ottimale, ad uno stesso costo di effort per impostare il processo di data load.

- Identificare le chiavi complete dei dati che si analizzano. Con l'analisi preventiva delle anagrafiche dei percorsi pianificati, per portare un esempio concreto relativo al progetto, si è compreso, ragionando a livello di business e confermando le ipotesi con alcune query sui relativi dati, che la chiave che identifica uno specifico itinerario pianificato era composta da due campi distinti. Si è infatti verificato che le pianificazioni di uno stesso codice itinerario divergevano per diverse parti a seconda del giorno della settimana in cui venivano svolti. Senza questa informazione, la modellazione dei dati e degli algoritmi applicati sarebbe risultata totalmente errata.
- Comprendere e analizzare i dati nella loro interezza, da subito, anche se non servono nell'immediato. Il processo di data profiling permette di comprendere il business su cui si sta lavorando e, di conseguenza, gli elementi fondamentali per ottenere un prodotto efficace. Considerare fin da subito tutti i dati a disposizione (quando questi sono in quantità modeste) permette di ottenere un modello più coerente e con meno discrepanze dalla realtà. Questa esperienza ha quindi suggerito che è preferibile comprendere tutte le tipologie di dato a disposizione e escluderne una parte dal processo di caricamento solo in caso questo causi un aggravio di effort. È inoltre importante sottolineare che, d'altra parte, impostare un processo di data load su dati non necessari potrebbe portare a una perdita di tempo nel caso in cui, una volta che questi diventassero utili, emergesse-

ro necessità particolari di manipolazione del dato, ovviamente non prevedibili inizialmente.

3.4 Progettazione

3.4.1 Stack Architettuale

Come mostrato in figura 3.4, lo stack architettuale scelto per sviluppare la soluzione è così composto:

- Browser Web come strumento per l'utilizzo del front end
- OBIEE 12c come strumento di dashboarding
- Oracle MapViewer
- Oracle MapBuilder per la definizione dei metadati grafici
- Oracle Database 12c per la gestione e lo storage dei dati
- Macchina virtuale WMWare che ospita tutti gli ambienti (information layer, staging area e datamart di produzione)

La macchina virtuale su cui è ospitata la soluzione verrà utilizzata sia per il processo di caricamento dei dati, sia per quello di sviluppo della soluzione, sia, infine, come ambiente su cui funzionerà il prodotto mostrato agli utenti.

3.4.2 Data Loading

I dati delle rilevazioni GPS sono stati forniti sotto forma di file CSV. Ogni file rappresenta la rilevazione di un singolo ordine di servizio. Il dataset contenente i tracciati GPS ammonta a più di 50Gb di files perché consiste nell'intera banca dati aziendale relativa ai dati

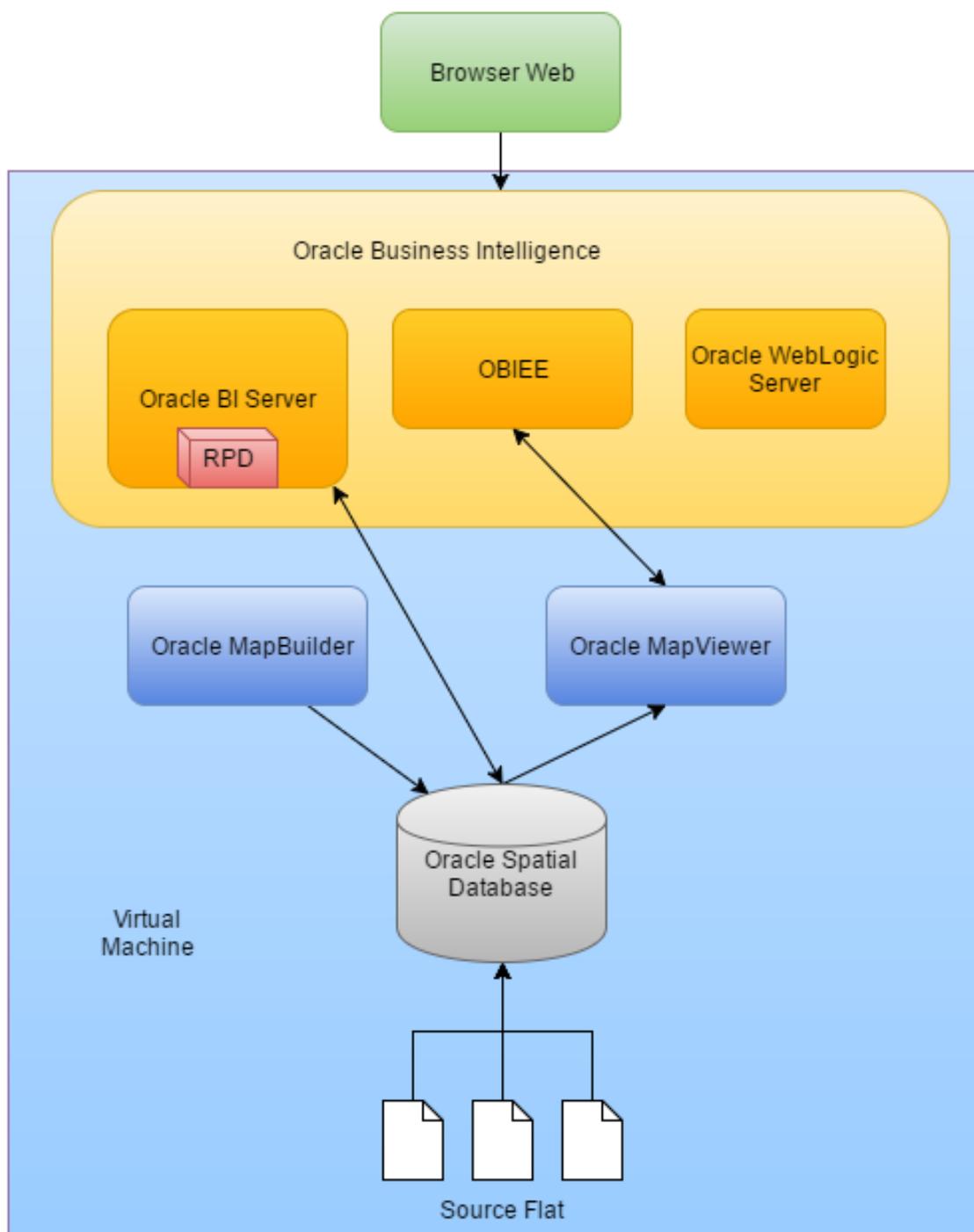


Figura 3.4: Schema dei componenti utilizzati

sensoristici dei mezzi. Per questo motivo è stato necessario eseguire una selezione “a monte” dei file da caricare, riducendo *drasticamente* i tempi di dataload. Utilizzando l’anagrafica degli ordini di servizio si sono selezionati solo i filename contenenti codici degli OdS eseguiti nel periodo temporale di interesse e relativi ai tracciati pianificati oggetto dell’analisi. Questa strategia ha permesso di ridurre di più di un ordine di grandezza la quantità di dati caricati in tabella.

3.4.3 Architettura del Front-End

La soluzione di front-end per questo progetto consiste in una fusione fra una web application custom e il front-end di OBIEE. In figura 3.5 viene mostrata un’architettura di massima che esemplifica le interazioni fra i componenti interessati. Il browser accede alla sezione *analytics* di OBIEE, come per una qualsiasi altra dashboard standard. Al suo interno viene però caricata una web application, hostata sullo stesso webserver WebLogic.

Di seguito si specificano le funzioni dei singoli componenti principali:

- Pagina JSP: contiene la struttura HTML della WebApp, generata parzialmente in automatico in base ai risultati dell’interrogazione a una servlet che fornisce i dati per strutturare correttamente le voci dei menù.
- Script Javascript: ogni JSP include uno script javascript, con rapporto 1:1, che ingloba tutta la logica applicativa. Le principali funzionalità gestite dallo script sono:
 - Gestione degli oggetti MapViewLayer, ovvero le variabili che rappresentano ogni singolo layer visualizzato sull’oggetto mappa, in particolare: istanziazione, configurazio-

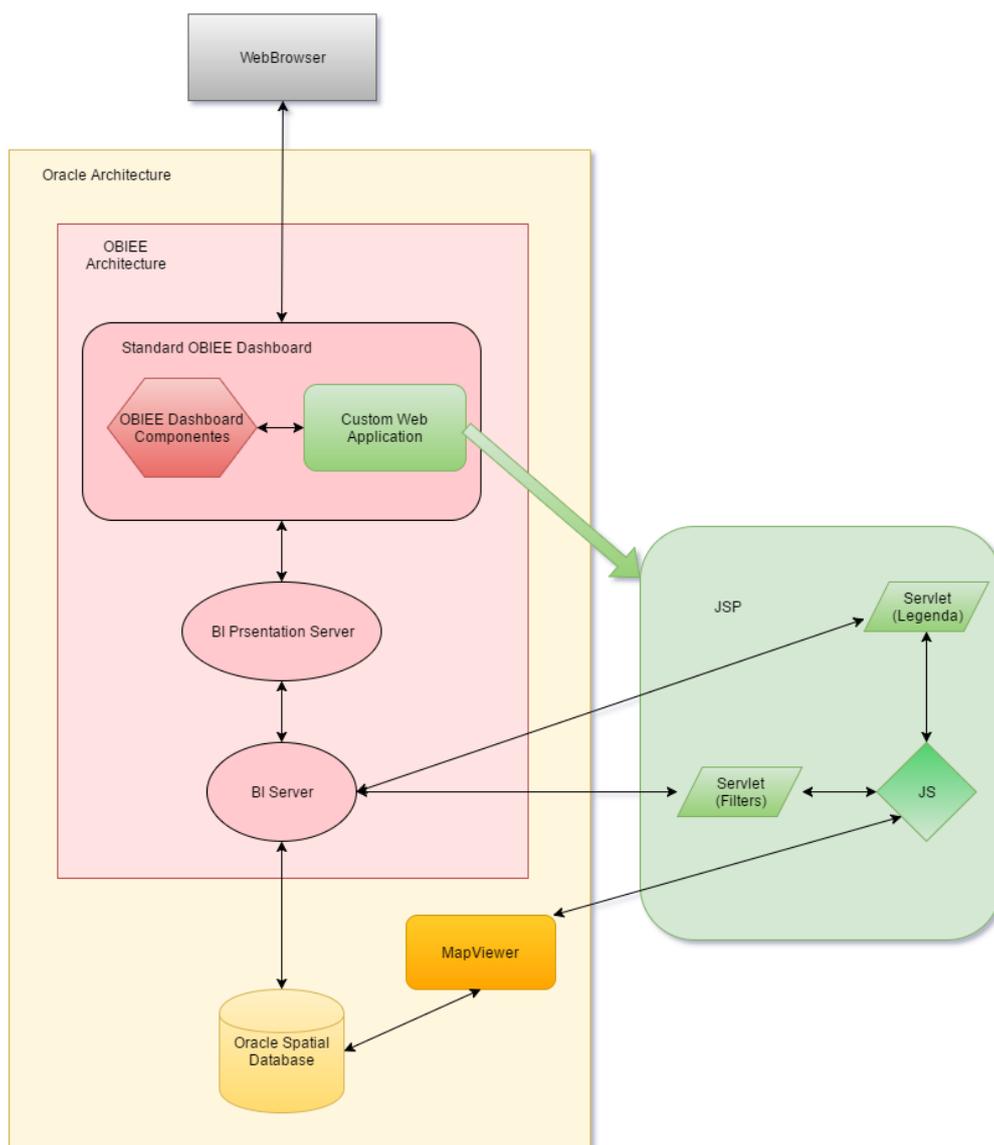


Figura 3.5: Dettaglio dell'architettura relativa al front-end

ne dei parametri, caricamento dei dati da MapViewer e l'aggiunta del tema all'oggetto mappa

- Gestione dei componenti javascript del front-end (ad esempio i selettori per i periodi temporali)

- Gestione dei filtri sui dati
 - Definizione dei KPI da visualizzare
 - Definizione del livello di aggregazione dei dati
 - Gestione delle legende dinamiche
 - Gestione dinamica dei filtri disponibili
 - Gestione della generazione dinamica delle tabelle necessarie per la creazione delle mappe di calore
 - Gestione degli eventi generati dal front-end (click sulle shape rappresentate, lancio dei grafici OBIEE relativi ai dati rappresentati, ecc.)
- Servlet dei filtri: una volta interrogata, si occupa di generare la lista di filtri disponibili per i dati rappresentati, sfruttando il modello OBIEE presente sul BI Server.
 - Servlet delle Legende: sfruttando il modello OBIEE, calcola dinamicamente gli intervalli che delimitano le classi dei KPI correntemente visualizzati su mappa. Questi valori vengono utilizzati sia per la generazione del componente di front-end, sia come parametri per l'extrapolazione dei dati dalle sorgenti.
 - MapViewer: funge da layer di intermezzo fra il front-end e la base dati, consistendo in pratica in una diversa via di accesso alle basi dati. Il componente infatti non utilizza il BI Server ma interroga direttamente le tabelle fisiche presenti sul OSDB (Oracle Spatial Data Base). Nelle tabelle di MapViewer sono salvate le query di tipo spatial necessarie all'extrapolazione dei dati, lo script javascript genera la chiamata istanziando la query SQL di tipo parametrico. Il risultato consiste in una serie di Features of Interest (*FOI* [26]) a cui vengono associa-

ti ulteriori attributi e che vengono aggiunti all'oggetto layer, istanziato dallo script javascript.

3.5 Sviluppo e test dell'applicazione

Nelle prossime sezioni verranno illustrate le due principali fasi del processo di sviluppo del sistema: la fase di ETL che include la manipolazione dei dati e l'applicazione degli algoritmi di location intelligence e la fase di sviluppo del front-end, dove viene definita la modalità di visualizzazione e fruizione dei dati per l'utente.

3.5.1 Overview del processo di ETL

Prima di cominciare lo sviluppo del processo di ETL, risulta estremamente importante prevedere una fase di progettazione dei flussi. Questa attività consiste nel definire, con un approccio top-down, tutte i singoli moduli dei flussi. Si parte costruendo un flow-chart composto dalle macro-fasi e che elenca le principali fonti dati utilizzate e le correlazioni fra le diverse fasi; questo permette di individuare ciò che è parallelizzabile e ciò che invece è legato da vincoli di dipendenza. Una volta elencati i macroflussi e le parti che li compongono, si approfondisce ogni singolo modulo scendendo più nel dettaglio e arrivando ad elencare tutte le operazioni atomiche, descrivendole a livello logico. A questo livello di dettaglio è anche utile definire quali sono le specifiche fonti dati sorgente e le tabelle di destinazione dei dati elaborati. Astruendo dall'ambito del processo di ETL, questa fase consiste, nella pratica, nella definizione di una WBS [44] specifica del singolo processo. Di conseguenza è possibile correlare ogni macro-fase a specifici requisiti funzionali e, a livello di dettaglio, scendere fino a individuare i singoli moduli atomici, così da facilitare, di molto, la fase implementativa. L'ottimizzazione del processo

si ottiene individuando il giusto trade-off fra astrazione e specificità, considerando che in caso di definizione troppo di dettaglio, si avrà un alto numero di modifiche dettate da motivazioni di carattere tecnico.

I benefici ottenuti da questo processo sono molteplici: affrontando la fase di progettazione in più persone, si otterrà una *soluzione condivisa* fra i membri del team e ottenuta grazie a un contributo di know-how e conoscenza indubbiamente maggiore di quello che si avrebbe affrontando direttamente la fase di sviluppo singolarmente e ne beneficerà anche l'omogeneità del prodotto; si faciliterà l'individuazione di vincoli di priorità che intercorrono fra le diverse fasi, permettendo una miglior organizzazione del lavoro e ottimizzando la parallelizzazione delle attività. Infine, nel caso dell'insorgere di problematiche e malfunzionamenti, sarà molto più semplice individuare i componenti interessati dal problema, isolando l'area interessata dalla modifica e risparmiando un ingente quantitativo di tempo

Di seguito, nell'immagine 3.6, si vuole fornire una schematizzazione semplificata del documento sopra descritto, per facilitare la comprensione del lettore, verranno quindi illustrate le fasi principali del processo di ETL cercando di fornire una visione d'insieme per poi, successivamente, entrare più nel dettaglio tecnico e operativo delle parti più interessanti.

- *Caricamento dei dati sull'Interface Layer*: in questa fase, indicata sullo schema dalla lettera A, vengono importati i flat file all'interno del database. Tutti i dati vengono gestiti come stringhe di testo per evitare problematiche dovute alle conversioni che vengono rimandate alle fasi successive.
- *Generazione dei dati geografici e delle dimensioni OdS e Route*: nella fase B, a partire dai dati caricati sull'interface layer, vengono generate le colonne contenenti i dati di tipo geometrico e le anagrafiche dei percorsi consuntivati (OdS, ordini di

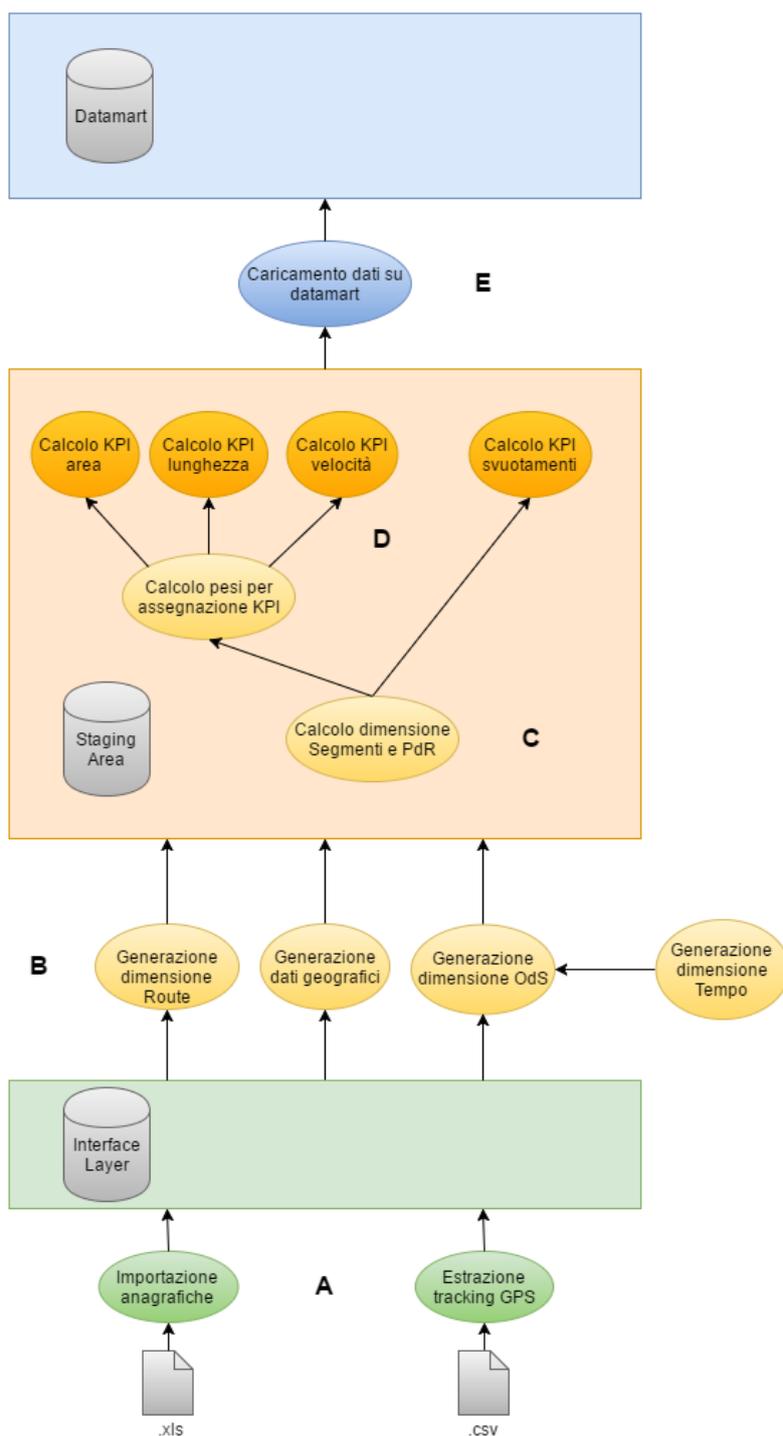


Figura 3.6: Schema delle principali fasi del processo di ETL

servizio) e dei percorsi pianificati (Route). La parte relativa alla dimensione tempo viene generata in automatico.

- *Calcolo delle dimensioni Segmento e PdR*: dalle informazioni relative ai percorsi pianificati vengono generate le dimensioni dei punti di raccolta (PdR) e dei segmenti che li congiungono.
- *Calcolo KPI*: nella fase E vengono applicati gli algoritmi necessari per il calcolo dei KPI distribuiti per ogni singolo segmento o, nel caso degli svuotamenti, per ogni singolo PdR.
- *Pubblicazione*: l'ultima parte del processo consiste nel caricamento su datamart dei dati calcolati.

Fase A - importazione dei dati

Per caricare i dati dei tracking su database Oracle è stato utilizzato *SQLoader*, uno strumento a riga di comando che, ricevendo in input parametri come il nome del file dei dati e la query di inserimento in una specifica tabella, esegue l'inserimento. La difficoltà di questa soluzione sta nel fatto che il file contenente i parametri deve specificare, di volta in volta, tutti i singoli filename dei dati. Per questo motivo si è utilizzato uno script di powershell che, data una lista di filename, modifica automaticamente il file di configurazione di *SQLoader* prima di eseguire ogni singolo comando di caricamento. In questo modo è stato possibile automatizzare il caricamento dei soli dati di interesse senza doverli identificare manualmente.

Per il caricamento su tabella di tutti gli altri dati, memorizzati su file Excel, si è utilizzato una funzionalità del tool *SQL Developer*. Per ogni sheet si è creata una tabella sull'interface layer, con lo stesso nome della sorgente sia per la tabella che per i singoli campi. Tutti i tipi di dato sono gestiti come stringhe per evitare problematiche di conversione o compatibilità. In questo caso, nonostante la numero-

sità dei tracciati pianificati, non è stato necessario sfruttare tecniche di selezione del dato o ottimizzazione.

Fase B - Trasformazioni geometriche

La fase di Spatial-ETL è quella che si occupa della creazione dei dati di tipo geometrico, prendendo come input una serie di stringhe corrispondenti alle coordinate geografiche ed eseguendo le opportune trasformazioni, sfruttando le librerie fornite dall'infrastruttura Oracle.

I primi due step eseguono i fondamentali processi di caricamento delle dimensioni principali, quello dei tracciati pianificati e degli ordini di servizio, necessari come anagrafiche di base per tutte le successive operazioni.

Nel primo caso, riguardante le cosiddette *route*, il processo inizia con una fase di data-check per migliorare la qualità del dato e prosegue poi con operazioni di data manipulation per ottenere tipi di dato che possano facilitare le operazioni successive. Per portare un esempio pratico, uno dei campi dell'anagrafica dei pianificati consisteva in una stringa contenente una serie di codici che indicano i giorni della settimana in cui viene eseguito lo specifico itinerario. Grazie alla fase di data profiling si è capito che per identificare un itinerario non basta il codice di quest'ultimo, bensì il giorno della settimana è parte integrante della chiave, perchè la sequenza di contenitori da svuotare cambia a seconda di quest'ultima variabile. Per questo motivo, si sono duplicate le righe di ogni itinerario e per ognuno si è associato uno dei giorni della settimana presenti nella stringa iniziale, associando ad ognuno un numero da 1 a 7 per facilitare alcune tipologie di calcolo.

Il caricamento degli ordini di servizio, oltre alle trasformazioni viste per i percorsi pianificati (*route*), comprende una fase di check di lookup table. L'operazione consiste nell'andare a confrontare tutti

i dati di una (o più) specifiche colonne con tutte le diverse occorrenze presenti in una tabella esterna, solitamente (ma non necessariamente) una anagrafica. L'obiettivo è identificare quali record della tabella in esame *non* sono presenti in anagrafica. Di norma, si sceglie di identificare e isolare questi record per poter eventualmente effettuare un'analisi delle cause dell'anomalia e si modifica il valore che ha generato l>alert con un fittizio, come spiegato nel prossimo paragrafo.

Dati Dummy Quando si definisce una dimensione anagrafica è fondamentale aggiungere un record in più, opportunamente definito, che verrà utilizzato da tutte le tabelle che avranno un riferimento all'anagrafica in oggetto, quando queste conterranno un riferimento errato o non esistente. Questa tecnica permette di garantire il vincolo di foreign key e assicurare la corretta manipolazione e aggregazione dei dati delle fact table; viene definita *aggiunta dei fittizi* o "fittizzazione".

Tabelle di errore Quando, all'interno di un processo di ETL, si eseguono check sui dati, è estremamente utile isolare le righe hanno presentato errori. I motivi di questa prassi sono generalmente due: analizzando i record in cui si sono rilevati problemi è possibile acquisire nuovo patrimonio informativo utilizzabile per ottimizzare l'intero processo, inoltre, tramite processi di pulizia del dato è talvolta possibile correggere i dati e re-immetterli nel dataset risultato, così da limitare la perdita di patrimonio informativo. Per raggiungere questi obiettivi si salvano tutti i record con anomalie in tabelle specifiche, una per ogni sorgente, aggiungendo opportuni metadati che tengano traccia del tipo di errore rilevato.

Caricamento dati geografici puntuali Il primo tipo di trasformazione *spatial* consiste nella conversione delle coordinate, fin qui

memorizzate come stringhe, in veri e propri oggetti geometrici, denominati *SDO_POINT*. Queste strutture, oltre a registrare il valore delle coordinate, inglobano anche tutte le informazioni necessarie per rappresentarle correttamente su una mappa geografica.

Il processo è suddiviso in tre parti distinte per ognuna delle due tipologie di dati, essendo per la maggior parte operazioni simili, verranno illustrate insieme, specificando però che le due parti di flussi hanno alcune sostanziali differenze, allo stato attuale di implementazione. A livello di best practices per questo tipo di attività, sarebbe opportuno standardizzare le fasi, con opportuni parametri, per ridurre al minimo le customizzazioni fra dati di pianificato e dati di consuntivato, dal momento che la tipologia del dato è la medesima.

La prima delle tre fasi consiste in una serie di trasformazioni aventi l'obiettivo primario di eliminare dati malformati:

- *Data Validation*: queste trasformazioni servono individuare dati mal strutturati, dovuti solitamente ad errori tecnici delle apparecchiature di rilevamento.
- *Lookup*: vengono eseguiti una serie di test di lookup table per garantire la selezione solo dei dati relativi alle anagrafiche presenti nel sistema. Per evitare la perdita di eccessivo contenuto informativo viene data la possibilità di analizzare anche i dati che risultano in errore durante questo controllo, reinserendoli dopo l'aggiunta di un opportuno valore fittizio nella foreign key che ha fallito il check di lookup table.
- *Data manipulation*: sempre in questa fase vengono effettuate alcune manipolazioni dei dati per ottimizzarne l'uso. Per portare un esempio concreto, per rendere confrontabili i dati riguardanti il giorno della settimana in cui viene svolto un pianificato, si convertono anche qui le stringhe in numeri da 1 a 7.

- *Controlli semantici*: per domini come le date di calendario, i check di data validation non sono sufficienti a garantire la correttezza dei dati. Questi ultimi permettono di eseguire controlli sull'input generalmente tramite ReGex, ma, per esempio, con questi strumenti non è possibile far sì che il 29 febbraio sia validato solo per gli anni effettivamente bisestili. Per questo motivo si sono implementati anche dei così detti *controlli semantici* che implementano logiche di verifica di tipo avanzato.

L'output di questa prima trasformazione consiste in una tabella di tipo *WT_* ancora priva di colonne di tipo *spatial* ma con dati sufficientemente *puliti* per poter generare shape partendo da essi.

Nella seconda fase si procede alla creazione vera e propria dei dati di tipo *SDO_GEOMETRY*: i dati sorgente hanno come unica informazione geografica le coordinate di una serie di punti del territorio; il primo passaggio consiste quindi nella trasformazione di questi dati in *shape* vere e proprie, mantenendo la tipologia *puntuale*. La conversione si esegue sfruttando la funzione *execute immediate* di Oracle che permette di eseguire la query all'interno di una stored procedure, automatizzando in questo modo il processo. L'unica variabile utilizzata è quella relativa al SRID (vedi: 2.4.2) che viene impostato basandosi sui parametri di configurazione di una opportuna tabella di appoggio. Il valore *2001* è un codice (denominato *SDO_TYPE*) che specifica che il valore di tipo geometrico è *un punto*. Alla fine di questo processo si avranno tabelle identiche al passaggio precedente con, in più, una colonna aggiuntiva di tipo geometrico.

Nella prima delle tre fasi si sono eseguite trasformazioni di filtraggio per escludere dati errati, in questa terza e ultima trasformazione viene eseguita una ulteriore fase di selezione dei dati basata, questa volta, su analisi di tipo geografico. Un esempio di selezione di questo tipo è controllo degli outlier delle coordinate. Si può infatti eseguire una verifica che identifichi, data una serie di punti successivi,

quando vi sono elementi che distano oltre un certo limite dal punto precedente o successivo. Questo problema può essere dovuto a imperfezioni del rilevatore GPS che può registrare dati molto distanti dalla locazione reale, dati corretti e quindi non scartati inizialmente ma rilevati da questo tipo di controllo.

Creazione shape multilinea Come già illustrato, i dati sorgente consistono in una serie ordinata di rilevazioni GPS che rappresentano un percorso, pianificato o consuntivato, di un mezzo sulla superficie terrestre. Per poter eseguire analisi e operazioni sulle shape rappresentanti questi percorsi è necessario unire i punti fino ad ora generati, seguendo il giusto ordine, così da ottenere rappresentazioni geometriche del percorso, sotto forma di linea orientata [45] [46] [47].

Come accennato nella sezione 2.4, Oracle fornisce un tipo di dato `SDO_GEOMETRY` compatibile con ogni tipologia di forma geometrica, differenziando queste ultime con il parametro `SDO_MTYPE`. L'operazione che si deve effettuare è quindi racchiudere tutti i punti che formano uno specifico tracciato pianificato o consuntivato in un singolo dato `SDO_GEOMETRY`, mantenendo l'ordine, in modo da costruire la relativa linea orientata.

Fase D - Calcolo KPI

L'obiettivo dell'algoritmo di calcolo dei KPI è fornire una serie di indicatori che permettano di misurare il grado di efficienza dell'esecuzione dei percorsi. Un tipo di analisi *ad alto livello* fornirebbe, per ogni tipologia di KPI definito, un singolo valore per ogni percorso consuntivato. In questo modo però si fornirebbe all'utente un'informazione eccessivamente aggregata che permetterebbe di individuare i tracciati pianificati con più problemi ma senza essere di alcun aiuto nel determinare la causa delle anomalie riscontrate. Per questo motivo, si è sviluppato un metodo di calcolo che permetta di ottene-

re valori a granularità inferiore al singolo ordine di servizio, ovvero i singoli tratti che vanno da un punto di raccolta al successivo. In questo modo sarà possibile rappresentare su mappa i percorsi pianificati andando a marcare le singole sottoparti con colori corrispondenti al valore del KPI calcolato per quel singolo tratto di strada, così non solo sarà possibile individuare quali percorsi pianificati presentano più anomalie ma anche in quale zona territoriale queste si verificano, dando così un'informazione molto più utile alla risoluzione del problema.

L'ostacolo principale a questo tipo di analisi sta nel fatto che non si può assumere che i tragitti tracciati dai mezzi siano coerenti con i percorsi pianificati, per questo motivo non è pensabile suddividere questi ultimi in segmenti *standard* ed effettuare confronti con le relative sottoparti dei tracking gps perchè non è garantita la loro esistenza.

La soluzione sta nel generare dinamicamente una serie di segmenti diversi a seconda dell'andamento del singolo tracking di ordine di servizio, con agli estremi sempre dei punti di raccolta pianificati ma, questa volta, non per forza immediatamente consecutivi. La generazione dinamica dei segmenti permette di avere ogni volta parti di pianificato e consuntivato *confrontabili*, cioè con lo stesso inizio e fine, dipendente da quali punti di raccolta sono stati raggiunti dallo specifico ordine di servizio. Si elencano di seguito i principali passaggi dell'algoritmo sviluppato con l'obiettivo di chiarirne la logica. In seguito verranno illustrate le parti implementative più significative.

1. Tutti i passaggi da 2 a 7 saranno effettuati iterativamente per ogni singolo tracciato pianificato (j) e ripetendo le operazioni per ogni singolo OdS (i) corrispondente al j selezionato.

```
1
2     foreach (j) do
3         foreach ( (i) of j ) do
4             2 to 7
5
```

2. Si *sovrappongono* tutti i *punti* rappresentanti dei PdR del pianificato j al tracciato effettivo i.
 - (a) Si ottengono tutti gli n PdR *toccati* dal tracciato consuntivato i
3. Si divide il tracciato effettivo i in $n-1$ segmenti $(A \dots \Omega)$ formanti l'insieme E
4. Si divide il tracciato pianificato j in $n-1$ segmenti $(A \dots \Omega)$ formanti l'insieme K
 - (a) Ogni segmento $(A, B) \in K$ comprenderà 1 o più PdR del tracciato pianificato j, risultando a tutti gli effetti un aggregato di PdR
 - (b) Per ogni segmento $\in K$ è necessario tenere tracciato di *quanti* e *quali segmenti base* di j $(\alpha, \beta \dots \omega \in j)$ contiene per poter successivamente ripesare i valori di KPI ottenuti sui singoli tratti *da PdR a PdR successivo* che rappresentano il livello di aggregazione minimo dell'analisi
5. A questo punto del processo abbiamo che
 - (a) $|E| = |K|$
 - (b) Ogni segmento $(A, B \dots \Omega) \in E$ e $(A, B \dots \Omega) \in K$ iniziano e finiscono nello stesso punto geografico, ognuno rappresentante uno specifico punto di raccolta
6. Per ogni coppia di segmenti $[A \in E, A \in K]$ si eseguono le seguenti operazioni che generano i KPI
 - (a) Si calcola l'area sottesa fra i segmenti A_E e A_K con il metodo descritto nella sezione 3.3.1
 - (b) Si calcola la differenza di lunghezza fra A_E e A_K

- (c) Si calcola la velocità media del segmento A_E
7. L'ultima operazione necessaria consiste nell'associare ogni segmento $(A, B \dots \Omega) \in K$ all'insieme dei *segmenti base* di j che li compongono. Una volta ottenuta questa serie di accoppiamenti, si assegna a ogni coppia $[A, \alpha], [A, \beta], [B, \gamma]$ un valore x che corrisponde, in termini di *lunghezza* a $\frac{\alpha}{A}, \frac{\beta}{A}, \frac{\gamma}{B} \dots$. Questo valore $0 < x \leq 1$ verrà utilizzato per soppesare i valori di KPI calcolati ai singoli *segmenti base* ($\alpha, \beta \dots \omega \in j$)
- (a) Essendo $(A, B \dots \Omega) \in K$ delle aggregazioni di *segmenti base*, sarà garantito che l'unione dei *segmenti base* contenuti in tutti i segmenti $(A, B \dots \Omega) \in K$ sarà *sempre uguale* all'unione dei *segmenti base* che compongono j .

L'output finale del processo di ETL consta di tre tabelle, una per ogni tipologia di KPI, tutte con la medesima cardinalità. Di seguito si trattano nello specifico alcuni passaggi significativi.

Suddivisione dei dati al minimo livello di granularità La prima fase consiste nella creazione dei segmenti base ($A, B, \Gamma \dots \Omega$): si tratta di una trasformazione di tipo geometrico da punto a linea, questa volta composta da un singolo segmento. Ogni riga della tabella risultante è quindi identificata univocamente dalla chiave composta del tracciato pianificato, unito all'ID del punto di raccolta e dall'ID del suo *precedente*. Il dataset così creato sarà quello di granularità minima su cui si baseranno tutte le rappresentazioni di dettaglio su mappa.

Successivamente vengono effettuati i passaggi che permettono di ottenere le aggregazioni (denominate $A, B \dots \Omega$) dei segmenti base, diversi a seconda dell'andamento del tracciato consuntivato. L'output di questa fase, quindi, consiste in un elenco di tracciati pianificati ripetuti, uno per ogni ordine di servizio (coerente con l'associazione

data dall'anagrafica), con la differenza che invece di essere composti da segmenti che vanno da un punto di raccolta al successivo, sono formati da segmenti aggregati. Il procedimento tecnico consta dei seguenti passaggi:

1. individuare, per ogni OdS, quali PdR risultano "toccati" dal tracciato consuntivato. Si noti che l'algoritmo deve selezionare *sempre* anche il primo e ultimo PdR della lista per garantire che vengano *sempre considerati tutti i PdR*, anche in caso di ampio discostamento del tracking gps nella parte iniziale e finale del percorso;
2. creare un set di segmenti composti/aggregati (A, B, Γ ... Ω) *concatenando* le parti base ($\alpha, \beta \dots \omega$). La suddivisione dei segmenti base varia per ogni singolo OdS, in base a quali PdR risultano raggiunti dal tracciato GPS;

L'obiettivo della fase successiva è suddividere le shape lineari, rappresentanti i tracking gps concatenati, in sottosegmenti ognuno in relazione 1:1 con quelli creati nella fase precedente. L'importante differenza fra i due processi sta nel fatto che in questo caso si lavora con una serie di punti geolocalizzati *omogenei*. Non si ha nessuna diversificazione fra una rilevazione che rappresenta un punto di raccolta e tutte le altre che compongono il percorso.

È quindi necessario identificare una rilevazione GPS consuntivata che faccia matching 1:1 con i punti geografici che rappresentano i PdR *agli estremi* dei segmenti A,B...Γ. Sfruttando la funzione OracleSpatial "SDO_NN" si va ad identificare, il punto GPS consuntivato (per ogni singolo OdS) più vicino ad ognuno dei PdR precedentemente selezionati. A causa di limiti prestazionali dell'infrastruttura che ospita la soluzione, è stato necessario impostare un opportuno partizionamento geografico per ridurre il numero di confronti da effettuare, oltre a un importante lavoro di tuning relativo agli indici di

tipo geometrico. Una volta individuate le singole rilevazioni GPS, si costruiscono, con la stessa metodologia vista per i tracciati pianificati, i sottosegmenti del tracciato consuntivato originale, con i punti trovati come estremi.

Calcolo dei KPI Una volta definiti le suddivisioni minime dei dati su cui poter applicare gli algoritmi, si procede al calcolo vero e proprio dei valori che verranno utilizzati come KPI, i primi sono relativi alle criticità dei percorsi: area, lunghezza e velocità mentre l'ultimo si riferisce alla regolarità degli svuotamenti dei singoli punti di raccolta. È fondamentale sottolineare che a questo punto del processo si otterranno dei valori che si riferiscono ai segmenti aggregati ma l'output richiede che siano quelli base ad avere valori ad essi associati. Si deve quindi suddividere ognuno dei KPI calcolati per ognuna delle sottoparti che compongono gli aggregati. Per farlo, si è deciso di utilizzare un proporzionamento basato sulla lunghezza.

Esemplificando, se:

- A è lungo 100 metri;
- A è composto da (α, β, γ) ;
- $\alpha = 20$ metri;
- $\beta = 30$ metri;
- $\gamma = 50$ metri;

Vengono calcolati 3 valori: $\alpha = 0.20$, $\beta = 0.30$, $\gamma = 0.50$ che vengono moltiplicati per il KPI associato a A per ottenere le parti relative a α , β e γ . La tabella con questi moltiplicatori viene utilizzata per ottenere le tabelle con la granularità massima (dettaglio del singolo segmento base associato a ogni singolo ordine di servizio) che verrà utilizzata nelle fact table.

Il primo dei KPI definiti in fase di progettazione consiste nell'area di discostamento fra i percorsi pianificati e quelli consuntivati, vista la sua particolarità si vuole illustrare un passaggio fondamentale necessario per poter applicare gli algoritmi geometrici: le funzioni di calcolo dell'area di Oracle (`SDO_AREA()` [48]) permettono di ottenere l'area unicamente di figure *chiuse* ma l'obiettivo di questa fase è definire il discostamento di due linee che non formano questo tipo di forma grafica. Per ovviare a questa problematica si è definito un processo che prima costruisce un nuovo tipo di geometria compatibile con i requisiti della `SDO_AREA` applicando, in cascata, un'operazione di buffering che minimizza l'errore del calcolo. Il calcolo della differenza di lunghezza viene eseguito applicando funzioni *spatial* basate sui segmenti calcolati in precedenza mentre per ottenere i valori di velocità si combinano opportunamente lunghezza del tratto di percorso e i tempi di inizio e fine.

Per i dati relativi agli svuotamenti dei contenitori la logica risulta totalmente diversa da quella utilizzata nei calcoli precedenti. In questo caso, infatti, non si vogliono analizzare dei tratti di lineari bensì di tipo puntuale, riferiti ai singoli punti di raccolta. Parliamo di PdR (punti di raccolta) e non di contenitori perchè si è deciso, in fase di modellazione, di eseguire l'aggregazione in fase di ETL esponendo sul DataMart un dato non ulteriormente separabile. Il motivo di questa assunzione sta nel fatto che il tipo di analisi che si vuole ottenere è di tipo geografico, si vogliono cioè individuare i luoghi dove avvengono anomalie, un'informazione di estremo dettaglio non fornirebbe nessun valore aggiunto, richiedendo però un più alto livello di complessità. Il calcolo del KPI risulta in questo caso più semplice: la riga di fact table rappresenta infatti il numero di contenitori presenti nel PdR nella pianificazione e il numero di contenitori effettivamente svuotati. Nello specifico di questo PoC, si rappresenta un valore tanto più alto quante sono gli OdS che hanno *saltato* lo svuotamento

e viene mostrato all'utente in forma percentuale: 0% significa che, durante gli OdS selezionati, sono sempre stati svuotati tutti i bidoni di quello specifico punto di raccolta, 100% indica che non sono mai stati svuotati, nonostante fosse pianificato di farlo.

Fase E - Pubblicazione

La fase di caricamento dei dati sul DataMart utilizza un approccio misto, estremamente legato alla tipologia di progetto che è caratterizzata da un unico caricamento dei dati iniziale senza sincronizzazioni successive.

- *Dimension Table statiche:*
 - Tutte le dimension table/lookup table delle anagrafiche principali (ordini di servizio, percorsi pianificati, tempo) rimarranno stabili per tutta la vita del PoC, quindi si sceglie l'opzione di utilizzarle nel DataMart con dei sinonimi, ovvero degli alias/collegamenti che permettono di utilizzare i dati presenti sullo schema di staging area da quello di DM, rendendo questo layer "trasparente"; nel caso in cui si decidesse in seguito di creare una vera e propria tabella con i dati anche nel terzo ambiente non sarà necessaria nessuna modifica alle query.
 - Per queste dimensioni non si creano chiavi surrogate dal momento che le PK originali sono di tipo numerico, e composte da uno o due campi. Vengono inoltre eseguiti per ognuna check di PK di conseguenza si assume che non sia necessario l'utilizzo di surrogate.
- *Dimension Table dei segmenti:*

- A causa della natura estremamente custom dei dati di questa dimensione si è deciso di copiare i dati da SA a DM in modo tale da permettere un eventuale cambiamento nella loro definizione ed elaborazione senza impattare direttamente sui dati di produzione.
- A causa della dimensione della PK (6 colonne) e della possibilità di modificare la struttura di questa DT, si è scelto di applicare una chiave surrogata in modo da poter facilitare un qualsiasi tipo di modifica sui dati e sulla loro composizione.
- *Dimension Table dei PdR*: anche in questo caso si è optato per una PK surrogata, nonostante le criticità risultino minori, e di conseguenza si esegue un disaccoppiamento fra i dati di datamart e quelli di staging area.
- *Fact Table*: per le fact table si sono fatte diverse scelte progettuali influenzate dal funzionamento della parte front-end
 - Nella staging area vengono generate 4 tabelle, una per ogni KPI calcolato.
 - Nel processo di caricamento vengono sostituite le chiavi multiple con i relativi surrogati eseguendo dei check di lookup table sulle DT del datamart, in caso di fallimento si utilizza un valore fittizio, opportunamente inserito nelle tabelle dimensionali.
 - 3 dei 4 indicatori, quelli riferiti ai tratti di percorso, vengono raggruppati in una nuova FT, denominata F_CRITICAL. Questa scelta ha l'obiettivo di semplificare la fase di configurazione del front-end: come verrà illustrato nella sezione

3.5.2, è necessario definire a priori la query di interrogazione i dati geografici presenti sulle fact table. Utilizzando `F_CRITICAL` si snellisce questa fase di configurazione sia in fase di creazione che di modifica successiva.

- Il rimanente indicatore, avendo dimensioni diverse, viene rappresentato in una FT specifica.

3.5.2 Sviluppo del front-end

Nella sezione di progettazione (3.4.3) è stata descritta l'architettura del modulo di front-end. In questa sezione, più orientata allo sviluppo vero e proprio, si vogliono fornire alcune informazioni più specifiche riguardo alcune importanti caratteristiche della web-application. Inizialmente viene fatto un distinguo importante sulle due diverse modalità con cui l'applicazione interroga le fonti dati, una più classica, che sfrutta OBIEE, per la gestione dei dati, e una specifica per la componente geografica che permette di generare le mappe con cui l'utente interagisce. Infine, verrà descritta l'interfaccia grafica illustrandone le sezioni principali e la modalità di utilizzo.

Interrogazione delle fonti dati

L'applicazione interroga le fonti dati in due modi distinti: tramite il modello OBIEE e il BI Server oppure direttamente sulle fonti dati. Si utilizza il BI server per ottenere informazioni sfruttando l'architettura e l'ottimizzazione di OBIEE che utilizza pesantemente il caching. Le query eseguite con questo metodo risultano efficienti e allo stesso tempo non impattano sulle prestazioni, permettendo una buona scalabilità del prodotto. Uno degli obiettivi primari del modulo di front-end consiste nell'integrare una webapp custom con le dashboard standard di OBIEE che sfruttano il relativo modello. Grazie a questa integrazione è possibile tenere sincronizzato ciò che

viene mostrato sulle mappe e le informazioni presenti nei relativi grafici.

Il caricamento degli elementi di tipo Spatial viene eseguito interrogando direttamente lo spatial database attraverso un componente denominato MapViewer (descritto nella sezione (vedi: 3.4.1)). Il componente viene interrogato e, utilizzando i parametri ricevuti dalla chiamata si occupa di eseguire interrogazioni SQL di tipo spatial direttamente sul database, sfruttando query SQL parametriche precedentemente salvate sulle tabelle dei metadati. Le query restituiscono un dataset di oggetti spatial che viene gestito da MapViewer, con opportuni metodi di caching, per poter essere poi renderizzato sulla mappa inclusa nella web application. Ogni oggetto, denominato FOI, può essere associato a specifici attributi, estrapolati dalle basi di dati, che saranno utilizzati dalla stessa web app o dalla dashboard di OBIEE per generare grafici e report standard. Tutte le specificità di tipo grafico e di renderizzazione sono preconfigurate in MapViewer e salvate su tabelle di metadati. L'oggetto logico che racchiude tutte le informazioni e i metadati relativi a un singolo layer viene denominato *theme* e rappresenta un layer informativo, come descritto nella sezione 1.4.2.

Configurazione di un layer su MapBuilder

Di seguito si descrivono i passaggi principali della configurazione di un layer su MapBuilder, tool utilizzato per la generazione dei metadati che andranno a definire la veste grafica dei temi di analisi visualizzati su mappa, con l'obiettivo di chiarirne meglio la composizione. Ad ogni tema che si vuole visualizzare è associata una query SQL che sarà utilizzata da MapViewer per interrogare la base dati. Quest'ultima sarà di tipo parametrico, permettendo la selezione di un subset di dati agendo sull'interfaccia grafica, ma mantenendo-

ne inalterata la tipologia. In figura 3.7 viene mostrata l'interfaccia grafica del tool che ne permette la definizione.

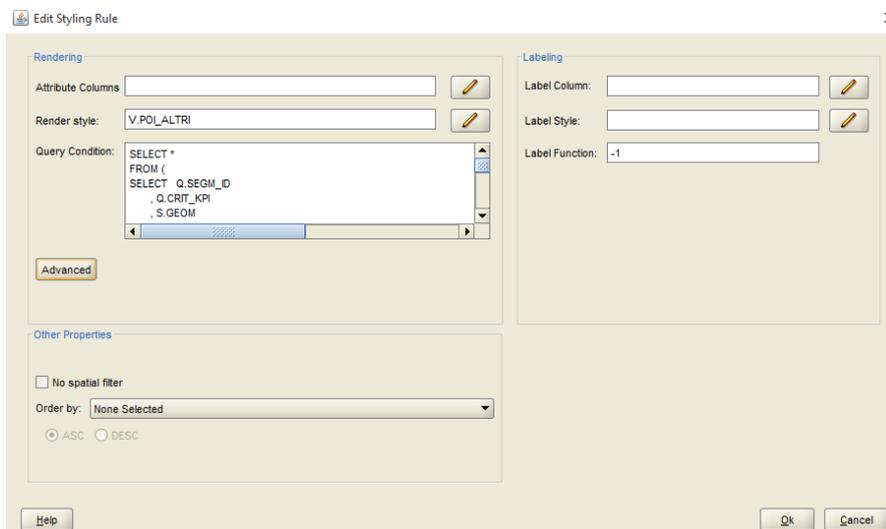


Figura 3.7: Esempio di interfaccia di MapBuilder dove viene definita la query per recuperare i dati da visualizzare

Un *theme*, ovvero ogni singola rappresentazione di dati geografici, utilizza degli *style*, cioè gli elementi grafici effettivamente rappresentati sulle mappe (linee, punti, marker, ecc.). A ogni *style* è associato un oggetto grafico e una colorazione specifica. Gli *style* possono però essere anche di tipo *advanced* e consistere nello stesso tipo di marker che però, in questo caso, viene colorato in modo differente basandosi sul valore del dato che rappresenta. Questa impostazione è definita nella sezione “Edit Rendering Rules” (figura 3.8) dove si definisce lo *style* (che determina il tipo di marker) e la colonna (*diversa da quella di tipo Geometry*) il cui valore determina la colorazione (*classe*) del marker.

L'obiettivo principale della query associata a un Theme di MapViewer consiste nel recuperare una serie di oggetti *geometry* per rappresentarli graficamente su una mappa. È però spesso necessario fornire una serie di informazioni relative all'oggetto che si rappresen-

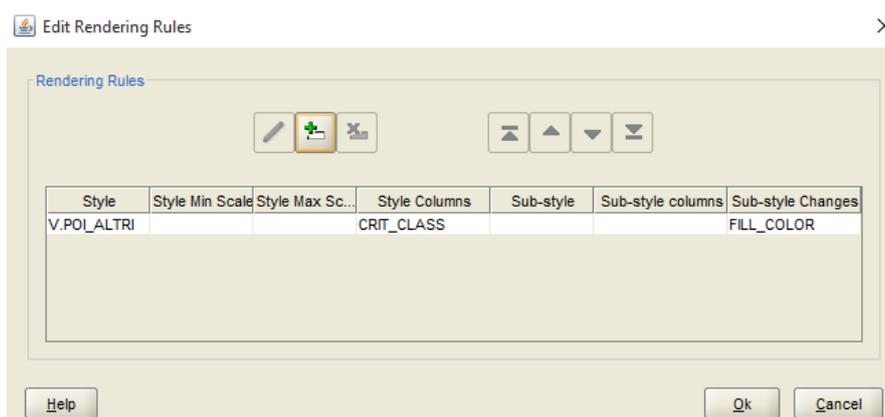


Figura 3.8: MapBuilder rendering rules

ta, per questo motivo nella select list dell'SQL statement vengono inserite una serie di colonne aggiuntive. Per permettere l'associazione fra i dati recuperati e l'oggetto grafico, è necessario elencare tutti i campi della select list che si vogliono rendere disponibili, associando ad ognuno un *name* identificativo (si veda un esempio in figura 3.9). Tramite questo metodo, Mapviewer restituirà al componente che ha originato l'interrogazione un dataset formato da diversi campi che saranno utilizzati dal front-end per mostrare all'utente le relative informazioni.

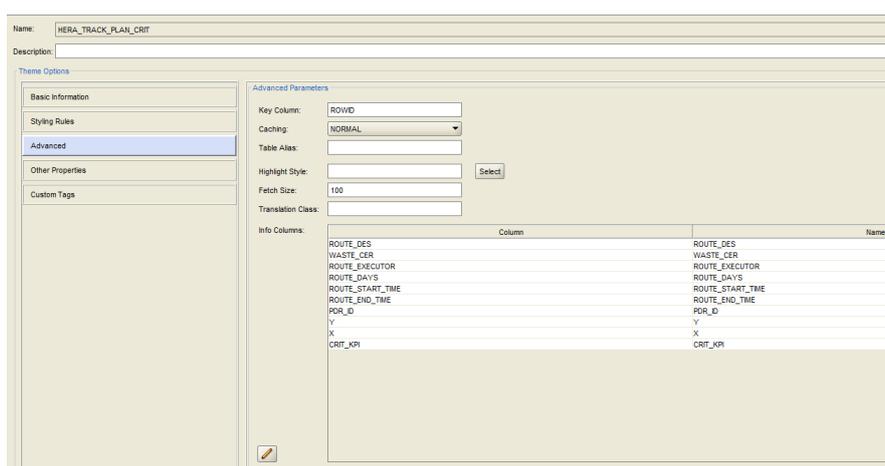


Figura 3.9: MapBuilder additional attributes

3.5.3 Fase di test

Lo scopo dei test non è trovare i difetti ma *prevenirli*. L'obiettivo della fase di test è organizzare il sistema in modo che non ammetta i difetti, questi ultimi sono un problema di gestione causato da processi difettosi. Nonostante il prodotto che si sta sviluppando sia un PoC, una attenta e strutturata fase di test risulta imprescindibile, perchè si possa parlare di una soluzione corretta e ben progettata. Strutturando attentamente i test case, sarà inoltre possibile riutilizzare l'output di questa fase per l'implementazione del progetto pilota, trattandosi principalmente di una serie di operazioni a livello funzionale e sui dati, in entrambi i casi *technology independent* e sostanzialmente di tipo gestionale. Si definisce quindi una macro-suddivisione dei test specificandone i passaggi maggiormente significativi:

- *Rispetto del modello*: I primi test effettuati sono quelli riguardanti il rispetto del modello e dei relativi vincoli di modellazione:
 - Test di Primary Key su tutte le fact table e sulle tabelle delle dimensioni
 - Test di Foreign Key fra le righe di fact table e le dimensioni e fra le diverse tabelle di uno snowflake schema
 - Controllo della presenza di doppie risalite sulle gerarchie delle dimensioni
 - Verifica dell'assenza di double counting
- *Correttezza dei dati*: La fase di controllo della correttezza dei dati risulta una delle più complicate perchè non si vuole individuare un'anomalia oggettiva del dato, bensì errori relativi al livello estensionale. Per questo motivo l'automatizzazione dei test risulta problematica e di difficile modellazione. Il metodo

più semplice per effettuare questo tipo di verifiche consiste nel generare interrogazioni che riguardano un subset di dati di cui si conosce il valore atteso e verificare che il risultato sia quello previsto.

- *Prestazioni*: Questi test hanno l'obiettivo di verificare che le prestazioni del sistema siano sufficienti a garantire una buona usabilità del prodotto. È importante sia valutare se esistono, e quali sono, le operazioni che richiedono il maggior impegno computazionale e di conseguenza maggiori tempi di attesa, sia verificare se, eseguendo contemporaneamente troppe richieste di tipo diverso, la web application può presentare problemi di gestione di troppi dati contemporanei.

- *Tempi di esecuzione*

- * Tempo di caricamento e visualizzazione della dashboard
- * Tempo di caricamento di ogni singolo tema
- * Tempo di caricamento di temi multipli selezionati
- * Tempo di applicazione di un filtro e di una serie di filtri combinati

- *Test di carico* In questo caso si definiscono una serie di prove con l'obiettivo di mettere in difficoltà le prestazioni del sistema per testarne la reazione, valutarne gli impatti e individuare, preventivamente, metodi per evitare problematiche future.

I test di carico riguardano sia le prestazioni del database, aumentando la quantità di dati e la quantità di query effettuate in un ridotto intervallo di tempo, sia dell'interfaccia grafica che deve renderizzare i dati ottenuti tramite Map-

Viewer, gestendone in numero maggiore a quello previsto e con un'alta frequenza di aggiornamento e rielaborazione.

- * Selezione di tutti i temi contemporaneamente
 - * Attivazione e disattivazione ripetuta e continua di uno o più temi
 - * Applicazione e eliminazione di uno o più filtri in modo continuo e ripetuto
 - * Significativo aumento dei dati presenti su datamart con l'obiettivo di simulare sia una maggior dimensionalità delle fact table sia delle tabelle di anagrafica degli ordini di servizio
- *Accesso ai dati:* In questa fase si vuole verificare la correttezza delle interrogazioni generate dal front-end, assicurandosi che vengano caricati tutti i dati presenti sul datamart e che rispettino correttamente i filtri applicati. Per effettuare questo tipo di test si modificano direttamente i dati del datamart tramite opportune query SQL atte a valorizzare una specifica porzione dei dati con valori fittizi, per poi effettuare sul front-end interrogazioni che selezionano solo la parte interessata dalla modifica e verificare che l'output sia quello previsto. Uno dei punti di attenzione evidenziati da questi test consiste in una peculiarità della web application determinata sia dal componente mapviewer sia dai metodi di gestione dell'interfaccia utente: Infatti, per ottimizzare le prestazioni, viene effettuato un uso costante della cache server-side di MapViewer. Questo fa sì che, in seguito a una modifica dei metadati di definizione dei temi, sia necessario lo svuotamento della cache per utilizzare i dati aggiornati.

Sull'interfaccia grafica, invece, la renderizzazione dei dati viene in parte gestita utilizzando la cache del browser per questo

motivo si possono talvolta verificare alcune piccole anomalie sui dati rappresentati, risolvibili con un semplice svuotamento della cache del browser utilizzato. Questo tipo di problematica è stato riscontrato solo dopo un prolungato utilizzo della web application contemporaneo a numerose modifiche del codice dell'applicativo front-end e dei dati e metadati del datamart, per questo motivo si è ritenuto accettabile nello specifico ambito del Proof of Concept che si è sviluppato. Nell'ottica di un rilascio in produzione agli utenti, sarà necessario definire una serie di contromisure automatizzate per evitare problematiche di questa natura.

- *Interfaccia utente*: Si sono eseguiti una serie di test atti a verificare il corretto funzionamento dell'interfaccia utente della web application, si elencano di seguito i più importanti
 - Funzionamento delle checkbox e dei radiobutton di selezione dei temi, dei layer mappa e dei filtri
 - Funzionamento dei pulsanti di navigazione fra i menù
 - Funzionamento dello zoom e dello spostamento sulla mappa, con conseguente visualizzazione dei dati corretti
 - Test su diverse macchine client, con risoluzioni e dimensioni dello schermo diverse

- *Compatibilità software della web application*: Il front-end dell'applicazione consiste in una web application, sviluppata in java e javascript. Questo tipo di applicazioni è molto influenzata dalla compatibilità dei browser su cui viene eseguita. A questo proposito, si sono effettuati tutti i test elencati nelle precedenti sezioni sui tre principali browser presenti sul mercato: Google Chrome, Mozilla Firefox e Microsoft Explorer, in modo

da assicurare la compatibilità con tutte le possibili piattaforme su cui potrebbe essere eseguito l'applicativo. Nonostante sia stata verificata la piena compatibilità, si è individuato un leggero miglioramento di prestazioni utilizzando Firefox che è stato quindi scelto per la fase di rollout e demo agli utenti.

SQL Test-Story Per implementare la verifica dei test che richiedono la definizione di una serie di query SQL direttamente sui dati del datamart, o nel caso si individuino anomalie delle quali si vuole comprendere la causa, si è utilizzato un utile approccio metodologico. Si è trovato estremamente efficace, infatti, la produzione di script SQL così strutturati:

1. commento che specifica il tipo di controllo che si vuole effettuare;
2. query SQL;
3. commento descrizione del risultato atteso;
4. eventuale commento con descrizione del risultato se diverso da quello atteso;

In questo modo si definiscono una serie di test, riproducibili in qualsiasi momento, con la specifica del contesto e del significato del risultato atteso. Questo tipo di script risulta utile sia per la stessa fase di test, per dare un ordine coerente alle operazioni e tenerne traccia, sia in caso di debug o ricerca di problematiche. È infatti utile avere a disposizione una traccia (una sequenza di test in successione) che permetta facilmente di individuare risultati anomali, isolati allo specifico, singolo, controllo.

Outlier e valori fuori range Vi è poi un diverso tipo di test che ha lo scopo di individuare valori errati perchè fuori dai range attesi. Questi controlli sono facilmente automatizzabili e dipendenti solo dal dominio del dato, di conseguenza molto più standardizzabili e stabili nel tempo. Alcuni esempi sono il controllo dei valori di velocità che devono essere strettamente maggiori di 0 e entro un certo valore massimo (una velocità media di 150 Km/h può verificarsi in un caso generico ma nello specifico del tipo di rilevazione delle velocità di mezzi di raccolta rifiuti, un valore simile indicherebbe un errore sui dati o sui calcoli). Simile è il discorso relativo agli outlier: effettuando query di analisi sui dati è possibile individuare misure calcolate che risultano molto diverse dai valori medi ottenuti. Un esempio di questa casistica è il calcolo della lunghezza dei percorsi pianificati e consuntivati. Effettuando questo tipo di test si sono individuati percorsi di lunghezza di due ordini di grandezza maggiore rispetto alla media, evidenziando quello che si è rivelato essere un errore specifico del singolo tracking gps che risultava errato sulla sorgente dati. Questo tipo di controlli sono fra i migliori candidati per essere automatizzati e gestiti in maniera programmatica.

Correttezza degli algoritmi e dei calcoli La verifica della correttezza degli algoritmi e dei calcoli effettuati sull'applicativo è una specifica casistica di errori sui dati. Come descritto nella sezione precedente, infatti, un valore errato di un dato può essere dovuto a un errore sulle sorgenti o da un'anomalia sui calcoli effettuati. Si intende ora concentrarsi sul testing di questa seconda evenienza. Per l'individuazione dell'errore, nel caso in cui questo rientri nei casi di out-of-range o outlier, l'unica difficoltà sta nell'indagare se il problema si trovi nelle sorgenti o nei calcoli. Quando invece il valore errato risulta "realistico", l'unica possibilità per testare la correttezza consiste la definizione di un set di dati creato ad-hoc, applicare gli

algoritmi e controllare che l'output sia quello atteso. Risulta quindi utile definire una serie di test-set con i quali verificare ogni algoritmo su input diversi, in modo sistematico. Nell'ottica futura di un'alimentazione programmatica con l'aggiunta progressiva di nuovi dati, questo tipo di test viene effettuato ripetutamente e prende il nome di *test di quadratura*, con l'obiettivo di verificare, quanto possibile, che ci sia corrispondenza fra i dati in input e quelli mostrati sulle dashboard di BI.

3.6 Deliverable progettuali e front-end

La webapp sviluppata in questo progetto ha la finalità di rendere facilmente fruibile all'utente un potente strumento di analisi geografica, cercando di fornire, velocemente, informazioni di tipo analitico e storico sulla totalità o un sottoinsieme dei dati presenti nel sistema. Una delle caratteristiche più importanti di questa interfaccia è la possibilità di eseguire, nella stessa dashboard, sia analisi di tipo aggregato e di insieme, sia analisi puntuali di dettaglio, così da permettere un flusso analitico completo a partire dalla situazione a livello macro fino agli eventi di interesse specifici.

Le analisi per aree geografiche estese sono ottimizzate grazie all'utilizzo di heatmap, mappe di calore che rappresentano il territorio con colorazioni differenti a seconda della tipologia di evento registrato in quelle zone. Grazie a questo tipo di visualizzazione, l'utente può individuare a colpo d'occhio quali sono le aree di interesse sulla totalità del territorio coperto dall'analisi, per potersi poi concentrare sui dati più significativi. Nella sezione dell'interfaccia in figura 3.10 si possono osservare tre diversi temi di analisi in formato heatmap, una per ogni KPI e un quarto utilizzato per visualizzare le caratteristiche peculiari del territorio come popolazione e PoI (Points of Interest).



Figura 3.10: Elenco dei tematismi disponibili sull'applicativo

Nel caso di analisi di dettaglio sono forniti tre ulteriori *theme* che visualizzano lo stesso tipo di informazione ma mostrando il dato al livello più “fine” (segmento di percorso o singolo punto di raccolta). In questo caso è possibile selezionare il singolo oggetto geografico ed ottenerne informazioni specifiche, sulla mappa stessa, così da facilitare l'indagine di anomalie una volta che si è individuata un'area critica o di interesse (se ne può vedere un esempio in figura 3.11).

Grazie allo strumento di selezione manuale, l'icona della lente di ingrandimento visibile in alto a sinistra nelle immagini dell'interfaccia, l'utente può modificare l'area visualizzata per concentrarsi sulle zone di maggior interesse. Una delle caratteristiche chiave di questa interfaccia di analisi è la possibilità di utilizzare contemporaneamente tutti i tematismi che si ritengono utili all'analisi, permettendo di correlare le informazioni e i fenomeni rappresentati su ognuno di essi.

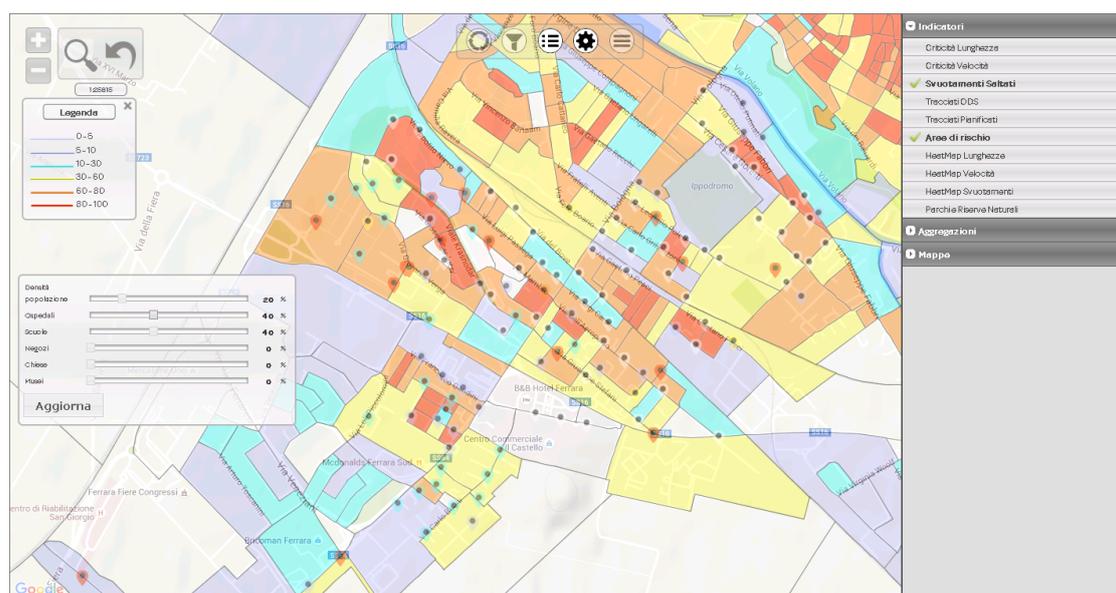


Figura 3.12: Mashup di tematismi per l'individuazione di fenomeni complessi

trate sui dati, riferite ai fenomeni che si stanno visualizzando sulla mappa. L'unione di questi due tipi di dashboard, uniti in una, permette di fornire uno strumento analitico completo, con un'interfaccia esteticamente attraente e allo stesso tempo semplice e immediata da utilizzare.

3.7 Sviluppi Futuri

Si sono definite una serie di possibili evoluzioni del prodotto, suddividendole in tre tipologie di analisi definite in letteratura:

1. *Analisi descrittive*: l'obiettivo di questo tipo di analisi è rispondere alla domanda "Cosa è successo?", fornendo quindi informazioni dello stato attuale (o passato) di ciò che viene analizzato, utilizzando strumenti di location intelligence.

2. *Analisi predittive*: una volta analizzata la situazione attuale può essere di fondamentale utilità riuscire a determinare cosa avverrà in futuro. A questo scopo vengono definite una serie di analisi di tipo predittivo che hanno l'obiettivo di fornire indicazioni su ciò che deve ancora verificarsi, analizzando le cause dell'insorgere di problematiche. Si può in questo caso parlare di *GeoSpatial Analytics*, caratterizzata in particolar modo dal quantitativo massivamente maggiore di dati da utilizzare [50, 51].
3. *Analisi prescrittive*: la naturale evoluzione delle analisi di tipo predittivo sono denominate *prescrittive*, ovvero si vuole sviluppare uno strumento che possa fornire indicazioni (*prescrivere*, appunto) su come ottenere risultati migliori. Una delle precondizioni necessarie per introdurre uno strumento di questo tipo all'interno dei processi aziendali consolidati è la *fiducia*: gli utenti devono avere avuto prova della correttezza dei "calcoli" effettuati da uno strumento. I sistemi di tipo predittivo possono assolvere a questa necessità nel caso in cui si riesca a verificare la veridicità delle predizioni effettuate. Una volta definiti strumenti predittivi di verificata efficacia, si può evolvere lo strumento aggiungendo la prescrittività che ha, fra le sue caratteristiche peculiari, la mancanza di una componente umana. Il modello è costruito dalle capacità e dalla conoscenza umana ma tutte le raccomandazioni sono generate dallo strumento basandosi sugli algoritmi e sui dati analizzati [52].

Conclusioni

Abbiamo visto che la location intelligence si pone l'obiettivo di fondere le potenzialità dei sistemi DataWarehouse e GIS, abilitando l'analisi di insieme di grosse moli di dati correlati ai luoghi da cui questi sono generati e permettendo, quindi, di catturare il loro contesto, le loro specificità e, soprattutto, le caratteristiche dell'ambiente da cui provengono. La novità, il valore aggiunto, sta nella possibilità di integrare informazioni che sono sempre state a disposizione delle aziende con materiale informativo esterno, legato ai territori e alle loro caratteristiche. Quello che prima era comprensibile solo per chi svolgeva le proprie analisi immerso nell'ambiente oggetto di studio ora può venire concretizzato in informazioni analizzabili *insieme* a quegli stessi dati che prima era complicato distinguere gli uni dagli altri, perchè spesso mancava *il contesto*.

Unire il contesto, l'environment, ai dati, permette una comprensione più profonda di questi ultimi, abilitando allo stesso tempo a gestire in modo centralizzato questa nuova mole di informazioni generata da queste nuove correlazioni. Strumenti di analisi geografica come i GIS esistono da decenni ma hanno sempre mancato di una capacità fondamentale, elemento caratterizzante della location intelligence: l'analisi aggregata e storica dei dati che, correlando le informazioni derivanti dalla geolocalizzazione, permette di far emergere evidenze impossibili da individuare sul singolo dato, o su uno specifico istante di tempo o singolo luogo.

É grazie a strumenti di location intelligence che ogni azienda, oggi, ha la possibilità di tenere sotto controllo sia le proprie performance sia il territorio su cui lavora, con la possibilità di individuare facilmente le criticità, localizzarle, contestualizzarle e comprenderne le cause, così da poter più facilmente intraprendere azioni correttive; tutto questo grazie a strumenti analitici integrati, sempre aggiornati e capaci di catturare tutte le sfaccettature di un evento, di un'evidenza, portata dai dati e spiegata e resa comprensibile da informazioni ad essi correlate.

Nell'ambito di questa tesi è stato progettato e realizzato un sistema completo che, prendendo in input dati operazionali, li ha arricchiti della componente geografica per poi analizzarli ed elaborarli con algoritmi sviluppati appositamente. Questo processo ha permesso inoltre di definire KPI specifici del dominio di business, utili a soddisfare le necessità di analisi del committente e a misurare in modo oggettivo le performance della rete di servizi aziendali. Per l'utente finale è stata sviluppata un'interfaccia di semplice utilizzo che abilita ad analisi aggregate ad alto livello e, contemporaneamente, permette di applicare un flusso di drill-down fino al singolo dettaglio alla minima granularità, così da soddisfare necessità d'uso diversificate. Il sistema è inoltre riuscito a rendere evidente al committente le potenzialità di strumenti di location intelligence, risultando un elemento chiave per la decisione di investire ulteriormente su questa tecnologia, così da arricchire il portafoglio applicativo aziendale di strumenti di analisi geografica predittiva e prescrittiva.

Questo lavoro di tesi ha dimostrato come un sistema di location intelligence può fornire un importantissimo valore aggiunto a una grande azienda, portando non solo un nuovo strumento e nuove capacità di analisi ma, più importante, nuove domande che prima le persone non sapevano di poter porre e che, ora, possono avere risposta.

Bibliografia

- [1] Matteo Golfarelli et al. “Lily: A geo-enhanced library for location intelligence”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. A cura di Ladjel Bellatreche e Mukesh K. Mohania. Vol. 8057 LNCS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 72–83. ISBN: 9783642401305. DOI: 10.1007/978-3-642-40131-2_7.
- [2] Business Week. “Location Intelligence : The New Geography of Business”. In: *BUSINESS WEEK RESEARCH SERVICES* (2007).
- [3] Brennon Martin. “Bringing Location Intelligence to the Enterprise”. In: (2010).
- [4] LT Moss e S Atre. “Business intelligence roadmap: the complete project lifecycle for decision-support applications”. In: (2003).
- [5] Dawn M. Turner. *What is Venture Management?*
- [6] Wikipedia.org. *John Snow*.
- [7] Wikipedia.org. *Epidemia di colera a Broad Street del 1854*.
- [8] H Brody et al. “Map-making and myth-making in Broad Street: the London cholera epidemic, 1854”. In: *The Lancet* (2000).

- [9] E A Parkes. “Mode of communication of cholera by John Snow, MD: second edition - London, 1855, pp 162.” In: *International journal of epidemiology* 42.6 (2013), pp. 1543–52. ISSN: 1464-3685. DOI: 10.1093/ije/dyt193.
- [10] Paul A Longley et al. *Geographical Information Systems and Science*. 3rd. Vol. 83. 3. Wiley Publishing, 2011, p. 537. ISBN: 0470870001. DOI: 10.2307/215736.
- [11] Vittorio Maniezzo. *Introduzione ai Sistemi Informativi Geografici (GIS) - slide corso Sistemi di Supporto alle Decisioni*. 2013.
- [12] P L N Raju. “Fundamentals of geographical information system”. In: *Satellite Remote Sensing and GIS Applications in Agricultural Meteorology - Proceedings of the training workshop* (2003), pp. 103–120.
- [13] Wikipedia.org. *Geodesia*.
- [14] F Canters. *Small-scale map projection design*. 2002.
- [15] Vittorio Maniezzo. *Introduzione alle Mappature Proiettive - slide corso Sistemi di Supporto alle Decisioni*. 2013.
- [16] Paolo Mogorovich. *Nozioni di Cartografia*.
- [17] Wikipedia.org. *WGS84*.
- [18] M Kumar. “World Geodetic System 1984: A modern and accurate global reference frame”. In: *Marine Geodesy* (1988).
- [19] Wikipedia.org. *Proiezione universale trasversa di Mercatore*.
- [20] Yin Liang e Hong Zhang. “Design and Implementation of Enterprise Spatial Data Warehouse”. In: *Research and Practical Issues of Enterprise Information Systems II Volume 1*. Boston, MA: Springer US, 2007, pp. 75–83. DOI: 10.1007/978-0-387-75902-9_8.

-
- [21] P Rigaux, M Scholl e A Voisard. *Spatial databases: with application to GIS*. 2001.
- [22] Oracle. *Oracle Spatial Users' Guide and Reference*. 2016.
- [23] Y Bédard, T Merrett e J Han. “Fundamentals of spatial data warehousing for geographic knowledge discovery”. In: *Geographic data mining and (2001)*.
- [24] S Rivest, Y Bédard e P Marchand. “Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP)”. In: *GEOMATICA-OTTAWA-* (2001).
- [25] Marco Vignoli. *Location Intelligence: un'applicazione per la pubblica sicurezza*. 2010.
- [26] Ravi Kothuri, Albert. Godfrind e Euro. Beinat. *Pro Oracle Spatial for Oracle database 11g*. Apress, 2007, p. 787. ISBN: 1590598997.
- [27] Roy Ernest. *Introduction to SQL Server Spatial Data*. 2015.
- [28] EPSG. *EPSG Geodetic Parameter Dataset*.
- [29] Oracle. *Coordinate System Transformation Functions in Oracle*.
- [30] OGC. *Open GIS Consortium*.
- [31] OGC. “OpenGIS simple features specification for SQL”. In: *OpenGIS Project Document 99 49* (1999), pp. 49–99.
- [32] ISO/IEC. *ISO/IEC 13249-3:2003*.
- [33] OGC Portal. *Geographic Markup Language (GML) Version 3 - OGC Portal*. 2002.
- [34] Arcgis.com. *ST_Geometry*.

-
- [35] N Beckmann, HP Kriegel e R Schneider. “The R*-tree: an efficient and robust access method for points and rectangles”. In: *ACM SIGMOD* (1990).
- [36] Matteo Golfarelli. *Classificatori - Slide del corso di Data Mining*. 2015.
- [37] A Guttman. “R-trees: a dynamic index structure for spatial searching”. In: (1984).
- [38] IEEE Computer Society. “IEEE Standard Glossary of Software Engineering Terminology”. In: (1990).
- [39] Project Management Institute. *Project Management Body of Knowledge (PMBOK® GUIDE)*. 4th. Project Management Institute, Inc., 2001. ISBN: 978-1-933890-51-7.
- [40] RK Wysocki. “Effective project management: traditional, agile, extreme”. In: (2011).
- [41] Marco A. Boschetti. “Scoping Process Group - slide del corso Project Management”. In: (2014).
- [42] Wikipedia.org. *Distanza Euclidea*.
- [43] Matteo Golfarelli e Stefano Rizzi. *Data Warehouse Design: Modern Principles and Methodologies*. 1^a ed. New York, NY, USA: McGraw-Hill, Inc., 2009. ISBN: 0071610391, 9780071610391.
- [44] RC Tausworthe. “The work breakdown structure in software project management”. In: *Journal of Systems and Software* (1979).
- [45] Anna Moore. *How to Convert a Text file (XY Table format) to a Shapefile for use in ArcGIS*. 2009.
- [46] Simon Greener. *Generating multi-points from single point records in Oracle Spatial*. 2007.

-
- [47] Simon Greener. *Building Lines into Polygons in Oracle Locator / Spatial*. 2011.
- [48] Oracle. *SDO_GEOM Package - Oracle Help Center*.
- [49] Istat. *Zone di censimento - Istat*. 2016.
- [50] Raju Vatsavai e Budhendra Bhaduri. “Geospatial Analytics in the era of Big Data and Extreme Scale Computing”. In: (1970).
- [51] William Vorhies. *A Step-by-Step Plan for Getting Your Company Started with Predictive Analytics*.
- [52] Brad Harkavy. *Prescriptive Modeling – What Amazon is Teaching Grid Operators*.

