

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN INGEGNERIA INDUSTRIALE
CICLO XXIII

Sviluppo e validazione sperimentale di
un framework software per la
trasformazione di dati RFID in
informazioni a valore aggiunto nella
supply chain fashion

Development of a software framework for
transforming RFID data in value-added
information for the fashion supply chain

Coordinatore:

Chiar.mo Prof. Antonio Rizzi

Dottorando:

Paolo Simonazzi

Indice

INDICE	2
INDICE DELLE FIGURE	4
INDICE DEI LISTATI	6
INTRODUZIONE	8
1. EPC NETWORK E BUSINESS INTELLIGENCE	11
1.1. EPCglobal Network	11
1.1.1. Descrizione	11
1.1.2. Tecnologia RFID	13
1.1.3. EPCIS	14
1.1.4. Discovery Services	20
1.2. Business Intelligence e Data Warehouse	22
1.2.1. Business Intelligence.....	22
1.2.2. Data Warehousing	23
1.2.3. Dashboard.....	24
2. IL PROGETTO PILOTA	25
2.1. Le attività di RFID lab nel settore fashion	25
2.1.1. RFID lab e il Board of Advisors	25
2.2. RFID Fashion Pilot	28
2.2.1. Obiettivi del progetto	28
2.2.2. Panoramica della supply chain	30
2.2.3. Architettura software	33
2.2.4. Hardware utilizzato nel progetto.....	38

3. SVILUPPO SOFTWARE	41
3.1. Tecnologie impiegate	41
3.1.1. Fosstrak EPCIS.....	41
3.1.2. Oracle Glassfish Application Server	43
3.1.3. Hibernate ORM	45
3.1.4. MySQL RDBMS.....	46
3.1.5. Google Web Toolkit	48
3.1.6. Isomorphic SmartGWT.....	50
3.2. Sviluppo software	51
3.2.1. Architettura del sistema	51
3.2.2. Data warehouse.....	53
3.2.3. Aggiornamento data warehouse	59
3.2.4. Dashboard.....	66
3.2.4.1. Applicazione web	66
3.2.4.2. Modulo Track&Trace	68
3.2.4.3. Modulo Inventory	74
3.2.4.4. Modulo Fitting	80
3.2.4.5. Modulo Replenishment	85
3.2.4.6. Modulo History	88
3.2.5. Esportazione dati	90
4. LA SPERIMENTAZIONE	92
4.1. Le statistiche della sperimentazione	92
4.2. Informazioni a valore aggiunto	93
4.2.1. Le tipologie di informazioni a valore aggiunto	93
4.2.2. Esempio I: Rapporto tra prove in camerino e vendite.....	96
4.2.3. Esempio II: Valutazione dell'impatto sul fatturato	98
5. CONCLUSIONI	100
BIBLIOGRAFIA.....	103

Indice delle figure

Figura 1 - EPCglobal Standards, (c) GS1, riprodotta con autorizzazione.....	13
Figura 2 - Architettura EPCIS.....	16
Figura 3 - Eventi EPCIS.....	17
Figura 4 - Diagramma della supply chain.....	31
Figura 5 - La supply chain del progetto	32
Figura 6 - Architettura software	34
Figura 7 - Architettura software nel dettaglio.....	35
Figura 8 - Architettura di rete.....	37
Figura 9 - Architettura di rete nel punto vendita.....	38
Figura 10 - Architettura di SmartGWT	51
Figura 11 - Diagramma dell'architettura software.....	53
Figura 12 - Diagramma di funzionamento dell'aggiornamento automatico	62
Figura 13 - Schermata iniziale Dashboard.....	66
Figura 14 - Particolare della schermata di aggiornamento.....	67
Figura 15 - Schermata Track&Trace	69
Figura 16 - Schermata Inventory.....	74
Figura 17 - Schermata Inventory in modalità dettagliata.....	77
Figura 18 - Schermata Fitting	81
Figura 19 - Schermata Replenishment.....	85
Figura 20 - Schermata History.....	89
Figura 21 - La metafora dell'“iceberg” RFID	94

Figura 22 - Distribuzione dei tempi tra prova e vendita di un capo	97
Figura 23 - Rapporto tra numero di prove e vendite	97
Figura 24 - Distribuzione del tempo tra ripristino e vendita di un capo	99

Indice dei listati

Listato 1 - Alcuni parametri di configurazione di MySQL.....	48
Listato 2 - Tabella dw_epcis_ean13	54
Listato 3 - Tabella dw_epcis_ext.....	54
Listato 4 - Tabella dw_epcis_color	55
Listato 5 - Tabella dw_epcis_receiving.....	55
Listato 6 - Tabella dw_epcis_last_process	56
Listato 7 - Tabella dw_view_tracktrace.....	57
Listato 8 - Tabella dw_view_inventory.....	57
Listato 9 - Tabella dw_view_inventory_total.....	58
Listato 10 - Tabella dw_view_replenishment.....	59
Listato 11 - Tabella dw_log	61
Listato 12 - Tabella dw_log_query.....	61
Listato 13 - Definizione della procedura update_all	65
Listato 14 - Definizione procedura create_view_tracktrace	74
Listato 15 - Definizione procedura create_view_inventory_total	79
Listato 16 - Definizione procedura update_view_inventory_total.....	80
Listato 17 - Definizione procedura create_view_fitting_epcs.....	84
Listato 18 - Estratto dalla definizione della procedura create_view_fitting	85

Introduzione

La tecnologia RFID (Radio Frequency Identification) rappresenta una innovazione di grande impatto non soltanto sul piano operativo, ma anche nella gestione delle informazioni acquisite dai processi logistici. L'attuale scenario competitivo globale impone la necessità di ridefinire il modello dei rapporti tra le imprese partecipanti alla catena logistica, in un'ottica di crescente collaborazione e condivisione delle informazioni. È questo il contesto ottimale per l'introduzione di sistemi basati su RFID per la raccolta e la condivisione dei dati concernenti i processi logistici, secondo gli standard definiti da EPCglobal. Le applicazioni RFID infatti possono essere considerate la porta d'ingresso verso una nuova fase di sviluppo della società dell'informazione, spesso denominata "Internet degli Oggetti", nella quale la rete globale internet collega non soltanto computer o altri terminali di comunicazione, ma anche gli oggetti e i beni di consumo appartenenti al nostro ambiente quotidiano.

Il presente elaborato riguarda un'attività di sperimentazione nata dalla collaborazione tra l'Università degli Studi di Parma e diverse aziende nazionali e multinazionali del territorio locale. Tale progetto di ricerca, denominato RFID Fashion Pilot, si propone di reingegnerizzare tramite l'introduzione della tecnologia RFID una supply chain basilare che comprende un centro di distribuzione ed un punto vendita nel settore tessile e dell'abbigliamento. Le aziende partecipanti, nello spirito del Board of Advisors, condividono i costi del progetto, le scelte progettuali e il know-how ricavato dalle sperimentazioni. Il

progetto, oltre a valutare la fattibilità tecnica della proposta, ha l'obiettivo di verificare le potenzialità della tecnologia RFID in relazione alla tracciabilità dei prodotti, all'ottimizzazione dei processi logistici, all'aumento dell'affidabilità dei dati di inventario e all'incremento di fatturato grazie alla visibilità delle informazioni derivante dall'EPCglobal Network, basata sugli EPCIS (EPC Information Services). Nel corso del progetto viene tracciato mediante tecnologia RFID il percorso logistico di imballaggi secondari (colli o case) di tutte le referenze all'interno della supply chain di riferimento: lo standard EPCIS fornisce le specifiche per la memorizzazione su database di tali dati relativi ad eventi RFID. La mole di informazioni raccolte in questo modo, unita alla difficoltà della loro interpretazione, rende necessario l'utilizzo di adeguati framework di elaborazione e Business Intelligence (BI) per presentare i dati in forma comprensibile all'utente finale, estraendo indicatori rilevanti delle performance della supply chain. La BI permette di utilizzare i dati residenti nei sistemi informativi aziendali trasformandoli in informazioni utili a definire e supportare le strategie di business e i processi decisionali.

La presente trattazione ha per oggetto proprio lo sviluppo e la validazione sperimentale di moduli software per la Business Intelligence, in grado di generare informazioni a valore aggiunto a partire dai dati RFID acquisiti lungo la catena logistica. Il software realizzato si occupa di estrarre i dati grezzi tramite i protocolli standard dell'EPCglobal Network, trasformarli in informazioni utili che vengono memorizzate e mantenute aggiornate in una Data Warehouse dedicata, ed infine presentare tali informazioni all'utente tramite un'interfaccia grafica di tipo web. Questa interfaccia, definibile come Dashboard o cruscotto logistico, offre all'utente la possibilità di interrogare il sistema di Data Warehouse per ottenere risultati sia di carattere aggregato, relativi alle prestazioni dell'intera catena logistica, che di carattere analitico, relativi ai singoli colli movimentati o ai singoli processi.

La Dashboard si compone di svariati moduli, ciascuno atto a presentare una particolare tipologia di indicatori prestazionali o informazioni logistiche, come ad esempio dati di tracciabilità di colli e lotti, giacenze, dati di vendita, gestione di ripristini e riordini di prodotto. Tramite l'interfaccia è inoltre possibile

valutare pressoché in tempo reale le prestazioni dei dispositivi RFID installati nei vari punti di lettura sia nel CeDi che nel punto vendita.

1. EPC Network e Business Intelligence

1.1. EPCglobal Network

1.1.1. Descrizione

L'EPCglobal Network è un'infrastruttura globale per lo scambio di dati basata su Electronic Product Code (EPC), un sistema di identificazione univoco per le merci. Lo sviluppo di EPC ed EPC Network è guidato da EPCglobal¹, l'organizzazione internazionale che presiede alla definizione e allo sviluppo degli standard relativi alla cosiddetta "Internet of Things". L'EPC Network può infatti essere descritto come un'infrastruttura ubiqua per collegare gli oggetti fisici alla rete informatica globale (Cole et al., 2008), realizzando così la visione dell'Internet of Things.

EPCglobal definisce l'EPC Network come un sistema basato su Internet per l'accesso a grandi quantità di dati logistici, che possono essere condivisi tra partner commerciali autorizzati (EPCglobal Inc., 2004). L'obiettivo dell'EPC Network è infatti quello di offrire la visibilità completa del flusso dei prodotti

¹ Sito web: <http://www.gs1.org/epcglobal>

lungo tutta la supply chain, garantendo un miglioramento dell'efficienza e una razionalizzazione dei processi logistici, attraverso la rintracciabilità e la tracciabilità delle merci dotate di codice EPC.

La visione di EPCglobal non è vincolata ad una particolare tecnologia, in quanto la sua realizzazione è indipendente dalla particolare implementazione tecnologica utilizzata per veicolare l'informazione contenuta nell'EPC (Brock, 2001). Tuttavia, l'Electronic Product Code è stato espressamente concepito per garantire l'interoperabilità della tecnologia RFID nelle applicazioni logistiche globali (Floerkemeier, 2009); il suo sviluppo è stato quindi fin dalle origini legato a quello della tecnologia RFID, che resta tuttora l'implementazione tecnologica di riferimento su cui si basa l'EPC Network per collegare gli oggetti fisici con i sistemi informativi (Sarma, 2005).

L'EPCglobal Network è fondato su una serie di standard accomunati dall'obiettivo di facilitare l'innovazione e lo scambio di informazioni nella supply chain, e favorire lo sviluppo di un mercato aperto e competitivo per gli attori coinvolti; si tratta quindi di un'architettura comune ed integrata all'interno della quale coesistono varie iniziative promosse da EPCglobal. È possibile quindi affermare che l'EPC Network segue il paradigma SOA (Service Oriented Architecture), caratterizzato da un sistema di moduli indipendenti ed intercomunicanti, organizzati su vari livelli (Erl, 2004).

All'interno di questo framework si possono individuare cinque moduli principali:

- Electronic Product Code (EPC): il codice numerico univoco ed universale usato per identificare gli elementi nella catena logistica, mediante tecnologia RFID.
- Auto-ID System: il sistema di identificazione automatica basato su tag RFID che vengono letti da appositi dispositivi (reader RFID).
- Middleware: lo strato software che si colloca tra l'hardware dei lettori RFID e le applicazioni software di alto livello che utilizzano i dati acquisiti.
- Object Naming Service (ONS) e altri Discovery Service (DS): i servizi software che consentono la localizzazione sulla rete Internet delle

informazioni relative ad un dato EPC.

- EPC Information Services (EPCIS): il modulo che registra tutte le informazioni che circolano attraverso l'EPC Network, e le rende disponibili agli attori della supply chain.

Nel seguito sono descritti in maniera più approfondita gli elementi sui quali si fondano i moduli di Business Intelligence realizzati nel progetto.

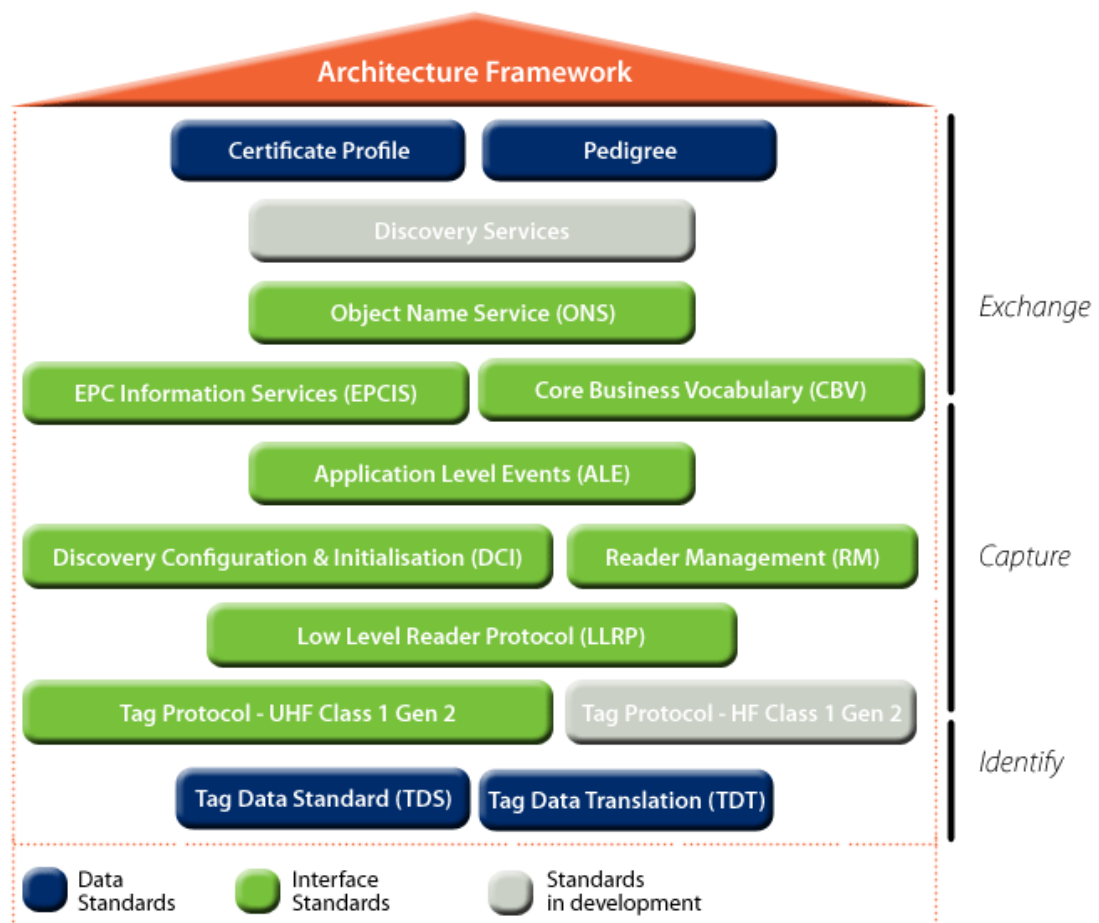


Figura 1 - EPCglobal Standards, (c) GS1, riprodotta con autorizzazione

1.1.2. Tecnologia RFID

L'auto identificazione (Auto-ID) è un insieme di tecnologie e strumenti con lo

scopo di automatizzare l'identificazione degli oggetti. L'Auto-ID permette infatti alle macchine di rilevare autonomamente e senza intervento umano le informazioni che consentono di identificare determinati oggetti. Tra le tecnologie impiegate nei sistemi Auto-ID vi sono i codici a barre, i sistemi di riconoscimento ottico, di riconoscimento vocale e, recentemente, di riconoscimento a radiofrequenza.

L'identificazione a radiofrequenza o RFID (Radio frequency identification) è una tecnologia che permette l'identificazione automatica e la raccolta dei dati attraverso le frequenze radio (Domdouzis, 2007). Uno dei primi lavori pubblicati sull'RFID è quello di H. Stockman, nato dalle ricerche in campo radio svolte durante la Seconda Guerra Mondiale (Stockman, 1948).

La caratteristica principale di tale tecnologia è la possibilità di associare un identificatore univoco e altre informazioni, per mezzo di un microchip, a qualsiasi oggetto, e di permetterne la lettura mediante un dispositivo senza fili. L'implementazione più comune della tecnologia RFID infatti prevede di memorizzare le informazioni dell'oggetto all'interno di un microchip collegato ad un'antenna (l'insieme di microchip e antenna è denominato transponder, o tag), e di far viaggiare oggetto e tag insieme. Al sollecito da parte di un'antenna idonea, il microchip invierà le proprie informazioni ad un dispositivo di ricezione (detto reader), il quale provvederà a renderle disponibili ai sistemi informativi.

Il formato standard per l'informazione identificativa contenuta nei tag RFID, come si è accennato, è l'Electronic Product Code, gestito a livello mondiale da EPCglobal Inc., in congiunzione con le varie organizzazioni GS1 nazionali (Indicod-Ecr² per l'Italia).

1.1.3. EPCIS

Nelle applicazioni RFID tradizionali, come il controllo degli accessi o

² <http://indicod-ecr.it>

l'identificazione del bestiame, i tag circolavano in un sistema chiuso e le informazioni generate venivano utilizzate soltanto da una singola applicazione. Pertanto, non vi era la necessità di distribuire i dati tra più organizzazioni diverse. In tempi recenti, analogamente a quanto è avvenuto per i sistemi informativi del passato, che si sono evoluti da applicazioni locali a sistemi ad elevata integrazione accessibili da Internet, anche la tecnologia RFID viene sempre più spesso utilizzata in catene logistiche molto articolate e composte da molteplici impianti di produzione, magazzini e punti vendita (Floerkemeier, 2007). Questo processo evolutivo ha introdotto la necessità di un'interfaccia comune per il trasferimento delle informazioni acquisite tramite RFID tra sistemi diversi: i servizi EPC Information Services sono nati appunto per rispondere a questa esigenza.

L'EPCIS, infatti, è il componente dell'architettura EPC Network che costituisce il principale veicolo di scambio delle informazioni condivise tra più partner, e le sue specifiche comprendono sia la definizione di una interfaccia standard di comunicazione, che il formato dei dati stessi. Le principali funzioni di questo modulo, all'interno della struttura dell'EPC Network, sono:

- ricezione dei dati RFID dallo strato "Filtering and Collection Middleware", ovvero dal software che controlla i dispositivi RFID e acquisisce i dati dal campo;
- traduzione di questi dati grezzi in eventi significativi per i processi di business;
- pubblicazione in rete degli eventi di processo, in modo da renderli disponibili ai partner autorizzati.

A queste tre distinte funzionalità corrispondono tre differenti moduli in cui può essere suddivisa l'architettura interna dell'EPCIS:

- EPCIS Capturing Application
- EPCIS Repository
- EPCIS Accessing Application

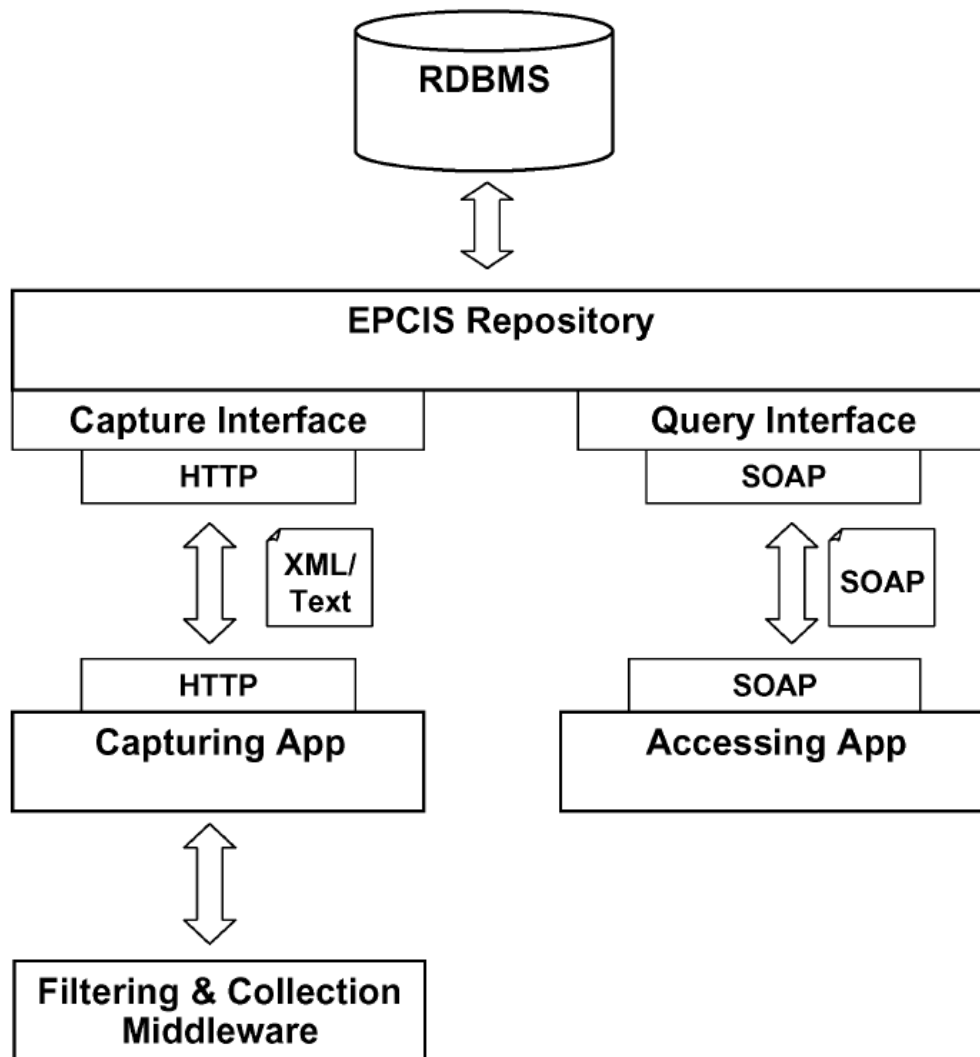


Figura 2 - Architettura EPCIS

Il primo è denominato EPCIS Capturing Application, e si occupa di interfacciarsi con il middleware per acquisire i dati di lettura RFID. I dati acquisiti vengono poi interpretati e trasformati in eventi significativi per i processi di business (ad esempio, il ricevimento di un pallet di merci). Questi “eventi EPCIS” (EPCIS events) seguono una struttura definita da EPCglobal e parte dello standard EPCIS: attualmente la specifica di riferimento è la versione 1.0.1, rilasciata nel novembre 2007. Le quattro tipologie di eventi previste sono:

- **Object event:** rappresenta un evento che coinvolge uno o più oggetti denotati da un codice EPC;

- Aggregation event: rappresenta un evento accaduto a uno o più oggetti, denotati da un codice EPC, fisicamente aggregati tra loro (ad esempio i colli di uno stesso pallet);
- Quantity event: rappresenta un evento riferito a una quantità di oggetti che condividono una classe EPC comune;
- Transaction event: rappresenta un evento in cui uno o più oggetti denotati da un codice EPC vengono associati o dissociati con una o più transazioni aziendali, codificate ad esempio tramite un identificativo di bolla d'ordine o DDT (documento di trasporto).

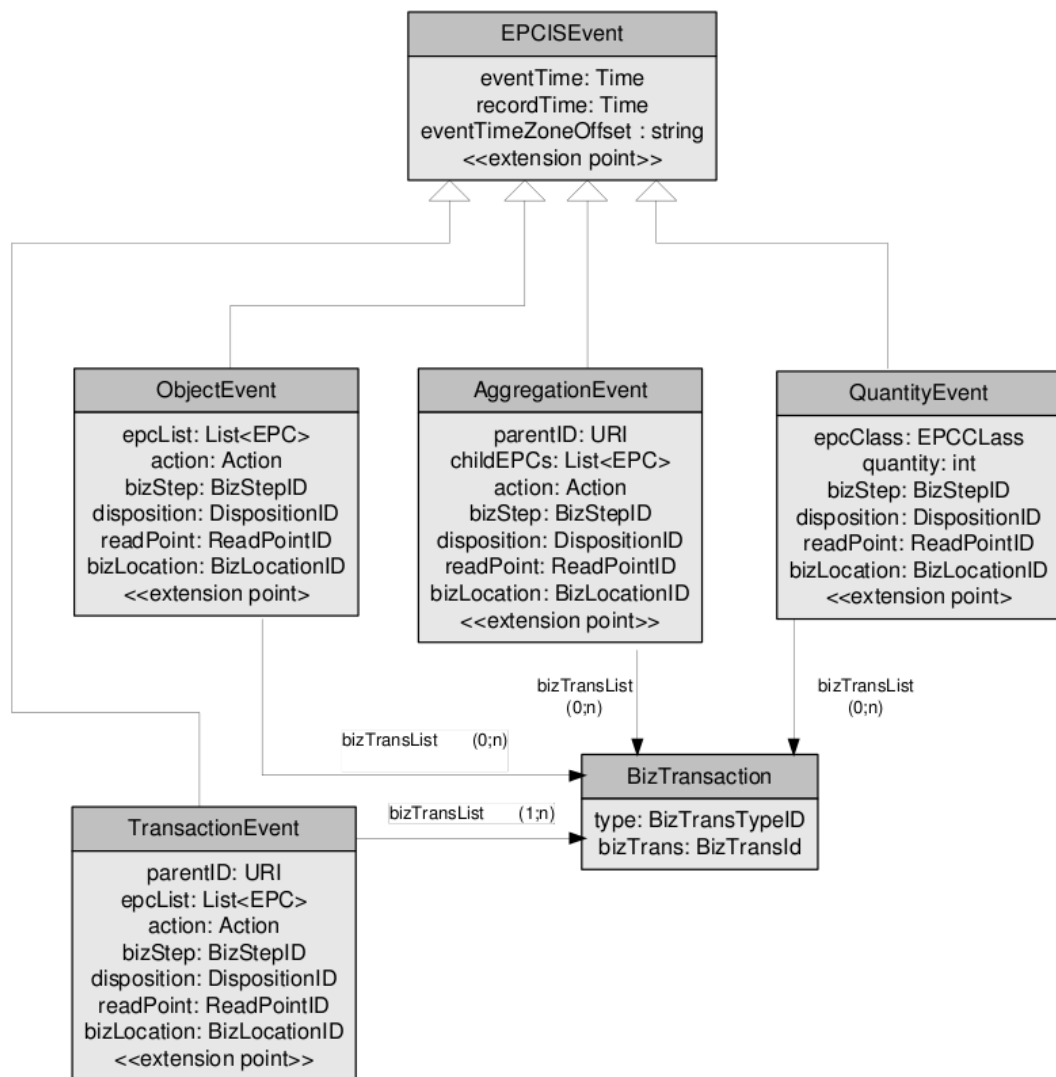


Figura 3 - Eventi EPCIS

Ciascun evento ha una serie di campi che rappresentano quattro dimensioni chiave di ogni evento EPCIS: gli oggetti o altre entità soggette all'evento; la data e l'ora dell'evento; la locazione dell'evento; il contesto di business. La prima dimensione varia a seconda del tipo di evento, mentre le dimensioni temporale e spaziale presentano due aspetti: uno "prospettivo" ed uno "retrospettivo", rappresentati da due differenti campi. In aggiunta ai campi delle quattro dimensioni principali, gli eventi possono portare ulteriori informazioni descrittive in campi aggiuntivi. Lo standard EPCIS riporta un solo campo descrittivo, bizTransactionList, che indica il particolare contesto aziendale all'interno del quale si è verificato l'evento. Lo standard EPCIS definisce quali debbano essere i tipi di dato da inserire nei campi, nonché le specifiche sintattiche da adottare nel loro inserimento. Questi tipi di dato sono:

- Primitive: le tipologie di dato primitive descritte dallo standard sono tre:
 - int: un numero intero;
 - time: una data ed un'ora in una precisa timezone;
 - EPC: un codice EPC.
- Dati di tipo Action: esprimono la relazione tra l'evento e il ciclo di vita dell'entità descritta. Tale campo è un'enumerazione che può assumere soltanto tre valori:
 - add: l'entità in questione è stata creata o aggiunta;
 - observe: l'entità in questione non ha subito mutamenti, ma è stata semplicemente rilevata dal sistema;
 - delete: l'entità in questione è stata rimossa.
- Dati di tipo Location:
 - ReadPointID: i Read Point sono i luoghi fisici in cui si registra l'evento, e sono determinati dalla EPCIS Capturing Application;
 - BusinessLocationID: una Business Location rappresenta il luogo in cui si succedono una serie di eventi.
- Dati legati al reader:
 - PhysicalReaderID: reader inteso come dispositivo fisico;
 - LogicalReaderID: reader inteso come unità logica, in quanto un

singolo punto di lettura può essere attrezzato con uno o più reader fisici.

- **BusinessStep**: denota un passo all'interno di un processo di business. Pertanto questo campo specifica il contesto in cui ha avuto luogo un evento.
- **Disposition**: la Disposition denota lo stato di un oggetto, ovvero specifica la condizione, successiva all'evento, degli oggetti coinvolti (ad esempio "venduto").
- **BusinessTransaction**: identifica una particolare Business Transaction, ad esempio una bolla d'ordine, ed è identificata da una coppia di valori: **BusinessTransactionTypeID** e **BusinessTransactionID**

Il secondo modulo è l'EPCIS Repository, la cui funzione è rendere persistenti gli eventi EPCIS, ad esempio memorizzandoli in un database relazionale. Gli standard di EPCglobal non forniscono alcuna specifica riguardo alla tecnologia da utilizzare per implementare il Repository, ma soltanto le interfacce di comunicazione e le strutture dati: questo garantisce la massima flessibilità e modularità, consente la distribuzione del lavoro di sviluppo tra più organizzazioni e permette una evoluzione indipendente, e dunque più rapida, dei singoli componenti.

Oltre alla modularità, un'altra caratteristica fondamentale che ha guidato lo sviluppo dell'EPCIS è l'estendibilità di tutti i suoi componenti (EPCIS specification, 2007): sia i tipi di dato che le interfacce infatti possono essere estesi con nuovi campi e funzionalità dai fornitori dei servizi EPCIS, senza per questo alterare lo standard.

L'EPCIS Repository offre due interfacce distinte per la comunicazione con moduli esterni:

- EPCIS Capture Interface
- EPCIS Query Interface

L'EPCIS Capture Interface è l'interfaccia di collegamento con le EPCIS Capturing Application definite sopra, e consente l'inserimento di nuovi eventi EPCIS nel

repository.

L'EPCIS Query Interface, viceversa, permette l'estrazione degli eventi EPCIS immagazzinati nel Repository. Lo standard definisce un linguaggio di interrogazione che permette di filtrare gli eventi in base a numerosi parametri, e che sarà discusso nel seguito della trattazione.

Il terzo ed ultimo modulo EPCIS è denominato Accessing Application. Appartengono a questa categoria tutte le applicazioni che estraggono ed utilizzano le informazioni contenute nell'EPCIS Repository tramite l'EPCIS Query Interface. Sono quindi da considerarsi Accessing Application i WMS (Warehouse Management System), così come i sistemi di Business Intelligence e qualunque applicazione basata sui dati forniti dall'EPC Network. Anche la Dashboard oggetto di questa trattazione rientra nella definizione di Accessing Application.

1.1.4. Discovery Services

Un altro componente fondamentale dell'architettura EPC Network è costituito dai Discovery Services, la cui funzione è quella di localizzare all'interno della rete i servizi EPCIS che contengono informazioni relative a un dato codice EPC. Una prima tipologia di Discovery Services è rappresentata dall'ONS. L'ONS (Object Naming Service) fornisce un registro centralizzato che stabilisce un'associazione statica tra un EPC e il produttore della merce identificata da quell'EPC. Le applicazioni basate su EPC Network (sia Capturing Application che Accessing Application) possono utilizzare l'ONS per ottenere l'indirizzo del servizio EPCIS che contiene le informazioni del produttore: tali informazioni sono generalmente di natura statica, quali lotto, data di scadenza, o peso. La funzione dell'ONS è concettualmente affine a quella dei sistemi DNS (Domain Name System), che forniscono il servizio di instradamento su Internet: così come il DNS viene interrogato dalle applicazioni con un nome di dominio internet, e restituisce l'indirizzo IP del server associato, così l'ONS viene

interrogato con un codice EPC e restituisce l'indirizzo del server EPCIS associato.

Data questa affinità funzionale, l'ONS condivide la stessa architettura del DNS, costituita da una gerarchia di nodi distribuiti, ciascuno dei quali contiene solo un sottoinsieme delle informazioni di instradamento, e per le informazioni che non possiede inoltra le richieste ad altri nodi. Al vertice di questa gerarchia vi sono i cosiddetti Root Server. Anche per il modulo ONS le specifiche dell'EPC Network non sono vincolate ad una particolare implementazione tecnologica; generalmente è conveniente sfruttare la struttura esistente del sistema globale DNS, estendendola per svolgere anche le funzioni dell'ONS. Questa estensione consiste nell'inserire nei database DNS anche le associazioni tra codici EPC (o super-classi di codici EPC) e indirizzi EPCIS relativi. Interrogando opportunamente i server DNS si otterranno anche queste associazioni ONS.

Il design dei sistemi ONS permette di gestire esclusivamente l'associazione tra EPC ed EPCIS del produttore dei beni: non è consentito specificare altri EPCIS, come ad esempio quelli relativi ad altri attori della filiera logistica.

Per questa ragione, le specifiche dell'EPC Network sono state estese per comprendere meccanismi più generali di Discovery Service. Questi servizi consentono alle applicazioni di individuare tutti gli EPCIS della catena logistica. I DS sono quindi in grado di fornire informazioni anche di tipo dinamico relative agli EPC, come ad esempio dati di tracciabilità aggiornati in tempo reale, valori di inventario lungo tutti i nodi della supply chain, date di spedizione, ricevimento e altro. In questa ottica dunque l'ONS può essere considerato come una versione specializzata di Discovery Service, che gestisce soltanto informazioni relative all'EPCIS del produttore.

La definizione dei DS, al momento attuale, non è ancora stata formalizzata da EPCglobal, e dunque non fa parte delle specifiche standard dell'EPC Network; si tratta quindi di una funzionalità prevista ma non ancora sviluppata (Armenio, 2007). Numerosi studi sono stati condotti riguardo all'architettura e implementazione dei Discovery Service, anche da parte di organizzazioni internazionali come Verisign, ma attualmente solo alcuni sono stati provati sul campo in supply chain reali, e nessuno è stato accolto come standard

(Kurschner, 2008; Verisign whitepaper; Cantero, 2008; Beier, 2006).

1.2. Business Intelligence e Data Warehouse

1.2.1. Business Intelligence

La presente trattazione ha per oggetto lo sviluppo di moduli software per la Business Intelligence, in grado di generare informazioni a valore aggiunto a partire dai dati RFID acquisiti dal campo. Il termine Business Intelligence (BI) si riferisce alle tecniche informatiche usate per l'analisi dei dati di business quali vendite, ricavi, inventari. Il primo utilizzo di questo termine in letteratura si riscontra in H. P. Luhn (1958), che lo definiva come "La capacità di cogliere le relazioni tra i dati in maniera tale da fornire una guida per il raggiungimento degli obiettivi desiderati". La BI è quindi un processo di trasformazione dei dati grezzi in conoscenza utile per prendere decisioni strategiche attraverso informazioni aggiornate e rilevanti. La portata del termine Business Intelligence è stata in seguito espansa da H. Dresner (Power, 2007), fino ad includere tutti i "concetti e metodi per il supporto decisionale basato su informazioni". Gli strumenti di Business Intelligence nella sua accezione moderna includono:

- la generazione di report
- il data mining
- il Business Performance Management (BPM)
- i Decision Support System (DSS)
- l'analisi statistica e predittiva
- l'Online Analytical Processing (OLAP)
- la Market Basket Analysis (MBA)

Ciascun sistema di BI ha obiettivi specifici condizionati dalla visione aziendale e dalla gestione strategica dell'azienda. In generale le funzioni della Business Intelligence comprendono l'analisi della situazione corrente dell'azienda e dell'andamento delle performance, la generazione di stime previsionali e la messa a punto di strategie finanziarie, di marketing, di controllo di gestione. Le fonti di informazione con cui alimentare i sistemi di Business Intelligence hanno generalmente provenienza interna, derivano cioè dai sistemi informativi aziendali come Customer Relationship Management (CRM), contabilità, Warehouse Management System (WMS). Tutti questi dati di input, e i risultati delle analisi di Business Intelligence, vengono immagazzinati nei sistemi informatici secondo una logica di Data Warehousing.

1.2.2. Data Warehousing

Il concetto di Data Warehouse venne formalizzato in letteratura da B. Devlin e P. Murphy (1988), che descrivevano un modello architetturale per la gestione dei dati utilizzati dai sistemi di supporto decisionale. Con il termine Data Warehouse (DW) si intende un archivio informatico contenente i dati di un'organizzazione, organizzato in maniera tale da ottimizzare l'esecuzione di analisi di Business Intelligence. Le caratteristiche principali di un DW, rispetto ai generici sistemi database, sono le seguenti:

- i dati confluiscono nel DW da più fonti diverse, e vengono ivi integrati e resi omogenei;
- i dati sono archiviati in modo da agevolarne l'elaborazione: non vengono quindi impiegate tecniche di normalizzazione o di eliminazione delle ridondanze;
- i dati vengono inseriti in maniera incrementale, senza cioè eseguire operazioni di rimozione o modifica dei dati esistenti; anche le applicazioni che usufruiscono dei dati lo fanno in modalità di sola lettura;

- i dati comprendono un valore temporale, e possono quindi essere utilizzati per valutazioni storiche che coprono orizzonti temporali anche molto estesi; per contro, questi dati vengono in genere aggiornati periodicamente e non rispecchiano la situazione in tempo reale.

1.2.3. Dashboard

Il termine Dashboard è usato nell'ambito delle tecnologie informatiche per designare una classe di interfacce utente per presentare in maniera integrata informazioni provenienti da più fonti (Eckerson, 2006). L'espressione allude metaforicamente al cruscotto di un veicolo ("dashboard", appunto, in inglese), che ha l'analogia funzione di presentare in maniera intuitiva i dati sulle prestazioni. La dashboard appare oggi come la modalità primaria di visualizzazione dei dati forniti dai sistemi di Business Intelligence (Few, 2006). Una dashboard può presentarsi sotto forma di applicazione software stand-alone, oppure come applicazione web accessibile tramite browser. In entrambi i casi l'applicazione deve prevedere un'interfaccia di controllo per eseguire query sui dati e modificarne le modalità di visualizzazione. Spesso la dashboard fa uso di grafici, istogrammi, diagrammi a torta o tabelle con indicatori grafici.

2. Il progetto pilota

2.1. Le attività di RFID lab nel settore fashion

2.1.1. RFID lab e il Board of Advisors

Il laboratorio RFID Lab³ è un laboratorio di eccellenza a livello mondiale del Dipartimento di Ingegneria Industriale dell'Università degli Studi di Parma. Il laboratorio è attivo da più di tre anni, ed è stato il primo centro nato in Italia per la ricerca, sperimentazione e sviluppo della tecnologia RFID e delle sue applicazioni ai processi aziendali.

Assieme all'RFID Research Center dell'Università dell'Arkansas (USA) e al LogDynamics Lab dell'Università di Brema (Germania), RFID Lab è uno dei membri fondatori della Global RF Lab Alliance (GRFLA)⁴. GRFLA è un network di laboratori RFID, che si propone di raggruppare, a livello mondiale, centri di ricerca attivi sul tema dell'identificazione in radiofrequenza. Annunciato formalmente durante il convegno RFID Journal LIVE! 2007, GRFLA realizza collaborazioni nelle attività di ricerca sull'RFID e nella condivisione dei risultati.

³ <http://www.rfidlab.unipr.it>

⁴ <http://www.grfla.org>

Nel maggio 2008, il laboratorio RFID Lab ha avviato le attività di ricerca in ambito fashion mediante la costituzione di un gruppo di lavoro, denominato Board of Advisors (BoA) Fashion, sulla base della triennale esperienza del Board of Advisors nel settore food e largo consumo (FMCG), che ha visto unito un panel di 22 aziende potenziali utilizzatrici della tecnologia RFID (produttori, operatori logistici e distributori)⁵.

Il meccanismo del BoA presenta notevoli vantaggi per le aziende che ne fanno parte. In primo luogo, il BoA permette di sviluppare e condividere un adeguato know-how preliminare per identificare le opportunità e massimizzare i benefici che la tecnologia RFID può fornire per applicazioni aziendali. Inoltre, tramite il BoA, la ricerca preliminare può essere proficuamente gestita in outsourcing, grazie ad un ente specializzato come RFID Lab, e condivisa con altri stakeholder, consentendo all'azienda significativi risparmi in termini di risorse tecniche, umane ed economiche. In concreto, la partecipazione al Board offre alle aziende la possibilità di definire, seguire e acquisire i risultati delle attività di ricerca, di partecipare ad attività collaterali (quali seminari e workshop), di essere costantemente aggiornata sullo stato dell'arte delle applicazioni e della tecnologia, e di effettuare benchmarking con partner e concorrenti del settore. Le attività di ricerca avviate nell'ambito Fashion si propongono di affrontare i diversi ambiti relativi all'applicazione della tecnologia RFID al mondo dell'abbigliamento, spaziando dall'analisi delle prestazioni tecnologiche all'impatto sui processi di supply chain e di punto vendita, dalla contraffazione alla tutela della privacy del consumatore. Il BoA si è rivelato un sistema estremamente virtuoso per favorire il trasferimento tecnologico, il benchmarking e l'aggiornamento continuo sulle tematiche inerenti al sistema EPC e alla tecnologia RFID, che presenta caratteristiche particolarmente favorevoli nel settore fashion, quali:

- assenza di metalli e acqua;
- minore incidenza del costo del tag rispetto al costo dell'item;
- automazione dei flussi logistici e produttivi (ricevimenti, preparazione

⁵ Il sito web del progetto pilota FMCG è disponibile all'indirizzo <http://www.rfidlogisticspilot.com>

- ordini, spedizioni);
- integrazione e visibilità sui flussi fisici di prodotto con clienti e fornitori;
- efficienza nella gestione del punto vendita (ricevimento, inventari, riduzione dello stock out, servizio al cliente, taccheggio);
- lotta alla contraffazione.

Per l'anno 2009, al BoA Fashion hanno aderito 13 importanti aziende del settore tessile, calzaturiero, abbigliamento sportivo, accessori e altro. Tra queste, le seguenti hanno autorizzato la comunicazione del proprio marchio:

- AEFPE (Pollini, Moschino, Alberta Ferretti)
- ARTSANA (Chicco)
- BRANDED APPAREL ITALIA (Lovable, Fila, Playtex, DIM, Unno)
- DOLCE & GABBANA INDUSTRIA
- LOTTO SPORT ITALIA
- IMAX (gruppo Max Mara Fashion Group)
- MIROGLIO (Elena Mirò, Diana Gallesi, Caractère)
- TRUSSARDI (Trussardi, TRU Trussardi)

Le attività di ricerca del BoA Fashion si sono orientate inizialmente verso lo sviluppo del progetto "L'impatto della tecnologia RFID nella supply chain fashion" e verso la realizzazione di RFID Fashion Store. Il progetto di ricerca "L'impatto della tecnologia RFID nella supply chain fashion" si è posto come obiettivo quello di progettare e verificare la fattibilità tecnico-economica di una soluzione basata su tecnologia RFID per ottimizzare i principali processi logistici e di punto vendita della supply chain fashion. RFID Fashion Store è invece un punto vendita retail completo realizzato presso RFID Lab, in cui sono state integrate le tecnologie RFID per la gestione dei processi di punto vendita: dal ricevimento merci alla cassa, dalla gestione dell'inventario in tempo reale all'antitaccheggio.

Il laboratorio conta su una partnership tecnologica con circa una trentina di aziende. Questi partner tecnologici forniscono le tecnologie hardware e

software necessarie per la sperimentazione e ricevono un riscontro su prestazioni, compatibilità e possibilità di integrazione nei processi di business. Altre partnership complementari riguardano partner istituzionali e media partner.

2.2. RFID Fashion Pilot

2.2.1. Obiettivi del progetto

Dopo il progetto pilota RFID nel largo consumo (RFID Logistics Pilot), conclusosi nel 2008 e premiato da RFID Italia Award 2009 come miglior progetto di filiera, il laboratorio RFID Lab dell'Università di Parma ha avviato un'analogha sperimentazione della tecnologia RFID nell'ambito fashion. Il progetto, denominato RFID Fashion Pilot, è stato avviato nel settembre 2009 e ha permesso di realizzare una supply chain pilota in cui testare l'implementazione della tecnologia RFID e dell'internet degli oggetti nel settore tessile e dell'abbigliamento.

A questa prima fase del progetto pilota, durata complessivamente circa un anno, hanno partecipato le seguenti aziende aderenti al Board of Advisors Fashion del laboratorio RFID Lab:

- Branded Apparel Italia (gruppo DBA)
- Dolce & Gabbana Industria con i suoi due operatori logistici DHL e TNT
- Imax e Diffusione Tessile (Gruppo Max Mara)
- Miroglio Fashion
- Norbert Dentressangle
- Trussardi

Al termine di questa prima campagna sperimentale è stato deciso di proseguire la sperimentazione anche per la stagione autunno/inverno 2010. Le aziende che partecipano al proseguimento dell'attività in questa seconda fase sono:

- Branded Apparel Italia (gruppo DBA)
- Benetton
- Imax e Diffusione Tessile (Gruppo Max Mara)
- Miroglio Fashion
- Norbert Dentressangle

RFID Fashion Pilot è stato guidato dalle medesime scelte progettuali che caratterizzano lo spirito delle altre iniziative del Board of Advisors: gli obiettivi e le soluzioni adottate hanno sufficiente generalità per essere di interesse comune; le aziende partecipanti condividono i costi del progetto, portando così a una notevole riduzione dell'investimento da parte di ciascuna; le aziende, inoltre, siano esse partner o competitor, acconsentono a condividere il know-how ed i risultati della ricerca.

Il laboratorio RFID Lab ha svolto il ruolo di coordinatore scientifico del progetto, anche in collaborazione con le università mondiali che aderiscono a GRFLA. Nel progetto sono stati inoltre coinvolti partner tecnologici di RFID Lab quali Impinj, Motorola, MTI, Psion Teklogix, Skeye, Tele Servizi, UPM Raflatac e Zebra, che hanno fornito i dispositivi hardware e l'infrastruttura software necessaria alla sperimentazione. L'integrazione del progetto è stata seguita da Id-Solutions, spinoff dell'Università degli Studi di Parma e Alliance Partner di RFID Lab. Infine, RFID Fashion Pilot ha beneficiato del supporto tecnico di EPC Global tramite Indicod ECR, che ha fornito strumenti e standard della piattaforma EPCGlobal per la codifica e lo scambio delle informazioni.

Le aziende partner del progetto hanno messo a disposizione i propri siti e i capi di abbigliamento su cui effettuare le sperimentazioni. La supply chain interessata dal progetto è costituita dal centro di distribuzione (CeDi) dell'azienda Miroglio fashion, sito in Pollenzo (CN), e da un punto vendita Elena Mirò, monomarca, sito all'interno dell'outlet Fidenza Village di Fidenza (PR). Durante la prima fase del progetto sono stati taggati circa 15.000 capi di

abbigliamento relativi alla stagione primavera/estate 2010.

Il progetto si prefiggeva numerosi obiettivi. In primo luogo, grazie alla realizzazione di una supply chain pilota, RFID Fashion Pilot ha costituito uno scenario per valutare la fattibilità tecnica dell'implementazione della tecnologia RFID nel contesto del fashion, misurando il livello di accuratezza ottenibile nell'identificazione dei capi nei diversi processi logistici. In secondo luogo, il progetto ha permesso di quantificare l'impatto economico della tecnologia sui processi logistici e di punto vendita, in termini di riduzione di ore uomo, miglioramento dell'accuratezza dei dati di inventario, disponibilità di nuovi dati dai camerini, miglioramento del servizio al cliente, incremento del fatturato. La sperimentazione sul campo ha dunque permesso di quantificare i benefici della tecnologia RFID già evidenziati nel progetto di ricerca "L'impatto della tecnologia RFID nella supply chain fashion", a cui si è accennato sopra. Il progetto pilota aveva inoltre l'obiettivo di realizzare la tracciabilità completa ed in tempo reale, al livello del singolo capo, lungo tutta la supply chain interessata.

2.2.2. Panoramica della supply chain

La supply chain del pilota si compone di due attori: il CeDi di Miroglio Fashion, situato a Pollenzo (CN); e il punto vendita Elena Mirò, all'interno dell'outlet "Fidenza Village" di Fidenza (PR). La figura seguente ne mostra una rappresentazione schematica:

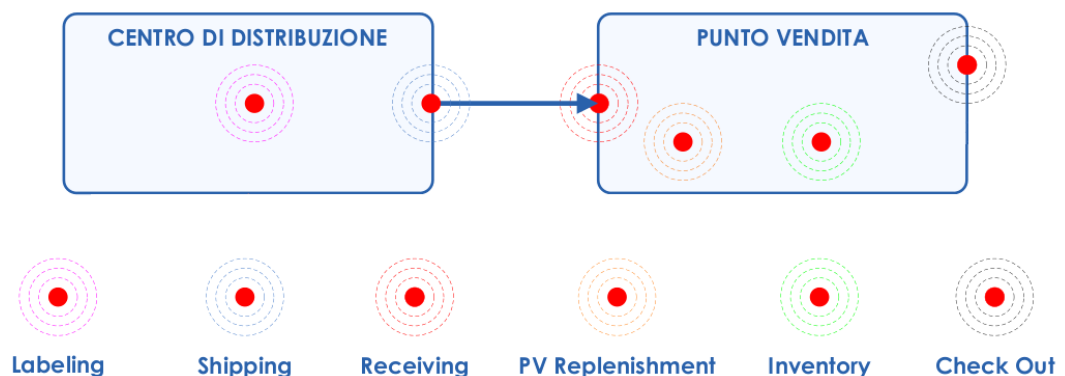


Figura 4 - Diagramma della supply chain

I processi monitorati possono essere così distinti:

- Processi presso il centro di distribuzione:
 - Ricondizionamento (Slap&Ship)
 - Spedizione (Shipping)
- Processi presso il punto vendita:
 - Ricevimento (Receiving)
 - Riassortimento (Replenishment)
 - Prove di camerino (Fitting)
 - Vendita (Check-out)
 - Inventario (Inventory)

La figura che segue illustra questa schematizzazione.

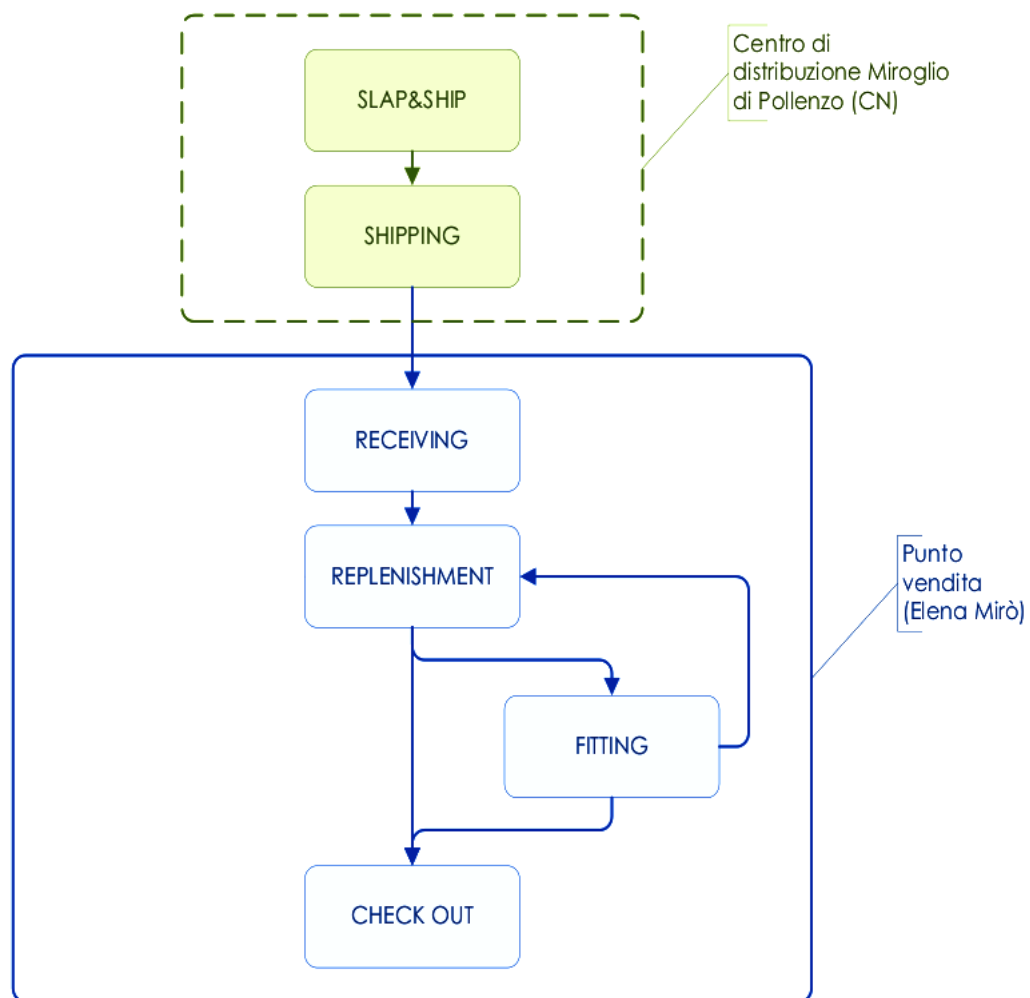


Figura 5 - La supply chain del progetto

I tag RFID, di tipo UHF GEN2, sono stati applicati su tutti e soli i capi destinati al punto vendita prescelto, tramite un processo di etichettatura svolto presso il CeDi di Pollenzo. Anche il processo di spedizione dal CeDi al punto vendita viene gestito mediante tecnologia RFID, effettuando un controllo puntuale della merce in uscita. All'interno del punto vendita di Fidenza, la prima rilevazione dei capi spediti avviene in fase di ricevimento della merce nel retronegoziio, dove i capi vengono presi in carico nella giacenza di retronegoziio, dopo aver controllato la conformità all'ordine. Quindi, un successivo punto di rilevazione RFID è stato collocato nel passaggio tra l'area di retronegoziio e l'area di vendita, allo scopo di leggere il capo ad ogni passaggio ed assegnarlo all'appropriata area di stock.

Questa rilevazione consente, all'interno del punto vendita, la gestione di due stock separati: uno relativo alla riserva e uno relativo all'area vendita. Un'ultima rilevazione dei capi taggati è quindi effettuata durante la vendita del capo alla cassa del punto vendita. Un capo può inoltre essere identificato mediante i processi di inventory o fitting, corrispondenti rispettivamente all'esecuzione di un inventario fisico del negozio e alla prova del capo in camerino da parte di un cliente.

Tutti gli attori della catena logistica hanno quindi potuto seguire il flusso fisico dei capi fino alla cassa del punto vendita, e di accedere ai dati condivisi ad essi relativi mediante l'Internet degli oggetti. Infatti, i processi logistici sono stati costantemente monitorati tramite una Dashboard web, sviluppata appositamente per il progetto pilota, che permette di effettuare specifiche analisi dei flussi di prodotto.

2.2.3. Architettura software

Il diagramma seguente mostra una panoramica di alto livello dell'architettura software del progetto. In esso si possono distinguere i due attori coinvolti (CeDi e punto vendita), oltre alla sede di RFID Lab, che ospita l'infrastruttura di elaborazione dati e Business Intelligence oggetto specifico di questa trattazione.

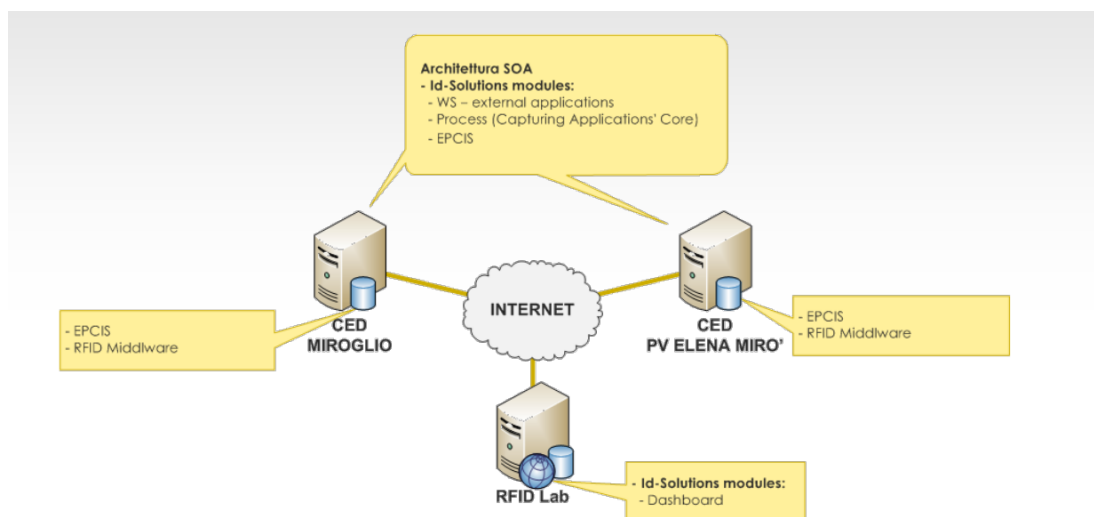


Figura 6 - Architettura software

Nella figura è possibile notare come gli attori comunichino tra loro tramite la rete pubblica Internet. Questa comunicazione è stata resa sicura e confidenziale mediante l'adozione di due accorgimenti principali. Il primo è l'utilizzo, per tutte le connessioni, di protocolli di rete cifrati. In particolare nel caso dello scambio dati EPCIS il protocollo usato è TLS (Transport Layer Security) (RFC 4346), mentre per la manutenzione dei server remoti è stato usato il protocollo SSH (secure Shell) (RFC 4252).

Il protocollo TLS (successore di SSL o Secure Sockets Layer), così come SSH, è stato sviluppato per prevenire l'intercettazione dei messaggi trasmessi, la loro manomissione e la falsificazione delle identità dei partecipanti. Entrambi i protocolli fanno uso di crittografia a chiave asimmetrica (Ferguson, 2003). Il secondo accorgimento consiste nell'impostazione di opportune regole di firewall sui server di CeDi e punto vendita, che filtrano tutto il traffico di rete permettendo la comunicazione solo tra specifici indirizzi IP e porte TCP. Nella fattispecie, i due server di controllo dei processi permettono l'accesso esclusivamente alle porte 8443 (sulla quale è in ascolto il servizio EPCIS con protocollo HTTPS) e 22 (la porta standard dedicata ad SSH secondo l'RFC 4250), ed esclusivamente dagli indirizzi IP pubblici dei due server stessi e del server collocato in RFID Lab.

Sempre in figura è possibile distinguere i moduli logici principali in cui è suddiviso il sistema: RFID Middleware, EPCIS e Dashboard. I primi due risiedono nei server di CeDi e punto vendita: RFID Middleware designa l'insieme dei software di controllo dei processi logistici reingegnerizzati mediante RFID, mentre l'EPCIS è il componente standard dell'architettura EPC Network, discussa sopra. Il modulo designato in figura come Dashboard rappresenta l'insieme dei software di analisi dei risultati e Business Intelligence, compresa la dashboard web che fornisce l'interfaccia utente.

Il diagramma riportato sotto rappresenta l'architettura in maniera più dettagliata:

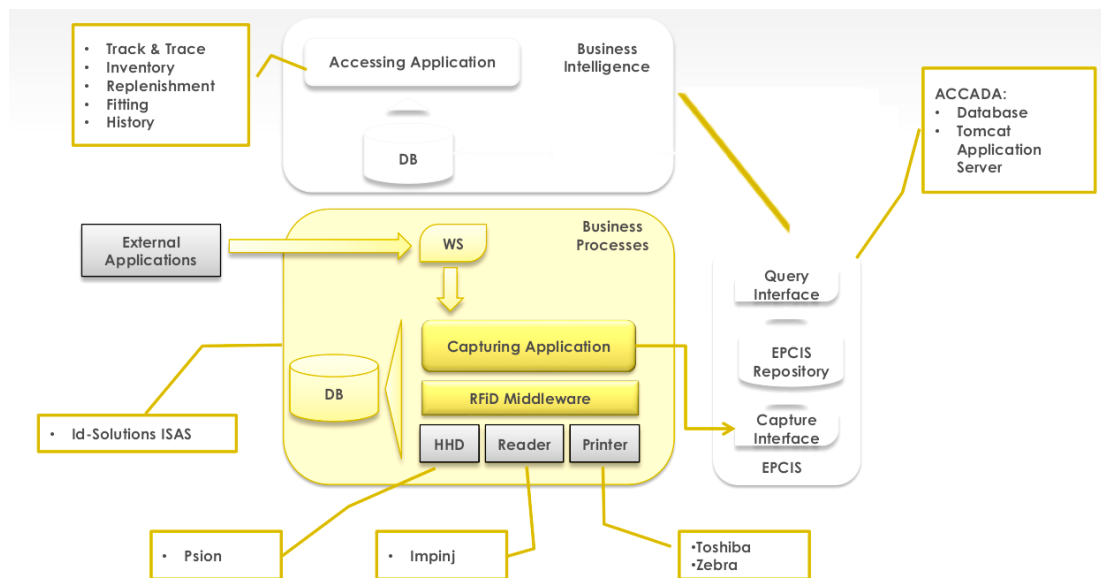


Figura 7 - Architettura software nel dettaglio

A questo livello di dettaglio è possibile distinguere i vari strati logici che compongono un sistema conforme alle specifiche EPC Network: in particolare Auto-ID Infrastructure, Middleware RFID, EPCIS Capturing Application, EPCIS Accessing Application e EPCIS Repository.

Lo strato hardware di acquisizione dei dati RFID, il middleware e le Capturing Application sono stati raggruppati in un modulo logico chiamato Business Processes, in quanto la loro funzione combinata è quella di controllare i processi

logistici e generare i dati RFID. Questi processi di business si occupano infatti di controllare i dispositivi RFID ed altri dispositivi esterni (come computer di cassa o terminali di etichettatura non RFID), fornire un'interfaccia di controllo agli operatori di magazzino e punto vendita, integrarsi con i sistemi informativi pre-esistenti e memorizzare i dati raccolti, sotto forma di eventi EPCIS, in un repository EPCIS. Per assolvere a queste funzioni i moduli sfruttano un database di supporto, mostrato in figura, sul quale è stato realizzato uno schema dei dati opportuno e non aderente ad alcuno standard EPCglobal, in quanto destinato esclusivamente ad uso interno dei processi software. Sempre in figura è possibile individuare come la comunicazione con applicazioni esterne (i sistemi informativi residenti in CeDi e punto vendita) avvenga tramite Web Service. Anche il modulo Capturing Application, che invia gli eventi RFID al repository EPCIS, comunica tramite web service, realizzando così il paradigma SOA (Service Oriented Application).

Le tecnologie utilizzate per la realizzazione del modulo Business Process comprendono un database MySQL e varie applicazioni JavaEE5 ospitate su server Oracle Glassfish, oltre ad applicazioni dotate di interfaccia grafica in ambiente Windows, realizzate in linguaggio Microsoft .NET.

Il modulo Business Processes comunica con un modulo EPCIS realizzato seguendo le specifiche EPCglobal Network: gli eventi in ingresso vengono quindi ricevuti tramite l'interfaccia di Capturing, sono memorizzati in un EPCIS Repository ed esposti alle Accessing Application tramite l'interfaccia di Query. Il sistema EPCIS implementato nel progetto è la piattaforma open source Fosstrak, che si basa su database MySQL e server Apache Tomcat.

L'ultimo livello che costituisce il sistema è quello delle Accessing Application. Questo modulo, designato come Business Intelligence nel diagramma, ha la funzione di elaborare gli eventi EPCIS per ottenere informazioni a valore aggiunto sui processi logistici, che vengono esposte all'utente tramite un'interfaccia web (Dashboard). Si tratta del modulo oggetto specifico di questa trattazione, e verrà quindi analizzato in dettaglio nel prosieguo. Si anticipa qui che le tecnologie utilizzate dal livello di Business Intelligence comprendono varie applicazioni in linguaggio Java, un server Oracle Glassfish e un database

MySQL.

Mentre i primi due moduli, Business Processes ed EPCIS, risiedono sui server di CeDi e punto vendita, in una configurazione replicata e pressoché identica, il sistema Business Intelligence risiede sul terzo server utilizzato nel progetto, presso RFID Lab.

L'infrastruttura di rete costruita per realizzare questa architettura software è rappresentata nella figura seguente:

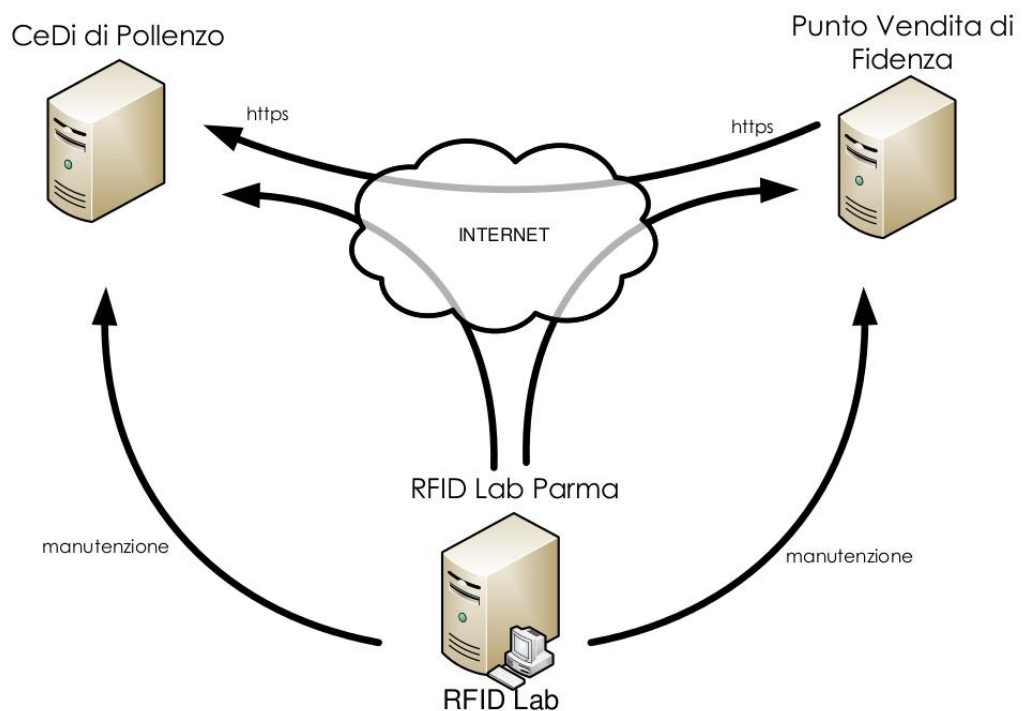


Figura 8 - Architettura di rete

Il diagramma illustra i tre server coinvolti nel progetto e i collegamenti di rete tra di essi. In esso si distinguono i flussi di comunicazione cifrati in protocollo HTTPS e le connessioni di manutenzione in protocollo SSH da RFID Lab ai server remoti, che sono stati esposti in precedenza.

Per quanto riguarda, infine, la configurazione di rete locale dei siti di Pollenzo e Fidenza, di particolare interesse è la struttura realizzata nel punto vendita:

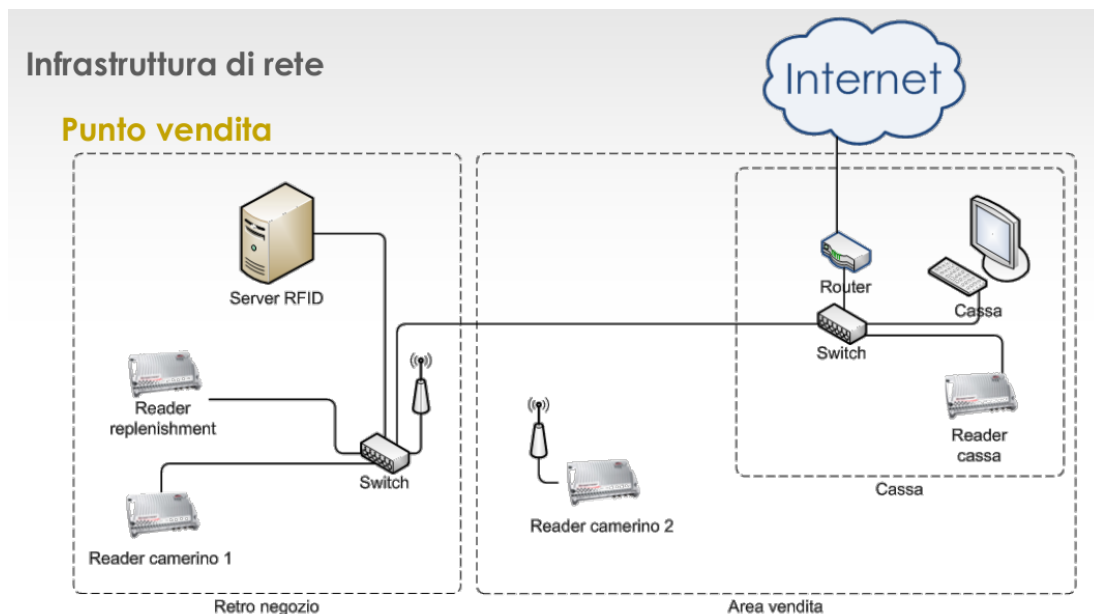


Figura 9 - Architettura di rete nel punto vendita

La figura mostra le aree di retronegozio e area vendita, e i rispettivi dispositivi connessi in rete. Si può notare come il server e due lettori RFID siano collegati tramite cavi Ethernet ad uno switch posto in retronegozio; a questo switch ne è collegato un altro, situato in area vendita, a cui sono a loro volta connessi un terzo lettore RFID e il computer di cassa. La particolare conformazione del negozio ha richiesto il collegamento dell'ultimo lettore RFID, collocato in camerino, mediante connessione wireless. Anche i terminali mobili sono connessi in wireless. La copertura WiFi è fornita da un access point situato in retronegozio.

Tutti i dispositivi fanno parte della medesima sottorete LAN. L'accesso alla rete Internet esterna avviene tramite un modem-router ADSL situato in cassa.

2.2.4. Hardware utilizzato nel progetto

L'infrastruttura software del progetto è ospitata su tre server dedicati:

- Server per gestione dei processi RFID installato presso il CEDI di Pollenzo (CN), denominato RFP1

- Server per gestione dei processi RFID installato presso il PV di Fidenza (PR), denominato RFP2
- Server per le applicazioni di Business Intelligence installato presso RFID Lab a Parma

Tutti i server condividono le medesime specifiche tecniche⁶:

- Modello Fujitsu TX120
- CPU Intel Xeon 3070
- 4GB DDR2 SDRAM
- 2x HDD SAS 146GB, 10000 RPM, su slot hot-swap 2.5", in configurazione RAID 1

Il sistema operativo installato sui server di Pollenzo e Fidenza è Ubuntu 9.10 64bit Server Edition con kernel GNU/Linux 2.6.31. Sul server di RFID Lab è invece installato Microsoft Windows Server 2003 64bit, adattato da un'installazione pre-esistente.

Per quanto riguarda la reingegnerizzazione delle etichette, si è scelto di adottare la stessa tipologia di cartellino preesistente, che è stato però dotato di tag RFID. I tag utilizzati, forniti dall'azienda UPM, sono di tipo passivo UHF GEN2 ed equipaggiati con chip Impinj Monza 3. La scelta di tale soluzione tecnica è stata motivata dalle elevate prestazioni riscontrate nei test.

Il restante hardware, costituito da lettori e stampanti RFID e da terminali di supporto, viene elencato di seguito suddiviso nei vari processi logistici in cui è impiegato:

- Ricondizionamento:
 - Stampante RFID Zebra R2844
- Spedizione:
 - Terminale mobile RFID Skeye Integral2
 - Lettore RFID Impinj Speedway con 4 antenne near field
 - Terminale touchscreen Asus Eee Top

⁶ Reperibili all'indirizzo <http://www.fujitsu.com/downloads/PRMRGY/catalog-tx120.pdf>

- Ricevimento:
 - Terminale mobile RFID Psion Workabout Pro G2
- Ripristino:
 - Lettore RFID Impinj Speedway con 4 antenne near field
 - Terminale touchscreen Asus Eee Top
- Vendita:
 - Lettore RFID Impinj Speedway con 2 antenne near field
- Prove di camerino:
 - 2 Lettori RFID Impinj Speedway con 4 antenne near field (uno per ciascun camerino)
- Inventario:
 - Terminale mobile RFID Psion Workabout Pro G2
 - Lettore RFID Impinj Speedway Revolution con 1 antenna near field, reso trasportabile con l'aggiunta di una batteria per l'alimentazione e di un adattatore di rete wireless
 - Terminale palmare HP

3. Sviluppo software

3.1. Tecnologie impiegate

3.1.1. Fosstrak EPCIS

Il modulo EPCIS utilizzato nel progetto è basato sulla piattaforma Fosstrak (precedentemente conosciuta sotto il nome di Accada)⁷. Fosstrak è una piattaforma con licenza open source LGPL 2.1 che implementa le specifiche EPCglobal Network. Il coordinatore del progetto è C. Roduner di ETH Zurich, ma numerosi contributi provengono sotto forma di *patch* da individui e organizzazioni esterni, secondo i principi della collaborazione open source. Il sistema Fosstrak consiste di tre moduli distinti: Reader, Filtering and Collecting Middleware, ed EPCIS (Floerkemeier, 2007). Il framework EPCIS di Fosstrak ha superato tutti i test di compatibilità con le specifiche EPCIS versione 1.0.1, ed è stato ufficialmente certificato da EPCglobal nel marzo 2008⁸. Nel progetto esaminato in questa dissertazione è stato utilizzato esclusivamente il modulo EPCIS Repository, nella sua versione 0.4.2. Questa particolare implementazione supporta vari protocolli di binding per le interfacce Capture e

⁷ <http://fosstrak.org>

⁸ Come riportato nell'elenco ufficiale delle certificazioni all'indirizzo http://www.epcglobalinc.org/certification/sw_cert/EPCIS_DetailReport_Fosstrak.pdf

Query definite dallo standard:

- XML per il Data Definition Layer
- HTTP/HTTPS per l'interfaccia Capture
- SOAP HTTP/HTTPS per l'interfaccia Query
- HTTP/HTTPS per l'interfaccia Query Callback

Per quanto riguarda l'interfaccia di Capture, l'implementazione di Fosstrak utilizza una Servlet Java residente su server Apache Tomcat 6⁹. Questa servlet riceve gli eventi EPCIS dal livello Capturing Application sotto forma di richieste HTTP POST. Gli eventi sono codificati nel formato XML previsto dalle specifiche EPCglobal.

L'interfaccia di Query risiede anch'essa sul medesimo server Apache Tomcat ma non fa uso di Servlet Java, bensì è esposta come Web Service. Questo web service è creato con il framework Apache CXF¹⁰ e segue le specifiche standard di JavaEE JAX-WS¹¹ e JAXB¹². Le query provenienti dalle Accessing Application e le relative informazioni di risposta sono codificate in messaggi SOAP e veicolate dal protocollo HTTPS.

Oltre alle interfacce esposte verso l'esterno, il funzionamento interno di Fosstrak prevede altri due moduli, che inseriscono ed estraggono gli eventi EPCIS memorizzati nel database, denominati rispettivamente Capture Operations e Query Operations. Entrambi si occupano di codificare o decodificare gli eventi tra il formato XML (usato per la comunicazione con l'esterno) e i record del database interno. Il sistema RDBMS utilizzato è MySQL. L'installazione standard di Fosstrak EPCIS è stata adattata alle esigenze del progetto con alcuni interventi di configurazione sul server Tomcat. Uno in particolare è l'abilitazione del protocollo cifrato HTTPS per entrambe le interfacce di Query e Capture, con la creazione del relativo certificato SSL a chiave pubblica. Il certificato è stato poi esportato in formato RFC 1421, per essere importato negli altri server che devono comunicare con l'EPCIS. Il

⁹ <http://tomcat.apache.org>

¹⁰ <http://cxf.apache.org>

¹¹ <https://jax-ws.dev.java.net>

¹² <https://jaxb.dev.java.net>

certificato a chiave pubblica creato utilizza un algoritmo di cifratura RSA a 1024 bit (Rivest, 1978).

3.1.2. Oracle Glassfish Application Server

Sia i software di controllo dei processi che la dashboard di Business Intelligence presentata qui sono realizzati in forma di applicazioni JavaEE che risiedono su Application Server Glassfish v2. Un Application Server è una particolare tipologia di software che offre un'infrastruttura di supporto per lo sviluppo e l'esecuzione di applicazioni server di classe enterprise, caratterizzate generalmente da numerosi livelli logici. Un application server è generalmente organizzato in moduli autonomi, ciascuno aderente a standard ben definiti. I componenti più comuni sono:

- gestore degli utenti e dell'autenticazione
- contenitore di componenti lato server (servlet, enterprise bean)
- gestore di pool di connessioni a database e transazioni
- server HTTP per contenuti statici
- interfaccia di configurazione e monitoraggio delle prestazioni

Lo standard Sun Java Enterprise Edition¹³ (abbreviato Java EE) copre un insieme di numerose e varie specifiche, tra cui JAXB, JAX-WS, JAXP, StAX, JSP, JSF, JPA, CDI, JMX. Il fondamento di tutte queste tecnologie resta comunque la specifica Java Standard Edition (JavaSE).

Glassfish è l'Application Server di riferimento per l'implementazione delle specifiche Sun Java Enterprise Edition, in quanto viene sviluppato e distribuito direttamente da Sun (recentemente acquisita da Oracle) sotto il nome di Sun Java System Application Server 9. Il marchio Glassfish designa in particolare l'edizione "Community" del Sun Java System AS, distribuita con licenza open source e senza contratto di supporto incluso. La versione v2.1.1 utilizzata nel

¹³ <http://java.sun.com/j2ee>

progetto ha visto il primo rilascio a fine 2009.

Glassfish è stato selezionato per il progetto principalmente grazie a due caratteristiche che lo contraddistinguono rispetto all'offerta di altri prodotti, come Jboss¹⁴ o IBM WebSphere¹⁵. Innanzitutto la sua posizione di implementazione standard di Java EE lo rende la piattaforma privilegiata per lo sviluppo di software aderente agli standard tecnologici. L'aderenza agli standard rende l'applicazione facilmente portabile su altre piattaforme, e permette la più ampia libertà di scelta riguardo ai singoli componenti da utilizzare minimizzando i problemi di compatibilità. Glassfish inoltre offre un'infrastruttura di amministrazione completa, che comprende un'interfaccia utente grafica, un'interfaccia da riga di comando per il controllo remoto e il completo supporto a tecnologie di monitoring automatizzato come SNMP (RFC 3411).

L'istanza di Glassfish installata sul server presso RFID Lab, che ospita la Dashboard per la Business Intelligence, è stata adattata alle esigenze del progetto con alcuni interventi di configurazione ed ottimizzazione delle prestazioni. In particolare è stato aumentato il quantitativo di memoria RAM a disposizione della virtual machine Java, portando il parametro Heap Size a 768 MB. Inoltre è stata ottimizzata la gestione delle risorse statiche da servire tramite HTTP, quali immagini e file Javascript, dato l'uso intensivo che ne fa l'interfaccia web. Si è dunque abilitata la compressione di queste risorse in formato standard GZIP, ed anche una politica di caching che riduca o azzeri la necessità da parte dei client HTTP di richiedere queste risorse al server ad ogni successiva connessione dopo quella iniziale, considerato che le risorse vengono modificate molto raramente.

¹⁴ <http://www.jboss.org>

¹⁵ <http://www.ibm.com/websphere>

3.1.3. Hibernate ORM

Hibernate¹⁶ è una libreria open source per il linguaggio Java che fornisce un servizio di Object-Relational Mapping (ORM) (Subramanian, 1999) , per la mappatura e la rappresentazione di un database relazionale tradizionale in un modello orientato agli oggetti. Lo sviluppo di Hibernate è stato inizialmente guidato da un team internazionale di volontari coordinato da G. King; in seguito il progetto è passato sotto l'egida di Jboss ed è diventato un componente aderente alle specifiche Java EE (Iverson, 2004). La licenza è tuttora di tipo open source, in particolare GNU LGPL. La versione utilizzata nel progetto è la 3.2. La funzione principale del framework è la mappatura tra classi Java e tabelle di un database relazionale, compresa la conversione tra tipi di dato Java e SQL. Hibernate si occupa di generare automaticamente il codice SQL adattandolo alla particolare implementazione del database utilizzato, garantendo anche la portabilità tra database diversi. La libreria risolve i problemi di integrazione tra il modello relazionale e quello orientato agli oggetti noti in letteratura come Object-Relational Impedance Mismatch (Behm, 1997), astruendo il codice di accesso al database con una API (Application Programming Interface) di più alto livello e orientata agli oggetti.

Le classi Java che corrispondono alle tabelle del modello relazionale sono, in conformità alla specifica JPA (java Persistence API) di Java EE, semplici oggetti (Plain Old Java Objects o POJO) eventualmente annotati con annotazioni Java 5. Hibernate implementa al proprio interno numerose strategie per l'ottimizzazione delle prestazioni, soprattutto per quanto riguarda il caricamento di vaste collezioni di record da database, come il Lazy Loading (Fowler, 2003). La configurazione di Hibernate è stata effettuata tramite una combinazione di annotazioni Java 5 e il file di configurazione standard previsto dalle specifiche JPA, denominato persistence.xml. Inoltre è stato sfruttato il componente di Connection Pool offerto dall'ambiente Glassfish, che consente di ottimizzare l'accesso al database gestendo un insieme di connessioni assegnate

¹⁶ <http://www.hibernate.org>

dinamicamente man mano che giungono le richieste di accesso.

3.1.4. MySQL RDBMS

Il database relazionale scelto per implementare il sistema Data Warehouse sul quale si basano le analisi di Business Intelligence e l'interfaccia Dashboard realizzate è MySQL¹⁷. MySQL è propriamente un Relational Database Management System (RDBMS) (Schumacher, 2005), ovvero un sistema server capace di gestire l'accesso concorrente di più utenti ad uno o più database. MySQL è distribuito con licenza open source GNU LGPL, così come gli altri software utilizzati nel presente lavoro. Inizialmente il progetto era sponsorizzato dalla società svedese MySQL AB, in seguito acquisita da Sun/Oracle¹⁸. La scelta di utilizzare MySQL è stata motivata principalmente dalla sua diffusione e popolarità sia presso progetti open source che organizzazioni di alto profilo e su scala globale come Google (Urlocker, 2005) e Facebook (Callaghan, 2009). La diffusione di MySQL garantisce anche un'ampia disponibilità di risorse per la documentazione ed il supporto. Le principali caratteristiche di MySQL sono il supporto a svariati Storage Engine, intercambiabili e selezionabili per ciascuna tabella del database; e il raggruppamento di numerose operazioni di commit provenienti da transazioni differenti. Nel progetto si è scelto in particolare di utilizzare come Storage Engine InnoDB¹⁹, che fornisce supporto alle transazioni di tipo ACID (Atomicity, Consistency, Isolation, Durability). L'acronimo ACID compare in letteratura in . Reuter e T. Hearder (1983), che definisce una serie di regole formali, implementabili da un sistema database, atte a garantire un insieme di proprietà che nel complesso garantiscono l'integrità dei dati sia in fase di memorizzazione che di restituzione all'utente. Queste quattro proprietà sono appunto definite

¹⁷ <http://www.mysql.com/products/database>

¹⁸ Press release: "Sun Microsystems Announces Completion of MySQL Acquisition; Paves Way for Secure, Open Source Platform to Power the Network Economy", Sun Microsystems Press release, February 26, 2008

¹⁹ <http://www.innodb.com>

Atomicità, Consistenza, Isolamento e Durevolezza.

Il sistema MySQL RDBMS è stato installato sul medesimo server in RFID Lab che ospita Glassfish e i restanti moduli di Business Intelligence: le risorse hardware del server sono state valutate sufficienti a sostenere il carico di lavoro complessivo, e questa stima è stata confermata durante la campagna sperimentale. Per garantire la scalabilità in caso di carichi di lavoro maggiori, rimane comunque possibile spostare MySQL su una macchina dedicata, collegata in rete locale ad alta velocità, con variazioni minime alla configurazione del sistema.

Le prestazioni del DBRMS, ed in particolare del sottosistema InnoDB, sono state regolate tramite alcuni parametri, di cui si ritiene interessante riportare un estratto.

```
max_heap_table_size = 64M
```

Memoria massima allocata per una singola tabella. Limitata per evitare un consumo di risorse eccessivo da parte di un'unica tabella.

```
sort_buffer_size = 8M
```

Memoria dedicata al buffer per le operazioni di ordinamento dei record (tramite i comandi ORDER BY e GROUP BY). Il valore impostato è sufficiente ad evitare l'utilizzo del disco fisso per queste operazioni, anche in caso di un elevato numero di record.

```
thread_cache_size = 8
```

Numero di thread da mantenere in memoria per il riutilizzo anche dopo la disconnessione del client. Il valore impostato permette di ridurre al minimo la necessità di creare nuovi thread, che è un'operazione costosa. I requisiti dell'applicazione comunque non richiedono un valore particolarmente alto, in quanto il numero di client connessi contemporaneamente è generalmente basso.

```
thread_concurrency = 8
```

Questo parametro indica il numero di thread da mantenere in esecuzione contemporaneamente. Il valore consigliato è 4 volte il numero di CPU, che nel server utilizzato è pari a 2.

```
query_cache_size = 96M
```

Memoria allocata per il caching dei risultati delle query. Questo parametro risulta estremamente importante per ottimizzare l'applicazione, in quanto la varietà delle

query eseguite è limitata, e i dati delle tabelle cambiano raramente (l'intervallo minimo è di un'ora, corrispondente al tempo di aggiornamento periodico della data warehouse). Queste due condizioni consentono lo sfruttamento ideale della memoria cache, che quindi è stata aumentata rispetto alle impostazioni di default.

```
query_cache_limit = 64M
```

Memoria massima utilizzabile da un singolo risultato in cache. Questo limite è imposto per impedire che un'unica query con un risultato di grandi dimensioni occupi interamente la memoria cache, impedendone l'utilizzo per le altre query.

```
innodb_buffer_pool_size = 2G
```

Memoria allocata al buffer di InnoDB per la gestione di indici e dati.

L'applicazione utilizza il formato InnoDB per la memorizzazione dei dati, e questo parametro è tra quelli che influenzano maggiormente le prestazioni. Il valore è stato impostato al 50% della memoria RAM disponibile nel sistema, ovvero il massimo valore che non pregiudica il funzionamento dell'application server Glassfish, che è ospitato sullo stesso hardware di MySQL.

```
innodb_file_io_threads = 6
```

Numero di thread utilizzati dal sistema per le operazioni di input/output. Il server Windows beneficia di un valore superiore a 4, che è il default per i sistemi Unix.

Listato 1 - Alcuni parametri di configurazione di MySQL

3.1.5. Google Web Toolkit

L'interfaccia di Dashboard, utilizzata dall'utente per l'accesso ai dati generati durante il progetto pilota e la presentazione delle informazioni di Business Intelligence, è stata realizzata sotto forma di applicazione web, ovvero un'applicazione accessibile da browser interamente basata sulle tecnologie web come HTML, Javascript ed il protocollo di comunicazione HTTP (o la sua variante cifrata HTTPS). L'applicazione è ospitata sul server Oracle Glassfish introdotto in precedenza, ed è dunque basata sulle specifiche Java EE. Esistono numerosissimi framework per Java EE a supporto della realizzazione di applicazioni web (Johnson, 2005): nel progetto in esame è stato impiegato

Google Web Toolkit (GWT) (Kutschera, 2009), scelto per le peculiarità che saranno esposte nel seguito.

GWT è uno strumento di sviluppo per la costruzione e l'ottimizzazione di applicazioni web, particolarmente indicato per applicazioni complesse e ricche di funzionalità. Il suo obiettivo principale è migliorare la produttività dello sviluppatore astruendo le varie implementazioni delle tecnologie web nei browser esistenti e permettendo l'utilizzo di un unico linguaggio di programmazione (in questo caso Java) per la costruzione dell'intera applicazione. GWT è sviluppato su iniziativa di Google e distribuito con licenza open source LGPL. Viene impiegato attualmente in numerosi progetti anche su vasta scala, compresi quelli della stessa Google²⁰. La versione utilizzata nel progetto è la 2.0.4.

Il Google Web Toolkit offre una serie di strumenti che lo sviluppatore può scegliere di utilizzare nella sua interezza, o soltanto in parte. Il principale è una libreria di programmazione che consente di scrivere software in linguaggio Java, che viene poi compilato in codice Javascript estremamente ottimizzato e compatibile con la maggior parte dei browser esistenti. Un altro vantaggio di questa maniera di produrre applicazioni Javascript è la possibilità di utilizzare strumenti di debug, refactoring ed IDE (integrated Development Environment) avanzati e maturi pensati per l'ambiente Java. Tra le ottimizzazioni effettuate automaticamente durante il processo di compilazione del codice Java in Javascript si ricordano l'in-lining dei metodi, la rimozione del codice inutilizzato, la suddivisione del codice su più file e la loro compressione per velocizzare il caricamento dell'applicazione. La libreria fornisce inoltre un supporto completo alla localizzazione delle applicazioni in più lingue, alla comunicazione tra lato client e server in vari formati, al logging e allo unit-testing.

Gli altri strumenti offerti da GWT comprendono uno strumento per l'analisi delle prestazioni dell'applicazione e un'applicazione grafica per la costruzione delle interfacce utente.

²⁰ Come affermato in <http://code.google.com/intl/it/webtoolkit/overview.html>

3.1.6. Isomorphic SmartGWT

SmartGWT è un framework basato su Google Web Toolkit, che fornisce un'architettura completa per lo sviluppo di applicazioni web, estendendo la libreria base di GWT con numerosi componenti per l'interfaccia utente. SmartGWT è sviluppato da Isomorphic Software e distribuito in due edizioni: una gratuita, su licenza open source LGPL, e l'altra commerciale, che offre un contratto di supporto e garanzie legali aggiuntive. Nel progetto si è optato per la versione open source, data la natura di progetto pilota. Il principale vantaggio derivante dall'utilizzo di questo framework è l'immediata disponibilità di un gran numero di componenti UI (User Interface), come caselle di testo, tabelle, finestre, che coprono una gamma estremamente ampia di casi d'uso. Questi componenti sono progettati per accelerare i tempi di risposta e minimizzare il carico di lavoro del server, migliorando nel complesso sia l'esperienza utente che le prestazioni del sistema. Sebbene l'integrazione dei componenti di SmartGWT si estenda anche al lato server, fornendo ad esempio la gestione delle transazioni database, anche utilizzandone soltanto una parte il framework si rivela comunque efficace e produttivo. L'approccio di SmartGWT è interamente orientato agli oggetti e mantiene, così come la libreria GWT su cui si basa, la compatibilità con la maggior parte dei browser e la robustezza ai bug di questi ultimi. Questo approccio facilita inoltre l'estensione dei componenti esistenti con funzionalità personalizzate. Anche l'aspetto grafico dei componenti è completamente personalizzabile, con la creazione di "temi" applicabili all'intera applicazione senza la necessità di modificare il funzionamento interno dei componenti.

Si ritiene interessante mostrare in figura l'architettura del sistema SmartGWT, in quanto anche l'architettura della Dashboard, realizzata avvalendosi di questo framework, ricalca questa struttura generale:

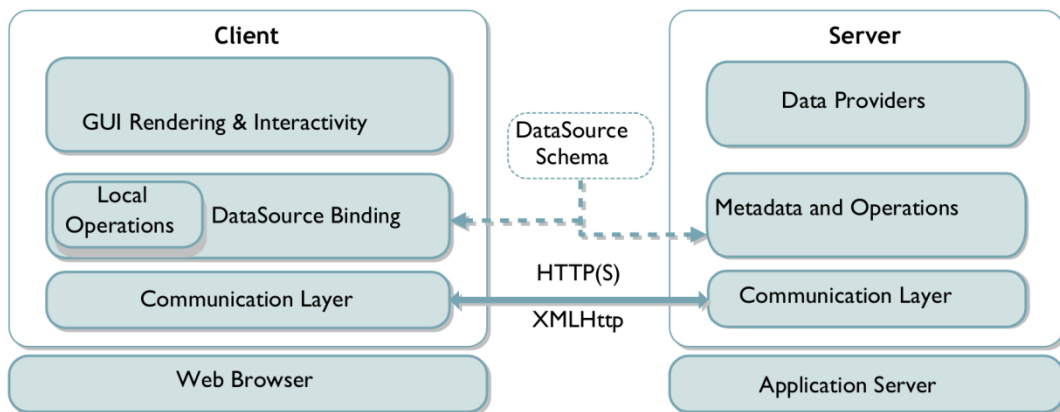


Figura 10 - Architettura di SmartGWT

3.2. Sviluppo software

3.2.1. Architettura del sistema

Lo sviluppo software del framework oggetto di questa dissertazione, grazie al quale è possibile elaborare e visualizzare informazioni ad elevato valore aggiunto per la supply chain fashion, si è concentrato prevalentemente sul livello di Accessing Application dell'architettura EPCglobal Network. Tale applicazione presenta all'utente un'interfaccia web composta da svariati moduli, ciascuno dei quali fornisce una particolare vista della supply chain e permette l'interrogazione dei dati secondo numerosi parametri di filtraggio.

I moduli o sezioni in cui è suddivisa questa interfaccia, detta Dashboard, sono:

- Track&Trace, per visualizzare le date in cui un dato capo ha subito i vari processi logistici;
- Inventory, per mostrare informazioni di giacenza all'interno dei vari attori;

- Fitting, per mostrare informazioni sulle prove in camerino;
- Replenishment, per visualizzare segnalazioni sui capi da ripristinare o riordinare;
- History, per visualizzare la successione temporale di tutti i processi logistici subiti da un dato capo.

Nel complesso, dunque, le informazioni disponibili sulla Dashboard permettono di compiere un'analisi delle prestazioni e delle potenzialità del sistema RFID impiegato nel progetto, facendo dell'applicazione un importante strumento di valutazione a disposizione dei membri del Board of Advisors Fashion (presentato nel capitolo 2). Questi moduli presentano la stessa struttura di impaginazione grafica, con una sezione laterale per impostare i parametri di interrogazione e una schermata occupante lo spazio rimanente per visualizzare i dati.

I dati mostrati dalla Dashboard sono estratti da un sistema di data warehouse (i cui principi generali sono stati esposti nel capitolo 1) nel quale confluiscono i dati aggregati di tutti i punti nodali della catena logistica. Questi dati sono ottenuti, secondo il principio delle Accessing Application, interrogando i sistemi EPCIS remoti degli attori che partecipano alla supply chain, in questo caso il CeDi di Pollenzo e il punto vendita di Fidenza. Queste query EPCIS vengono effettuate periodicamente, a intervalli di tempo configurabili, e gli eventi EPCIS così ottenuti sono elaborati da opportune procedure prima di essere inseriti nel sistema data warehouse. L'interfaccia web quindi non si basa su dati aggiornati in tempo reale, poichè l'aggiornamento della data warehouse avviene periodicamente. L'intervallo di tempo utilizzato per la maggior parte del progetto è stato fissato a un'ora. Questa funzione di aggiornamento periodico della data warehouse è svolta da un componente software indipendente e distinto dall'applicazione web.

Nelle prossime sezioni saranno presentati lo schema di database impiegato per la data warehouse, il componente di aggiornamento della stessa, e la Dashboard web.

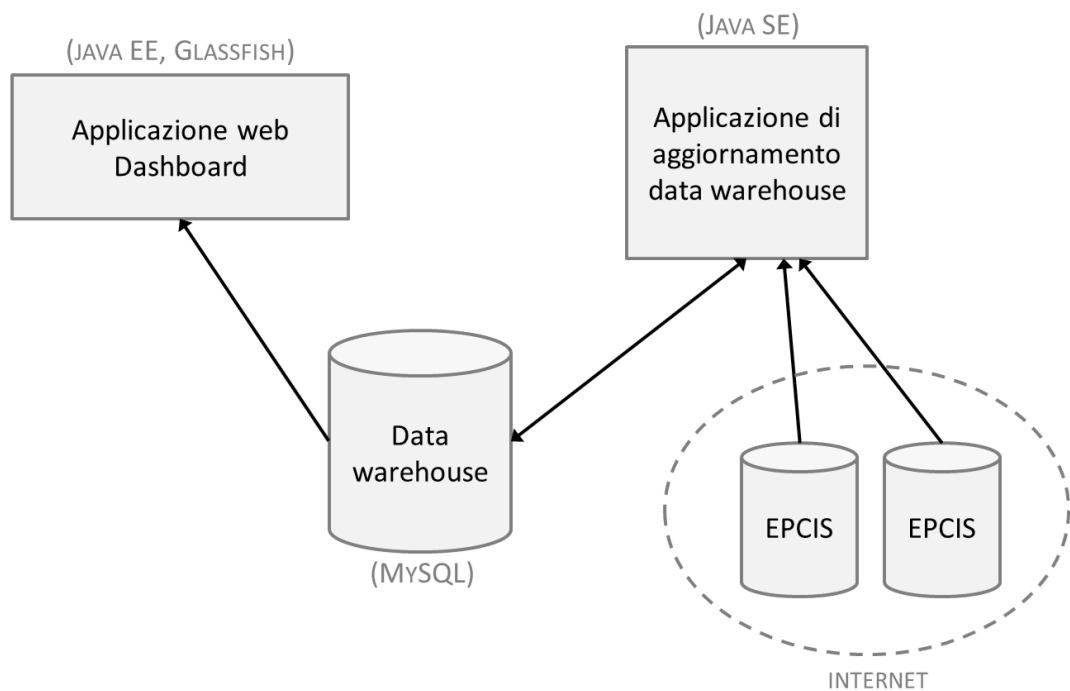


Figura 11 - Diagramma dell'architettura software

3.2.2. Data warehouse

Il sistema di data warehouse è basato sul MySQL RDBMS le cui caratteristiche e la cui configurazione sono state illustrate in precedenza. Lo schema di questo database prevede diverse tabelle la cui struttura, in ottemperanza ai principi del datawarehousing, non è normalizzata secondo le regole del modello relazionale (Codd, 1990). Le varie tabelle infatti presentano numerosi campi ridondanti: ad esempio i dati anagrafici relativi ad un capo, come codice EAN, taglia, modello, colore sono riportati integralmente in diverse tabelle. Questo permette da un lato la storicizzazione delle informazioni, riportando eventuali variazioni nel tempo dei valori di questi campi, e dall'altro la semplificazione delle query e l'ottimizzazione dei tempi di risposta.

Di seguito sono riportate, a titolo esemplificativo, le definizioni della struttura di alcune tabelle dedicate alle anagrafiche dei capi e ai dati provenienti dai processi logistici.

```

CREATE TABLE `dw_epcis_ean13` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `ean13` varchar(16) NOT NULL,
  `epc` varchar(64) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `index_epc` (`epc`),
  KEY `index_ean13` (`ean13`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 2 - Tabella dw_epcis_ean13

```

CREATE TABLE `dw_epcis_ext` (
  `ean13` varchar(20) NOT NULL,
  `season` varchar(50) DEFAULT NULL,
  `model` varchar(50) DEFAULT NULL,
  `color` varchar(50) DEFAULT NULL,
  `client` varchar(50) DEFAULT NULL,
  `serial` varchar(50) DEFAULT NULL,
  `model_description` varchar(50) DEFAULT NULL,
  `measure_description` varchar(50) DEFAULT NULL,
  `measure` varchar(50) DEFAULT NULL,
  `upper_price_description` varchar(50) DEFAULT NULL,
  `upper_price_currency` varchar(50) DEFAULT NULL,
  `upper_price` varchar(50) DEFAULT NULL,
  `upper_price_flag` varchar(50) DEFAULT NULL,
  `lower_price_description` varchar(50) DEFAULT NULL,
  `lower_price_currency` varchar(50) DEFAULT NULL,
  `lower_price` varchar(50) DEFAULT NULL,
  `lower_price_flag` varchar(50) DEFAULT NULL,
  `discount` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ean13`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 3 - Tabella dw_epcis_ext

```

CREATE TABLE `dw_epcis_color` (
  `color` varchar(50) NOT NULL,
  PRIMARY KEY (`color`)
)

```

```
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Listato 4 - Tabella dw_epcis_color

```
CREATE TABLE `dw_epcis_receiving` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `id_player` varchar(64) DEFAULT NULL,  
  `event` varchar(64) DEFAULT NULL,  
  `epc` varchar(64) DEFAULT NULL,  
  `event_time` datetime DEFAULT NULL,  
  `event_timezone` int(11) DEFAULT NULL,  
  `parent_epc` varchar(64) DEFAULT NULL,  
  `quantity` int(11) DEFAULT NULL,  
  `event_action` varchar(64) DEFAULT NULL,  
  `disposition` varchar(64) DEFAULT NULL,  
  `read_point` varchar(64) DEFAULT NULL,  
  `biz_location` varchar(64) DEFAULT NULL,  
  `biz_transaction` varchar(64) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `index_epc` (`epc`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Listato 5 - Tabella dw_epcis_receiving

```
CREATE TABLE `dw_epcis_last_process` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `id_player` varchar(64) DEFAULT NULL,  
  `event` varchar(64) DEFAULT NULL,  
  `epc` varchar(64) DEFAULT NULL,  
  `event_time` datetime DEFAULT NULL,  
  `event_timezone` int(11) DEFAULT NULL,  
  `parent_epc` varchar(64) DEFAULT NULL,  
  `quantity` int(11) DEFAULT NULL,  
  `event_action` varchar(64) DEFAULT NULL,  
  `disposition` varchar(64) DEFAULT NULL,  
  `read_point` varchar(64) DEFAULT NULL,  
  `biz_location` varchar(64) DEFAULT NULL,  
  `biz_transaction` varchar(64) DEFAULT NULL,  
  `biz_step` varchar(64) DEFAULT NULL,
```

```

`ean13` varchar(16) DEFAULT NULL,
`model` varchar(50) DEFAULT NULL,
`measure` varchar(50) DEFAULT NULL,
`color` varchar(50) DEFAULT NULL,
`description` varchar(50) DEFAULT NULL,
`season` varchar(8) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `index_epc` (`epc`),
KEY `index_time` (`event_time`),
KEY `index_bizloc` (`biz_location`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 6 - Tabella dw_epcis_last_process

Sempre al fine di migliorare le prestazioni in fase di interrogazione del database, sono state realizzate tabelle specifiche per ciascun modulo della Dashboard, con una struttura che ricalca precisamente quella delle informazioni visualizzate nell'interfaccia. Questo permette di minimizzare il numero di query necessarie per comporre i dati da mostrare nell'interfaccia, che in molti casi si riducono ad una soltanto. Si riportano qui le tabelle corrispondenti ai moduli di Track&Trace, Inventory e Replenishment.

```

CREATE TABLE `dw_view_tracktrace` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `epc` varchar(64) NOT NULL,
  `ean13` varchar(16) DEFAULT NULL,
  `slapship` datetime DEFAULT NULL,
  `slapship_count` int(11) DEFAULT NULL,
  `shipping` datetime DEFAULT NULL,
  `shipping_count` int(11) DEFAULT NULL,
  `receiving` datetime DEFAULT NULL,
  `receiving_count` int(11) DEFAULT NULL,
  `replenishment` datetime DEFAULT NULL,
  `replenishment_count` int(11) DEFAULT NULL,
  `fitting` datetime DEFAULT NULL,
  `fitting_count` int(11) DEFAULT NULL,
  `trash` datetime DEFAULT NULL,
  `trash_count` int(11) DEFAULT NULL,

```

```

`model` varchar(50) DEFAULT NULL,
`measure` varchar(50) DEFAULT NULL,
`color` varchar(50) DEFAULT NULL,
`description` varchar(100) DEFAULT NULL,
`checkout` datetime DEFAULT NULL,
`checkout_count` int(11) DEFAULT NULL,
`inventory` datetime DEFAULT NULL,
`oblivion` datetime DEFAULT NULL,
`inventory_backstore` datetime DEFAULT NULL,
`replenishment_backward` datetime DEFAULT NULL,
`replenishment_forward` datetime DEFAULT NULL,
`season` varchar(8) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `index_epc` (`epc`),
KEY `index_ean13` (`ean13`),
KEY `index_model` (`model`),
KEY `index_measure` (`measure`),
KEY `index_color` (`color`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 7 - Tabella dw_view_tracktrace

```

CREATE TABLE `dw_view_inventory` (
  `time_stamp` datetime NOT NULL,
  `warehouse` int(11) DEFAULT '0',
  `backstore` int(11) DEFAULT '0',
  `store` int(11) DEFAULT '0',
  `sold` int(11) DEFAULT '0',
  `oblivion` int(11) DEFAULT '0'
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 8 - Tabella dw_view_inventory

```

CREATE TABLE `dw_view_inventory_total` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `ean13` varchar(16) NOT NULL,
  `warehouse` int(11) NOT NULL DEFAULT '0',
  `backstore` int(11) NOT NULL DEFAULT '0',
  `store` int(11) NOT NULL DEFAULT '0',

```

```

`sold` int(11) NOT NULL DEFAULT '0',
`time_stamp` datetime NOT NULL,
`model` varchar(64) DEFAULT NULL,
`measure` varchar(64) DEFAULT NULL,
`color` varchar(64) DEFAULT NULL,
`description` varchar(64) DEFAULT NULL,
`oblivion` int(11) NOT NULL DEFAULT '0',
`season` varchar(8) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `ean_index` (`ean13`),
KEY `time_index` (`time_stamp`),
KEY `model_index` (`model`),
KEY `measure_index` (`measure`),
KEY `color_index` (`color`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 9 - Tabella dw_view_inventory_total

```

CREATE TABLE `dw_view_replenishment` (
  `ean13` varchar(16) NOT NULL,
  `warehouse` int(11) DEFAULT NULL,
  `backstore` int(11) DEFAULT NULL,
  `store` int(11) DEFAULT NULL,
  `sold1` int(11) DEFAULT NULL,
  `sold2` int(11) DEFAULT NULL,
  `sold3` int(11) DEFAULT NULL,
  `sold5` int(11) DEFAULT NULL,
  `sold7` int(11) DEFAULT NULL,
  `sold15` int(11) DEFAULT NULL,
  `sold30` int(11) DEFAULT NULL,
  `model` varchar(50) DEFAULT NULL,
  `measure` varchar(50) DEFAULT NULL,
  `color` varchar(50) DEFAULT NULL,
  `description` varchar(50) DEFAULT NULL,
  `type1` int(11) DEFAULT NULL,
  `type2` int(11) DEFAULT NULL,
  `type3` int(11) DEFAULT NULL,
  `type5` int(11) DEFAULT NULL,
  `type7` int(11) DEFAULT NULL,
  `type15` int(11) DEFAULT NULL,
  `type30` int(11) DEFAULT NULL,

```

```

`season` varchar(8) DEFAULT NULL,
PRIMARY KEY (`ean13`),
KEY `ean13_index` (`ean13`),
KEY `model_index` (`model`),
KEY `measure_index` (`measure`),
KEY `color_index` (`color`),
KEY `type_index` (`type1`),
KEY `type2_index` (`type2`),
KEY `type3_index` (`type3`),
KEY `type5_index` (`type5`),
KEY `type7_index` (`type7`),
KEY `type15_index` (`type15`),
KEY `type30_index` (`type30`)
)
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Listato 10 - Tabella dw_view_replenishment

Questa strategia di ottimizzazione segue i principi delle data warehouse, e pur apportando benefici considerevoli nei tempi di risposta alle interrogazioni, comporta però un incremento della complessità di elaborazione dei dati in fase di inserimento nel database. La sezione seguente illustra il componente che si occupa dell'inserimento dei dati nella data warehouse.

3.2.3. Aggiornamento data warehouse

L'applicazione di aggiornamento dei dati è stata denominata DatawarehouseETL (Extract, Transform and Load), secondo la terminologia classica del data warehousing. La funzione di questo modulo software consiste nell'interrogazione periodica dei due EPCIS coinvolti nel progetto, nell'elaborazione degli eventi EPCIS così ottenuti e nell'inserimento dei dati trasformati nel sistema data warehouse descritto in precedenza. Il software è realizzato sotto forma di applicazione in linguaggio Java ed è completamente indipendente dall'applicazione web Dashboard: non è infatti ospitato all'interno dell'Application Server, ma viene eseguito direttamente dal sistema operativo.

Questo approccio è stato scelto dopo aver valutato anche un prototipo alternativo di applicazione Java EE gestita dall'Application Server. La decisione di orientarsi su un'implementazione svincolata dall'Application Server è stata presa considerando i vantaggi di stabilità che ne derivano. In primo luogo, l'aggiornamento deve essere richiamato periodicamente, e il framework offerto dal sistema operativo Windows Server per l'esecuzione periodica, denominato "Scheduled Tasks", si è rivelato pienamente affidabile e configurabile secondo le esigenze del progetto. Il framework inoltre gestisce particolari situazioni critiche, come quella che si verifica quando l'applicazione si trova ancora in esecuzione al momento in cui dovrebbe scattare la successiva esecuzione periodica: non è quindi necessario implementare direttamente nel codice la gestione di queste criticità, come sarebbe invece indispensabile utilizzando le infrastrutture EJB Timer offerte dall'Application Server Glassfish (Cecchet, 2002). In secondo luogo, la possibilità di svincolare l'applicazione dall'Application Server, e quindi dalla Dashboard, permette di proteggere ciascun componente dall'eventuale malfunzionamento dell'altro, il che si rivela utile nelle fasi di sviluppo e beta-testing, quando il software presenta ancora instabilità.

L'applicazione viene configurata tramite una serie di parametri memorizzati in opportune tabelle sul database. I parametri, aggiornabili durante l'esecuzione del sistema, comprendono l'intervallo di aggiornamento periodico, il numero massimo di eventi EPCIS recuperabili in una singola esecuzione (limitabile per ragioni di prestazioni), e alcuni flag per abilitare o disabilitare specifiche funzionalità. Anche gli indirizzi delle interfacce di Query EPCIS da interrogare sono configurabili in questo modo: il modulo è dunque adattabile dinamicamente nel caso di cambiamenti nella supply chain, ed inoltre è in grado di supportare una qualunque implementazione di EPCIS aderente agli standard previsti da EPCglobal per le Query Interface.

L'applicazione non offre un'interfaccia utente grafica, ma viene eseguita da riga di comando tramite uno script batch. È comunque possibile controllarne il funzionamento tramite un sistema di logging che tiene traccia di tutti gli eventi in due formati: in formato testuale su file e come record inseriti in particolari

tabelle del database. Le due tabelle in questione, `dw_log` e `dw_log_query`, tracciano data e ora di esecuzione degli aggiornamenti, la loro durata, l'utente che ha avviato l'esecuzione, le query eseguite e il loro esito, e il numero di eventi EPCIS recuperati.

```
CREATE TABLE `dw_log` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `user_id` varchar(64) DEFAULT NULL,  
  `time_start` datetime DEFAULT NULL,  
  `time_end` datetime DEFAULT NULL,  
  `timezone` int(11) DEFAULT NULL,  
  `query_num` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Listato 11 - Tabella dw_log

```
CREATE TABLE `dw_log_query` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `request_id` bigint(20) DEFAULT NULL,  
  `query_name` varchar(64) DEFAULT NULL,  
  `epcis` varchar(1024) DEFAULT NULL,  
  `time_start` datetime DEFAULT NULL,  
  `time_end` datetime DEFAULT NULL,  
  `time_sync` datetime DEFAULT NULL,  
  `timezone` int(11) DEFAULT NULL,  
  `status` varchar(64) DEFAULT NULL,  
  `result_num` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Listato 12 - Tabella dw_log_query

Il funzionamento dell'applicazione può essere illustrato dal seguente diagramma di flusso:

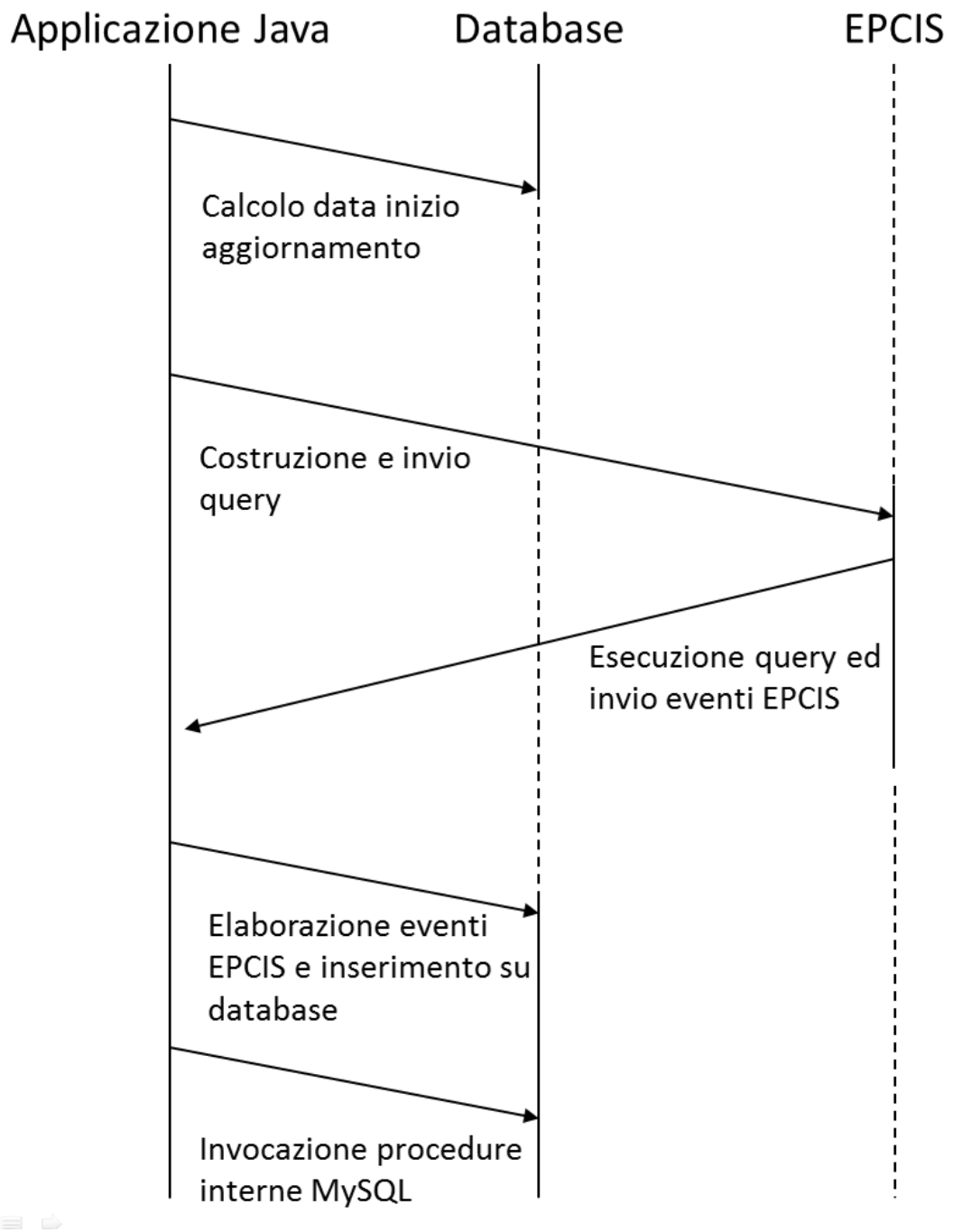


Figura 12 - Diagramma di funzionamento dell'aggiornamento automatico

Il primo passo consiste nel caricamento della configurazione da database e nella costruzione della Query EPCIS. La Query viene inviata sfruttando il framework Fosstrak illustrato in precedenza, in particolare il modulo di Query Interface, e segue quindi le specifiche standard. L'aggiornamento dei dati avviene in

maniera incrementale: questo significa che ad ogni esecuzione il sistema recupera ed elabora dall'EPCIS soltanto i nuovi eventi, ovverosia quelli con data e ora successive alla data dell'evento più recente già elaborato. Questo elimina la necessità di reimportare ad ogni esecuzione l'intero contenuto dell'EPCIS, il che porterebbe naturalmente ad un notevole consumo di risorse. La query EPCIS, di tipo SimpleEventQuery, è stata dunque configurata con i seguenti parametri standard EPCglobal (EPCglobal specification, 2007):

- GE_eventTime: gli eventi richiesti devono avere data successiva o uguale a quella specificata qui.
- orderBy: indica il campo in base al quale ordinare gli eventi richiesti.
- orderDirection: indica la direzione dell'ordinamento (ascendente o discendente).
- eventCountLimit: indica il massimo numero di eventi restituiti dalla query.

Nel caso in cui ci siano più eventi del limite fissato, i restanti verranno acquisiti nella successiva esecuzione dell'aggiornamento.

Una volta che gli eventi sono stati acquisiti da EPCIS, il sistema passa alla loro elaborazione per renderli utilizzabili per l'aggiornamento della data warehouse. Questa elaborazione consiste di diverse fasi, la prima delle quali è l'iterazione sull'intera collezione di eventi EPCIS (sotto forma di oggetti Java). In base alla tipologia di evento vengono generati uno o più record sulle tabelle della data warehouse. Gli eventi vengono inizialmente discriminati in base al campo BizStep, che indica il processo logistico relativo all'evento: a seconda del valore assunto da questo campo viene generato un record nella tabella della data warehouse corrispondente al processo designato. Ad esempio, un evento con un BizStep corrispondente al processo di spedizione viene convertito in un record della tabella dw_epcis_shipping, che contiene i dati relativi al processo di spedizione. Il campo Event viene invece utilizzato per discriminare il tipo di evento: nel caso in cui sia pari a ObjectEvent, indica che l'evento in esame è di tipo "Object" (cfr. Capitolo 1), e il record generato nella data warehouse rappresenta la rilevazione RFID di uno o più capi nella fase di spedizione,

avvenuta ad un certo orario. Se invece l'evento è di tipo TransactionEvent, viene generato un record in un'opportuna tabella che contiene la relazione tra un collo e il pallet che lo contiene. Un altro passo dell'elaborazione consiste nell'aggiornamento dei dati anagrafici dei capi: ad ogni tipologia di capo, identificata da un codice EAN13, corrispondono infatti numerose caratteristiche come colore, taglia, modello, stagione ed altro. Quando il processo di aggiornamento individua un nuovo codice EAN13, non ancora presente nella data warehouse, si occupa anche di popolare il relativo record con tutte le caratteristiche ad esso associate, nella tabella dw_epcis_ext; le caratteristiche sono contenute nell'evento EPCIS sotto forma di estensioni, che utilizzano il meccanismo per l'estendibilità previsto dallo standard EPCIS 1.0.1.

Una volta completata l'iterazione su tutti gli eventi acquisiti, l'applicazione passa all'ultima fase dell'aggiornamento, che consiste nell'invocazione di diverse "stored procedure" su database. Queste procedure, realizzate nel linguaggio nativo di MySQL 5.1, sono definite all'interno dello schema realizzato per il progetto, e si occupano di rigenerare tutte le tabelle che ospitano i dati visualizzati dai vari moduli dell'interfaccia web. Come si è accennato in precedenza, infatti, i dati interrogabili dall'utente tramite Dashboard sono contenuti in apposite tabelle che ne ricalcano la struttura: tali dati sono il risultato di una fase di precalcolo e trasformazione svolta appunto dalle stored procedure invocate al termine dell'aggiornamento. L'invocazione di tutte le procedure avviene tramite un unico comando in linguaggio SQL inviato dall'applicazione Java al database attraverso il framework Hibernate discusso in precedenza. Questo comando esegue un'unica procedura, che al suo interno procede poi all'invocazione in successione di tutte le procedure specializzate necessarie all'aggiornamento. In questo modo è stata realizzata una separazione della logica interna al database dall'applicazione Java, il che elimina la necessità di dover modificare e ricompilare anche il codice Java ogniqualvolta vengono apportate modifiche alle procedure del database. La definizione di questa procedura è riportata nel listato seguente:

```
CREATE PROCEDURE `update_all`()
```

```

BEGIN
call update_extensions();
call create_view_tracktrace();
call update_view_inventory_total(NOW());
call create_view_inventory_total('2010-03-29 00:00:00', NOW(), 1, 'DAY');
call create_view_replenishment();
call create_view_fitting();
call drop_oldest_logs(15);
END;

```

Listato 13 - Definizione della procedura update_all

Poiché i nuovi eventi EPCIS ottenuti con l'aggiornamento possono andare a modificare i dati già esistenti nelle tabelle cui attinge l'interfaccia web, come ad esempio valori medi calcolati su un insieme di record, l'aggiornamento di queste tabelle non può avvenire in maniera incrementale: tali tabelle vengono dunque cancellate e ricreate completamente dalle stored procedure. Anche per questa ragione, il popolamento di queste tabelle si è rivelato in fase di test la parte più dispendiosa, in termini di risorse del sistema, di tutto il processo di aggiornamento. Proprio per ottimizzare per quanto possibile le prestazioni di questa fase, è stato deciso di implementarla sotto forma di stored procedure MySQL, che vengono eseguite internamente al database e riducono quindi il consumo di risorse rispetto all'esecuzione, ad esempio, di codice Java esterno (che comporterebbe un overhead dovuto alla comunicazione remota con il database).

Alla luce di questo considerevole impatto sulle prestazioni, appare ancora più critica l'importanza di eseguire questo aggiornamento ad intervalli periodici, precalcolando in anticipo tutti i dati necessari alla Dashboard, piuttosto che eseguire questi calcoli in tempo reale nel momento in cui vengono richiesti dall'utente, il che comporterebbe tempi di attesa intollerabili (dell'ordine dei minuti o decine di minuti sull'hardware utilizzato nel progetto).

3.2.4. Dashboard

3.2.4.1. Applicazione web

In questa sezione è illustrata l'interfaccia utente offerta dall'Accessing Application oggetto di questa dissertazione. Si tratta di una applicazione web che rientra nella particolare categoria di interfacce visuali definite Dashboard (cfr. Capitolo 1), ovvero cruscotti per la visualizzazione di informazioni ricavate da sistemi di Business Intelligence. Come accennato nella panoramica sull'architettura del sistema, la Dashboard è un'applicazione accessibile da browser che fa uso delle tecnologie standard per il web, come i linguaggi HTML e Javascript e il protocollo HTTP. Il software è stato realizzato prevalentemente in linguaggio Java e si basa sui framework Google Web Toolkit e SmartGWT. È ospitato su server Glassfish e accede al sistema data warehouse con database MySQL. All'accesso tramite browser, l'applicazione presenta una schermata analoga a quella mostrata in figura:

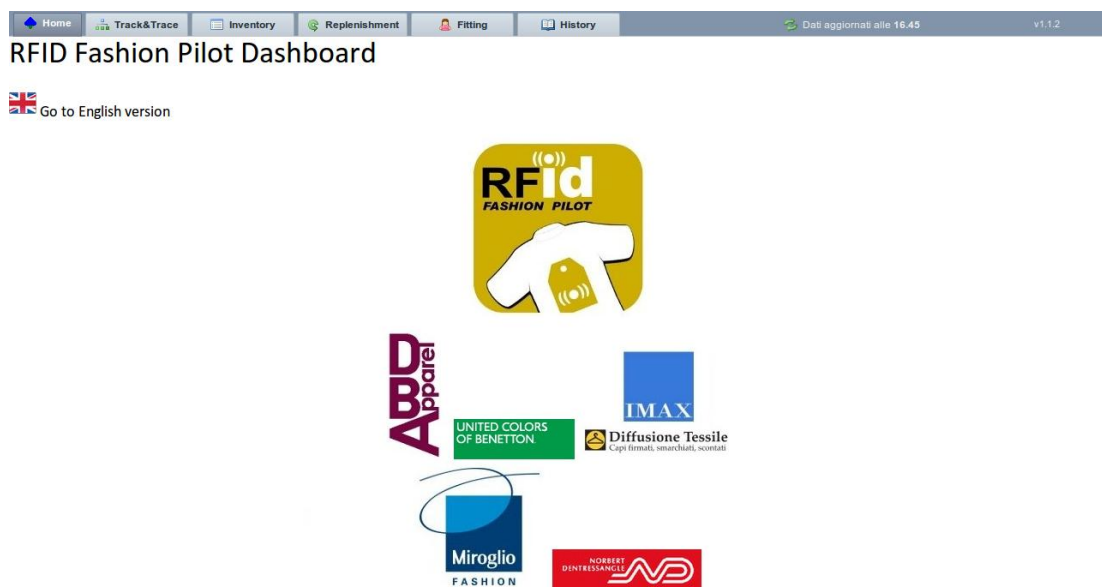


Figura 13 - Schermata iniziale Dashboard

Nella parte superiore compare una serie di sezioni o “tab” corrispondenti ai vari moduli in cui è suddivisa la Dashboard, ciascuno dei quali offre una vista focalizzata su un particolare aspetto di Business Intelligence, come le giacenze inventariali o la cronologia dei processi logistici subiti da ciascun collo. Sempre nella parte superiore, a destra, è presente un indicatore dello stato dell’aggiornamento della data warehouse:

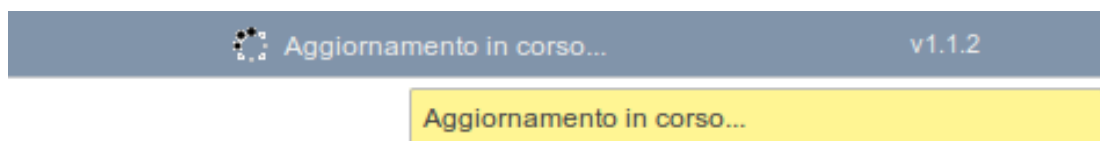


Figura 14 - Particolare della schermata di aggiornamento

Questo messaggio fornisce un’indicazione di quanto sono recenti i dati mostrati dall’applicazione, poiché riporta l’orario dell’ultimo aggiornamento; inoltre segnala quando un aggiornamento è in corso. Infine, nell’angolo a destra, compare il numero di versione dell’applicazione (giunto attualmente a 1.1.2). La schermata iniziale, corrispondente alla tab etichettata come “Home”, riporta i loghi delle aziende aderenti al progetto RFID Fashion Pilot, oltre ad un controllo che permette di cambiare la lingua utilizzata nell’interfaccia. L’interfaccia web infatti è stata completamente localizzata in italiano e inglese: oltre alla lingua, la localizzazione ha interessato anche il formato delle date e dei valori numerici. I moduli di Business Intelligence veri e propri, ai quali si accede dalle restanti tab, condividono come già accennato il medesimo layout grafico. In ottemperanza al paradigma della programmazione orientata agli oggetti, che è stato impiegato in tutta l’architettura software, tutti i moduli di interfaccia sono stati implementati come sottoclassi di una medesima classe comune, tramite il meccanismo dell’ereditarietà (Cox, 1985). Questo ha permesso di includere tutti gli elementi comuni in un’unico oggetto; questi elementi vengono ereditati dagli

oggetti che corrispondono ai vari moduli, i quali si occupano soltanto di implementare il loro contenuto specifico lasciando invariata la struttura comune.

L'oggetto che rappresenta la struttura dei diversi moduli fornisce alcuni elementi comuni: un pannello a sinistra con i parametri impostabili per le query sui dati, e a fianco un pannello che occupa lo spazio restante e mostra le informazioni richieste. Tale pannello presenta nella parte più alta un'area di notifica in cui compaiono eventuali messaggi di errore interno dell'applicazione, o altri suggerimenti per l'utente. Lo spazio sottostante mostra un riassunto dei parametri di query impostati. L'area rimanente è interamente occupata da una tabella che mostra le informazioni richieste. La tabella viene aggiornata alla pressione del pulsante "Query". L'altro pulsante presente permette di esportare i dati estratti dalla query in formato Comma Separated Value (CSV), che può essere facilmente importato in altre applicazioni per effettuare altre analisi o visualizzazioni. Sia la tabella dei risultati che il pannello di query sono implementati in maniera specifica da ciascun modulo.

Essendo la Dashboard, per sua natura, un'applicazione particolarmente orientata alla gestione di grandi moli di dati, è stato possibile sfruttare in maniera estremamente efficace le funzionalità offerte dalla libreria SmartGWT per il trattamento dei dati e la loro visualizzazione.

3.2.4.2. Modulo Track&Trace

Il modulo Track&Trace permette di conoscere tutte le informazioni di tracciabilità raccolte dal sistema RFID, fino al livello di dettaglio del singolo capo. Queste informazioni sono organizzate in una tabella che mostra per ogni riga un capo e, per ciascun processo logistico, la data più recente in cui il capo ha subito quel processo. Qualora la data manchi, significa che quel capo non ha ancora subito quel particolare processo. Questo modulo permette quindi di ricostruire la sequenza temporale con cui i capi si muovono attraverso la catena logistica in esame. Tramite opportuni filtri e ordinamenti per colonna è inoltre

immediato valutare ad esempio quali capi sono stati spediti dal CeDi ma non ancora ricevuti in punto vendita, o quali capi sono stati sia provati che venduti. La schermata di Track&Trace è mostrata in figura:

The screenshot shows the Track&Trace application interface. At the top, there is a navigation bar with tabs for Home, Track&Trace, Inventory, Replenishment, Fitting, and History. The main area displays a table of product tracking data. The table has the following columns: Prodotto, Stagione, Modello, Taglia, Colore, Slap&Shi, Shipping, Receivini, Repl. FW, Repl. BK, Fitting, Inv. Neg., Inv. Retr., Non Trov, Checkou, and Trash. The data rows show various product IDs, seasons (IO), models (e.g., 5577F05, 5484F22), sizes (e.g., 44, 51, 33), colors (e.g., 10, 33), and dates (e.g., 9/11/2011, 10/11/2011). A sidebar on the left contains filters for 'FILTRA PRODOTTI' (EAN13, Mod/Tg/Col, DDT), 'Stagione', 'Modello', 'Taglia', 'Colore', and 'INTERVALLO TEMPORALE' (Da: 0:00, To: 16:54). A 'Query' button and 'Esporta risultati (.CSV)' are also visible.

Figura 15 - Schermata Track&Trace

Il pannello laterale di query presenta alcuni parametri in base ai quali filtrare i risultati, raggruppati in sezioni per facilitarne l'utilizzo all'utente. Una volta impostati i parametri, premendo il pulsante Query vengono visualizzati i risultati nel pannello principale dell'applicazione. Non è indispensabile selezionare alcun parametro: i valori impostati di default non effettuano alcun filtraggio e premendo il tasto Query senza modificarli verranno restituiti tutti i dati nell'intervallo di tempo di default (un giorno). Le sezioni e i parametri sono:

- Filtra prodotto: contiene i parametri di filtro sui prodotti, suddivisi in tre tipologie esclusive. La selezione di una tipologia nasconde tutti i parametri non relativi ad essa, ed impedisce all'utente la loro impostazione.
 - EAN13: permette di selezionare uno o più codici EAN13. Saranno

selezionati tutti i capi associati ad uno qualunque di questi codici (logica OR).

- Taglia/Modello/Colore: offre vari menu che permettono di selezionare uno o più modelli, taglie, colori e/o stagioni. Saranno selezionati tutti i capi con uno qualunque di questi modelli, e una qualunque di queste taglie, e uno qualunque di questi colori, e una qualunque di queste stagioni (logica AND fra i tre campi).
- DDT: permette di selezionare uno o più documenti di trasporto. Saranno selezionati tutti i capi associati ad uno qualunque di questi DDT (logica OR).
- Intervallo temporale: contiene i parametri di filtro su un intervallo tra due date. Questo intervallo di tempo è relativo a uno o più processi logistici selezionabili dal menu presente in questa sezione. Ad esempio, selezionando il processo di Ricevimento e impostando come data iniziale le 06.00 del 20/06/2010 e data finale le 23.59 del 22/06/2010, saranno selezionati tutti i capi che hanno subito il processo di Ricevimento in punto vendita in un qualunque orario compreso nell'intervallo indicato. Data e ora possono essere immesse numericamente dall'utente, oppure selezionate da un calendario grafico che compare in un pop-up cliccando sull'icona apposita.

Il pannello principale mostra i risultati richiesti in seguito alla pressione del pulsante Query. Nella parte alta è presente un riassunto dei parametri di query impostati e il numero totale dei risultati. Al di sotto si trova la tabella che riporta i risultati di tracciabilità. La tabella ha i seguenti campi:

- Prodotto: riporta sia il codice EAN13 che una descrizione per facilitarne l'individuazione; passando con il puntatore del mouse sulla riga appare inoltre il codice EPC completo di seriale che individua quel capo
- Stagione
- Modello
- Taglia

- Colore
- Slap&Ship
- Shipping
- Receiving
- Replenishment Forward
- Replenishment Backward
- Fitting
- Inventario in negozio
- Inventario in retronegozio
- Non trovato
- Checkout
- Trash

Cliccando con il tasto sinistro del mouse su una riga della tabella, l'applicazione mostra il modulo di History con i risultati relativi al codice EPC di quel capo. Questa funzione risulta utile soprattutto in fase di verifica, per approfondire l'esame dei processi subiti da uno specifico capo, senza la necessità di digitarne manualmente il codice EPC nel modulo History.

I dati utilizzati per visualizzare nell'interfaccia questa tabella sono elaborati nel corso dell'aggiornamento periodico della data warehouse, come avviene per tutti gli altri moduli della Dashboard. Nel caso del modulo di Track&Trace i dati provengono dalla tabella `dw_view_tracktrace`. Tutti i parametri eventualmente selezionati dall'utente tramite interfaccia vengono inviati al server, che compone una query opportuna per estrarre i record da questa tabella. I record della tabella `dw_view_tracktrace` vengono creati durante l'aggiornamento periodico della data warehouse descritto nelle sezioni precedenti.

La procedura interna nel linguaggio di MySQL denominata `create_view_tracktrace`, invocata durante l'aggiornamento, è responsabile del popolamento di questa tabella in particolare. Questa procedura estrae dalle tabelle che contengono i dati relativi ai vari processi (come `dw_epcis_shipping`, `dw_epcis_receiving` e altre) la data più recente in cui è stato registrato ogni

processo per ciascun capo presente nel sistema. Tutti questi record vengono composti tramite l'operazione di JOIN tra le varie tabelle sul campo epc, che contiene il codice EPC univoco del capo, e inseriti nella tabella dw_view_tracktrace. In un passo successivo i record della tabella vengono aggiornati con le informazioni anagrafiche relative ai capi, estratte dalle tabelle di anagrafica già illustrate (come dw_epcis_ext e dw_epcis_model). Di seguito si riporta il codice di definizione della procedura:

```
CREATE PROCEDURE `create_view_tracktrace`()
    MODIFIES SQL DATA
BEGIN
SET autocommit=0;
START TRANSACTION;

delete from dw_view_tracktrace;
insert into dw_view_tracktrace
    (epc,
     slapship,
     shipping,
     receiving,
     replenishment,
     replenishment_backward,
     replenishment_forward,
     fitting,
     checkout,
     trash,
     inventory,
     inventory_backstore,
     oblivion)
select
    epc,
    max_slapship.max_event_time,
    max_shipping.max_event_time,
    max_receiving.max_event_time,
    max_replenishment.max_event_time,
    max_replenishment_backward.max_event_time,
    max_replenishment_forward.max_event_time,
    max_fitting.max_event_time,
    max_checkout.max_event_time,
    max_trash.max_event_time,
```

```

        max_inventory.max_event_time,
        max_inventory_backstore.max_event_time,
        max_oblivion.max_event_time
from
    (select epc from dw_epcis_last_process) as all_epcis
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_slapship group by epc) as max_slapship using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_shipping group by epc) as max_shipping using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_receiving group by epc) as max_receiving using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_replenishment group by epc) as max_replenishment using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_replenishment_backward group by epc) as max_replenishment_backward using
(epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_replenishment_forward group by epc) as max_replenishment_forward using
(epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_fitting group by epc) as max_fitting using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_checkout group by epc) as max_checkout using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_trash group by epc) as max_trash using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_inventory group by epc) as max_inventory using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_inventory_backstore group by epc) as max_inventory_backstore using (epc)
    left join (select epc, MAX(event_time) as max_event_time from
dw_epcis_oblivion group by epc) as max_oblivion using (epc)
where epc LIKE 'urn:epc:tag:sgtin-96%' order by epc asc;

update dw_view_tracktrace v, dw_epcis_ean13 e set v.ean13 = e.ean13 where v.epc =
e.epc;
update dw_view_tracktrace v, dw_epcis_ext e set v.color = e.color, v.measure =
e.measure, v.model = e.model, v.description = e.model_description, v.season =
e.season where v.ean13 = e.ean13;

COMMIT;
SET autocommit=1;
END;

```

3.2.4.3. Modulo Inventory

Il modulo Inventory fornisce informazioni sulle giacenze inventariali. L'utente può accedere ai dati storici di giacenza per tutti i giorni di durata della campagna sperimentale, nonché ai dati del giorno corrente aggiornati pressoché in tempo reale (ovvero con un ritardo massimo di un'ora, corrispondente all'intervallo di aggiornamento periodico della data warehouse).

Un esempio di schermata di Inventory è mostrato in figura:

The screenshot shows the 'Inventory' module interface. On the left, there are filters for 'FILTRA PRODOTTI' (Stagione, Modello, Taglia, Colore) and 'INTERVALLO TEMPORALE' (Da: 19/11/2010, A: 29/11/2010). The main area displays a summary table and a detailed table.

Summary Table:

	Min	Max	Med	Ultimo
Magazzino	635	764	692,73	635
Retronegozio	1674	1903	1768,91	1674
Negozio	1635	1860	1741,45	1799
Venduto	6535	7017	6839,27	7017
Non Trovato	121	217	164,91	125

Detailed Table:

Data	Magazzino	Retronegozio	Negozio	Venduto	Non Trovato
19/11/2010	764	1770	1860	6535	217
20/11/2010	764	1736	1781	6661	210
21/11/2010	764	1724	1695	6764	209
22/11/2010	764	1722	1659	6805	211
23/11/2010	754	1722	1635	6844	210
24/11/2010	635	1903	1726	6865	121
25/11/2010	635	1894	1712	6880	129
26/11/2010	635	1774	1620	6891	130
27/11/2010	635	1817	1705	6966	127
28/11/2010	635	1722	1764	7004	125
29/11/2010	635	1674	1799	7017	125

Figura 16 - Schermata Inventory

I valori di giacenza per ogni giornata sono relativi al magazzino del CeDi, all'area vendita e al retronegozio; sono inoltre riportati i dati di venduto e infine i capi che si trovano nello stato denominato "Non trovato". Quest'ultimo valore si riferisce al numero di capi per cui il sistema non è riuscito a determinare la posizione nel punto vendita. Tali capi potrebbero trovarsi in area vendita

oppure in retronegozio. Le ragioni per cui un capo viene classificato come “Non trovato” sono due. La prima è un funzionamento non accurato del processo RFID di Replenishment, il quale è deputato al trasferimento logico dei capi tra area vendita e retronegozio, ed aggiorna la locazione logica di ciascun capo presente a sistema informativo. Il processo di Replenishment rileva il passaggio dei capi da un varco RFID posto nel corridoio di comunicazione tra le due zone del punto vendita, e modifica la locazione logica dei capi (il campo EPICS BizLocation) spostando in area vendita quelli che attualmente risultano in negozio e viceversa. Nel caso però in cui il varco RFID rilevi contemporaneamente capi appartenenti a due zone diverse, si verifica una situazione di indeterminatezza che richiede la conferma manuale dell’operatore che sta effettuando il trasporto fisico dei capi, tramite apposita interfaccia su schermo posizionata nel varco. Qualora l’operatore non effettui questa conferma manuale entro un certo tempo, i capi passano nello stato di “Non trovato”. La seconda ragione dipende dal processo di Inventario effettuato manualmente in punto vendita tramite lettore RFID: qualora infatti vi siano dei capi attesi in inventario in quanto già presenti nel sistema informativo, ma non rilevati durante la lettura RFID, questi passano nello stato di “Non trovato”. Si tratta in entrambi i casi di situazioni limite che durante la sperimentazione si sono verificate in una ridotta percentuale di capi, nondimeno il modulo Inventory permette di tracciare anche questo particolare stato. Anche il modulo Inventory presenta un pannello laterale con alcuni campi in base ai quali l’utente può filtrare i risultati. I campi sono suddivisi in due sezioni, analogamente a quanto descritto per il modulo Track&Trace:

- Filtra prodotti: contiene i parametri di filtro sui prodotti, che permettono di selezionare uno o più modelli, taglie, colori e/o stagioni. Saranno selezionati tutti i capi con uno qualunque di questi modelli, e una qualunque di queste taglie, e una qualunque di queste stagioni, e uno qualunque di questi colori (logica AND fra i tre campi).
- Intervallo temporale: consente di specificare le date di inizio e fine del periodo da mostrare. La tabella dei risultati mostrerà i valori di inventario soltanto per i giorni compresi tra le due date.

Il pannello principale mostra i risultati richiesti in seguito alla pressione del pulsante Query. Il pannello è suddiviso in tre sezioni: la prima è un riepilogo dei parametri di query impostati, analogamente agli altri moduli; la seconda è una tabella che fornisce alcune statistiche sui dati inventariali, come giacenze minime, massime e medie; la terza infine è la tabella dei risultati principale. Ogni riga della tabella corrisponde a un giorno, e riporta i dati di giacenza per quel giorno. Nel caso del giorno corrente i dati saranno parziali e riferiti all'ultimo aggiornamento della data warehouse. Le colonne della tabella sono le seguenti:

- Data
- Magazzino
- Retro negozio
- Negozio
- Venduto
- Non trovato

Cliccando su una riga di questa tabella è possibile accedere ad una ulteriore modalità di visualizzazione dei dati di inventario. In seguito al clic del tasto sinistro del mouse, infatti, compare una finestra che occupa quasi interamente la schermata, che riporta il dettaglio delle giacenze per il giorno selezionato. Questa finestra è mostrata nella figura seguente:

Inventory

Intervallo: 19/11/2010 - 29/11/2010

Prodotto: Nessun filtro / Taglia: Nessun filtro / Colore: Nessun filtro / Stagione: Nessun filtro

20/11/2010

Stagione: 20/11/2010

Nessun filtro

Modello: Risultati: 5116

Nessun filtro

Taglia: 8300737410: G356F19911 20 41 19 0 0 0 1 0

Nessun filtro

8300737050: QE45F02713 33 45 19 0 0 0 1 0

Colore: 2500022464: PK51A00116 22 44 19 0 0 0 1 0

Nessun filtro

2500023716: G282F18823 62 43 19 0 0 0 1 1

INTERVALLO TEMPO

Da: 2500024089: G000F08588 16 43 19 0 0 0 2 0

19 Nov

A: 2500024089: G000F08588 37 43 19 1 0 0 2 0

29 Nov

2500023709: G260F18588 13 41 19 0 0 0 5 1

2500022675: G000F08588 62 43 19 0 0 0 1 0

2500024089: G000F08588 19 39 19 0 0 0 1 0

2500024089: G000F08588 37 39 19 0 1 1 1 0

2500022457: G000F08588 16 39 19 0 0 0 1 0

2500024089: G000L08588 59 XXL 19 0 0 1 3 0

2500024089: G000F08588 37 41 19 0 0 0 1 0

2500022457: G000F08588 16 47 19 0 0 1 2 0

2500022675: G000F08588 62 47 19 0 0 0 1 0

2500022457: G000F08588 16 51 19 0 0 0 1 0

2500023716: G282F18823 62 41 19 0 0 0 1 0

2500024097: G282F18823 41 39 19 0 0 0 1 0

2500024095: G271F18852 41 41 19 0 0 0 1 0

2500024095: G271F18852 41 49 19 0 0 0 2 0

2500024095: G271F18852 41 45 19 0 0 0 1 0

2500024095: G271F18852 41 43 19 0 0 0 1 0

2500023716: G282F18823 62 37 19 0 0 0 1 0

2500023716: G282F18823 62 47 19 0 0 0 2 0

Figura 17 - Schermata Inventory in modalità dettagliata

Questo pannello presenta in alto il giorno selezionato e il numero di risultati totali, oltre al pulsante per l’esportazione dei dati in formato CSV. Al di sotto si trova una tabella che riporta una riga per ciascun prodotto (identificato dal codice EAN), con i valori di giacenza relativi a quel prodotto per quel giorno. Oltre alle colonne di giacenza per magazzino, retro negozio, area vendita, venduto e “Non trovato”, analoghe a quelle della schermata principale del modulo Inventory, questa tabella mostra anche i dettagli anagrafici del prodotto, quali modello, taglia, colore.

In definitiva, quindi, il modulo Inventory fornisce due modalità di visualizzazione dei dati di giacenza: una più generale, che mostra le giacenze complessive per ciascun giorno, da cui si può passare ad un’altra più dettagliata che per ciascun giorno riporta i dati di giacenza di ogni prodotto.

I dati utilizzati per visualizzare queste due tipologie di tabelle sono elaborati nel corso dell’aggiornamento periodico della data warehouse, come avviene per tutti gli altri moduli della Dashboard, e provengono dalle tabelle dw_view_inventory. Il codice di definizione delle due procedure è riportato di

seguito.

```
CREATE PROCEDURE `create_view_inventory_total`(startdate DATETIME, enddate
DATETIME, intval integer, unitval varchar(10))
    MODIFIES SQL DATA
BEGIN
declare thisDate timestamp;
declare nextDate timestamp;
declare i INTEGER;
declare warehouse_count INTEGER;
declare backstore_count INTEGER;
declare store_count INTEGER;
declare sold_count INTEGER;
declare oblivion_count INTEGER;
set thisDate = startdate;
SET i = -1;

SET autocommit=0;
START TRANSACTION;

delete from dw_view_inventory;
repeat
    select
        case unitval
            when 'MICROSECOND' then timestampadd(MICROSECOND, intval, thisDate)
            when 'SECOND'      then timestampadd(SECOND, intval, thisDate)
            when 'MINUTE'      then timestampadd(MINUTE, intval, thisDate)
            when 'HOUR'        then timestampadd(HOUR, intval, thisDate)
            when 'DAY'         then timestampadd(DAY, i + intval, thisDate)
            when 'WEEK'        then timestampadd(WEEK, intval, thisDate)
            when 'MONTH'       then timestampadd(MONTH, intval, thisDate)
            when 'QUARTER'     then timestampadd(QUARTER, intval, thisDate)
            when 'YEAR'        then timestampadd(YEAR, intval, thisDate)
        end into nextDate;
    insert into dw_view_inventory (time_stamp, warehouse, backstore, store, sold,
oblivion)
        select nextDate, sum(it.warehouse), sum(it.backstore), sum(it.store),
sum(it.sold), sum(it.oblivion)
        from dw_view_inventory_total it where CAST(it.time_stamp AS DATE) =
nextDate;
    SET i = i + 1;
UNTIL nextDate >= enddate
```

```

        end repeat;

COMMIT;
SET autocommit=1;
END;

```

Listato 15 - Definizione procedura create_view_inventory_total

```

CREATE PROCEDURE `update_view_inventory_total`(thisdatetime DATETIME)
    MODIFIES SQL DATA
BEGIN
    DECLARE done INT DEFAULT 0;
    declare thisdate DATE;
    declare ean13_cur VARCHAR(16);
    declare season_cur VARCHAR(8);
    declare model_cur VARCHAR(50);
    declare measure_cur VARCHAR(50);
    declare color_cur VARCHAR(50);
    declare description_cur VARCHAR(50);
    declare warehouse_count INTEGER;
    declare backstore_count INTEGER;
    declare store_count INTEGER;
    declare sold_count INTEGER;
    declare oblivion_count INTEGER;
    DECLARE cur1 CURSOR FOR select distinct(lp.ean13), lp.season, lp.model, lp.measure,
    lp.color, lp.description from dw_epcis_last_process lp where lp.ean13 is not null;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    SET autocommit=0;
    START TRANSACTION;
    set thisdate = CAST(thisdatetime AS DATE);
    delete from dw_view_inventory_total where CAST(time_stamp AS DATE) = thisdate;

    OPEN cur1;
    REPEAT
        FETCH cur1 INTO ean13_cur, season_cur, model_cur, measure_cur, color_cur,
        description_cur;
        IF NOT done THEN
            select count(lp.epc) into warehouse_count from dw_epcis_last_process lp
            where lp.biz_location = 'urn:epc:id:sgln:95100046.102.1' and lp.ean13 = ean13_cur;
            select count(lp.epc) into backstore_count from dw_epcis_last_process lp
            where lp.biz_location = 'urn:epc:id:sgln:95100046.220.1' and lp.ean13 = ean13_cur;

```

```

        select count(lp.epc) into store_count from dw_epcis_last_process lp where
lp.biz_location = 'urn:epc:id:sgln:95100046.225.1' and lp.ean13 = ean13_cur;
        select count(lp.epc) into sold_count from dw_epcis_last_process lp where
lp.biz_location = 'urn:epc:id:sgln:95100046.500.1' and lp.ean13 = ean13_cur;
        select count(lp.epc) into oblivion_count from dw_epcis_last_process lp
where lp.biz_location = 'urn:epc:id:sgln:95100046.230.1' and lp.ean13 = ean13_cur;

        insert into dw_view_inventory_total
(ean13, season, model, measure, color, description, warehouse, backstore,
store, sold, oblivion, time_stamp)
select
ean13_cur, season_cur, model_cur, measure_cur, color_cur, description_cur,
warehouse_count, backstore_count, store_count, sold_count, oblivion_count,
thisdate;
    END IF;
UNTIL done END REPEAT;
CLOSE cur1;

COMMIT;
SET autocommit=1;
END;

```

Listato 16 - Definizione procedura update_view_inventory_total

3.2.4.4. Modulo Fitting

Il modulo denominato Fitting presenta tutte le informazioni raccolte dal sistema RFID riguardo le prove dei capi effettuate dai clienti nei quattro camerini presenti nel punto vendita. Il modulo si presenta come in figura:

Intervallo	Prodotti										Conteggio prove per camerino				Venduto
Intervallo: 1	Prodotti	Stagior	Modello	Taglia	Colore	Camerino 1	Camerino 2	Camerino 3	Camerino 4						
2500024370038	MAGLIA/PULL	I9	M658LA74AM	XXL	42	0	0	0	0	0	0	0	0	2	
2500024382963	GIACCA	I9	R282F06883	41	1	0	0	0	0	0	0	0	0	2	
2500024411595	CANOTTA	I9	M412L0824M	XXL	33	0	0	0	0	0	0	0	0	2	
8300738738131	IMBOTTITO	I0	F230F00414	45	33	0	0	0	0	0	0	0	0	2	
2500022748990	GIACCONE	I9	8033F16510	39	1	0	0	0	0	0	0	0	0	1	
2500022749034	GIACCONE	I9	8033F16510	47	1	0	0	0	0	0	0	0	0	1	
2500023279165	GONNA	I9	1718F07230	49	42	0	0	0	0	0	0	0	0	1	
2500023318376	MAGLIA/PULL	I9	MB54L082UM	XXL	2	0	0	0	0	0	0	0	0	1	
2500023647285	MAGLIA/PULL	I9	M800L014EM	L	2	0	2	0	0	0	0	0	0	1	
2500023663339	CAPPOTTO	I9	0673F12958	43	33	0	0	0	0	0	0	0	0	1	
2500024089718	CAMICETTA/B	I9	G000L08588	S	59	0	0	0	0	0	0	0	0	1	
2500024095788	CAMICETTA/B	I9	G274F08922	41	32	0	0	0	0	0	0	0	0	1	
2500024149870	GONNA	I9	1624FA1681	49	32	0	0	0	0	0	0	0	0	1	
2500024164200	GIACCONE	I9	8033F07099	39	1	0	0	0	0	0	0	0	0	1	
2500024187667	CAMICETTA/B	I9	5623F12220	41	33	0	2	0	0	0	0	0	0	1	
2500024367823	T-SHIRT	I9	G409L08588	XXL	40	0	0	0	0	0	0	0	0	1	
2500024367861	T-SHIRT	I9	G448L09812	XL	81	0	0	0	0	0	0	0	0	1	
2500024368554	T-SHIRT	I9	MB52L162UM	XXL	75	0	0	0	0	0	0	0	0	1	
2500024368882	MAGLIA/PULL	I9	M251L109EM	S	73	0	1	0	0	0	0	0	0	1	
2500024369827	MAGLIA/PULL	I9	M627LA70VM	XL	11	0	0	0	0	0	0	0	0	1	
2500024370519	MAGLIA/PULL	I9	M742LA68UM	XL	19	0	0	0	0	0	0	0	0	1	
2500024370526	MAGLIA/PULL	I9	M742LA68UM	XXL	19	0	0	0	0	0	0	0	0	1	
2500024370625	T-SHIRT	I9	M783LA165M	XXL	1	0	0	0	0	0	0	0	0	1	
2500024370717	MAGLIA/PULL	I9	M787LA12XM	XL	1	0	1	0	0	0	0	0	0	1	
2500024370724	MAGLIA/PULL	I9	M787LA12XM	XXL	1	0	0	0	0	0	0	0	0	1	
2500024371943	CAPPOTTO	I9	0691F02009	39	12	0	0	0	0	0	0	0	0	1	
2500024381881	PANTALONE	I9	P510F08951	43	62	0	0	0	0	0	0	0	0	1	
2500024385520	CAMICETTA/B	I9	5685FA8588	47	16	0	2	0	0	0	0	0	0	1	
2500024385674	CAMICETTA/B	I9	5705F36107	47	1	0	1	0	0	0	0	0	0	1	
2500024387012	ABITO	I9	7933F18951	41	33	0	0	0	0	0	0	0	0	1	
2500024410215	TOP	I9	D644F02538	51	2	0	0	0	0	0	0	0	0	1	

Figura 18 - Schermata Fitting

I dati mostrati da questo modulo permettono di conoscere il numero di prove effettuate per ciascuna tipologia di capo nei vari camerini, confrontate con il numero di capi venduti recentemente. Questi dati consentono di mettere in evidenza la correlazione tra la prova di un capo in camerino e la vendita del capo stesso, un'informazione interessante per analizzare il comportamento dei clienti e valutare il gradimento del capo.

Analogamente a quanto avviene per gli altri moduli della Dashboard, anche il modulo Fitting presenta un pannello laterale con varie opzioni di filtraggio dei dati, organizzate in due sezioni:

- Filtra prodotti: contiene i parametri di filtro sui prodotti secondo due modalità alternative. La prima è per codice EAN13, e offre la possibilità di selezionare dal menu a tendina uno o più codici di interesse. La seconda offre quattro menu a tendina che permettono di selezionare uno o più modelli, taglie, colori e/o stagioni. Saranno selezionati tutti i capi con uno qualunque di questi modelli, e una qualunque di queste taglie, e una qualunque di queste stagioni, e uno qualunque di questi colori

(logica AND fra i tre campi).

- Intervallo temporale: consente di specificare l'intervallo temporale sul quale conteggiare i capi provati e venduti. A differenza di altri moduli in cui è possibile specificare liberamente una data d'inizio e una data di fine periodo, qui viene offerta all'utente solo la possibilità di specificare l'ultimo giorno o gli ultimi due, tre, cinque, sette, quindici o trenta giorni.

Alla pressione del pulsante Query, i risultati vengono visualizzati nel pannello principale, il quale è suddiviso in due sezioni: la prima riassume i parametri di filtro impostati, e la seconda contiene i dati in formato tabulare. Ogni riga della tabella corrisponde a una tipologia di capo (identificata da un codice EAN univoco), e mostra per quel capo il numero di prove nei quattro camerini e la quantità venduta. Tutti i valori si riferiscono al periodo temporale selezionato (il comportamento di default è mostrare i dati riferiti all'ultimo giorno). Più in dettaglio, le colonne della tabella sono le seguenti:

- Prodotto: riporta codice EAN13 e una descrizione testuale
- Stagione
- Modello
- Taglia
- Colore
- Prove in camerino 1
- Prove in camerino 2
- Prove in camerino 3
- Prove in camerino 4
- Venduto

I dati utilizzati per alimentare questa tabella vengono elaborati nel corso dell'aggiornamento periodico della data warehouse, come avviene per tutti gli altri moduli della Dashboard. In particolare per questo modulo i dati provengono da una serie di tabelle denominate `dw_view_fitting1`, `dw_view_fitting2`, `dw_view_fitting3`, `dw_view_fitting5`, `dw_view_fitting7`,

dw_view_fitting15, dw_view_fitting30. Analogamente a quanto esposto per gli altri moduli, anche in questo caso l'elaborazione dei dati è interamente delegata alla fase di aggiornamento periodico, in modo da ridurre il più possibile i tempi di attesa in seguito alle richieste dell'utente: nel caso del modulo di Fitting, però, i dati variano a seconda del periodo di tempo selezionato, e la natura dell'elaborazione richiesta non consentiva di filtrare dinamicamente i dati in base all'intervallo temporale specificato dall'utente. Per questo motivo si è reso necessario calcolare separatamente i conteggi di prove e venduto relativi ai possibili intervalli temporali, e i dati vengono memorizzati in altrettante tabelle che condividono la medesima struttura. Ad esempio, dw_view_fitting3 contiene i dati relativi agli ultimi tre giorni, mentre dw_view_fitting15 contiene gli stessi dati ma relativi agli ultimi 15 giorni.

L'elaborazione avviene anche in questo caso tramite procedure MySQL interne al database. Le procedure interessate sono due: una esegue i conteggi relativi ad un intervallo temporale specificato come parametro in ingresso alla procedura stessa; l'altra si occupa di invocare ripetutamente la prima, passandole di volta in volta l'intervallo temporale, e inserisce i dati nella tabella opportuna. Di seguito si riporta la definizione della prima procedura e un suo esempio di invocazione, relativo al calcolo dei dati per gli ultimi 7 giorni.

```
CREATE PROCEDURE `create_view_fitting_epcs`(d DATE)
    MODIFIES SQL DATA
BEGIN

delete from dw_view_fitting;
insert into dw_view_fitting (epc, sold)
    select epc, event_time from dw_epcis_last_process
    where epc LIKE 'urn:epc:tag:sgtin-96%' and biz_location =
'urn:epc:id:sgln:95100046.500.1' and CAST(event_time AS DATE) >= d;
insert into dw_view_fitting (epc, fitting1)
    select epc, event_time from dw_epcis_fitting
    where epc LIKE 'urn:epc:tag:sgtin-96%' and read_point =
'urn:epc:id:gid:95100046.101.2' and CAST(event_time AS DATE) >= d;
insert into dw_view_fitting (epc, fitting2)
    select epc, event_time from dw_epcis_fitting
    where epc LIKE 'urn:epc:tag:sgtin-96%' and read_point =
```

```

'urn:epc:id:gid:95100046.101.3' and CAST(event_time AS DATE) >= d;
insert into dw_view_fitting (epc, fitting3)
    select epc, event_time from dw_epcis_fitting
    where epc LIKE 'urn:epc:tag:sgtin-96%' and read_point =
'urn:epc:id:gid:95100046.102.2' and CAST(event_time AS DATE) >= d;
insert into dw_view_fitting (epc, fitting4)
    select epc, event_time from dw_epcis_fitting
    where epc LIKE 'urn:epc:tag:sgtin-96%' and read_point =
'urn:epc:id:gid:95100046.102.3' and CAST(event_time AS DATE) >= d;
update dw_view_fitting v, dw_epcis_ean13 e set v.ean13 = e.ean13 where v.epc =
e.epc;
update dw_view_fitting v, dw_epcis_ext e set v.season = e.season, v.color =
e.color, v.measure = e.measure, v.model = e.model, v.description =
e.model_description where v.ean13 = e.ean13;

END;

```

Listato 17 - Definizione procedura create_view_fitting_epcs

```

call create_view_fitting_epcs(date_7);
delete from dw_view_fitting7;
OPEN cur1;
REPEAT
    FETCH cur1 INTO ean13_cur, season_cur, model_cur, measure_cur, color_cur,
description_cur;
    IF NOT done THEN
        select count(d.epc) into fitting1_cur from dw_view_fitting d where
d.fitting1 is not null and d.ean13 = ean13_cur;
        select count(d.epc) into fitting2_cur from dw_view_fitting d where
d.fitting2 is not null and d.ean13 = ean13_cur;
        select count(d.epc) into fitting3_cur from dw_view_fitting d where
d.fitting3 is not null and d.ean13 = ean13_cur;
        select count(d.epc) into fitting4_cur from dw_view_fitting d where
d.fitting4 is not null and d.ean13 = ean13_cur;
        select count(d.epc) into sold_cur from dw_view_fitting d where d.sold is
not null and d.ean13 = ean13_cur;
        insert into dw_view_fitting7
        (ean13, season, model, measure, color, description, fitting1, fitting2,
fitting3, fitting4, sold)
        select
        ean13_cur, season_cur, model_cur, measure_cur, color_cur, description_cur,
fitting1_cur, fitting2_cur, fitting3_cur, fitting4_cur, sold_cur;

```


generico, focalizzata sulla situazione attuale e priva dei dati storici; rispetto al generico inventario, però, questa interfaccia mette in evidenza visivamente le situazioni che richiedono un intervento da parte dell'operatore. Nella tabella infatti le righe (corrispondenti alle tipologie di capi) sono colorate in maniera differente quando vi è la necessità di ripristinare il capo in area vendita oppure di procedere al riordino. La necessità di ripristino sorge quando un prodotto, che sia stato venduto almeno una volta nell'ultimo periodo, non è più presente in area vendita ma è disponibile nel retro negozio: questa situazione, nota come rottura di stock, determina una riduzione del fatturato dovuta alla mancata vendita dei capi non esposti sugli scaffali, oltre a una mancata soddisfazione delle esigenze del cliente, e comporta dunque un danno considerevole (ECR Europe, 2003; Pramatarì e Milotis, 2008). Se il capo non è disponibile neppure come scorta di retro negozio, allora si rende necessario il riordino presso il CeDi. Uno degli obiettivi del progetto in esame è proprio quello di ridurre il più possibile queste situazioni di out of stock, tramite una notifica tempestiva agli operatori di punto vendita della necessità di ripristinare certi capi sugli scaffali. Questo modulo permette di valutare immediatamente la situazione delle giacenze e individuare le situazioni anomale. Oltre al modulo di Replenishment è stato sviluppato un altro strumento specifico, basato sugli stessi dati della Dashboard, che risulta di utilizzo più immediato da parte degli operatori, e viene trattato altrove in questo capitolo. Il modulo di Replenishment infatti, come tutta l'interfaccia Dashboard, va inteso principalmente come uno strumento di consultazione e di Business Intelligence, piuttosto che come un'applicazione operativa da utilizzare nel corso dei processi logistici. Analogamente agli altri moduli della Dashboard, anche questo presenta un pannello laterale con tre sezioni per il filtraggio dei dati:

- **Filtra prodotti:** contiene i parametri di filtro sui prodotti che permettono di selezionare uno o più modelli, taglie, colori e/o stagioni. Saranno selezionati tutti i capi con uno qualunque di questi modelli, e una qualunque di queste taglie, e una qualunque di queste stagioni, e uno qualunque di questi colori (logica AND fra i tre campi).
- **Analisi vendite:** consente di specificare l'intervallo temporale sul quale

conteggiare i capi venduti. Analogamente al modulo di Fitting, viene offerta all'utente la possibilità di specificare l'ultimo giorno o gli ultimi due, tre, cinque, sette, quindici o trenta giorni. Questo parametro influisce esclusivamente sul valore di venduto, e non sui valori di giacenza che sono sempre aggiornati ai dati più recenti.

- Attività da eseguire: questo parametro permette di selezionare una o più tipologie di attività tra ripristino e riordino, oppure nessuna attività necessaria.

Premendo il pulsante Query vengono caricati i dati richiesti nel pannello principale dell'interfaccia, oltre a uno specchietto riassuntivo dei parametri di filtro. La tabella principale presenta le seguenti colonne:

- Prodotto: riporta codice EAN13, una descrizione testuale, taglia, colore, stagione e modello
- Magazzino: giacenza attuale in magazzino
- Retro negozio: giacenza attuale in retro negozio
- Negozio: giacenza attuale in area vendita
- Venduto (ultimi giorni): quantità venduta negli ultimi N giorni (dove N è specificato dall'utente come parametro di interrogazione)

Ciascuna riga corrisponde a una tipologia di prodotto. Lo sfondo della riga viene evidenziato con una colorazione particolare a seconda della situazione: in caso vi sia la necessità di un ripristino la riga è arancio, è gialla in caso di necessità di riordino, bianca altrimenti. Qualora la giacenza di negozio o retronegozio sia zero, il valore della casella è evidenziato in rosso.

Come avviene per gli altri moduli della Dashboard, anche i dati su cui si basa il Replenishment sono elaborati nel corso dell'aggiornamento periodico della data warehouse, al fine di ottimizzare i tempi di risposta percepiti dall'utente. La tabella su database in cui sono memorizzati i dati mostrati nell'interfaccia è `dw_view_replenishment`. A differenza del modulo di Fitting, in cui viene utilizzata una diversa tabella per ciascun intervallo temporale selezionabile

dall'utente, in questo caso si è preferito includere i valori di venduto, che variano a seconda dell'intervallo, come altrettante colonne nella medesima tabella. A seconda del valore selezionato dall'utente, l'interfaccia mostra di volta in volta i valori presi dalla colonna opportuna: nel caso ad esempio in cui si specifichi un valore di venduto negli ultimi 7 giorni, verranno mostrati i dati della colonna sold7.

3.2.4.6. Modulo History

L'ultimo modulo presentato è quello denominato History, il quale permette di consultare un elenco di tutti i processi logistici subiti da un singolo capo, con le relative date. Questo modulo può essere utilizzato direttamente, oppure vi si può accedere tramite il modulo di Track & Trace. Come è stato esposto nella sezione relativa, infatti, cliccando su una riga della tabella dei risultati di Track & Trace si passa automaticamente al modulo History per il capo corrispondente alla riga selezionata. Questo evita all'utente di dover recuperare ed inserire manualmente il codice EPC che identifica il capo. Il modulo History ha principalmente una funzione diagnostica del sistema RFID, e può essere utilizzato per verificare eventuali situazioni anomale o per conoscere in dettaglio la sequenza temporale dei processi logistici in cui è stato individuato il capo in esame. Rispetto al modulo Track & Trace infatti, che mostra esclusivamente la data più recente per ciascun processo, qui sono riportate tutte le date in cui un capo ha subito un certo processo: ciò è significativo soprattutto nel caso dei processi di prova in camerino o di ripristino, che tipicamente si verificano più volte nella vita di un capo.

Una schermata esemplificativa del modulo di History è mostrata in figura:

Data	Processo
28/11/2010 15:48	ReplenishmentBackward
28/11/2010 10:55	ReplenishmentForward
27/11/2010 19:01	ReplenishmentBackward
26/11/2010 18:40	ReplenishmentForward
24/11/2010 17:04	ReplenishmentBackward
24/11/2010 11:58	Inventory Backstore
24/11/2010 10:40	Receiving
19/11/2010 15:06	Slap&Ship

Figura 20 - Schermata History

Anche questo modulo presenta un pannello laterale per la selezione dei parametri di filtraggio. In questo caso però è presente un unico parametro, ovvero una casella di testo che consente di specificare il codice EPC, completo quindi di seriale univoco, che individua il capo di interesse. Premendo il pulsante Query vengono mostrati i risultati nel pannello principale dell'interfaccia. Questo pannello presenta una tabella con due colonne: tipologia di processo e data.

A differenza degli altri moduli della Dashboard, il modulo History non necessita di una fase di elaborazione dei dati specifica durante l'aggiornamento periodico. Le informazioni visualizzate sono dunque ricavate direttamente ed in tempo reale da alcune tabelle di utilizzo generale presenti nella data warehouse. In particolare, si tratta delle tabelle relative a ciascun processo logistico, come dw_epcis_receiving, dw_epcis_slapship e altre. Non è stata dunque sviluppata una tabella specifica per memorizzare i risultati di questo modulo, né una procedura interna del database.

3.2.5 Esportazione dati

Le informazioni elaborate durante il processo di aggiornamento della data warehouse vengono rese disponibili all'utente non soltanto attraverso l'interfaccia grafica di Dashboard, ma possono anche essere esportate in formati standard adatti alla condivisione con altre applicazioni. Sono state realizzate due modalità di esportazione differenti. La prima è accessibile tramite un pulsante Esporta nell'interfaccia web, collocato in prossimità del pulsante Query in tutti i moduli della Dashboard. Alla pressione di questo tasto l'applicazione genera un file in formato CSV (Comma Separated Values) contenente tutti i dati risultato della richiesta, così come verrebbero mostrati nell'interfaccia. Una volta che il file è stato generato, il browser ne propone lo scaricamento all'utente, che può così memorizzare una copia del file sul computer locale. Il file CSV contiene record in formato testuale, uno per riga, con i singoli campi separati dal carattere ','.

La seconda modalità di esportazione dei dati è stata realizzata appositamente per il modulo Replenishment della Dashboard. L'applicazione web infatti mette a disposizione un particolare indirizzo URL, accessibile tramite il consueto protocollo HTTP, che corrisponde ad un documento in formato XML contenente i dati del modulo Replenishment. Questo file XML viene generato dall'applicazione sulla base degli stessi dati utilizzati anche dall'interfaccia grafica del modulo Replenishment, filtrandoli però in modo da includere soltanto i record corrispondenti a capi che necessitano di ripristino o riordino. Il documento XML conterrà dunque l'elenco di capi non presenti in area vendita, e quindi da ripristinare a scaffale, oppure non presenti né in area vendita né in retro negozio, e quindi da riordinare presso il CeDi. Questo elenco è stato generato in formato XML per essere utilizzato da una applicazione apposita su terminale mobile: si tratta di uno strumento a disposizione degli operatori di punto vendita per controllare in maniera immediata i capi da ripristinare o riordinare. Questa applicazione su terminale mobile, realizzata in linguaggio .NET C#, accede via rete tramite protocollo HTTP all'indirizzo citato sopra, ottiene in risposta i dati in formato XML, e li utilizza per comporre l'interfaccia

grafica da mostrare all'utente.

4. La sperimentazione

4.1. Le statistiche della sperimentazione

In seguito alla fase progettuale dell'RFID Fashion Pilot, che ha visto la reingegnerizzazione mediante tecnologia RFID dei processi logistici di magazzino e punto vendita, nonché la definizione degli strumenti di Business Intelligence oggetto di questo lavoro, si è svolta la campagna sperimentale durante la quale sono stati testati sul campo i processi reingegnerizzati. In questa sezione si presenta un quadro delle metriche e delle statistiche raccolte nel progetto pilota, e si mettono in evidenza le principali informazioni a valore aggiunto che lo strumento di Dashboard mette a disposizione dell'utente.

Il periodo di riferimento per i risultati esposti di seguito va dal 2 Aprile 2010 al 31 Agosto 2010, corrispondente alla stagione Primavera/Estate della collezione Elena Mirò. Il volume complessivo dei capi monitorati in questo periodo è pari a 12690. Di questi, il 42% è rappresentato da capi stesi e accessori (T-Shirt, Camicette, Sciarpe, Bigiotteria), che tendenzialmente vengono inviati dal CeDi in colli ed esposti in area vendita su scaffali. Il 58% invece è rappresentato da capi appesi (giacche, pantaloni, camicie), che vengono inviati dal CeDi su stendini e sono in seguito esposti in area vendita appesi a grucce. I capi tracciati possono essere suddivisi in categorie a seconda di colore, taglia e modello: in totale sono

stati individuati 87 colori, 752 modelli e 46 taglie differenti, per un totale di 3818 varianti.

4.2. Informazioni a valore aggiunto

4.2.1. Le tipologie di informazioni a valore aggiunto

I vantaggi dell'implementazione di un sistema RFID vengono generalmente associati alla possibilità di automatizzare i processi di acquisizione delle informazioni. Tale visione risulta peraltro riduttiva, in quanto non tiene in dovuta considerazione le potenziali ricadute dell'adozione di questa tecnologia in ambiti non immediatamente coinvolti. L'impatto della tecnologia RFID, infatti, pervade non solo il settore dell'automazione industriale e dell'acquisizione automatica delle informazioni, ovvero i settori più evidentemente interessati a tale tecnologia; grazie alla condivisione delle informazioni tra gli attori della filiera un sistema RFID è infatti in grado di aggiungere notevole valore a processi non immediatamente riconducibili ad esso, mettendo a disposizione preziosi indicatori di prestazioni e altre informazioni di Business Intelligence. Questo concetto è illustrato nella figura seguente, che raffigura con la metafora dell'iceberg gli effetti più profondi e pervasivi della tecnologia RFID.

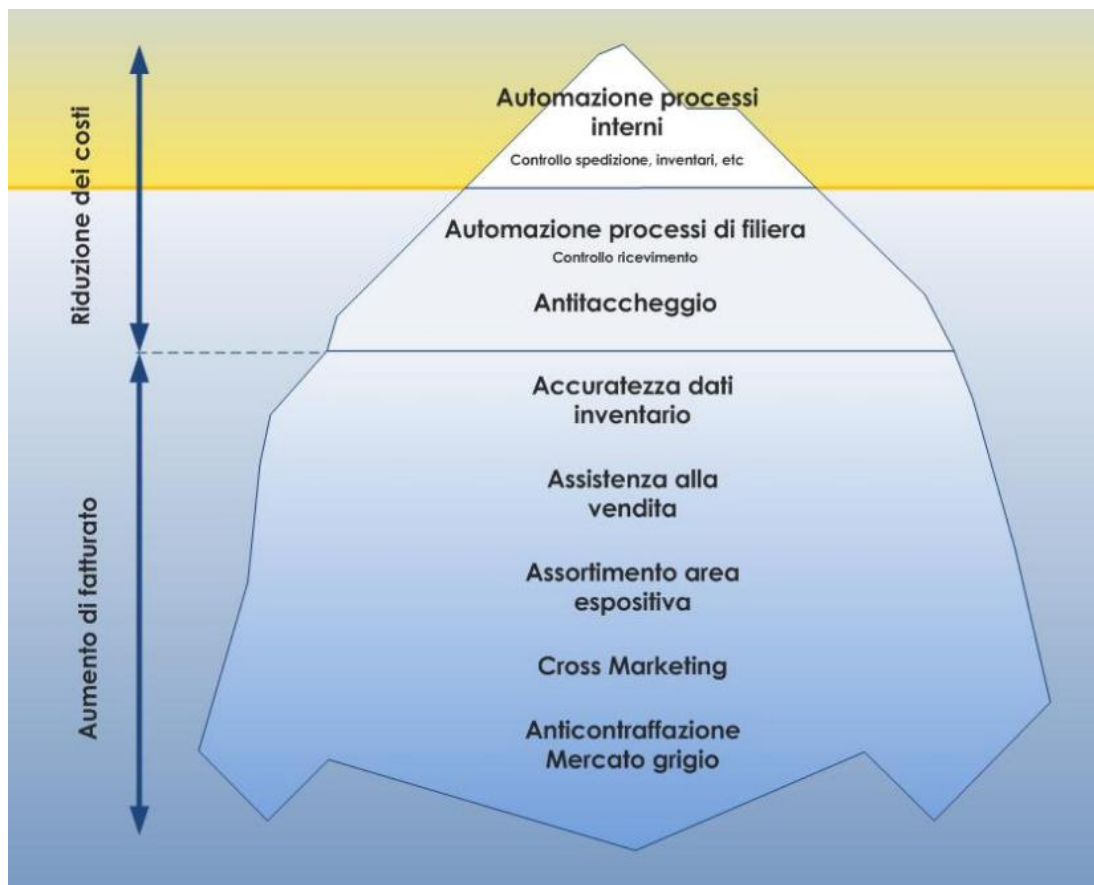


Figura 21 - La metafora dell'“iceberg” RFID

Nel progetto pilota oggetto di questa dissertazione, il modulo preposto alla generazione e alla visualizzazione di tali informazioni a valore aggiunto, che come detto rappresentano il potenziale meno evidente ma più vasto dell'adozione di un sistema RFID, è l'applicazione Dashboard.

Un primo campo nel quale la Dashboard realizzata può fornire un contributo sostanziale, grazie ai dati raccolti dal campo tramite RFID, è la gestione delle scorte. La visibilità dei flussi fisici dei prodotti offerta dalla tecnologia RFID lungo l'intera filiera rappresenta infatti un aspetto di interesse per il produttore di beni. Generalmente tale visibilità si ferma al momento della spedizione dell'ordine dalla banchina del CeDi; da questo istante in poi manca ogni riscontro sulle movimentazioni che il prodotto subisce lungo la supply chain, fino all'arrivo di un nuovo ordine del retailer. Grazie alla tecnologia RFID è invece possibile avere visibilità in tempo reale dei flussi di prodotto e dei dati di

vendita; è pertanto possibile sviluppare modelli previsionali della domanda che siano fondati su tali dati appunto e non solo su proiezioni. L'accesso ai dati inventariali e di vendita offerto dalla Dashboard, non solo al gestore di punto vendita ma anche al CeDi e al produttore, rappresenta uno strumento dal notevole potenziale per monitorare i consumi di prodotto associati alle vendite. La Dashboard, in particolare attraverso il modulo di Tracciabilità (Track & Trace) e Inventario (Inventories), rende infatti disponibili in tempo reale i flussi fisici di prodotto nella supply chain, compresi le spedizioni da CeDi e i ricevimenti in punto vendita. Altrettanto interessanti sono le potenzialità legate all'analisi delle vendite, che permette la determinazione puntuale dei capi da riassortire e offre statistiche di vendita temporali e per categoria. Di seguito viene presentata una panoramica generale di alcune tipologie di informazioni a valore aggiunto messe a disposizione dalla Dashboard per la Business Intelligence, illustrate con esempi.

- *Statistiche aggregate*: tramite il modulo Track & Trace, fissando un intervallo temporale, è possibile ottenere l'elenco dei capi spediti (selezionando il processo di Shipping), ricevuti (selezionando il processo di Receiving), venduti (selezionando Checkout e Trash) o che hanno subito altri processi nell'intervallo selezionato. È possibile anche filtrare ulteriormente i dati in base a modello, taglia, colore, stagione, codice EAN o bolla di trasporto. Un esempio di tali statistiche è il conteggio dei capi venduti giornalmente o settimanalmente nell'intervallo di tempo.
- *Distribuzione temporale dei quantitativi spediti*: sfruttando la possibilità di esportazione in formato CSV, si possono realizzare grafici basati sui dati ottenuti dalla Dashboard. Ad esempio esportando i dati di tracciabilità riferiti ai capi spediti in un intervallo temporale, e incrociandoli con i dati di anagrafica, si può ottenere per ciascun giorno dell'intervallo il numero di capi stesi e di capi attesi che sono stati spediti, e costruirne l'istogramma.
- *Distribuzione delle vendite per categoria*: analogamente alla procedura descritta sopra, è possibile costruire basandosi sui dati di tracciabilità riferiti al venduto un diagramma a torta che mostra la percentuale

rappresentata da ciascun modello sul totale dei capi venduti.

- *Controllo delle bolle di trasporto*: l'interfaccia può essere utilizzata anche per verificare la corrispondenza tra i quantitativi dichiarati nelle bolle di trasporto che accompagnano il ricevimento della merce e i capi fisicamente rilevati in punto vendita tramite lettura RFID. Interrogando la Dashboard per Documento di Trasporto (DDT) e raggruppando i risultati per modello e colore, infatti, si ottiene una struttura analoga a quella riportata sul documento, per un confronto immediato.

Nelle sezioni seguenti vengono analizzati in dettaglio altri indicatori particolarmente significativi in ottica di Business Intelligence e supporto decisionale, corredati dai risultati riscontrati durante la campagna sperimentale.

4.2.2. Esempio I: Rapporto tra prove in camerino e vendite

Tramite il modulo Fitting dell'interfaccia Dashboard è possibile analizzare la correlazione tra prove di un capo in camerino e vendita del capo stesso, utile per sviluppare modelli di comportamento dei clienti. Questo dato è ricavabile in virtù della tecnologia RFID che permette l'identificazione univoca dei singoli capi; non sarebbe stato possibile ottenerlo impiegando ad esempio una tradizionale tecnologia barcode che identifica esclusivamente la categoria (codice EAN) di prodotto. Una prima informazione di correlazione riguarda il tempo che intercorre tra la prova del capo e la sua vendita. Tale informazione si può ottenere esportando dal modulo Track & Trace le date di ultima prova in camerino e di vendita per ciascun capo, e calcolando la differenza tra le due nel caso in cui entrambe siano avvenute nello stesso giorno; un altro dato significativo è rappresentato dalla percentuale di capi che sono stati provati ma di cui non risulta la vendita entro il giorno, così come dei capi venduti senza essere stati provati. Passando ai dati aggregati, risulta interessante il calcolo del rapporto tra quantità vendute e provate. Questo dato è ricavabile dal modulo

Fitting messo a disposizione dalla Dashboard. Con riferimento alla campagna sperimentale del progetto pilota, i diagrammi seguenti mostrano appunto la distribuzione dei tempi medi intercorsi tra prova e vendita dei capi, e il rapporto tra numero di vendite e prove per modello durante l'intero periodo della sperimentazione:

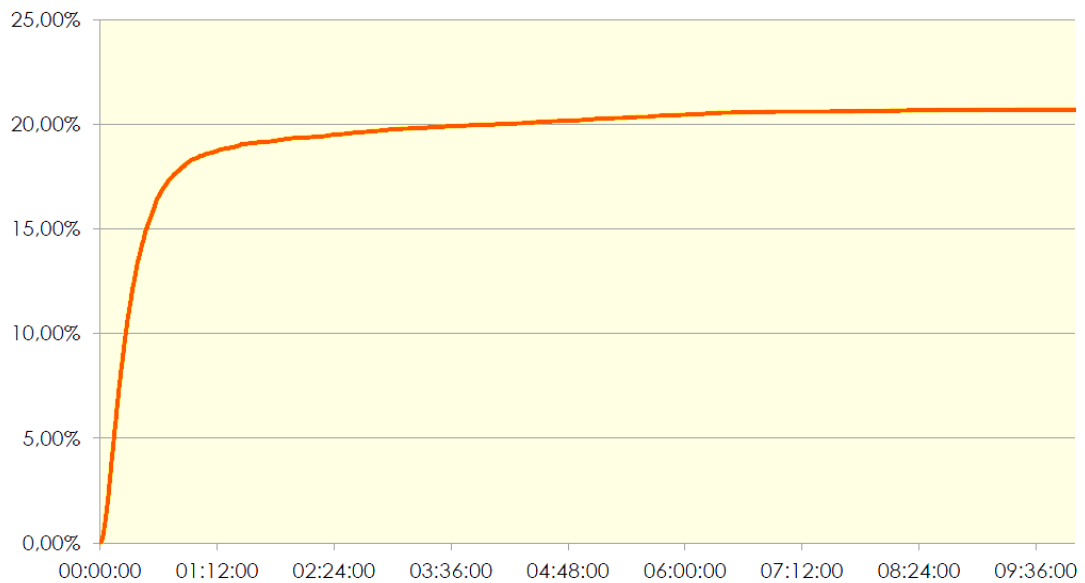


Figura 22 - Distribuzione dei tempi tra prova e vendita di un capo

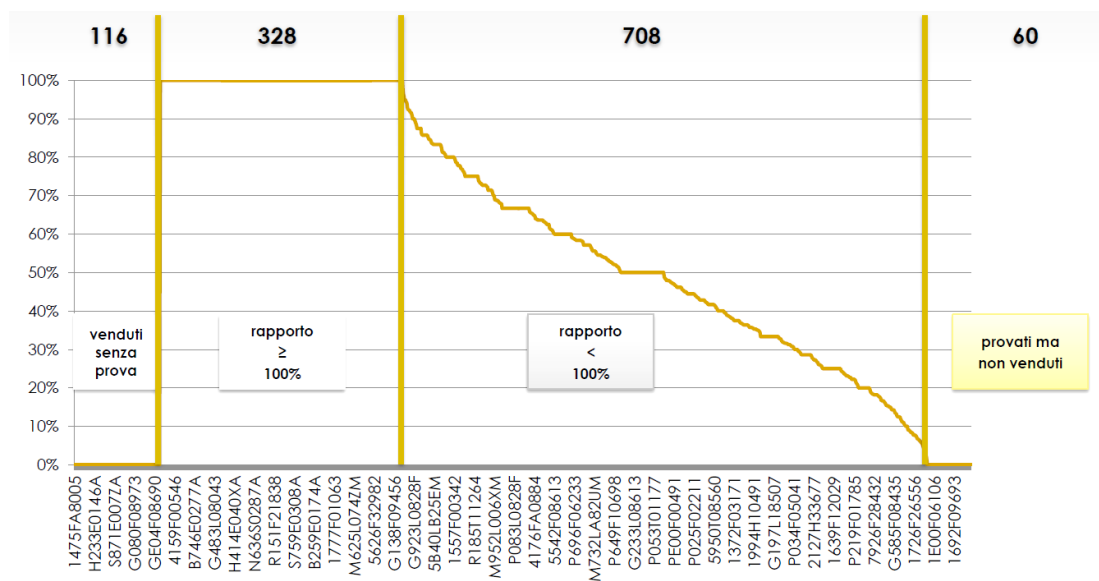


Figura 23 - Rapporto tra numero di prove e vendite

Dal primo diagramma si evince come nel 20% dei casi sia intercorsa meno di un'ora tra prova in camerino e vendita (ed è quindi più sicura una correlazione diretta tra i due eventi). Il secondo riporta come solamente il 5% dei capi non sia stato venduto dopo essere stato provato, e il 10% sia stato viceversa venduto senza essere provato. Questi ultimi prodotti sono quasi esclusivamente rappresentati da bigiotteria e occhiali.

4.2.3. Esempio II: Valutazione dell'impatto sul fatturato

Tra i vari indicatori che permettono di valutare l'impatto sul fatturato delle soluzioni RFID implementate nel progetto pilota, uno dei principali è rappresentato dall'efficacia degli strumenti di supporto alla vendita che sono stati realizzati, e il loro contributo all'aumento di fatturato complessivo. Tale indicatore è ricavabile dalla Dashboard a partire dai dati del modulo Track & Trace e dai dati storici di utilizzo dell'apposita applicazione per il ripristino (cfr. capitolo 3). L'applicazione permette agli operatori di punto vendita di generare una stampa con l'elenco dei prodotti mancanti in area vendita ma presenti nel magazzino di retronegozio (analoga alla visualizzazione del modulo Replenishment della Dashboard, e basata sugli stessi dati). In questo modo gli operatori possono procedere al ripristino esponendo i capi in area vendita. Esportando tali dati e selezionando i capi che hanno subito il processo di ripristino e sono stati in seguito venduti, è possibile costruire un diagramma come quello riportato qui.

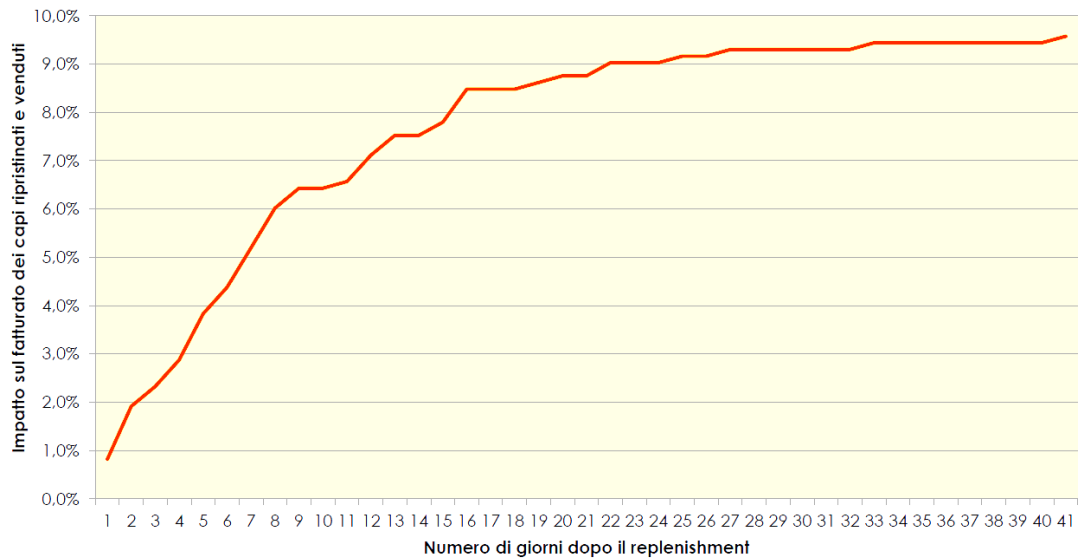


Figura 24 - Distribuzione del tempo tra ripristino e vendita di un capo

Il grafico mostra la percentuale dei capi che sono stati ripristinati entro un certo numero di giorni prima della vendita, sul totale dei capi venduti. Si può notare come questa percentuale si assesti sul 9-10% se si considerano i capi venduti 20 giorni o più dopo essere stati ripristinati. È possibile ritenere che questa percentuale corrisponda a un equivalente incremento del fatturato, assumendo che i capi non disponibili in area vendita (a causa del mancato ripristino) si traducano in mancate vendite.

Sul totale dei capi venduti durante i giorni in cui è stata attiva la funzionalità di ripristino, una frazione pari al 10,8% ha subito il processo logistico di ripristino, e di questi capi l'88,6% sono stati in seguito venduti.

5. Conclusioni

Nell'ambito del progetto pilota "RFID Fashion Pilot", che propone la reingegnerizzazione tramite tecnologia RFID dei processi logistici nel settore fashion, è stato sviluppato un framework software per l'elaborazione dei dati RFID e la loro trasformazione in informazioni di Business Intelligence a valore aggiunto. Il progetto pilota, frutto di una collaborazione tra RFID Lab e varie aziende nazionali ed internazionali del settore tessile e della moda, coinvolge un centro di distribuzione sito a Pollenzo ed un punto vendita outlet a Fidenza.

I moduli software realizzati consistono in un sistema di data warehouse mantenuto aggiornato con i dati raccolti dai processi RFID, e un'interfaccia web che fornisce all'utente una Dashboard logistica per la visualizzazione e l'interrogazione delle informazioni elaborate. Tutta l'infrastruttura di rete e lo scambio dei dati sono conformi alle specifiche EPCglobal Network, un'architettura standard a livello internazionale che garantisce l'interoperabilità tra partner logistici. Gli strumenti software utilizzati per l'implementazione del sistema comprendono MySQL RDBMS, Oracle Glassfish Application Server, la libreria di interfaccia con il database Hibernate, e le librerie in linguaggio Java Google Web Toolkit (GWT) e SmartGWT per la realizzazione di applicazioni web.

La Dashboard offre vari moduli focalizzati su differenti aspetti di Business Intelligence: Track&Trace per la tracciabilità dei capi lungo la catena logistica, Inventory per giacenze e inventari, Replenishment per le segnalazioni di

ripristino e riordino di capi, Fitting per le statistiche sulle prove in camerino, History per l'analisi dei processi RFID subiti da un singolo capo. Questi moduli si sono rivelati indispensabili per filtrare l'ingente mole di dati generata dai processi RFID e presentare all'utente finale informazioni coerenti, puntuali e utili a fornire un supporto decisionale e a valutare le prestazioni del sistema. In questo senso quindi l'obiettivo del progetto, ovvero estrarre informazioni a valore aggiunto dal sistema RFID, può dirsi pienamente raggiunto.

Tra le principali criticità riscontrate durante lo sviluppo software si ricorda l'elevato carico computazionale richiesto dall'elaborazione dei dati da visualizzare nell'interfaccia grafica: per mantenere i tempi di risposta alle richieste dell'utente nell'ordine dei decimi di secondo ci si è concentrati sull'ottimizzazione delle strutture dati e delle query su database, e si è organizzata l'architettura del sistema in maniera tale da spostare la maggior parte del carico computazionale sulla fase di aggiornamento periodico dei dati, che avviene ad intervalli di circa un'ora, piuttosto che sulle richieste in tempo reale effettuate dagli utenti. Un altro elemento critico nella realizzazione del software è stato la selezione e in seguito l'implementazione di un framework adeguato alla costruzione di un'applicazione web complessa e capace di gestire e visualizzare grandi quantità di dati senza cali di prestazioni: in questo la scelta di GWT si è rivelata adeguata.

Per quanto riguarda possibili sviluppi futuri del progetto, il principale limite consiste nella ridotta flessibilità del sistema così come è concepito. Qualora infatti la topologia della supply chain coinvolta dovesse modificarsi radicalmente, con l'aggiunta ad esempio di stadi intermedi tra CeDi e punto vendita, oppure i punti di lettura RFID e i relativi processi logistici dovessero cambiare, il framework software dovrebbe essere rivisto ed adattato di conseguenza. Altrettanto critica sarebbe la generalizzazione del sistema ad altri settori merceologici diversi da quello dell'abbigliamento, in quanto lo strumento si basa su una definizione aprioristica dei tipi di dato coinvolti. Sebbene l'inclusione ad esempio di nuovi processi logistici comporterebbe necessariamente una modifica del framework, è altresì vero che l'architettura estremamente modulare del sistema faciliterebbe questo tipo di modifiche; lo

sviluppo di un livello di generalizzazione superiore, in grado di adattare la Dashboard dinamicamente a configurazioni variabili della catena logistica, è quindi fattibile mantenendo l'attuale architettura di base, e potrebbe costituire un ambito di sperimentazione futura. Anche l'aderenza agli standard dell'EPC Network rende possibili futuri scenari di integrazione con sistemi informativi esterni e strutture di rete più ampie. In particolare l'acquisizione di tutti i dati che alimentano la data warehouse avviene esclusivamente tramite l'interrogazione di sistemi EPCIS che seguono le specifiche standard: questo rende già da ora il framework flessibile ed in grado di supportare dinamicamente l'aggiunta e la rimozione degli EPCIS coinvolti. Nonostante quindi la Dashboard logistica sia prettamente funzionale al progetto pilota, essa presenta comunque ottime prospettive di ampliamento e generalizzazione nell'ottica di una maggiore flessibilità, per diventare una soluzione di Business Intelligence applicabile ad una casistica sempre più ampia.

Il framework software realizzato rappresenta quindi un prototipo fortemente innovativo, che si è rivelato un componente fondamentale per sfruttare appieno le qualità dei sistemi RFID applicati alla logistica, e che mostra le potenzialità di innovazione della tecnologia RFID nel campo del Knowledge Management. I futuri sviluppi di queste tecnologie dipenderanno anche dall'adesione di tutti gli attori coinvolti nella catena logistica alla filosofia dell'EPC Network, ovvero un'architettura capace di arricchire di valore aggiunto le informazioni nella misura in cui queste sono condivise.

Bibliografia

- P. H. Cole, M. Harrison, D. C. Ranasinghe, (2008), "EPC Network Architecture", Networked RFID systems and lightweight cryptography, Springer, pp. 59-78.
- EPCglobal Inc, (2004), "The EPCglobal Network - Overview of Design, Benefits, & Security", white paper, EPCglobal Inc, (http://www.gs1nz.org/documents/TheEPCglobalNetworkfromepcglobTheEP_001.pdf).
- D. Brock, (2001), "The Electronic Product Code (EPC) - A Naming Scheme for Physical Objects", white paper, Auto-ID Labs (AUTO ID-WH-002).
- F. Thiesse, C. Floerkemeier, M. Harrison, F. Michaelles, C. Roduner, (2009), "Technology, Standards, and Real-World Deployments of the EPC Network", IEEE Internet Computing, vol. 13(2), pp. 36-43.
- S. Sarma, (2005), "A History of the EPC, RFID: Applications, Security, and Privacy", S. Garfinkel and B. Rosenberg, eds., Addison-Wesley, pp. 37-55.
- T. Erl, (2004), "Service-Oriented Architecture: Concepts, Technology, and Design", Prentice Hall.
- K. Domdouzis, B. Kumar, C. Anumba, (2007), "Radio-Frequency Identification (RFID) applications: A brief introduction", Advanced Engineering Informatics, vol. 21(4), pp. 350-355.
- H. Stockman, (1948), "Communication by Means of Reflected Power", Proceedings of the Institute of Radio Engineers, vol. 36(10), pp. 1196-

1204.

- C. Floerkemeier, C. Roduner, M. Lampe, (2007), "RFID Application Development With the Accada Middleware Platform", IEEE Systems Journal, vol. 1(2), pp. 82-94.
- EPCglobal, (2007), "EPC information services (EPCIS) version 1.0.1 specification", (www.epcglobalinc.org).
- F. Armenio et al., (2007), "The EPCglobal Architecture Framework", EPCglobal Inc., (http://www.epcglobalinc.org/standards/architecture/architecture_1_2-framework-20070910.pdf).
- C. Kurschner, C. Condea, O. Kasten, F. Thiesse, (2008), "Discovery Service Design in the EPCglobal Network: Towards Full Supply Chain Visibility", proceedings, The Internet of Things 1st International Conf.. Zurich, Switzerland.
- "The EPCglobal Network: Enhancing the Supply Chain", Verisign, Whitepaper, (<http://www.verisign.com/stellent/groups/public/documents/whitepaper/002109.pdf>).
- J.J. Cantero et al., (2008), "Traceability applications based on Discovery Services", IEEE International Conf. Emerging Technologies and Factory Automation, pp. 1332-1337.
- S. Beier, T. Grandison, K. Kailing, R. Rantza, (2006), "Discovery Services - Enabling RFID Traceability in EPCglobal Networks", Computer Society of India.
- H. P. Luhn, (1958), "A Business Intelligence System", IBM Journal.
- D. J. Power, (2007), "A Brief History of Decision Support Systems", version 4.0, DSSResources.COM, (<http://dssresources.com/history/dsshistory.html>).
- Barry A. Devlin, Paul T. Murphy, (1988), "An Architecture for a Business and Information System", IBM Systems Journal, 27(1), pp. 60-80.
- W. Eckerson, (2006), "Performance Dashboards: Measuring, Monitoring, and Managing Your Business", John Wiley & Sons.

- S. Few, (2006), "Information Dashboard Design", O'Reilly.
- Network Working Group of the IETF, (2006), "RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1".
- Network Working Group of the IETF, (2006), "RFC 4252, The Secure Shell (SSH) Authentication Protocol".
- N. Ferguson, B. Schneier, (2003), "Practical Cryptography", Wiley.
- Network Working Group of the IETF, (2006), "RFC 4250, The Secure Shell (SSH) Protocol Assigned Numbers".
- J. Lerner, J. Tirole, (2005), "The Economics of Technology Sharing: Open Source and Beyond", Journal of Economic Perspectives, vol. 19(2), pp. 99-120.
- Network Working Group of the IETF, (1993), "RFC 1421, Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures".
- R. Rivest, A. Shamir, L. Adleman, (1978), "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, vol. 21 (2), pp. 120–126.
- Network Working Group of the IETF, (2002), "RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks".
- M. Subramanian, V Krishnamurthy, (1999), "Performance Challenges in Object-Relational DBMSs", IEEE Data Eng. Bull., vol. 22(2), pp. 27-31.
- W. Iverson, (2004), "Hibernate: A J2EE Developer's Guide", Addison Wesley Professional.
- A. Behm , A. Geppert , K. R. Dittrich, (1997), "On the migration of relational schemas and data to object-oriented database systems", proceedings 5th International Conference on Technologies for Information Systems.
- M. Fowler, (2003), "Patterns of Enterprise Application Architecture", Addison-Wesley, pp. 200-214.
- R. Schumacher, A. Lentz, (2005), "Dispelling the Myths", MySQL AB,

(<http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>).

- Sun Microsystems, (2008), “Sun Microsystems Announces Completion of MySQL Acquisition; Paves Way for Secure, Open Source Platform to Power the Network Economy”, Sun Microsystems Press release.
- Z. Urlocker, (2005), “Google Runs MySQL”, TheOpenForce, (http://zurlocker.typepad.com/theopenforce/2005/12/googles_use_of_.html).
- M. Callaghan, (2009), “MySQL at Facebook”, O’Reilly.
- A. Reuter, T. Haerder, (1983), “Principles of Transaction-Oriented Database Recovery”, ACM Computing Surveys, vol. 15(4), pp. 287–317.
- R. Johnson, (2005), “J2EE development frameworks”, Computer , vol.38(1), pp. 107- 110.
- A. Kutschera, B. Spillmann, (2009), “Enterprise GWT: Combining Google Web Toolkit, Spring and Other Features to Build Enterprise Applications”, White paper, (<http://www.maxant.co.uk/whitepapers.jsp>).
- E. F. Codd, (1990), “The Relational Model for Database Management”, Second edition, Addison Wesley.
- E. Cecchet, J. Marguerite, W. Zwaenepoel, (2002), “Performance and scalability of EJB applications”, SIGPLAN Not. 37, 11, pp. 246-261.
- B.J. Cox, (1985), “Object oriented programming”, Addison-Wesley.
- ECR Europe, (2003), “Optimal Shelf Availability – Increasing Shopper Satisfaction at the Moment of Truth”, ECR Europe Publications, (<http://www.ecrnet.org>).
- K. Pramatarı, P. Miliotis, (2008), “The Impact of Collaborative Store Ordering on Shelf Availability”, Supply chain management: an international journal, vol. 13(1), pp. 49-61.