



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea Magistrale in Ingegneria Informatica

Stealth 3000,

Un gestionale per la filiera della moda:

**Implementazione della gestione degli ordini
intercompany**

Relatore: Prof. Moreno Muffatto

Laureanda: Christelle Kamdem Kenmogne

Matricola: 621882

Materia: Gestione strategica delle organizzazioni

Anno accademico: 2011/2012

*A mio papa e mia mamma che non vedo da sei anni!
A mio marito!*

*Un ringraziamento al mio relatore Professore
Moreno Muffatto per la passione che ha saputo
trasmettermi durante il corso di gestione strategica
delle organizzazioni e la sua disponibilità durante tutto
questo lavoro di tesi.*

*Un ringraziamento alla CSC Srl Italia che mi ha dato
la possibilità di questa grande esperienza e ai
colleghi.*

Introduzione

L'industria della moda europea (Italiana, francese e spagnola in primis) è conosciuta in tutto il mondo per la creatività, l'innovazione e lo stile. Grandi firme come Gucci, Versace, LVMH, Armani e Prada hanno estimatori in tutto il mondo, e marchi come Diesel, Zara, Etam e Benetton sono sempre più sinonimi di qualità e convenienza.

Nonostante la notorietà e il prestigio ottenuto a livello mondiale, negli ultimi anni il settore ha attraversato momenti di forte criticità e, dopo il periodo di grande espansione degli anni novanta caratterizzato dal lancio continuo di nuovi marchi e tassi di crescita sostenuti, oggi vede nell'aumentare dell'efficienza produttiva e nell'abbattimento dei costi le proprie priorità strategiche.

La sfida fondamentale, dunque, è rappresentata dalla capacità di tenere sotto costante controllo le singole fasi del processo manifatturiero, della scelta delle materie prime fino alla distribuzione dei prodotti finiti, per riuscire così a garantire, in tutte le fasi della produzione, la gestione ottimale di ogni singolo step.

Sottoposti alla pressione delle diverse forze in gioco, le aziende, hanno risposto con attività mirate, quali l'acquisizione e la gestione diretta dei punti vendita distribuiti sul territorio, finalizzate soprattutto alla possibilità di avere il controllo immediato e puntuale dell'intera catena distributiva. In tal modo, infatti, queste riescono ad avere una conoscenza molto più precisa e grado di modellare in modo puntuale la propria offerta. Se è vero, poi, che le grandi firme tendono ancora a mantenere la produzione a livello locale per sfruttare al massimo il valore insito nel marchio "made in", è anche vero che molte altre aziende stanno percorrendo, con sempre maggiore frequenza, la strada della delocalizzazione produttiva nei paesi a basso costo dell'Europa orientale o dell'Estremo oriente, aumentando, di fatto, la complessità e l'estensione geografica della propria catena del valore.

In tal contesto, all'Information Technology viene chiesto di trovare soluzioni che permettano di relazionare i dati di vendita dispersi sul territorio con processi a monte della supply chain. La disponibilità in tempo reale di informazioni puntuali, di anticipare le richieste (domande) e pianificare in modo efficace l'intero ciclo produttivo (offerta).

Questa possibilità è ancor più importante in un settore, quale quello del Fashion & Retail, che è in costante evoluzione e che, nell'arco di pochissimo tempo, è passato dalla produzione basata sulle tradizionali collezioni primavera/estate e autunno/inverno ad un'offerta che si declina in quattro e, a volte, cinque collezioni annue. Contestualmente, se nel passato le attività di approvvigionamento erano delegate essenzialmente all'esperienza e alla conoscenza personale dei buyers, oggi l'IT diviene uno dei principali fattori abilitanti per la gestione dei processi interni e dei flussi di informazioni, che spesso si esplicano su basi transnazionali. Le soluzioni, dunque, non possono che essere web oriented, al fine di consentire alle aziende di mantenere un controllo centrale sulla distribuzione, pur garantendo alle varie filiali distribuite sul territorio l'accesso diretto al sistema.

In un contesto di mercato complesso e dinamico, le aziende hanno bisogno non tanto di un prodotto quanto di soluzioni complete, in grado di rispondere in maniera rapida ed efficace ai propri bisogni,

basate su prodotti pensati e sviluppati ad hoc, e su un patrimonio di esperienze e competenze specifiche del settore. Le aziende del Fashion & Retail non possono più accontentarsi del vecchio paradigma, proprio dei primi sistemi ERP proprietari sorti a cavallo degli anni ottanta e novanta, ma devono, piuttosto, guardare a soluzioni aperte in grado di soddisfare la crescente necessità di lavorare in team distribuiti.

Le principali difficoltà con cui queste dovranno misurarsi, infatti, sono correlate soprattutto alla crescita esponenziale della dimensione delle stesse che, per avere successo, dovranno saper coniugare efficacemente la capacità di adeguarsi alle dinamiche del nuovo scenario economico mondiale (apertura di nuovi mercati e off-shore produttivo in primis) con la difesa del valore del proprio brand e del “made in”, che imprescindibilmente sono ancora sinonimo di qualità e stile.

In questa tesi s'intende esporre il più possibile nel dettaglio, nel rispetto delle disposizioni vigenti in merito del segreto aziendale stipulate tra l'Ateneo e le Aziende convenzionate (Articolo 3 comma 1, Convenzione di tirocinio di formazione ed orientamento), le attività che ho svolto durante il mio tirocinio semestrale presso CSC s.r.l Italia, leader globale nella fornitura di soluzioni e servizi IT innovativi nelle aree Business Solutions & Services e Global Outsourcing Services, attraverso le tre linee di servizio Consulting, Systems Integration e Outsourcing, e più in particolare nella Business Unit Fashion & Retail con sede centrale a Padova che elabora soluzioni e servizi per l'industria del settore della moda per conto di grandi aziende operanti sul mercato internazionale.

Nel primo capitolo presenteremo la realtà aziendale presso cui abbiamo trascorso sei mesi di tirocinio, i suoi principali progetti, partner, e Stealth 3000 con l'obiettivo di evidenziare il contesto in cui provengono tutte le esperienze che sono infine confluite nella realizzazione di questo scritto. Nel capitolo successivo, si illustreremo la tecnologia e gli strumenti per lo sviluppo di Stealth 3000. Nel terzo capitolo presenterò il progetto “Gestion” del gruppo “Fashion” e la sua struttura, cliente della CSC s.r.l. con una realtà aziendale abbastanza complessa per la sua espansione geografica, la varietà di applicativi che comporta il sistema informativo e infine la diversità di prodotti offerti. Nel capitolo successivo, parleremo di Stealth 3000 all'interno di Gestion, la sua installazione e l'interazione con altri applicativi. Dopo di che, nel capitolo cinque, affronteremo la ristrutturazione societaria del gruppo, le sue implicazioni e le sue conseguenze nella generazione di codice a barre per gli articoli del gruppo, offrendo una soluzione innovativa per la nuova gestione di quest'ultimi. Infine nel capitolo sesto, illustreremo la gestione degli ordini prima della incorporazione aziendale, e l'implementazione del nuovo flusso per gli ordini intercompany. La nostra analisi e implementazione vuole mettere insieme efficacia, integrabilità e qualità nel soddisfare le esigenze del cliente, dando una soluzione adatta al suo business, in continuo evoluzione. Per questi motivi, in conclusione faremo delle valutazioni e daremo qualche proposta per una migliore integrazione.

Indice

Introduzione	4
Capitolo 1: Presentazione dell'azienda CSC.....	9
1.1 CSC.....	9
1.2 Consulting.....	9
1.2.1 In Italia	10
1.3 Systems Integration.....	11
1.3.1 Approccio	11
1.4 Outsourcing	12
1.4.1 Costruire collaborazioni di lunga durata: i principi del Dynamic Sourcing.....	12
1.4.2 Gestire ambienti complessi multi-sourcing	12
1.5 Fashion & Retail.....	13
1.5.1 Stealth.....	13
1.5.2 Retail PRO	14
Capitolo 2: Stealth 3000	15
2.1 La codifica degli oggetti	15
2.2 La codifica dei soggetti	18
2.3 Industrializzazione prodotto.....	21
2.4 Ciclo attivo	21
2.5 Ordini clienti	22
2.6 Assegnazioni	22
2.7 Spedizioni	23
2.8 Statistiche.....	24
2.9 Provvigioni.....	24
2.10 Ciclo passivo	24
Capitolo 3: STEALTH 3000, la tecnologia, l'architettura e l'ambiente applicativo	27
3.1 Oracle Application Server 10g (OAS)	28
3.1.1 Principali componenti	28
3.2 Database Oracle 10g.....	29
3.2.1 Principali componenti	30
3.2.2 Analisi del processo di esecuzione di un istruzione SQL	32
3.3 Oracle PL/SQL Developer, PL/SQL	32
3.3.1 Struttura dei blocchi PL/SQL	33
3.3.2 Elementi di linguaggio.....	35
3.3.3 Gestione Eccezioni	37
3.3.4 Procedure e Funzioni	38
3.3.5 Packages.....	39
3.4 Oracle Forms Builder	39
3.5 Oracle Report Builder	41
3.6 Oracle JDeveloper 11g, ADF	43
3.6.1 Architettura ADF	43
3.6.2 Vantaggi di Oracle ADF Negli altri frame work.....	45
3.7 Strumenti di gestione & controllo e delivery.....	47

3.7.1 BPEL	47
3.7.2 Zipper-SmartDeliverer	47
3.7.3 PVCS	47
3.8 Personalizzazione	49
3.9 Stealth nel Contesto del SI Aziendale	50
Capitolo 4: Il Progetto GESTION del Gruppo FASHION	53
4.1 Architettura di GESTION	54
4.2 Installazione Stealth 3000 per FASHION.....	56
4.3 Stealth 3000 Multisocietà.....	58
Capitolo 5: Ristrutturazione societaria del Gruppo	59
5.1 Codice a barra.....	59
5.1.1 Checkpoint Systems Spa Open EPC Number.....	59
5.1.2 GESTION24.....	60
5.1.3 GTIN Global Trade Item Number	60
5.2 Flusso Ordini Intercompany	64
5.2.1 La struttura dell'ordine	65
5.2.2 Testata ordine, S3T_OCT.....	65
5.2.3 Riga ordine, S3T_OCR	68
5.2.4 Conferma ordine.....	69
5.2.5 Provvigione agente su riga.....	70
5.2.6 Avanzamento della Quantità per riga.....	70
5.2.7 Abbinamento righe	71
5.2.8 Storni su righe	71
5.3 Configurazione di Flussi Intercompany in Stealth	72
Capitolo 6: Nuovi flussi di ordini intercompany per FASHION	75
6.1 Definizioni.....	75
6.2 Descrizione dei processi introdotti	76
6.3 Punti d'intervento	77
6.4 Presentazione dei nuovi flussi	78
6.5 Descrizione dei dettagli tecnici.....	79
6.5.1 Modifiche alla struttura dati	79
6.5.2 Personalizzazione dei packages di gestione dei flussi intercompany	81
6.5.3 Integrazione con funzionalità esistenti	85
6.5.4 Nuovi flussi intercompany: codifica	86
Conclusione	93
Valutazione.....	93
Difficoltà incontrate	94
Proposte future	94
Bibliografia	95

Capitolo 1: Presentazione dell'azienda CSC

1.1 CSC

CSC è leader globale nella fornitura di soluzioni e servizi IT innovativi nelle aree Business Solutions & Services e Global Outsourcing Services, attraverso le tre linee di servizio Consulting, Systems Integration e Outsourcing.

Con headquarters a Falls Church (Virginia, USA), CSC si avvale della collaborazione di 94.000 professionisti che operano in oltre 90 Paesi. Ha realizzato ricavi per \$16,1 miliardi nel corso dell'anno fiscale chiuso ad aprile 2010. CSC ha contribuito sin dalla sua nascita all'evoluzione dell'IT e nel corso degli anni ha saputo affrontare le sfide e le opportunità presentate dai nuovi mercati e dalle nuove tecnologie, accompagnando i clienti lungo il cammino dell'innovazione. La storia inizia nel 1959, anno in cui viene fondata da Roy Nutt e Fletcher Jones, con un investimento iniziale di 100 dollari. Dieci anni dopo, nel 1968, CSC è la prima società di software ad essere quotata al New York Stock Exchange (NYSE).

Negli anni '70 la Systems Integration diventa il settore strategico in cui espandersi, anche tramite importanti acquisizioni che assicurano a CSC una significativa quota di mercato negli Usa, in Europa e in Australia, diventando protagonista a livello internazionale. Dopo cinquant'anni d'ininterrotta attività, CSC oggi è riconosciuta sul mercato per l'esperienza acquisita e l'affidabilità delle soluzioni e dei servizi offerti a clienti di tutto il mondo in 15 diversi settori merceologici. La focalizzazione per industry, unita ad un modello di delivery strutturato per linee di servizio orizzontali e ad un'organizzazione in grado di valorizzare le peculiarità delle singole country, rappresentano gli asset che consentono a CSC di garantire ad ogni cliente la soluzione più adeguata alle proprie esigenze (Best Total SolutionSM).

Tra i suoi maggiori Clienti, in Italia CSC è impegnata a generare valore per: Banca Intesa Sanpaolo, BNL, Reale Mutua, Generali, SOGEI, INPS, INAIL, Ministero dell'Interno, Ministero della Difesa, Diesel, Versace, Armani, Gucci, Telecom Italia, Poste Italiane, Enel, ENI, Vodafone. Da febbraio 2007, CSC fa parte della South & West Region insieme a Francia, Belgio, Lussemburgo, Spagna e Portogallo. CSC Italia garantisce ai propri clienti un approccio e una value proposition coordinati a livello di Region, in grado di combinare competenze ed esperienze sui processi di business con le tecnologie più innovative.

1.2 Consulting

CSC integra abilmente l'expertise acquisito in vari settori di mercato con la profonda competenza tecnica e tecnologica e la disponibilità di una vasta gamma di servizi differenziati, così da garantire i propri clienti una consulenza di business di altissimo livello. L'esperienza maturata a livello internazionale nella riorganizzazione e implementazione dei processi di business è alla base dei nostri servizi consulenziali, che vanno dall'interpretazione dell'idea o del progetto di business del cliente, alla risoluzione di problematiche insite nel processo organizzativo esistente, tenendo conto di tre dimensioni fondamentali come il budget, le persone e i processi. Grazie a un approccio innovativo e collaborativo e a una comprovata esperienza nella gestione dell'IT, CSC è in grado di valutare lo "stato di salute" dell'organizzazione aziendale e, sulla base delle best practice consolidate, interviene combinando l'esperienza strategica, tecnologica, operativa e organizzativa con approcci coordinati e testati, capaci di garantire al cliente risultati e competitività nel mercato di riferimento.

Nello specifico, tutti i servizi di consulenza CSC, siano essi in ambito di business consulting, di systems integration o di outsourcing, sono guidati dalla metodologia **CSC Catalyst**®. Catalyst consente infatti di coniugare il rigore metodologico con la flessibilità richiesta in tutte le situazioni: non si limita, quindi, alle fasi di progettazione dei prodotti e dei servizi, ma è in grado di supportare anche tutte le importanti fasi di manutenzione, erogazione e monitoraggio. La consulenza CSC, durante la fase di delivery dei servizi, opera anche attraverso una rete di **Competence Center**, strutture di eccellenza diffuse a livello mondiale, che operano in stretta collaborazione con le strutture commerciali di front-end e offrono supporto alla struttura di delivery locale.

1.2.1 In Italia

La nostra consulenziale in Italia è focalizzata su quattro principali linee di servizio:

- **Business Process Management**

La necessità di organizzare, governare e adattare sistemi organizzativi aziendali sempre più complessi e dinamici impone l'adozione di una visione strategica per processi che approcci olisticamente la gestione del patrimonio IT, delle risorse umane e delle specificità di business di ciascuna azienda.

Il Business Process Management (BPM) è un insieme di metodologie e tecnologie che consentono di analizzare, ridisegnare e rapidamente implementare nuovi e più efficienti processi di business.

Il BPM permette la progettazione di processi direttamente eseguibili e ne supporta un'evoluzione incrementale con impatti minimi sull'operatività corrente.

CSC è stata pioniera nella concettualizzazione ed evoluzione delle metodologie e degli standard del BPM e vanta oggi una lunga e consolidata esperienza nel supporto a organizzazioni medie e grandi che desiderano adottare con successo questa innovativa metodologia.

- **Program / Change Management**

Le aziende, nel contesto economico attuale, sono chiamate ad affrontare sfide sempre più complesse: la riduzione crescente del ciclo di vita dei prodotti, la centralità delle esigenze del Cliente o ancora, l'organizzazione flessibile della produzione, sono obiettivi imprescindibili per un'azienda che voglia mantenere un elevato grado di competitività sul mercato. In questo scenario, la tecnologia diventa il fattore abilitante per il raggiungimento degli obiettivi, ma il vero valore aggiunto sta nel definire come utilizzarla.

Questo è possibile sia attraverso la corretta programmazione e gestione degli step progettuali e l'integrazione tra le diverse fasi di progetto (*Program Management*), sia tramite l'analisi dei flussi informativi aziendali, del modello organizzativo e, soprattutto, attraverso l'adozione di opportune azioni che garantiscano il coinvolgimento di tutti gli attori necessari, affinché i cambiamenti originati vengano accettati a tutti i livelli organizzativi. (*Change Management*).

CSC vanta un'esperienza consolidata in ambito Program & Change Management, grazie alla quale favorisce il successo dei progetti IT dei clienti, supportando concretamente gli stessi nel programmare e raggiungere efficacemente gli obiettivi prefissi.

- **Business Continuity & Disaster Recovery**

I servizi informativi sono vitali per tutte le aziende, che ne dipendono fortemente per l'articolazione dei processi interni, produttivi e amministrativi, nonché per la gestione delle relazioni con l'esterno.

Le informazioni devono essere adeguatamente protette ed è indispensabile mantenere costante e operativo il livello di servizio offerto. I temi legati alla *Business Continuity* (continuità del servizio) e al *Disaster Recovery* (recupero dell'operatività a fronte di un evento catastrofico), pur esprimendo due concetti diversi, sono strettamente legati tra loro.

CSC ha realizzato efficaci piani operativi di BC&DR, tesi a permettere di gestire ogni emergenza e garantire la presentazione giornaliera delle offerte sul mercato, prevenire e ridurre al minimo le interruzioni di servizio, prevenire e ridurre al minimo le perdite finanziarie e di immagine, assicurare il pronto ripristino dei servizi operativi.

- **IT Strategy & Governance**

In uno scenario di mercato caratterizzato da instabilità e complessità, gli ingredienti principali per differenziarsi e creare vantaggio competitivo sono flessibilità, ricerca e innovazione, controllo dei costi. Il legame tra processi di business e IT, con le relative implicazioni, è un fattore decisivo per conseguire concreti innalzamenti della competitività aziendale.

In questo contesto, l'IT non è più chiamato a fornire una mera strategia tecnologica ma deve integrare capacità strategiche in ambito organizzativo, di sourcing, governance e comunicazione. La formulazione di un piano strategico dell'IT e il governo delle decisioni connesse, con la conseguente attribuzione dei poteri decisionali e delle responsabilità tra i diversi attori interni ed esterni all'IT stesso, devono garantire l'allineamento con il business. Per la formulazione delle strategie IT, CSC ha sviluppato la metodologia *Fusion* che, "fondendo" tecnologia e business, assicura l'allineamento "top down" tra le iniziative IT e le strategie di business, preservando la sincronizzazione dell'IT con l'operatività aziendale ordinaria. Completano la nostra offerta strumenti e soluzioni come *Balanced Scorecard Process*®, per supportare la governance dell'IT, e *Smart Spend*®, per la razionalizzazione dei costi IT.

1.3 Systems Integration

La nostra esperienza in materia di Systems Integration ci consente di implementare e integrare tecnologie collaudate e allo stato dell'arte per realizzare gli obiettivi aziendali desiderati dai clienti.

1.3.1 Approccio

Il nostro approccio al cambiamento aziendale e all'innovazione tecnologica è di tipo olistico. Ci dedichiamo innanzi tutto all'ascolto e alla comprensione delle specifiche esigenze del cliente, per poi disegnare, costruire e rendere operativi i complessi sistemi necessari ad ottenere il successo sul mercato. Grazie a un'efficace combinazione tra esperienza tecnica e gestionale e una profonda conoscenza dell'industria, collaboriamo con i team aziendali responsabili sia del business che della tecnologia per l'implementazione di sistemi e applicazioni che consentano al cliente di ottenere i risultati desiderati.

1.4 Outsourcing

1.4.1 Costruire collaborazioni di lunga durata: i principi del Dynamic Sourcing

Il Dynamic Sourcing di CSC soddisfa le sfide emergenti del business: sfruttare il cambiamento, trarre benefici dall'innovazione, costruire modelli di business collaborativi e ottenere risultati misurabili. Le esigenze e le priorità di business cambiano più velocemente che mai. Anche la tecnologia si evolve con rapidità, e molti contratti devono prendere in considerazione eventuali nuove caratteristiche e possibilità, dato che l'allineamento tra strategia di business e indirizzo tecnologico è un fattore cruciale.

1.4.2 Gestire ambienti complessi multi-sourcing

I clienti possono scegliere tra un'ampia offerta di soluzioni infrastrutturali, desktop, applicazioni, reti e gestione dei processi operativi, con opzioni che vanno da una risorsa unica a una catena del valore complessa e multi-vendor. Tali modelli multi-partner implicano una serie di questioni da affrontare. Come distribuire i processi end-to-end tra una miriade di fornitori? Come suddividere le responsabilità e attribuire obblighi ed eventuali inconvenienti? A chi, nel caso? Come ottimizzare il ruolo di ciascun partner, definendo in modo appropriato ogni attività e, cosa ancor più problematica, come cambiare ruoli e obiettivi?

L'approccio Dynamic Sourcing di CSC è basato su cinque punti fondamentali.

- **Focus sui Risultati di Business:**

Gli obiettivi tecnici e le misure all'interno di contratti di outsourcing stanno diventando sempre più secondari rispetto agli obiettivi di business che una strategia di sourcing complessiva punta portare a termine o a supportare.

- **Creare un Collaborative Sourcing Model:**

Il Dynamic Sourcing fa sì che ogni operazione di sourcing venga progettata per adattarsi a un modello di sourcing collaborativo, così da essere ottimale nel proprio campo.

- **Allineare i Propositi di Cliente e Fornitore:**

C'è un riconoscimento esplicito che molti dei presupposti iniziali della transazione cambieranno, come i livelli di performance richiesti e le aspettative economiche.

- **Costruire Meccanismi Adatti a una Trasformazione Sostenibile:**

L'approccio Dynamic Sourcing di CSC è costruito per sostenere anche ripetute trasformazioni lungo tutto il corso della relazione di outsourcing.

- **Assicurare un Perfetto Allineamento dei Sistemi di Governance:**

Il Dynamic Sourcing presuppone che sin dall'inizio ci sia un ambiente di multi-partner con servizi end-to-end, e processi di gestione e integrazione. Il Dynamic Sourcing fonde tutto ciò in un modello operativo innovativo e con un approccio radicalmente differente di governance.

1.5 Fashion & Retail

Oggi l'industria della moda si trova a competere in uno scenario socio-economico in rapida e costante trasformazione. La domanda, sempre più attenta ed esigente, ha portato i grandi marchi a ripensare i propri modelli produttivi, spingendoli a passare dalle tradizionali collezioni Primavera/Estate e Autunno/Inverno ad una estensione dell'offerta fino a quattro o cinque collezioni annue. Inoltre, il contesto di mercato altamente competitivo, ha determinato la necessità costante di controllare i costi, ottimizzando i processi produttivi lungo tutta la catena del valore.

In questo scenario, il successo delle imprese che operano in questo settore è strettamente legato alla disponibilità in tempo reale di informazioni puntuali e integrabili, che permettano alle aziende di anticipare le richieste (domanda) e pianificare in modo efficace l'intero ciclo produttivo (offerta).

Da oltre 20 anni, la Business Unit Fashion & Retail è un asset portante di CSC, rispondendo con efficacia ai bisogni di oltre 250 clienti del settore, è orgogliosa di annoverare tra i suoi clienti le principali case di moda internazionali. Ciò le ha permesso di osservare da vicino l'evoluzione del mercato e di anticiparne le necessità, recependo appieno le nuove tendenze sin dalla loro prima manifestazione e adeguando i propri servizi alle mutate dinamiche di mercato. Con sede direttiva a Padova, la BU Fashion & Retail fornisce consulenza e supporto in tutto il mondo alle principali case di moda europee ed internazionali. L'offerta CSC per il Fashion comprende servizi e soluzioni, proprie o di partner, centrate sulla suite Stealth, per le aree:

- product lifecycle management
- product definition
- procurement
- production planning and scheduling
- customer order fulfillment
- accounting
- data mining
- warehouse management
- sales force automation
- PoS management

L'ampia gamma di servizi spazia dal Change Management all'Outsourcing completo delle attività IT.

1.5.1 Stealth

Stealth è il **sistema informativo integrato** pensato e realizzato appositamente per il mondo del Fashion, in particolare per le aziende che si trovano a dover affrontare le problematiche di taglia/colore e stagionalità. **Stealth** è in grado di gestire i processi relativi alla *supply chain* e al

sales & distribution. E' dotato di un'**architettura applicativa** totalmente **integrata** e altamente modulare, interfacciabile con prodotti che gestiscono l'area Retail e con le più diffuse soluzioni di gestione contabile e amministrativa. Disponibile su piattaforma i5/OS, con hardware IBM i5 (database IBM DB2) o su piattaforma Linux, Unix, Windows con hardware IBM, HP, SUN (database ORACLE), è **una soluzione**:

- **web enabled**
- **multisocietaria**
- **multivaluta**
- **multilingua**

Stealth si colloca in una posizione di assoluta **eccellenza** per efficacia e produttività nel contesto dell'offerta applicativa specifica per le aziende del Fashion Italia e nel mondo.

I numeri di Stealth:

- oltre **250** clienti
- oltre il **60%** di market-share in Italia

1.5.2 Retail PRO

Quale fornitore globale di soluzioni Retail per l'industria del Fashion, CSC distribuisce e installa in tutto il mondo Retail Pro, una soluzione applicativa per la gestione dei punti vendita, attualmente impiegato in oltre 25.000 negozi in 60 Paesi Retail Pro consente di automatizzare tutti gli aspetti del POS: il controllo dell'inventario, la gestione dei rapporti con la clientela, le comunicazioni tra molteplici negozi, l'e-commerce e l'analisi della vendita al dettaglio. Con oltre 30 progetti internazionali realizzati in quest'ambito, CSC ha sviluppato un'offerta specifica di indiscussa qualità sia per i servizi di assistenza e consulenza, sia per la capacità di gestione in outsourcing.

1.5.3 Logimoda

E' una soluzione dipartimentale che risponde in modo completo alle esigenze legate alla gestione dei magazzini di materie prime e di prodotti finiti, in termini di movimentazione fisica della merce e del personale di magazzino, grazie ad una grande flessibilità derivante dall'architettura parametrica del pacchetto. La soluzione, che supporta immagazzinamento per coordinate, prelievo e carico, ottimizzazione spazi, processo con RFID, etc., soddisfa pienamente le esigenze legate alla gestione per box tecnico e box cliente, e consente di mantenere la tracciabilità del prodotto e degli eventi a esso legati. Logimoda permette di ottimizzare la gestione fisica dei magazzini, sia nell'ottica dei percorsi del personale, sia in quella della movimentazione guidata da macchine (PLC).

Inoltre CSC offre soluzioni specifiche funzionali a:

- Gestione via web degli ordini di riassortimento (stock, ATP) con visione completa del loro ciclo di vita, consentendo ai clienti una visione dettagliata dello stato di avanzamento degli ordini.
- Pianificazione della domanda.
- Gestione in remoto dell'avanzamento delle commesse da parte di terzisti, consentendo l'interazione diretta con il sistema informativo aziendale.
- Raccolta remota degli ordini cliente tramite agenti o che svolgono attività di show room.

Capitolo 2: Stealth 3000

Progettato per aziende che operano nei seguenti settori dell'abbigliamento, delle calzature di prodotti in pelle, di calze accessori e, in generale, per tutte quelle aziende che trattano prodotti caratterizzati dalla problematica della taglia/colore. E' un prodotto completamente nuovo, che abbina soluzioni tecnologiche ed architettrurali all'avanguardia con la solidità di un impianto funzionale ricco delle esperienze accumulate nelle più di 150 aziende clienti di Stealth_400. Stealth 3000 gestisce i processi aziendali di: Codifica degli Oggetti, Codifica Soggetti, Industrializzazione Prodotto, Ciclo attivo, Ciclo passivo, Magazzini, Costi. Stealth, sia nell'attuale versione su AS/400 che nella nuova versione su piattaforma Oracle, non si presenta come un ERP ma si configura come una prodotto "best of breed", ovvero come una delle componenti del sistema informativo che, relativamente alle aree coperte, offre la migliore soluzione specifica. Ciò è frutto di una precisa strategia di CSC che ritiene che le specificità del settore dell'abbigliamento possano essere meglio coperte da una offerta che privilegi la verticalità della soluzione, piuttosto che un'integrazione orizzontale tipica delle soluzioni ERP. Stealth quindi si caratterizza come prodotto autosufficiente, che ha in sé tutte le componenti funzionali necessarie a gestire autonomamente i processi coperti.

2.1 La codifica degli oggetti

Il processo di codifica degli oggetti prende in considerazione la loro rappresentazione, le loro tipologie, la caratterizzazione e le abilitazioni stagionali. Gli oggetti sono qualunque entità codificabile ad esempi: i modelli, le materie prime, i prodotti finiti, oggetti abbinabili, i prodotti intermedi. La rappresentazione degli oggetti si ottiene con due diverse strutture di codici:

- **Modello**



Rappresentano lo "stile".

Possono avere taglie o misure.

Non hanno indicazioni di tessuti e colori.

Non possono rappresentare oggetti fisici.

- **Parte**

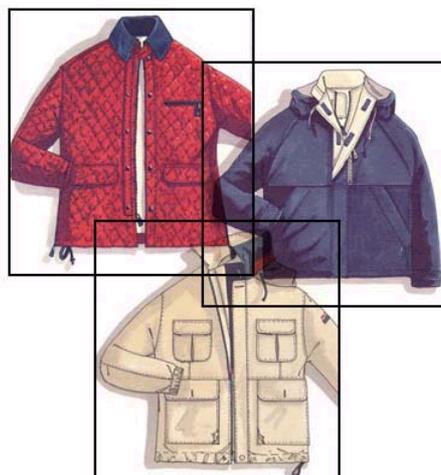


Rappresentano oggetti fisici (prodotti finiti o materiali).

Oggetti non fisici (aspetto, denominazione commerciale) e in tal caso si abbinano al modello andando a costituire l'oggetto (modello-parte).

Possano avere taglie o misure. Possano avere colore.

- **Modello/Parte**



Rappresentano oggetti fisici ottenuti come abbinamento di modelli a parti (prodotti finiti, intermedi o materiali).

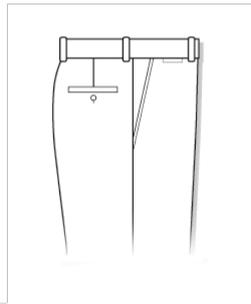
Ereditano le taglie o misure del modello. Possano avere colore.

- **Taglia & misura**

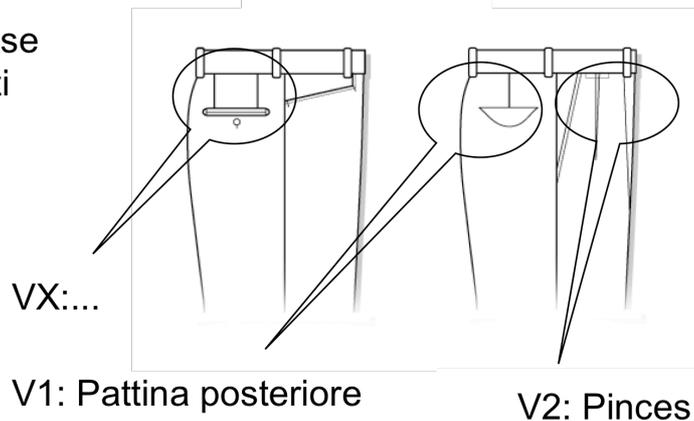
Le taglie sono entità predefinite, controllate ed esplicitamente abilitate all'oggetto. Le misure sono entità non predefinite, valide solo per l'oggetto a cui vengono attribuite. Le taglie sono organizzate in griglie. Ogni oggetto può avere 'n' griglie taglie abilitate, ognuna con 'n' taglie abilitate. Ogni taglia abilitata deve essere associata ad una fascia prezzo di vendita.

- **Variante**

Oggetto Base



Oggetto base
con Varianti



La variante permette di specificare delle variazioni fatte su un oggetto di base. Questo elemento favorisce una ampia codifica di oggetti minimizzando il numero di oggetti esistenti nel database. Sono delle Opzioni che differenziano l'Oggetto standard (es. Collo Pelliccia, Bottoni dorati, finito in fondo, ecc.). Sono utilizzabili per qualsiasi tipo di Modello, Formale e non Formale, vengono specificate in fase di gestione Ordine Cliente, Fornitore, Commessa. Possono essere di vario tipo:

- Implicite, quelle insite nel Modello
- Elementari, una singola Variante
- Cumulative, un insieme di Varianti.

Sono abilitati a vari livelli: Classe Modelli, Modelli, Modelli –Parte.

Oltre alla caratterizzazione anagrafica (Modello/parte/variante/colore/misura o griglia taglie e taglia) è possibile differenziare gli oggetti per: cartellino, etichetta, made in, stagione, bagno/partita, classe altezza, tipo lotto, drop, statura, stato, fornitore, matricola pezza, commessa, ordine fornitore.

- **Tipologie di Oggetti e loro caratterizzazione**

Prodotto Finito è un oggetto (Parte o Modello/Parte) caratterizzato dalla linea di vendita, a sua volta appartenente ad un Marchio e eventualmente una divisione produttiva o di acquisto. Il marchio e la Linea hanno rilevanza per il ciclo attivo, e la divisione produttiva ha rilevanza per il ciclo passivo. Ci sono due tipi di prodotto finito. Uno prodotto finito non formale che sono esplicitamente definiti nel sistema. Ciò significa che ogni occorrenza di prodotto valida deve essere definita nelle anagrafiche modello, parte e modello/parte. Secondo prodotto finito formale, per questo tipo non è richiesta la creazione dell'anagrafica Modello/Parte, ma questa viene dinamicamente verificata sulla base del meccanismo di abilitazione. Ciò significa che ogni occorrenza di prodotto valida deve essere definita nelle anagrafiche modello, parte, e Abilitazioni Prodotti.

Classi sono particolari tipi di modelli o di parti che servono a raggruppare caratteristiche comuni a più oggetti. Le classi sono distinte in varie tipologie in funzione del loro utilizzo. L'appartenenza di un oggetto ad una classe può essere stagionale o continuativa e l'utilizzo delle classi è ammesso solo per gli oggetti gestiti a Modello/Parte

Componenti, elementi abbinabili al modello oppure non. Ad esempio abbinabili al modello(non sono gestiti a taglie, solo codificato sulla Parte), gli aspetti finali del tessuto. Non abbinabile al modello(gestito come prodotto finale non formale), accessori, materiali intermedi.

- **Abilitazioni stagionali**, l'abilitazione stagionale di oggetto indica se l'abbinamento del modello alla parte ha valore continuativo. E l'abilitazione stagionale di colore indica invece se l'abilitazione dei colori all'oggetto (Parte o Modello/Parte) ha valore stagionale o continuativo. Il risulta quindi è la possibilità di gestire tutte le combinazioni di abilitazione colore rispetto alla stagionalità o meno dell'oggetto.

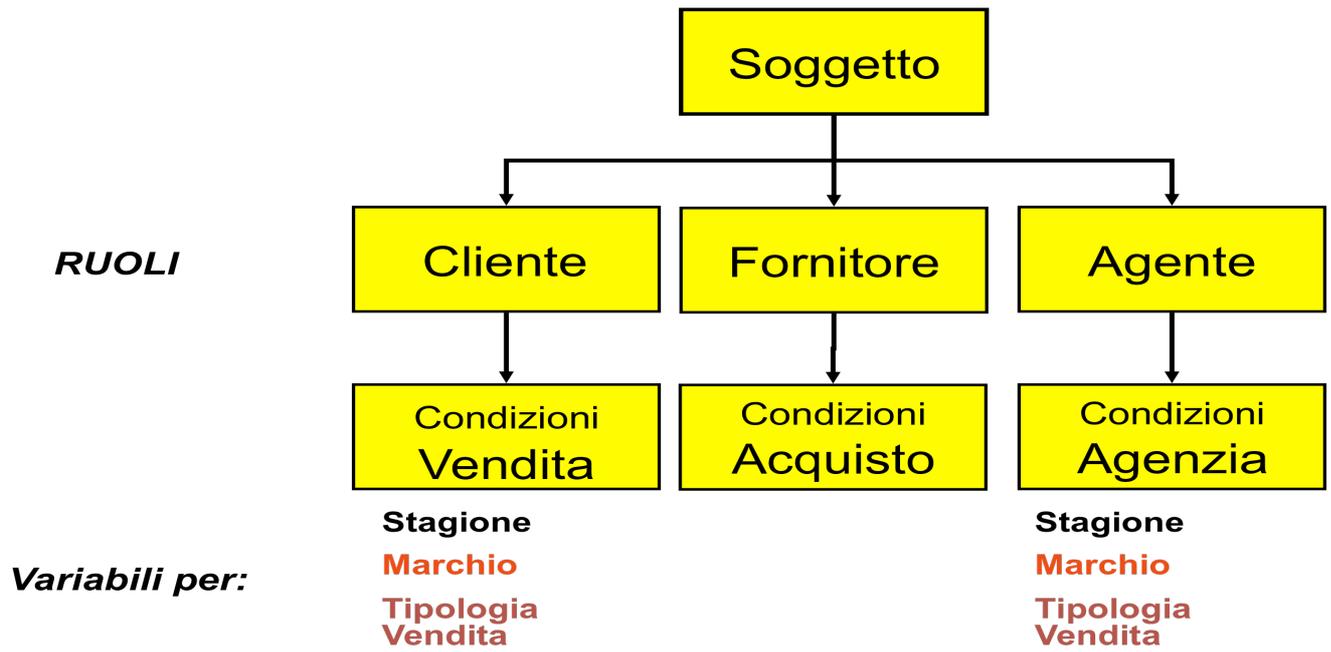
- **Eventi**

L'evento è una lista di prodotti. Viene abilitato stagionalmente, può o meno essere vincolato alle abilitazioni stagionali dei prodotti ad esso associati. l'Evento può essere associato ad un listino e possono essere abilitate classi di clienti.

2.2 La codifica dei soggetti

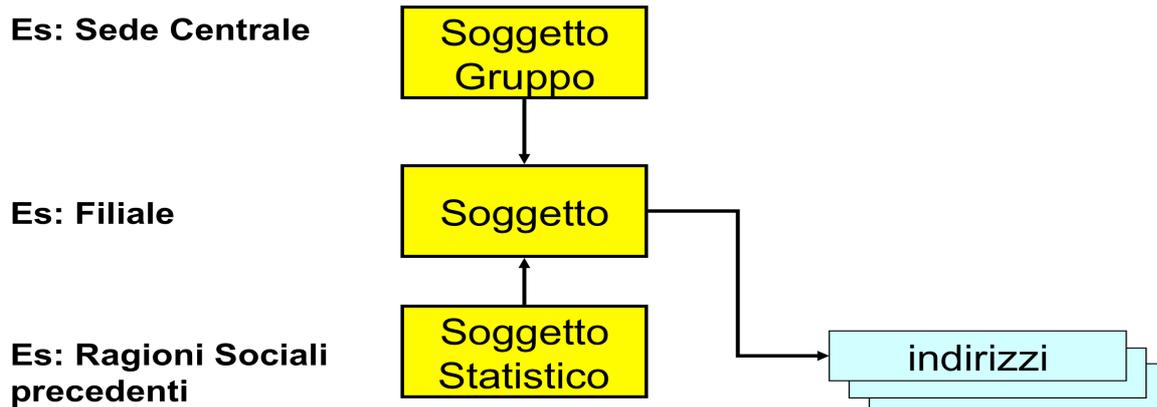
Un soggetto è una qualunque entità fisica, giuridica, gestionale, organizzativa con cui l'azienda intrattiene rapporti di business. Ad esempio i clienti, i fornitori, i reparti interni, gli agenti, i terzisti gli importatori ecc...Ad uno stesso soggetto possono corrispondere più tipologie di rapporti (ruoli) intrattenuti dall'azienda. La codifica può essere unica o differenziata per ruolo.

STRUTTURA LOGICO DEL DATA BASE SOGGETTI

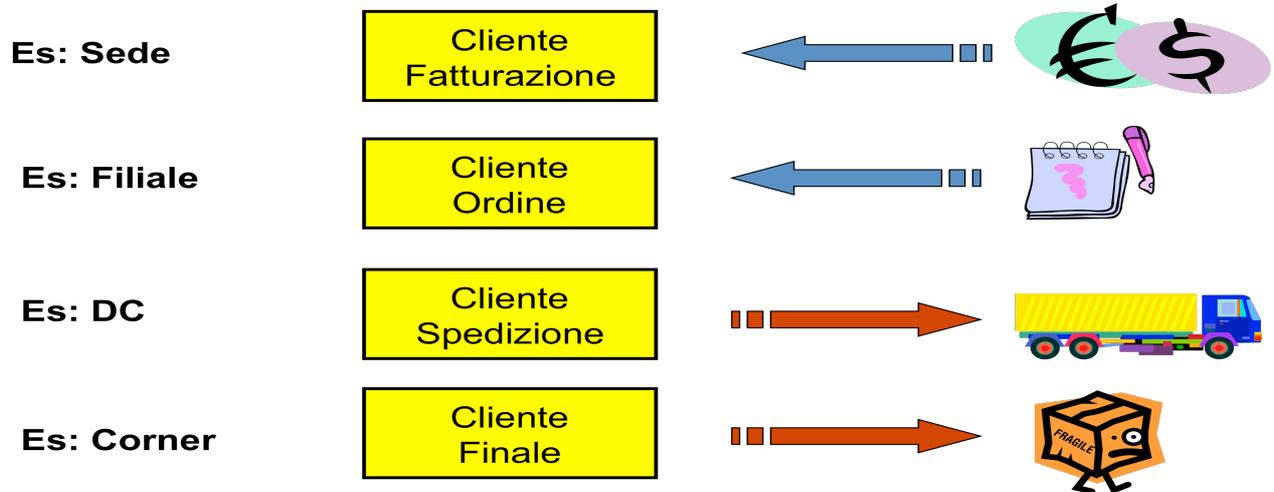


L'archivio soggetti contiene i dati anagrafici invariante rispetto ai ruoli (ragione sociale, partita IVA,...). Consente di rendere espliciti i legami per i soggetti che hanno più ruoli nei confronti dell'azienda. Gli archivi dei ruoli contengono dati che caratterizzano lo specifico ruolo.

Struttura Soggetti Attivi

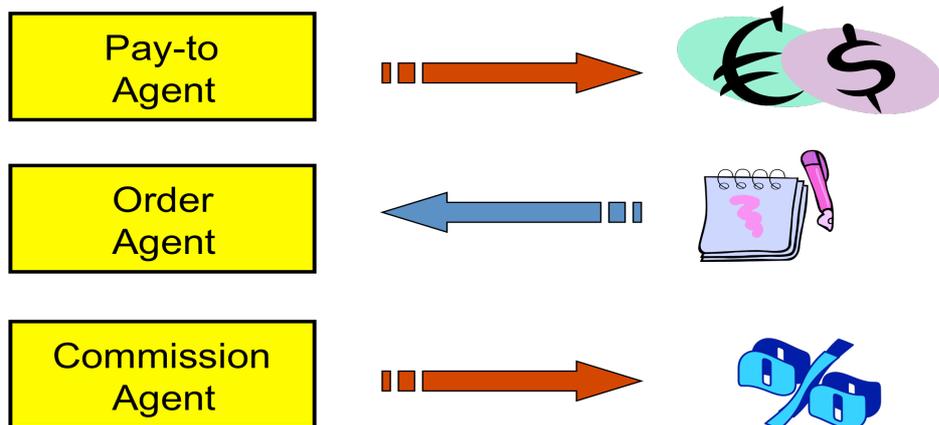


Struttura Soggetti Attivi



Per ogni soggetto viene specificato le operazione che sono determinate dal suo o i suoi ruoli nei confronti dell'azienda. I clienti, ad esempio, possono essere abilitati alla emissione di ordini, alla ricezione di fatture, ad essere destinatari di spedizione.

Struttura Agenti



2.3 Industrializzazione prodotto

Il processo di Industrializzazione Prodotto prevede la definizione dei dati tecnici attraverso la gestione di: Distinta Base e Cicli.

- **Distinta base**

La Distinta Base è l'elenco di componenti (materie prime, semilavorati, accessori, phantoms) che costituiscono il prodotto finito con i relativi consumi unitari. Le DiBa prevedono la gestione multi livello e la possibilità di specificare i consumi differenziati per taglia, drop/statura e misura, sono sempre stagionali, possono essere neutre o colorate, in versioni diverse, rilasciate o provvisorie.

- **Cicli**

I Cicli definiscono l'insieme delle lavorazioni necessarie alla realizzazione di un prodotto ed i relativi costi/tempi. Analogamente alle DiBa i Cicli sono stagionali, possono essere neutri o colorati, possono essere gestiti in versioni diverse.

- **Interfacce con CAD**

E' possibile collegare Stealth_3000 con sistemi di progettazione CAD per l'interscambio dei dati relativi alle anagrafiche prodotto e alle Diba.

2.4 Ciclo attivo

In Stealth 3000, il ciclo attivo si estende nella gestione dei listini di vendita, la gestione degli ordini, le proiezioni, le assegnazioni, le spedizioni, la fatturazione le statistiche e le provvigioni.

- **Listini di vendita**

La generazione dei Listini di vendita può avvenire per inserimento manuale, duplicazione calcolo da costi standard che possono essere applicate per percentuali di ricarico o di margine di contribuzione assolute o differenziate per prodotto o gruppi di prodotto. A fronte di una variazione prezzi di listino è possibile allineare automaticamente gli ordini clienti interessati, su cui non siano intervenute variazioni di prezzo manuali. I listini di vendita sono identificati con la stagione, il marchio, il mercato, la valuta e il codice listino.

Nel caso di prodotti NON Formali, i prezzi vengono definiti per prodotto. I prodotti gestiti a taglia possono avere prezzo unico o per fascia. In aggiunta è possibile dettagliare i prezzi per colore, misura, drop/statura, variante e etichetta/cartellino. Nel caso di prodotti Formali, i prezzi sono anche definibili per classe modello/classe parte, classe modello/ parte, modello/classe parte e variante.

L'evento può essere associato ad uno specifico listino. L'associazione al listino ha validità stagionale.

2.5 Ordini clienti

Molteplici tipologie di rapporti commerciali col cliente e differenti modalità di acquisizione dell'ordine. L'Ordine Cliente è caratterizzato da un TIPO che abilita o meno certe gestioni specifiche. I tipi di ordine gestiti sono Normale, Pronto, Distributore, Previsionale e Rilascio.

- **Ordine da pronto**

Questa tipologia si utilizza quando vi è necessità di partire dalla disponibilità per effettuare l'ordine (es. ordini da stock). Questi tipi d'ordine possono essere evasi e fatturati immediatamente. La disponibilità può essere sulla Giacenza, gli'Ordini Fornitore o le Commesse di Produzione.

- **Ordine distributore**

Questa tipologia si utilizza quando vi è necessità di gestire anche i dati relativi al rapporto tra il Distributore ed i suoi clienti finali, oltre quelli inerenti il rapporto tra il distributore e l'azienda. Questo tipo d'ordine consente di memorizzare anche le condizioni commerciali e i prezzi relativi ai clienti finali.

- **Ordine previsionale/impegno**

Questa tipologia si utilizza quando vi è necessità di creare un impegno o comunque una domanda per determinati prodotti non ancora collegata a specifici ordini cliente. L'ordine previsionale si comporta a tutti gli effetti come un ordine normale, ma non può né essere spedito, né essere fatturato.

- **Ordine di rilascio**

L'ordine di rilascio si utilizza a fronte di un ordine previsionale. Attraverso l'ordine di rilascio si vanno ad abbattere le quantità dell'ordine previsionale. Non necessariamente i prodotti e le quantità di un ordine di rilascio devono coincidere con prodotti e quantità abbattuti sull'ordine previsionale.

Gli ordini sono acquisiti tramite vari canali, acquisizione ordini Interna, acquisizione ordini da agenti/clienti e acquisizione ordini batch.

2.6 Assegnazioni

Le assegnazioni sono allocazioni di quantità di riga ordine cliente e di giacenza di magazzino, riga ordine fornitore e riga commessa di produzione relative allo stesso codice oggetto. E' possibile assegnare delle disponibilità in varie fasi della loro costituzione, all'inserimento della commessa o ordine fornitore, Sullo work-in-progress, sulla giacenza. Le assegnazioni standard allocano quantità a livello di codice oggetto come definito sulla riga ordine cliente, fornitore e commessa. Se il livello

di giacenza dell'oggetto è più dettagliato, l'assegnazione può, opzionalmente, operare a questo livello. Su una stessa riga ordine cliente possono insistere più assegnazioni, anche di tipo diverso (su righe ordine fornitore, su righe commessa produzione, su giacenze). Le assegnazioni degli oggetti gestiti a Taglia avvengono sulle singole taglie.

- **Stato assegnazione**

Ogni assegnazione ha associato uno STATO che la caratterizza, per mezzo dello STATO si possono condizionare certi processi. Ad esempio, solo le assegnazioni nello Stato "imballabile" sono trasformabili in colli. Gli stati gestiti dal sistema sono prenotabile, imballabile, movimentabile, prebollabile, spedibile. Ulteriori stati possono essere definiti dal cliente.

- **Modalità di assegnazione**

Le Assegnazioni possono essere generate secondo due modalità BATCH ("automatica") o INTERATTIVA (manuale). Le applicazioni consentono inoltre di gestirne, in modo puntuale o massivo, la relativa manutenzione e la eventuale cancellazione.

2.7 Spedizioni

Possono essere gestite diverse modalità di evasione delle spedizioni, spedizione di giacenza assegnata, spedizione di giacenza non assegnata e spedizione diretta fornitore – cliente. Per quanto riguarda il processo di spedizione, sono gestite diverse modalità;

Prelievo → spedizione

Prelievo → imballo → spedizione

Imballo → prelievo → spedizione.



Le spedizioni da assegnato possono essere pianificate in base a criteri che definiscono la spedibilità di un'ordine in funzione, ad esempio, della copertura, dello stato di evasione di ordini abbinati, dell'importo ecc.

- **Fatturazione attiva**

La fatturazione attiva prevede la gestione dell'emissione fatture in modalità interattiva e batch. E' possibile selezionare in modo parametrico il portafoglio evaso da fatturare e i criteri di raggruppamento ai fini dell'emissione fattura. Vengono gestite con funzionalità specifiche: fatturazioni in service, vendite di materiali ai terzisti.

Il sistema gestisce i **resi da cliente**, per questo si può emettere note credito per entrate a magazzino.

2.8 Statistiche

Il sistema consente ampia flessibilità di analisi dei dati di vendita attraverso il modulo di Ricerca e Sintesi e ai collegamenti con gli strumenti di Business Intelligence. Vengono comunque fornite le statistiche di base sul venduto e sul fatturato.

2.9 Provvigioni

Il sistema è in grado di calcolare le provvigioni sull'ordinato per agente d'ordine e agente di liquidazione.

2.10 Ciclo passivo

Copre tutti gli aspetti legati alla gestione della *supply chain*. La configurazione del sistema consente di gestire diverse modalità organizzative della produzione e degli acquisti: Produzione/Acquisti per Magazzino Produzione/Acquisti per Cliente. I processi gestiti comprendono Fabbisogni (MRP – PB), Pianificazione, Previsioni, Acquisti e Produzione.

- **Fabbisogni**

I fabbisogni possono essere guidati dalla richiesta reale o proiettata, dalla richiesta prevista, dal piano di produzione.

P.B. Product balance: Confronto tra Richiesta e Disponibilità per Prodotti, senza esplosione ai livelli inferiori (DiBa);

M.R.P.: Confronto Richiesta e Disponibilità per Prodotti e Materiali, con esplosione ai livelli inferiori (DiBa). (Pianificazione del fabbisogno materiali)

P.B. e M.R.P.: Possibilità di confronto a livello di Oggetto e anche di divisione produttiva, di tipologia di pianificazione e dalla data pianificata o della stagione.

Di grande importanza è il sistema di Pianificazione Fabbisogno Materiali o Materials Requirements Planning (MRP) che permette di programmare logiche di ordini automatici ai fornitori veramente

sofisticate, tanto da tener conto dei tempi di consegna e di messa in produzione del prodotto; questa metodologia permette di ottimizzare la rotazione dei materiali nei magazzini e la minimizzazione delle giacenze che impattano a livello contabile e fiscale. I suggerimenti per gli acquisiti e la produzione possono essere generati in automatico dall'elaborazione MRP – PB ed essere trasformati in Ordini e Commesse effettive.

- **Pianificazione**

E' possibile gestire un Piano di Produzione per Oggetti - Date Pianificate. Dal Piano di Produzione possono essere generate Ordini a Fornitore e Commesse di Produzione. Il Piano di Produzione può essere l'input per i Fabbisogni (MRP – PB).

- **Previsioni**

E' possibile immettere delle Previsioni per Oggetti - Date Pianificate. Dalle previsioni possono essere generati ordini a fornitore e commesse di produzione. Le previsioni possono essere l'input per i Fabbisogni (MRP – PB).

- **Acquisti**

Gli acquisti riguardano tutti gli oggetti gestiti: prodotti, materie prime, ecc. Possono essere immessi ordini a fornitore: Normali, di impegno (ordini previsionali, “quadro”) e di Rilascio. Gli Ordini a Fornitore “normali” possono essere immessi manualmente o generati dai suggerimenti prodotti dall'MRP – BP.

- **Produzione**

La produzione viene gestita tramite Commesse. E' possibile gestire le problematiche relativa alla lavorazioni interne o esterne (conto lavoro), al rifornimenti materiali per commessa o su magazzino sia in conto lavorazione che in conto vendita, al controllo avanzamento lavorazioni e alla gestione dei semilavorati con rientro e uscita o con trasferimenti diretti. Le Commesse di Produzione possono anche esse essere immesse manualmente o generate dai suggerimenti prodotti dall'MRP – BP. La gestione delle Commesse prevede l'assegnazione dei materiali alle commesse, il prelievo di materiali in conto lavoro, vendita, per rifornimento le commesse di taglio, gli ordini di lavoro e gli avanzamenti e i versamenti. La lanciabilità / fattibilità dei prodotti può essere controllata tramite le applicazioni di verifica. Le verifiche possono esaminare gli input diversi (ordini cliente, commesse, piano di produzione, suggerimenti MRP), tutti o parte dei Materiali previsti dalle DiBa, secondo il livello di criticità e le disponibilità effettive (Giacenze) e/o future.

- **Magazzini**

Le locazioni si identificano con una struttura gerarchica articolata in:

Magazzino, ubicazione, cella.

Magazzino e Ubicazione sono sempre necessari per identificare la locazione, le Celle sono facoltative. Il sistema è multi-magazzino e per ogni magazzino è possibile definire 'n' ubicazioni. Il sistema gestisce indifferentemente magazzini (ubicazioni) fisici e/o logici.

- **Costi**

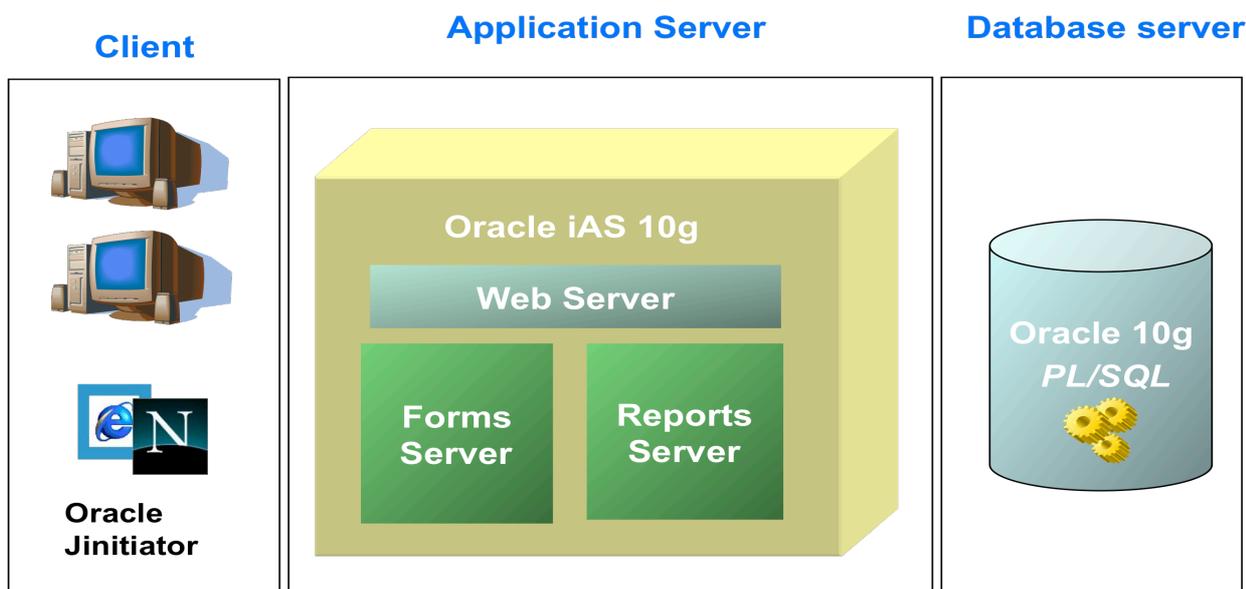
Il sistema gestisce sia i costi standard che i costi effettivi per prodotti finiti, materiali e semilavorati. I costi standard sono stagionali e possono essere differenziati per colore, taglia o misura. E' possibile gestire 'n' versioni di un costo standard. I costi standard possono essere calcolati automaticamente a partire dalle DiBa e dai Cicli e possono essere articolati in voci di costo variabili. I costi effettivi sono calcolati sulla base dei consuntivi delle commesse di produzione (consumi di materiale e lavorazioni) e sugli acquisti. Entrambi sono utilizzabili per le valorizzazioni di magazzino e per analisi dei costi di produzione/acquisto.

Capitolo 3: STEALTH 3000, la tecnologia, l'architettura e l'ambiente applicativo

Oracle e CSC hanno un accordo di partnership strategico sia a livello globale che a livello nazionale. Nell'ambito del Fashion questa partnership si è concretizzata nella scelta di Oracle come piattaforma tecnologica su cui realizzare Stealth 3000. Oltre a ciò, si è deciso, dove richiesto, di complementare l'offerta di Stealth 3000 con alcuni moduli delle Oracle Applications. Alla data attuale, questa offerta congiunta è limitata alla componente Amministrativa (Finanza e Controllo) delle Oracle Applications, che è l'unica componente sicuramente mancante, assieme alle Paghe e Stipendi, a Stealth 3000 per completare la copertura di tutte le aree aziendali. Questo accordo implica che, pur rimanendo due prodotti, due applicazioni distinte,

Oracle e CSC lavoreranno per fornire, dove appropriato, una modalità di interfacciamento già preconfigurata e gestita a standard. In altre parole, rispetto ad integrazioni con altri pacchetti contabili, il cliente che acquisisce i moduli contabili di Oracle Applications ha il vantaggio di avere i flussi di contabilizzazione già predisposti e gestiti a standard. Ciò vuol dire che CSC e Oracle si incaricheranno di mantenere aggiornate queste interfacce rispetto alla evoluzione delle rispettive releases. Inoltre, il processo di set up dell'applicazione sarà più rapido e mirato poiché a livello di dati anagrafici e tabellari la mappatura delle strutture dati è stata già fatta da Oracle e CSC, così come sono state già individuate le modalità di codifica dei dati tali da garantire la compatibilità tra le due applicazioni. Un ulteriore vantaggio di tale offerta è costituito dall'architettura tecnologica comune che consente una più facile realizzazione di funzioni aggiuntive o di accesso ai dati, oltre che una semplificazione e un abbassamento dei costi di manutenzione e di presidio del sistema.

Deployment Architettura a 3 livelli



3.1 Oracle Application Server 10g (OAS)

E' una infrastruttura completa che permette il deploy e la gestione di applicazioni Web realizzate con strumenti Oracle e di terze parti. Le tipologie di applicazioni supportate sono molteplici. Possono essere semplici pagine realizzate come JSP, applicazioni più complesse sviluppate con EJB, portali realizzati con Oracle Portal, applicazioni di data entry costituite con Developer (eg. Forms).

Dal punto di vista funzionale OAS è uno degli ambienti per applicazioni web più completi disponibili sul mercato e la sua integrazione presenta vantaggi per un utilizzo a livello di Enterprise. Questo parte cerca di presentarne gli elementi fondamentali accennando alle principali funzionalità e riportando qualche elemento dell'architettura.

3.1.1 Principali componenti

Prima di descrivere singolarmente i componenti è necessario introdurre le due tipologie di Server che OAS utilizza:

- **Infrastructure**
E' un server dedicato ai servizi di sistema quali l'autenticazione degli utenti, i DB contenenti le configurazioni applicative.
- **Middle Tier**
E' un *normale* Application Server che ospita applicazioni e servizi utilizzati dagli utenti finali. Nelle installazioni più complesse sono presenti più Mid Tier con configurazioni che possono anche essere differenti ed elementi posti in cluster. Questa è la tipologia usata da Stealth 3000.

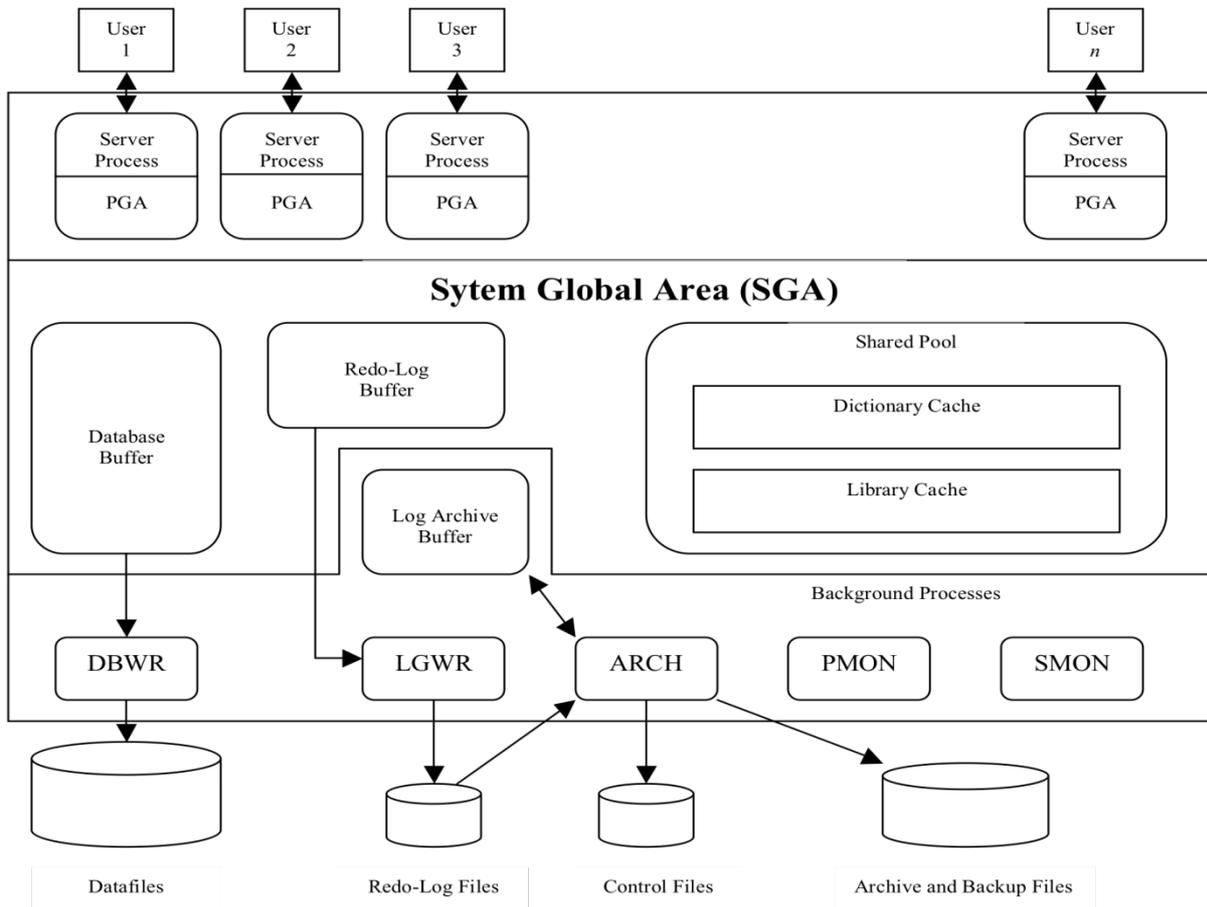
Ogni installazione attiva di OAS prende il nome di **OAS Instance**. I Mid Tier fanno riferimento ad una singola instance di infrastruttura. L'insieme dell'istanza di infrastruttura e di tutte le istanze di Mid Tier ad essa associate prendono il nome di **OAS Farm**. I servizi su entrambe le tipologie di Server sono realizzati con componenti di base analoghi adattati con configurazioni specifiche. I principali componenti sono:

- **HTTP Server**
OAS utilizza il diffusissimo ed ottimo **Apache**. Si tratta dell'HTTP Server più potente e diffuso con funzionalità assai complete (Ad esempio, virtual host, SSL, CGI, ecc) Oltre ai moduli standard di Apache sono presenti moduli specifici realizzati da Oracle quali il **mod_plsql** per accedere ai package PL/SQL, **mod_osso** per l'aggancio al Single Sign-On (SSO), **mod_oc4j** per il dialogo AJP con il Java Container OC4J.
- **Web Cache**
Rispetto ad altri programmi di cache la web cache di OAS si differenzia poiché ottimizzata per il caching di oggetti dinamici su DB. In questo modo non è necessario effettuare un nuovo accesso e la pagina contenuta in cache viene immediatamente restituita all'utente finale se non sono cambiati i dati sul DB. L'implementazione di tale meccanismo richiede uno scambio di informazioni verso la web cache con messaggi di Invalidation a fronte di modifiche sui dati.

- **Java Container**
OAS utilizza quale J2EE Java Container il programma **OC4J**. L'OC4J può ospitare su una J2SE pagine Java Server Pages (JSP), Servlet ed Enterprise Java Beans (EJB). Possono essere ospitate tutte le applicazioni Standard J2EE.
- **LDAP Server**
Il server LDAP realizzato da Oracle prende il nome di **OID**. Si tratta di un server LDAP standard v.3 e mantiene i dati relativi ad utenti ed applicazioni su uno schema Oracle.
- **DB**
Naturalmente non poteva mancare un'istanza di RDBMS Oracle! L'RDBMS viene utilizzato per ospitare i diversi schemi utilizzati dagli ambienti di OAS per memorizzare oggetti e configurazioni. Per tale ragione viene generalmente denominato **Metadata Repository**.

3.2 Database Oracle 10g

Un Oracle database server è costituito da due strutture: Database e Istanza.



Architettura Database Oracle

3.2.1 Principali componenti

Per Database si indicano tutti i file fisici in cui vengono memorizzati i dati. Ha il compito di controllare un'enorme quantità di dati in un ambiente multiutente, in modo da permettere l'accesso contemporaneo agli stessi dati da parte di più utenti provvedendo anche a fornire soluzioni efficienti per il recovery. Il database è formato da strutture di memorizzazione, logiche e fisiche. Le prime vengono utilizzate per immagazzinare, attraverso Tablespace, Block, Extent e Segment, e gestire (tabelle, indici, etc.) i dati, le seconde invece contengono le strutture fisiche (data file, redo log file, control file). I servizi offerti dalle strutture logiche del server sono indipendenti dalle strutture fisiche che le contengono, perché le strutture logiche possono essere progettate nello stesso modo, indipendentemente dall'hardware e dal sistema operativo impiegati.

Per Istanza si intendono l'insieme delle aree di memoria (System Global Area) e dei processi di background necessari ad accedere ai dati. Tali aree sono allocate nell'istanza Oracle ogni volta che viene fatta partire e rilasciate quando l'istanza viene terminata. Le aree di memoria di un server Oracle sono usate per contenere i dati, le informazioni del dizionario dei dati, i comandi SQL, il codice PL/SQL ecc. Ogni database deve obbligatoriamente fare riferimento almeno ad un'istanza. Le due maggiori strutture di memoria sono la System Global Area (SGA) e la Program Global Area (PGA). La PGA è un'area di memoria che contiene i dati di un singolo processo utente ed è allocata da Oracle quando un utente si collega al database e una sessione viene creata.

Un tablespace è una suddivisione logica del database, costituito fisicamente da uno o più files chiamati DATA-FILES (DF). Tutti gli oggetti del database vengono logicamente memorizzati in tablespace. Ogni database ha almeno un tablespace, il tablespace SYSTEM, che contiene il data dictionary. Altri tablespace possono essere creati e usati per differenti applicazioni o compiti o per semplificare e migliorare l'efficienza. Ad esempio si può creare un tablespace USERS di utilizzo generale e un tablespace UNDO per i segmenti di rollback. Un tablespace può appartenere ad un solo database.

La SGA è l'area di memoria assegnata all'avvio da Oracle, contiene le strutture della memoria per memorizzare i dati e controllare le informazioni. Gli utenti che si conatteranno al database condivideranno le informazioni nella stessa SGA. Le informazioni memorizzate nella System Global Area sono divise in diversi tipi di buffer: Database Buffer, Redo Log Buffer, Shared Pool, Large Pool e Java Pool.

Redo log file, registrano tutte le modifiche occorse ai dati. Ogni DB possiede almeno due file di redo log, perché Oracle scrive in questi file in maniera circolare: quando un file di redo log è pieno allora Oracle scrive in quello successivo, quando l'ultimo file di redo log è pieno allora Oracle ricomincia dal primo, assicurandosi però di memorizzare le informazioni nei data file prima di sovrascriverle. Se una qualsiasi anomalia non permette la scrittura delle modifiche occorse al database nei rispettivi data file, allora possiamo tranquillamente ottenere tali modifiche dai redo log file. Le modifiche ai dati, pertanto, non sono mai perse. I redo log file rappresentano la cronologia delle modifiche ai dati e sono di vitale importanza per l'integrità del DB. Oracle permette di avere più di una copia per ciascun redo log file: questa importante caratteristica è denominata multiplexing dei redo log file.

Ogni database ha almeno un control file che contiene informazioni circa la struttura fisica del DB, ovvero il nome, la data e l'ora di creazione e il percorso completo di ciascun data file e redo log file. È di vitale importanza e si consiglia di configurarne più di una copia: anche in questo caso parleremo di multiplexing di control file.

The shared pool è usata per memorizzare gli statement SQL recentemente utilizzati, questo consente ad un eventuale processo oracle, di reperire informazioni in maniera più veloce.

The database buffer è usata per memorizzare dati recentemente trattati. Tali dati sono letti e/o scritti nei data file (trattati successivamente).

Data and Library caches. Sono aree di cache utilizzate per contenere i dati in transito da e verso il database (data) e le istruzioni SQL e PL/SQL in esecuzione (library).

I processi di background, in una istanza oracle, eseguono funzioni comuni in relazione alla richiesta di un processo utente. Oltre ad effettuare tali richieste, i processi di background eseguono anche operazioni di input/output. Il numero ed il tipo di processi che vengono eseguiti dipendono dalla particolare configurazione di oracle, ma quelli elencati nel seguito sono sempre presenti.

- Database writer (DBW0) responsabile della scrittura dei dati dal database buffer (memoria) nei datafile (disco).
- Log writer (LGWR) responsabile della scrittura, degli eventi che si susseguono in una sessione di lavoro, dal log buffer (memoria) nel log file (disco).
- System monitor (SMON) responsabile del controllo della consistenza del database, se necessario inizia un recovery del database, quando esso è attivo.
- Process monitor (PMON) responsabile del rilascio delle risorse allocate da un eventuale processo utente fallito. In altri termini esso esegue le operazioni di “pulizia” deallocando le risorse inutilizzate.
- The checkpoint process (CKPT) responsabile dell’aggiornamento dello status del database, per quanto concerne i datafile, ogni volta che viene memorizzato permanentemente un record nel database (garantisce la sincronizzazione dei dati).
- The recoverer process (RECO) responsabile della risoluzione delle transazioni fallite in ambiente distribuito.
- The archiver process (ARC0) responsabile del salvataggio automatico delle copie dei redo log in un’area di memoria stabile specificata dall’amministratore del database. Tale processo risulta attivo se è impostata a “true” la proprietà di “log_archive”.
- The dispatcher process (D000) responsabile della gestione in termini di scheduling delle richieste utente.
- The lock process (LMS) responsabile della gestione dei lock sulle transazioni in ambiente distribuito.

Un processo listener è il processo che si occupa di gestire la comunicazione fra i processi client e un’istanza oracle. Un processo listener è caratterizzato da un nome, del database oracle che serve e da una porta di ascolto. E’ possibile definire un processo listener per ogni oracle database, o, più listener sullo stesso database.

Un’ oracle database viene poi univocamente determinato, in ambito distribuito, da un nome o SID, dall’indirizzo IP o nome a dominio dell’host che la ospita e dalla porta di accesso su cui è in ascolto il processo listener. Tutte le impostazioni che determinano il funzionamento dell’ambiente runtime del database Oracle 10g sono contenute nel file init.ora che viene letto all’atto di avvio di un’istanza Oracle. In esso si ritrovano informazioni relative alla posizione su file system di tutti i file di sistema, le informazioni relative al dimensionamento della SGA e quelle relative alle funzionalità abilitate. Riassumendo quindi possiamo affermare che l’architettura oracle si compone di uno o più processi utente, che rappresentano gli utilizzatori finali del DB (utilizzatori umani o entità software), e di un processo server, il cui compito è quello di accettare richieste da parte dei processi utente, e smistare poi tali richieste all’istanza oracle e così via.

3.2.2 Analisi del processo di esecuzione di un'istruzione SQL

Nella parte seguente analizzeremo sommariamente come viene processata un'istruzione SQL dal server Oracle e come i processi e i buffer sono coinvolti.

- Si assuma che un utente (lavorando con SQL*Plus) esegua un'istruzione **update** nella tabella **TAB** in modo che più di una tupla venga interessata dall'aggiornamento. L'istruzione viene passata al server dal processo **USER**. Quindi il server (o più raramente il processore di query) controlla se l'istruzione è già contenuta nella library cache, in modo che le corrispondenti informazioni (albero di decodifica, piano di esecuzione) possano essere riutilizzate. Se l'istruzione non viene trovata, viene decodificata e dopo aver verificato l'istruzione (privilegi utente, tabelle e colonne interessate) utilizzando i dati dalla cache del data dictionary, viene generato un piano di esecuzione della query dall'ottimizzatore di query. Insieme all'albero di decodifica, questo piano viene memorizzato nella library cache.
- Per gli oggetti interessati dall'istruzione (in questo caso la tabella **TAB**) viene controllato se i corrispondenti blocchi di dati esistono nel buffer del database. In caso negativo, il processo **USER** legge i blocchi di dati dai files fisici al buffer del database. Se non c'è abbastanza spazio nel buffer, i blocchi di altri oggetti utilizzati meno recentemente vengono scritti su disco dal processo **DBWR**.
- Le modifiche delle tuple interessate dall'**update** avvengono nel buffer del database. Prima che i blocchi di dati vengano modificati, una "immagine" delle tuple viene scritta sui segmenti di rollback dal processo **DBWR**.
- Mentre il buffer redo-log viene riempito a causa delle modifiche ai blocchi, il processo **LGWR** scrive sui file di redo-log le voci presenti nel redo-log buffer.
- Dopo che tutte le tuple (o i corrispondenti blocchi di dati) sono state modificate nel buffer del database, le modifiche possono essere confermate dall'utente attraverso un comando **commit**.
- Finché il comando **commit** non è stato inviato dall'utente, le modifiche possono essere annullate utilizzando il comando **rollback**. In questo caso, i blocchi modificati nel buffer del database vengono sovrascritti dai blocchi originali memorizzati nei segmenti di rollback.
- Se l'utente invia un comando **commit**, lo spazio allocato per i blocchi nel segmento di rollback può essere liberato e reso quindi disponibile per altre transazioni. Inoltre, i blocchi modificati nel buffer del database vengono sbloccati in modo che gli altri utenti possano ora leggere e modificare quei blocchi. La fine della transazione (più precisamente il **commit**) viene registrato nei files redo-log. I blocchi modificati vengono scritti su disco solo se si rende necessario altro spazio per altre transazioni.

3.3 Oracle PL/SQL Developer, PL/SQL

SQL è un linguaggio utilizzato per effettuare interrogazioni su database relazionali. Esso possiede caratteristiche complesse che consentono di estrapolare i dati in diverse forme in base alle nostre esigenze, tuttavia, essendo un linguaggio non procedurale, non è possibile interrogare i dati e al contempo eseguire operazioni su questi utilizzando esclusivamente il linguaggio SQL. Lo sviluppo di applicazioni di database richiede tipicamente costrutti di linguaggio simili a quelli che possono essere ritrovati nella programmazione di linguaggi C o C++ . Questi costrutti

sono necessari per l'implementazione di complesse strutture di dati e algoritmi. Una delle maggiori restrizioni del linguaggio di database SQL è quella di non poter compiere diversi compiti usando solo gli elementi forniti con questo linguaggio (che è stato pensato espressamente per le interrogazioni di database). PL/SQL (Procedural Language /SQL) è un'estensione procedurale dell'Oracle-SQL che offre costrutti di linguaggio del tutto simili a quelli dei linguaggi di programmazione per eccellenza. PL / SQL Developer è un ambiente di sviluppo integrato per lo sviluppo, il test, il debug e l'ottimizzazione per unità di programma memorizzato Oracle PL / SQL come i packages e trigger. Il programma dispone di guida sensibile al contesto e le descrizioni degli oggetti del database. Inoltre è incluso un completo debugger integrato, una multi-sessione o ambiente multi-threaded, un assistente di codice, un modello di libreria, una libreria macro, e una finestra di comando.

PL/SQL permette agli utenti e ai designer di sviluppare complesse applicazioni di database che richiedono l'uso di strutture di controllo e elementi procedurali come procedure, funzioni e moduli. Il costrutto di base in PL/SQL è il *block* (blocco). I blocchi permettono ai programmatori di combinare logicamente i comandi SQL in unità. In un blocco, costanti e variabili possono essere dichiarate, e le variabili possono essere utilizzate per memorizzare i risultati di una query. Le istruzioni in un blocco PL/SQL includono istruzioni SQL, strutture di controllo (loop), istruzioni di condizione (if-then-else), manipolazione delle eccezioni (controllo errori), e chiamate ad altri blocchi PL/SQL. I blocchi PL/SQL che specificano procedure e funzioni possono essere raggruppati in *packages* (pacchetti). Un package è simile a un modulo e ha un'interfaccia e un'implementazione a parte. Oracle offre diversi packages predefiniti, per esempio, routines di input/output, manipolazione di files, pianificazione di jobs, ecc.

Un'altra importante caratteristica di PL/SQL è che offre un meccanismo per processare i risultati delle query in un modo "orientato alle tuple", il che vuol dire, una tupla alla volta. A questo scopo, vengono utilizzati i *cursori*. Un cursore è fondamentalmente un puntatore al risultato di una query ed è impiegato per leggere i valori degli attributi delle tuple selezionate, inserendoli in variabili. Un cursore è tipicamente usato in combinazione con un costrutto loop tale che ogni tupla letta dal cursore può essere processata individualmente. Riassumendo, i più importanti obiettivi del PL/SQL sono:

- Incrementare le possibilità delle espressioni in SQL.
- Processare i risultati delle query "orientandosi alle tuple".
- Ottimizzare i comandi SQL combinati.
- Sviluppare applicazioni modulari per database.
- Riutilizzare il codice di programma.
- Ridurre i costi per la manutenzione e la modifica delle applicazioni.

3.3.1 Struttura dei blocchi PL/SQL

PL/SQL è un linguaggio strutturato in blocchi. Ogni blocco costituisce una denominata unità di programma, in più i blocchi possono essere nidificati. I blocchi che costituiscono procedure, funzioni o package devono avere un nome. Un blocco PL/SQL può avere una sezione di dichiarazione opzionale, una parte contenente istruzioni PL/SQL e una parte opzionali per la gestione delle eccezioni (errori). Quindi la struttura di un programma PL/SQL appare come segue

(le parentesi [] racchiudono parti opzionali):

[<Block Header>]

[declare

<Costanti>

<Variabili>

<Cursori>

<Eccezioni definite dall'utente>]

begin

<Istruzioni PL/SQL>

[exception <gestione eccezioni>]

end;

L'intestazione del blocco specifica se è una procedura, una funzione, o un package. Se non viene specificata nessuna intestazione, il blocco viene detto *anonimo*. Ogni blocco PL/SQL è costituito ancora da istruzioni PL/SQL. Quindi il blocco può essere nidificato come i blocchi nei linguaggi di programmazione classici. Lo scopo delle variabili dichiarate (per es., nella parte di programma nel quale si può fare riferimento alla variabile) è analogo allo scopo delle variabili nei linguaggi di programmazione come C o Pascal.

Dichiarazioni

Costanti, variabili, cursori ed eccezioni usate in un blocco PL/SQL devono essere dichiarati nella sezione di dichiarazione di quel blocco. Variabili e costanti possono essere dichiarate come segue:

<nome variabile> [**constant**] <tipo di dati> [**not null**] [**:=** <espressione>];

Tipi di dati validi sono i tipi di dati SQL e il tipo **boolean**. Il tipo boolean può essere solamente *vero*, *falso* o *null*. La clausola **not null** richiede che la variabile dichiarata debba sempre avere un valore diverso da *null*. <espressione> è usata per inizializzare la variabile. Se non viene specificata nessuna espressione, il valore *null* viene assegnato automaticamente.

La clausola **constant** indica

Al posto di uno specifico tipo di dati, si può fare anche riferimento al tipo di dato di una colonna di tabella (*dichiarazione ancorata*). Per esempio, `EMP.Empno%TYPE` si riferisce al tipo di dato della colonna `Empno` nella relazione `EMP`. Al posto di una singola variabile, può essere dichiarato un record che può memorizzare una tupla completa di una determinata tabella (o risultato di query). Per esempio, il tipo di dati `DEPT%ROWTYPE` specifica una record adatto a memorizzare tutti i valori degli attributi di una riga completa dalla tabella `DEPT`. Tali record sono tipicamente usati in combinazione con un cursore. Si può accedere a un campo di un record utilizzando la sintassi

<nome record>.<nome colonna>, per esempio, `DEPT.Deptno`. La dichiarazione di un cursore specifica un insieme di tuple (come risultato di una query) in modo che la dichiarazione di un cursore ha la sintassi:

cursor <nome cursore> [(<lista parametri>)] **is** <istruzione *select*>;

In PL/SQL un parametro ha la forma <nome parametro> <tipo parametro>. I possibili parametri sono **char**, **varchar2**, **number**, **date** e **boolean** e tutti i corrispondenti sotto-tipi come **integer**. I parametri sono utilizzati per assegnare valori alle variabili che sono date in un'istruzione **select**.

Se le tuple selezionate dal cursore devono essere modificate all'interno del blocco PL/SQL, deve essere aggiunta la clausola **for update** [(<colonna(e)>)] alla fine della dichiarazione del cursore. In questo caso le tuple vengono bloccate e gli altri utenti non vi possono accedere finché non è stato eseguito un comando **commit**. Prima che un cursore dichiarato possa essere utilizzato in istruzioni PL/SQL, il cursore deve essere aperto, e dopo aver processato le tuple selezionate il cursore deve essere chiuso.

Le eccezioni vengono usate per processare errori e avvisi che si incontrano durante l'esecuzione controllata delle istruzioni PL/SQL. Alcune eccezioni sono definite internamente, come ZERO_DIVIDE (divisione per 0). Altre eccezioni possono essere specificate dall'utente alla fine di un blocco PL/SQL. Le eccezioni definite dall'utente devono essere dichiarate usando <nome eccezione> **exception**.

3.3.2 Elementi di linguaggio

In aggiunta alla dichiarazione delle variabili, costanti e cursori, PL/SQL offre diversi costrutti di linguaggio come assegnazione di variabili, strutture di controllo (loop, if-then-else), chiamate a procedure e funzioni, ecc. Comunque, PL/SQL non consente l'utilizzo di comandi di definizione dati SQL, come l'istruzione **create table**. Per questo, PL/SQL fornisce speciali package. Inoltre, PL/SQL usa un'istruzione **select** modificata che richiede che ogni tupla selezionata sia assegnata a un record (o a una lista di variabili). Ci sono diverse alternative in PL/SQL per assegnare un valore a una variabile. Il modo più semplice è:

declare

```
counter integer := 0;
```

...

begin

```
counter := counter + 1;
```

I valori da assegnare a una variabile possono anche essere recuperati da un database usando un'istruzione **select**:

```
select <colonna(e)> into <lista di variabili corrispondenti>
```

```
from <tabella(e)> where <condizione>;
```

E' importante assicurarsi che l'istruzione **select** recuperi almeno una tupla! Altrimenti non è possibile assegnare i valori degli attributi alla lista di variabili specificata, che causerà un errore runtime. Se l'istruzione recupera più di una tupla, allora deve essere utilizzato un cursore. Inoltre, il tipo di dati delle variabili specificate deve coincidere con quello dei valori degli attributi recuperati. Per molti tipi di dati, PL/SQL esegue una conversione dei tipi automatica (per esempio, da **integer** a **real**). Al posto di una singola lista di variabili, può essere dato un record dopo la parola chiave **into**.

Anche in questo caso, l'istruzione **select** deve recuperare almeno una tupla!

PL/SQL fornisce loop **while**, due tipi di loop **for** e loop continui. Questi ultimi sono utilizzati in combinazione con i cursori. Tutti i tipi di loop sono utilizzati in una sequenza di istruzioni più volte. La specificazione di un loop avviene nello stesso modo dei linguaggi di programmazione come C o Pascal. Un loop while ha la forma

```
[<< <nome etichetta> >>]
while <condizione> loop
    <sequenza di istruzioni>;
end loop [<nome etichetta>];
```

Un loop può avere un nome. Dare un nome a un loop è utile quando i loop sono nidificati e i primi vengono completati incondizionatamente utilizzando l'istruzione **exit** <nome etichetta>. Mentre il numero di iterazioni di un loop **while** è sconosciuto fino al momento in cui il loop viene completato, il numero di iterazioni di un loop **for** può essere specificato usando due interi.

```
[<< <nome etichetta> >>]
for <indice> in [reverse] <estremo inferiore>..<estremo superiore> loop
    <sequenza di istruzioni>
end loop [<nome etichetta>];
```

Il contatore del loop <indice> viene dichiarato implicitamente. Lo scopo del contatore del loop è solo per il loop **for**. Esso sovrascrive lo scopo di ogni variabile che ha lo stesso nome al di fuori del loop. All'interno del loop **for**, <indice> può essere utilizzato come riferimento come se fosse una costante. <indice> può comparire in espressioni, ma non gli possono essere fatte assegnazioni. Utilizzando la parola chiave **reverse** si fa in modo che il processo di iterazione parta dall'estremo superiore fino a quello inferiore nell'insieme specificato.

Elaborazione di cursori: prima che un cursore possa essere usato, deve essere aperto utilizzando l'istruzione **open**

```
open <nome cursore> [( <lista di parametri> )];
```

L'istruzione **select** associata viene quindi processata e il cursore punta alla prima tupla selezionata. Le tuple selezionate possono essere processate una alla volta utilizzando il comando **fetch**

```
fetch <nome cursore> [( <lista di variabili> )];
```

Il comando **fetch** assegna i valori degli attributi selezionati dalla tupla corrente alla lista di variabili. Dopo un comando **fetch**, il cursore avanza alla successiva tupla nell'insieme del risultato dell'istruzione **select** ottenuto con il comando **open**.

Da notare che le variabili nella lista devono avere lo stesso tipo di dati dei valori delle tuple selezionate. Dopo che tutte le tuple sono state processate, si utilizza il comando **close** per chiudere e disabilitare il cursore.

```
close <nome cursore>;
```

Ogni loop può essere completato incondizionatamente usando la clausola **exit**:

```
exit [<etichetta blocco>] [when <condizione>];
```

Usando **exit** senza un'etichetta di blocco si provoca il completamento del loop che contiene l'istruzione **exit**. Una condizione può essere un semplice paragone di valori. In molti casi, comunque, la condizione fa riferimento a un cursore. Nell'esempio sopra, `%NOTFOUND` è un predicato che restituisce *false* se il comando **fetch** più recente ha letto una tupla. Il valore di `<nomecursore>%NOTFOUND` è *null* prima che la prima tupla venga letta. Il predicato restituisce *true* se il comando **fetch** più recente fallisce nel restituire una tupla, nel caso contrario restituisce *false*. `%FOUND` è l'opposto logico di `%NOTFOUND`.

I loop **for** di cursore vengono utilizzati per semplificare l'utilizzo di un cursore:

```
[<< <nome etichetta> >>]
  for <nome record> in <nome cursore> [(lista di parametri)] loop
<sequenza di istruzioni>
end loop;
```

Un record utilizzabile per memorizzare una tupla recuperata da un cursore viene implicitamente dichiarato. Inoltre, questo loop implicitamente esegue un **fetch** ad ogni iterazione, un **open** prima dell'ingresso nel loop e un **close** dopo che il loop è terminato. Se ad un'iterazione nessuna tupla viene recuperata, il loop viene automaticamente terminato senza un **exit**. E' anche possibile specificare una query al posto di <nome cursore> in un loop for:

```
for <nome record> in (<istruzione select>) loop
<sequenza di istruzioni>
end loop;
```

Dunque, un cursore non deve essere specificato prima dell'ingresso nel loop, ma è definito nell'istruzione **select**.

Per il controllo condizionale, PL/SQL offre costrutti **if-then-else** nella forma:

```
if <condizione> then <sequenza di istruzioni>
  [elsif] <condizione> then <sequenza di istruzioni>
...
[else] <sequenza di istruzioni> end if;
```

Partendo con la prima condizione, se risulta *true*, la corrispondente sequenza di istruzioni viene eseguita, altrimenti il controllo viene passato alla successiva condizione. Quindi, quello che sta dietro a questo tipo di istruzione PL/SQL è del tutto simile alle istruzioni if-then-else dei linguaggi di programmazione classici. Ad eccezione dei comandi di definizione dei dati come create table, tutti i tipi di istruzione SQL può essere usata nei blocchi PL/SQL, in particolare dolete, insert, update e commit. Da notare che in PL/SQL solo l'istruzione **select** del tipo **select** <colonna(e)> **into** è permessa, cioè, i valori degli attributi selezionati possono solo essere assegnati a variabili (finchè l'istruzione select viene utilizzata in una subquery). L'utilizzo dell'istruzione **select** come nell'SQL causa un errore di sintassi (*syntax error*). Se le istruzioni **update** o **delete** vengono utilizzate in concomitanza con un cursore, questi comandi possono essere ristretti alla tupla attualmente recuperata.

3.3.3 Gestione Eccezioni

Un blocco PL/SQL può contenere istruzioni che specificano routines di gestione delle eccezioni. Ogni errore o avviso durante l'esecuzione di un blocco PL/SQL causa una eccezione. Si può distinguere tra due tipi di eccezioni:

Eccezioni definite dal sistema. Eccezioni definite dall'utente (che deve essere dichiarata dall'utente nella parte di dichiarazione di un blocco dove l'eccezione è utilizzata/implementata). Il sistema definisce eccezioni che vengono automaticamente attivate quando i corrispondenti errori o avvisi vengono incontrati. Le eccezioni definite

dall'utente, contrariamente, devono essere attivate esplicitamente in una sequenza di istruzioni utilizzando

raise <nome eccezione>.

Dopo la parola chiave **exception** alla fine di un blocco, le routines di gestione delle eccezioni definite dall'utente vengono implementate. Un'implementazione ha la forma:

when <nome eccezione> **then** <sequenza di istruzioni>;

Dopo una parola chiave **when** una lista di nomi di eccezioni (implicitamente connesse con un operatore **or**) può essere specificata. L'ultima clausola **when** nell'eccezione può contenere il nome di eccezione **others**. Questo introduce la routine di gestione delle eccezioni di default, per esempio, un **rollback**.

Se un programma PL/SQL viene eseguito da una shell SQL*Plus, le routine di gestione delle eccezioni possono contenere istruzioni che visualizzano l'errore o l'avviso sullo schermo.

A questo scopo, può essere utilizzata la procedura **raise_application_error**. Questa procedura ha due parametri <numero errore> e <messaggio errore>. <numero errore> è un numero intero negativo definito dall'utente e deve essere compreso tra -20000 e -20999. <messaggio errore> è una stringa con una lunghezza fino a 2048 caratteri. L'operatore di concatenazione "||" può essere utilizzato per concatenare singole stringhe in un'unica stringa. Allo scopo di visualizzare variabili numeriche, queste variabili devono essere convertite in stringhe usando la funzione **to_char**. Se la procedura **raise_application_error** viene chiamata da un blocco PL/SQL, l'elaborazione del blocco termina e tutte le modifiche ai database vengono annullate, cioè, viene eseguito un implicito **rollback** in aggiunta alla visualizzazione del messaggio di errore.

Esempio:

```
if emp_sal*1.05 > 4000
then raise_application_error(-20010,'Aumento di stipendio per impiegato id '
||to_char(emp_no)||' è troppo alto');
```

3.3.4 Procedure e Funzioni

PL/SQL fornisce sofisticati costrutti di linguaggio per programmare procedure e funzioni come blocchi PL/SQL separati. Essi possono essere richiamati da altri blocchi PL/SQL, altre procedure e altre funzioni. La sintassi per la definizione di una procedura è

create [or replace] procedure <nome procedura> [(<lista di parametri>)] **is**
<dichiarazione>

begin

<sequenza di istruzioni>

[**exception**

<routine di gestione dell'eccezione>]

end [<nome procedura>];

Una funzione può essere specificata in modo analogo

create [or replace] function <nome funzione> [(<lista di parametri>)]

return <tipo di dati> **is**

...

La clausola opzionale **or replace** ri-crea la procedura/funzione. Una procedura può essere cancellata utilizzando il comando **drop procedure** <nome procedura> (**drop function** <nomefunzione>).

Contrariamente ai blocchi PL/SQL anonimi, la clausola **declare** non può essere usata nella definizione di procedure/funzioni.

I tipi di dati validi includono tutti i tipi di dati. Comunque, per **char**, **varchar2** e **number** senza lunghezza e precisione/scala, rispettivamente, possono essere specificati. Al posto di espliciti tipi di dati, impliciti tipi nella forma %TYPE e %ROWTYPE possono essere utilizzati anche se viene fatto riferimento a dichiarazioni con vincoli. Un parametro è specificato come segue:

<nome parametro> [**IN** | **OUT** | **IN OUT**] <tipo di dato> [{ := | **DEFAULT** } <espressione>]

La clausola opzionale **IN**, **OUT**, e **IN OUT** specifica il modo nel quale il parametro è utilizzato. Il modo di default di utilizzo di un parametro è **IN**. **IN** significa che al parametro si può fare riferimento all'interno del corpo della procedura, ma non può essere modificato. **OUT** significa che un valore può essere assegnato al parametro nel corpo della procedura, ma non si può fare riferimento al valore del parametro. **IN OUT** permette sia di far riferimento al parametro che l'assegnazione di valori a quest'ultimo. Tipicamente, è sufficiente usare il modo di default dei parametri.

3.3.5 Packages

E' essenziale per un buon stile di programmazione che i blocchi, procedure e funzioni logicamente correlati vengano combinati in moduli, e che ogni modulo fornisca un'interfaccia che permetta agli utenti e sviluppatori di utilizzarne le funzionalità. PL/SQL supporta il concetto di modularizzazione grazie al quale i moduli e altri costrutti possono essere organizzati in *packages*. Un package è costituito da una dichiarazione di package e da un corpo di package. La dichiarazione del package definisce l'interfaccia che è visibile ai programmatori di applicazione, e il corpo del package implementa la dichiarazione del package (simile all'header e source files nel linguaggio di programmazione C). Una procedura o funzione implementata in un package può essere richiamata da altre procedure e funzioni utilizzando l'istruzione <nome package> . <nome procedura> [(<lista di parametri>)].

3.4 Oracle Forms Builder

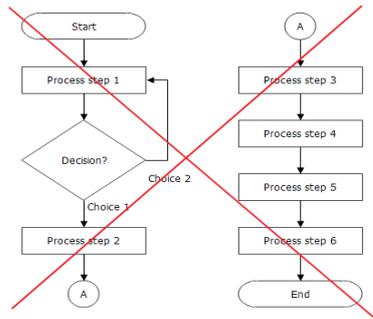
Oracle Forms è un prodotto software per la creazione, la messa in rete e l'esecuzione di "maschere" che interagiscono con un database Oracle. La creazione avviene utilizzando un IDE Forms Builder che fa parte della suite di applicazioni correlate Forms Developer (10gRel.2). La suite, a sua volta, è uno dei componenti dell'Oracle Fusion Middleware Forms Services (App. Server + servizi associati). Questi componenti forniscono in definitiva un completo **framework** per lo sviluppo rapido di applicazioni (RAD) web-deployed e database-oriented. In particolare, gli aspetti caratteristici dell'ambiente di sviluppo sono:

- Dichiaratività
- Metadata* driven
- Event handling programming (triggers)
- Wizards & Editors sia per la definizione del layout che della logica applicativa.

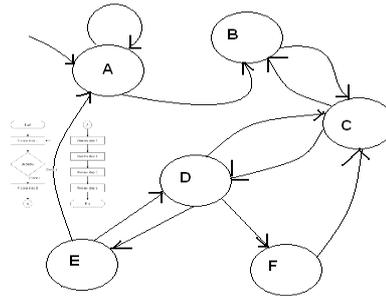
- Aspetto caratteristico dell'ambiente di esecuzione:
 - Event driven execution

Sequential VS Event handling programming

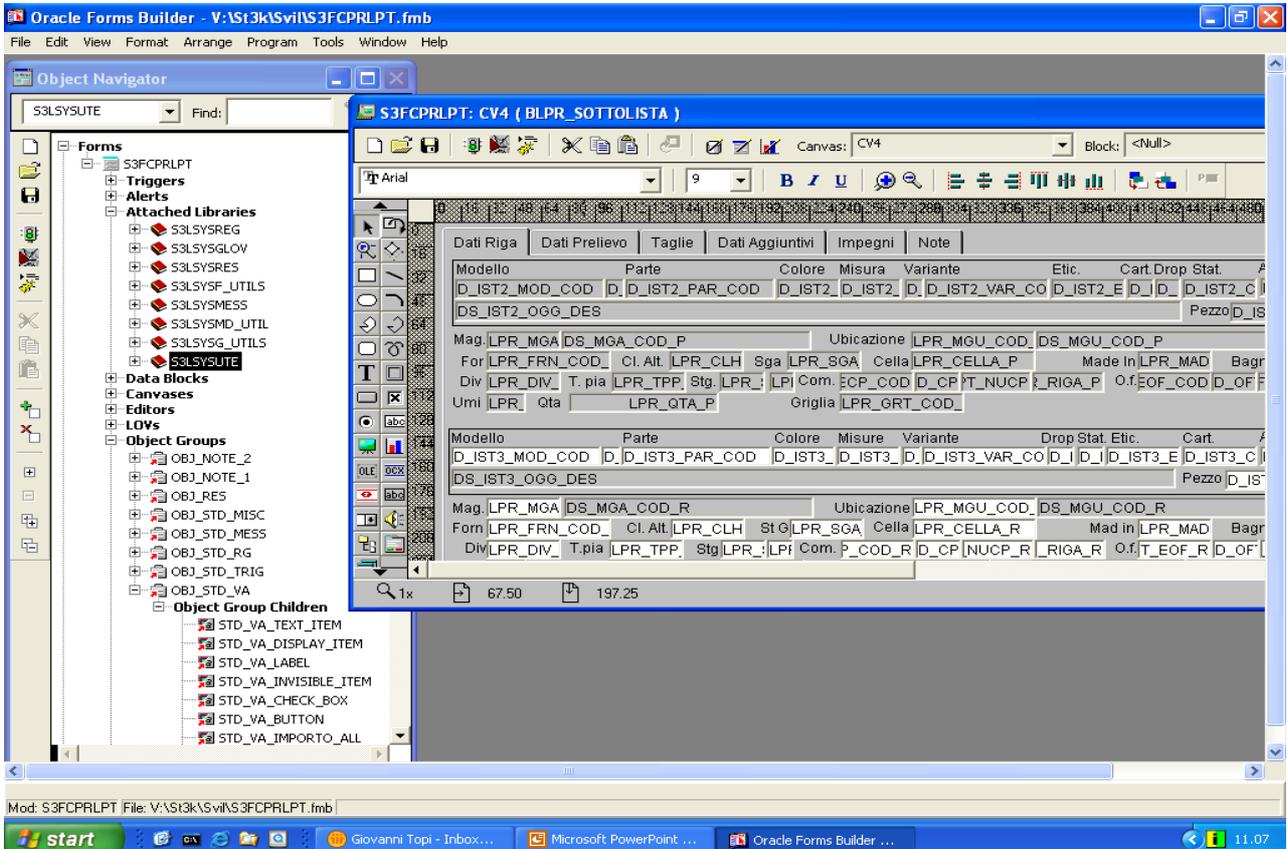
Sequential (flow chart)



Event driven (state diagram)



Una form (vista globalmente come "processo") non viene eseguita in questo modo.



3.5 Oracle Report Builder

Report Builder è il report-building componente di Oracle Reports Developer (un componente della Oracle Developer Suite), un potente strumento di reportistica enterprise che consente di sviluppare e implementare rapidamente sofisticati report Web e di carta da qualsiasi fonte di dati (tra cui un database Oracle, JDBC, XML, file di testo, e Oracle OLAP). Sfruttando le ultime tecnologie J2EE come JSP e XML, è possibile pubblicare i report in diversi formati (tra cui HTML, XML, PDF, testo delimitato, PostScript, PCL, e RTF) a qualsiasi destinazione (compresi e-mail, browser Web, OracleAS Portal, e il file system) in una soluzione scalabile, efficiente.

Riconoscere le differenze tra la pubblicazione Web e pubblicazioni cartacee, Report Builder offre la possibilità di sviluppare le produzioni di alta qualità per i requisiti di Web e e-business, così come alta fedeltà report stampati. Report Builder comprende:

- Procedura guidata user-friendly, che guida l'utente attraverso il processo di progettazione dei reports.
- Fonti di dati pluggabile PDSs, come JDBC e XML, che consentono di accedere ai dati da qualsiasi fonte per i reports.
- Un generatore di query con una rappresentazione grafica di SQL per ottenere i dati dei reports.
- Modelli di report predefiniti e stili di layout che possono essere personalizzati in caso di necessità.
- Un editore live che consente di modificare i layout di report di carta in modalità WYSIWYG.
- La possibilità di aggiungere output dinamico nel report di una pagina HTML inserendo i tag JSP all'interno di un documento HTML.
- Un costruttore grafico integrato per rappresentare graficamente i dati del report.
- Standard di formati di output report sono HTML, HTMLCSS, XML, PDF, RTF, foglio elettronico, PCL, PostScript e ASCII.
- La capacità di eseguire istruzioni SQL dinamiche all'interno di procedure PL / SQL.
- Il supporto per oggetti di database Oracle.
- Event-based reporting (esecuzione di report basati su eventi di database).

Un report è costituito da oggetti che insieme lo definiscono:

Oggetti del modello dati (query, i gruppi, le colonne, i collegamenti, i parametri utente).

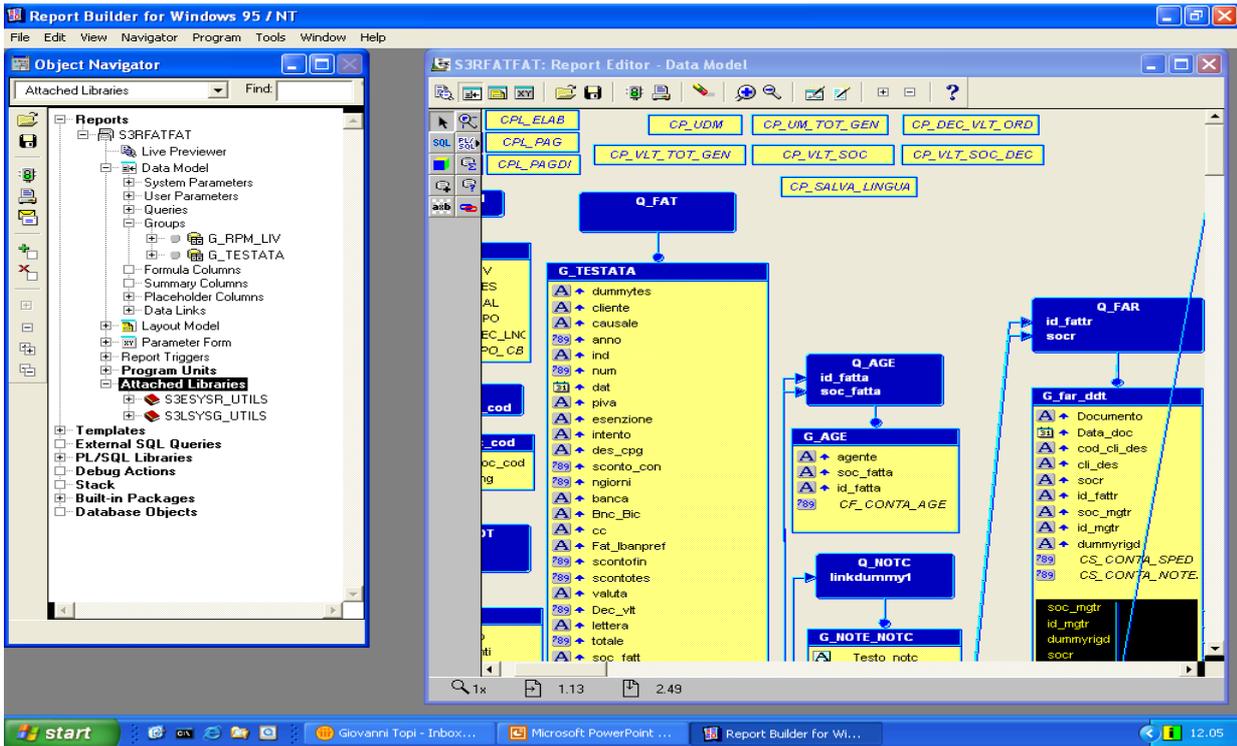
Oggetti del formato (frame ripetere, cornici, campi, boilerplate, ancore).

Parametri degli oggetti della form (parametri, campi, boilerplate).

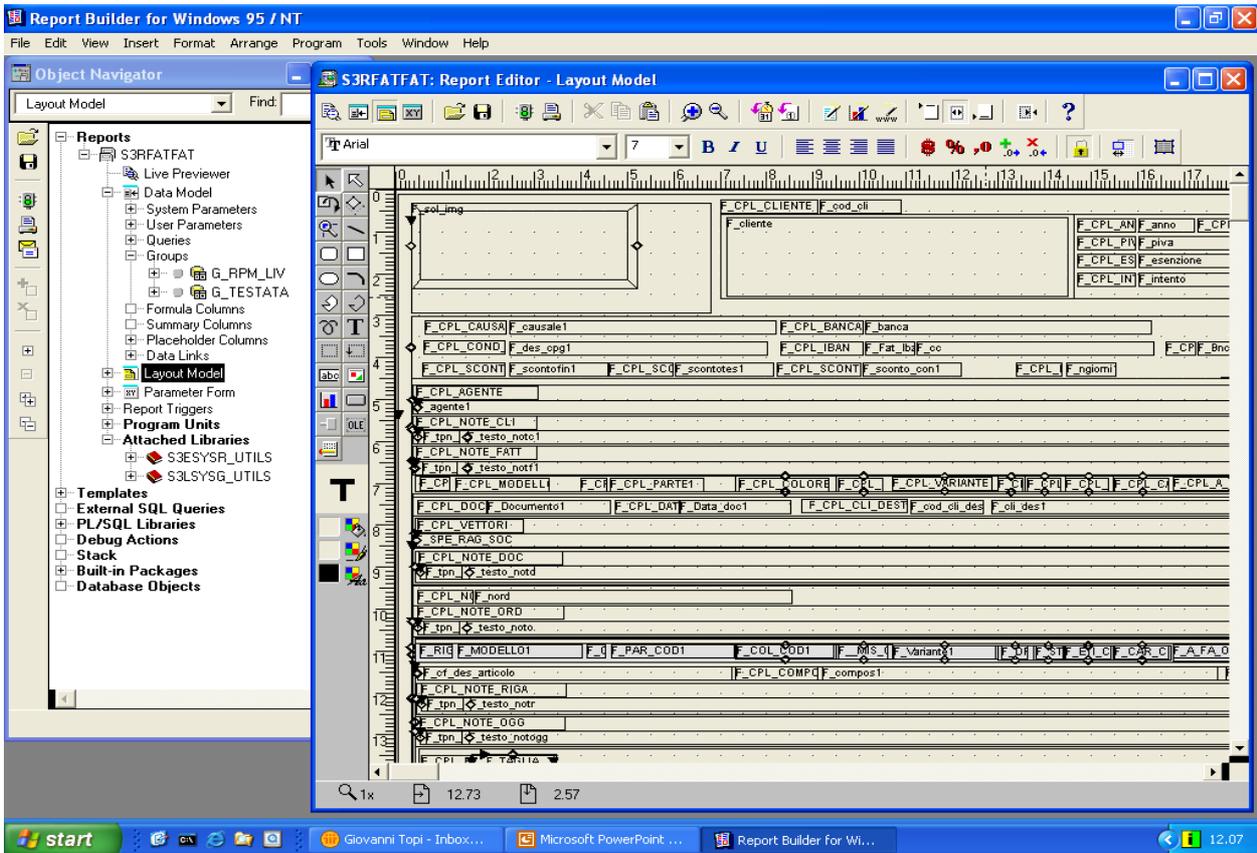
Oggetti PL/SQL (unità di programma, trigger) e riferimenti esterni come librerie PL/SQL.

Codice riportato nella visualizzazione Origine Web (JSP per report basati Web).

Utilizzando la finestra di ispezione Proprietà, è necessario definire le proprietà del report. La tassonomia del documento (classificazione) con proprietà (titolo, autore, oggetto e parole chiave) assiste nella catalogazione e la ricerca di un documento di report.



Oracle Reports Builder- Data Model



Oracle Reports Builder- Layout Model

3.6 Oracle JDeveloper 11g, ADF

Inoltre, la crescente necessità per le organizzazioni a costruire applicazioni composite che utilizzano i principi dello Service Oriented Architecture (SOA) ha costretto gli sviluppatori a creare applicazioni che sono estremamente agile. L'attuazione di queste buone pratiche in applicazioni agili di solito comporta la scrittura di un notevole quantità di codice di infrastruttura, aggiungendo un altro ostacolo per gli sviluppatori di costruire la loro prima applicazione Java EE. Oltre a fornire robusta, performante e applicazioni mantenibile, Oracle Application Development Framework fornisce anche codice per implementare un'infrastruttura agile per applicazioni basate su SOA eliminando lo sforzo e permettendo di aggiungere valore rispetto alla costruzione di infrastruttura.

Oracle ADF semplifica lo sviluppo Java EE riducendo al minimo la necessità di scrivere codice che implementa infrastrutture dell'applicazione che permette agli sviluppatori di concentrarsi sulle caratteristiche della domanda effettiva. Oracle ADF fornisce queste implementazioni di infrastrutture come parte del framework. Per riconoscere una serie di servizi di runtime non è sufficiente, Oracle ADF è anche incentrata sull'esperienza di sviluppo per fornire un approccio visuale e dichiarativo per lo sviluppo Java EE attraverso il tool di sviluppo Oracle JDeveloper 11g.

Oracle ADF implementa il Model-View-Controller modello di progettazione e offre una soluzione integrata che copre tutti gli strati di questa architettura con una soluzione a settori quali: Object / Relational Mapping, la persistenza dei dati, livello di controller riutilizzabili, framework di interfaccia Web utente, i dati binding con l'interfaccia utente, la sicurezza e la personalizzazione. Si estende oltre l'approccio Web MVC, ADF si integra inoltre con la SOA di Oracle e il framework WebCenter Portal semplificando la creazione di complete applicazioni composite. Per esempio, Oracle ADF rende più facile sviluppare applicazioni agili che espongono i dati come servizi accoppiando una interfaccia di servizio al built-in business services in ADF. Questa separazione di particolari business service implementa dettagli che vengono eseguita in Oracle ADF tramite metadati. L'uso di questa architettura basata su metadati consente agli sviluppatori di applicazioni di concentrarsi sulla logica di business e user experience, piuttosto che i dettagli di come i servizi sono accessibili.

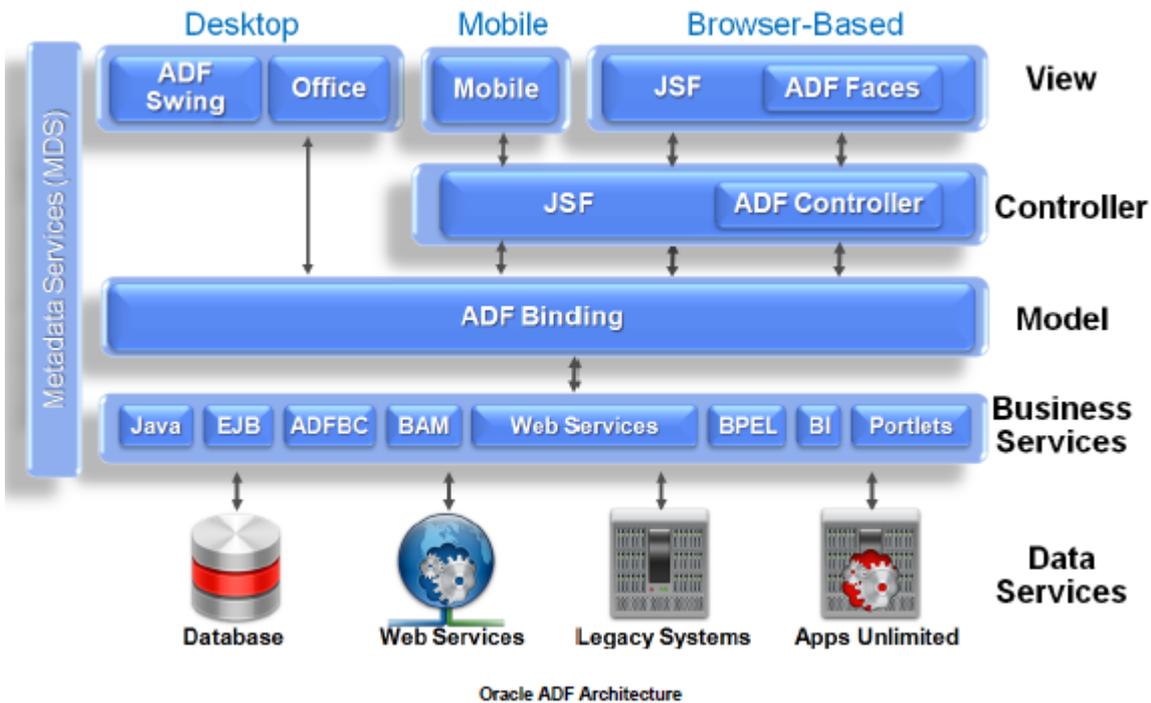
3.6.1 Architettura ADF

Oracle ADF si basa sul Model-View-Controller (MVC) design pattern. Un'applicazione MVC è suddiviso in:

- Uno strato modello (Layer MODEL) che gestisce l'interazione con fonti di dati e gestisce la logica di business.
- Uno strato di vista (Layer VIEW) che gestisce l'interfaccia utente dell'applicazione.
- Un controller (Controller) che gestisce il flusso dell'applicazione e funge da l'interfaccia tra il model e gli strati View.

Separando le applicazioni in questi tre strati semplifica la manutenzione e il riutilizzo di componenti tra le applicazioni. L'indipendenza di ogni strato dai risultati di altri in un loosely coupled, Service Oriented Architecture (SOA). Oracle ADF implementa MVC e separa ulteriormente il livello di modello dai Business Services per abilitare il servizio orientato allo sviluppo di applicazioni. La Oracle ADF architettura è basata su quattro strati.

Lo strato Business Services - consente di accedere a dati provenienti da fonti diverse e gestisce la logica di business. Il livello del modello - fornisce un livello di astrazione sopra dello strato di business service, consentendo agli strati View e Controller di lavorare con diverse implementazioni dei servizi alle imprese in modo coerente. Lo strato di controllo - fornisce un meccanismo per controllare il flusso dell'applicazione web. Il livello View - fornisce l'interfaccia utente dell'applicazione.



Oracle ADF consente agli sviluppatori di scegliere la tecnologia che preferiscono utilizzare in sede di attuazione di ogni strato. Il diagramma mostra le varie opzioni disponibili per gli sviluppatori durante la creazione di applicazioni Oracle ADF. La colla che integra le varie componenti di applicazioni Java EE e rende lo sviluppo in modo flessibile, è il modello Oracle ADF strato. EJB, Web Services, JavaBeans, JPA / EclipseLink / TopLink oggetti e molti altri possono essere utilizzati come servizi business per il modello Oracle ADF. Strati di vista possono includere interfacce basate su Web implementato con JSF, applicazioni Swing Desktop e MS front end Office, così come interfacce per dispositivi mobili.

3.6.1.1 The Business Services Layer

Lo strato Business Services gestisce l'interazione con uno strato di persistenza dei dati. Esso fornisce servizi come la persistenza dei dati, mappatura oggetti / relazionale, gestione delle transazioni, e la logica di esecuzione di business. Lo strato Servizi commerciali in Oracle ADF può essere implementato in una delle seguenti

opzioni: come semplici classi Java, EJB, servizi Web, oggetti APP, e componenti di Oracle ADF Business. Inoltre, i dati possono essere consumati direttamente da file (XML o CSV) e REST.

3.6.1.2 The controller Layer

Lo strato di controller gestisce il flusso di applicazioni e gestisce l'input dell'utente. Per esempio, quando si fa clic su un pulsante di ricerca in una pagina, il controllore determina l'azione da eseguire (fare una ricerca) e dove per la navigazione (la pagina dei risultati). Ci sono due opzioni di controllo per applicazioni web-based in JDeveloper: lo standard JSF controller o il controller di ADF, che estende la funzionalità del controller JSF. Qualunque sia il controller che si usa, in genere progettare il vostro flusso di domanda da parte del layout delle pagine e le regole di navigazione su un diagramma. Con il controller ADF è possibile interrompere il flusso dell'applicazione in unità più piccole, i flussi di lavoro riutilizzabili, comprendono componenti non visuali come le chiamate ai metodi e punti di decisione nel flusso e crea pagina "frammento" flussi che scorrono all'interno di una regione di una singola pagina contenente . Questo approccio favorisce il massimo riutilizzo dei frammenti di un'interfaccia utente semplificata e l'integrazione nei portali e le applicazioni mashup.

3.6.1.3 The View Layer

Il livello della Vista rappresenta l'interfaccia utente dell'applicazione. Oracle ADF supporta il multi-canale di accesso ai servizi di business che consente di riutilizzare i business services e accedervi da un client Web, un client-server un'applicazione desktop basata altalena, fogli di calcolo Microsoft Excel, o un dispositivi mobili come smart-phone. Per interfaccia basata su Web Oracle ADF offre un ricco set di oltre 150 Ajax abilitato componenti JSF che ha semplificato la creazione di interfacce utente dinamiche e accattivanti. Il modello a strati Il livello del modello collega i servizi alle imprese agli oggetti che li utilizzano in altri livelli. Oracle ADF fornisce un'implementazione strato modello che si trova sulla cima dei servizi alle imprese, fornendo una singola interfaccia che può essere utilizzato per accedere a qualsiasi tipo di servizi alle imprese. Il livello del modello è costituito da due componenti, controlli dei dati e le associazioni di dati, che utilizzano file di metadati per definire l'interfaccia. Controlli dati astratto i dettagli aziendali di implementazione del servizio da parte dei clienti. Associazioni di dati espongono i metodi di controllo di dati e attributi di componenti dell'interfaccia utente, fornendo una netta separazione della vista e modello. A causa della architettura dello strato di metadati modello.

3.6.2 Vantaggi di Oracle ADF Negli altri frame work

Le caratteristiche principali di Oracle ADF che lo rende unico tra gli altri framework Java EE sono:

- End-to-End Solution - Oracle ADF non si concentra su un solo strato dell'architettura Java EE. ADF fornisce una soluzione integrata e completa per ogni Java EE livello dal livello vista e data-binding, attraverso il business services e l'accesso ai dati, così come il supporto a ogni fase di sviluppo.
- Ambiente di sviluppo - Molti degli altri framework Java EE hanno una mancanza forte del supporto integrato dagli strumenti di sviluppo. Oracle JDeveloper fornisce supporti visivi e

un approccio dichiarativo per ridurre al minimo la necessità di scrivere codice del framework, che lo rende uno strumento perfetto per la costruzione di Oracle ADF-based. Questo approccio di sviluppo dichiarativo riduce anche la curva di apprendimento per gli sviluppatori familiari a strumenti come 4GL stile.

- Indipendenza della piattaforma – Altri framework bloccano gli sviluppatori a un fornitore di software specifico. Il runtime di Oracle ADF, tuttavia, può essere installato su diversi server compatibile con applicazioni Java EE e servizi business in grado di connettersi a qualsiasi database di SQL-92 compliant.
- Scelta della tecnologia - Gli sviluppatori hanno preferenze per il loro modo di implementare diversi livelli di un'applicazione. Oracle ADF supporta diverse tecnologie per ciascuno degli strati di applicazione e non applica una tecnologia specifica o uno stile di sviluppo specifico per lo sviluppatore.
- Impegno Tecnologia - E' importante notare che Oracle ADF è la scelta della tecnologia Oracle per la prossima generazione di applicazioni enterprise -Oracle Fusion Applications - ed è in uso continuo per scopi di sviluppo interno. Il prodotto è utilizzato per sviluppare applicazioni portale, applicazioni wireless e applicazioni web, e fornisce stack tecnologico sostenuto, e coerente.
- Metadata-Driven - Tutti i livelli del framework Oracle ADF offrono opzioni dichiarativo per lo sviluppo, configurato dai metadati XML, garantendo allo stesso tempo di codifica personalizzati ovunque sia necessario. È possibile scegliere di utilizzare in tutto o parte del quadro nelle applicazioni di costruire, rendendo l'applicazione molto più componenti riutilizzabili e flessibili. L'utilizzo di metadati consente inoltre regole per i campi dati associati da specificare a livello di modello. Etichette, la validazione, e le proprietà tooltip possono essere specificata nei metadati per ADF.
- Associazioni di dati - queste proprietà sono utilizzate indipendentemente dell'implementazione dell'interfaccia utente.
- Dichiarativa di personalizzazione - Oracle ADF fornisce una soluzione unica che consente a un'organizzazione di utilizzare una singola applicazione di base e personalizzarlo per soddisfare le esigenze di utenti diversi. Oracle ADF funziona in combinazione con un (MDS) strato che fornisce servizi di metadati per la personalizzazione dell'applicazione tramite due differenti livelli di implementazione: la prima, sarebbe “seeded customization” che si riferisce ad una personalizzazione delle applicazioni che sarebbe in vigore per chiunque acceda alla applicazione per un gruppo particolare. Il secondo è l'utente di personalizzazione spesso definito come personalizzazione in cui l'utente finale indica le personalizzazioni alla loro esperienza personale, che vengono poi persistito tramite il repository MDS.
- Riutilizzabilità rafforzata - JDeveloper + ADF offre il supporto per funzionalità riutilizzabilità superiori, tra cui: JSF template, flussi di lavoro riutilizzabili, il flusso di lavoro di modelli, servizi aziendali riutilizzabili, librerie ADF, regioni frammento JSF base, e molto, molto di più.
- Disponibilità source - Oracle fornisce il codice sorgente per il framework ADF per i clienti con una licenza di supporto. Avendo a disposizione la fonte può aiutare gli sviluppatori a comprendere i meccanismi alla base dei problemi del framework ed eseguire il debug nelle loro applicazioni.
- Supporto - Oracle ADF è un prodotto ufficiale di Oracle e come tale è seguito dalla organizzazione di supporto Oracle. Questo fornisce tutto il supporto di clock da una organizzazione stabile .
- Formazione - Oracle University offre regolarmente corsi su Oracle ADF e JDeveloper.

3.7 Strumenti di gestione & controllo e delivery.

3.7.1 BPEL

BPEL è lo strumento di Interfacciamento con il pacchetto esterno SAP. Stealth 3000 non implementando tutta la parte di Finance. La Business Unit ha prodotto questo software, per permettere la comunicazione e l'interscambio di dati con il strumento SAP, favorendo l'integrazione dei vari componenti del sistema informativo aziendale.

3.7.2 Zipper-SmartDeliverer

SmartDeliver è usato dopo la fase di sviluppo per la generazione e l'installazione dei programmi nei vari ambienti di test e di produzione per l'implementazione di nuove funzionalità. A fronte di una specifica richiesta del cliente di personalizzazione o di aggiunta di nuove utilità, segue uno studio funzionale (nella maggior parte dei casi insieme al cliente) per capire le esigenze, i punti d'impatto e la metodologia ad adottare. A questa poi, seguirà un'analisi tecnica per un'implementazione ottima delle funzioni. A seconda dell'intervento si può andare a modificare un package, una form o un report già esistente oppure partire da zero dello sviluppo di un modulo funzionale per l'applicazione usando gli strumenti Oracle citati sopra per produrre il codice necessario. Quando il programma è pronto in ambiente di sviluppo, per mezzo dello Zipper si genera i pacchetti da installare negli ambienti di Test (coppia molto vicina all'istanza Stealth 3000 del cliente). Durante la fase di testing, si valuta la funzionalità e le prestazioni del programma. Lo SmartDeliverer servirà in seguito per il rilascio finale in ambiente di Produzione, dal cliente.

3.7.3 PVCS

Originariamente sviluppato da Don Kinzer e pubblicato da Polytron nel 1985, attraverso una storia di acquisizioni e fusioni, il prodotto è stato a volte di proprietà di Sage Software, Intersolv e software Merant. Quest'ultima è stata acquisita da Serena Software nel 2004, che a sua volta è stata acquisita da Silver Lake Partners nel 2006. PVCS Version Manager (originariamente denominato Sistema Versione Polytron Control) è un pacchetto software di Serena Software, Inc. per il controllo di revisione dei file, in particolare i file di codice sorgente PVCS segue l'approccio "lock" per il controllo della concorrenza. PVCS può anche essere configurato per supportare più utenti contemporaneamente che tentano di modificare il file, in questo caso il secondo programmatore (cronologicamente parlando) avrà un 'Branch' creato per lui / lei in modo che entrambe le modifiche, invece di creare conflitto, apparirà come storie parallele per lo stesso file. Questo è diverso da CVS e Subversion dove il secondo programmatore ha bisogno di fondere prima i cambiamenti tramite il comando update e poi risolvere i conflitti (quando esistono) prima di fare commit. Il controllo versione ingegneristico si è sviluppato dai processi formali basati sui disegni cartacei.

Man mano che il software viene sviluppato e dispiegato, è sempre più probabile che versioni distinte dello stesso software siano dispiegate in posti diversi, e che gli sviluppatori del software lavorino privatamente allo sviluppo di aggiornamenti. I *bug* e altre questioni riguardanti il software sono spesso presenti solamente in certe versioni (a causa del fatto che man mano che il software evolve alcuni problemi vengono corretti e altri ne vengono rilevati). Pertanto, allo scopo di individuare e correggere i *bug*, è di importanza vitale per il programmatore poter recuperare e mandare in esecuzione diverse versioni del software per determinare in quali versioni il problema si

verifica. Può anche essere necessario sviluppare parallelamente due versioni del software (quando, per esempio, in una versione sono stati corretti dei bug, ma non ha nuove caratteristiche, mentre nell'altra versione vengono sviluppate le nuove caratteristiche).

Nella maggior parte dei progetti di sviluppo software, più sviluppatori lavorano in parallelo sullo stesso software. Se due sviluppatori tentano di modificare lo stesso file contemporaneamente, in assenza di un metodo di gestione degli accessi, essi possono facilmente sovrascrivere o perdere le modifiche effettuate contestualmente. La maggior parte dei sistemi di controllo di versione possono risolvere questo problema in due diversi modi. Questo problema riguarda solamente i sistemi di controllo di versione centralizzati, poiché i sistemi distribuiti permettono intrinsecamente più modifiche simultanee. Qualche operazione messa a disposizione da PVCS è:

Repository

Il repository è dove i file sono memorizzati, spesso su un server. Talvolta è chiamato anche depot.

Commit

Un **commit** (o, più raramente, **install**, **submit**, **check-in** o **ci**) si effettua quando si copiano le modifiche fatte su file locali nella directory (il software di controllo versione controlla quali file sono stati modificati dall'ultima sincronizzazione).

Modifica

Una **modifica** (**change**) rappresenta una specifica modifica ad un documento sottoposto al controllo di versione. La granularità delle modifiche considerate come cambiamenti varia tra i sistemi di controllo versione.

Change List

Su molti sistemi di controllo versione con commit di modifiche multiple atomiche, una **changelist** identifica un insieme di **changes** fatti in un singolo commit.

Check-Out

Un **check-out** (o **checkout** o **co**) effettua una copia di lavoro dal repository (può essere visto come l'operazione inversa dell'importazione).

Update

Un **update** (o **sync**) copia le modifiche fatte sul repository nella propria directory di lavoro (può essere visto come l'operazione inversa del commit).

Merge / Integrazione

Un **merge** o **integrazione** unisce modifiche concorrenti in una revisione unificata.

Revisione

Una **revisione** o **versione** è una versione in una catena di modifiche.

Import

Il termine **import** è usato per descrivere la copiatura dell'intero albero di directory locale sul repository.

Export

Un **export** è simile ad un **check-out** eccetto il fatto che crea un albero di directory vuoto senza metadati di controllo versione (spesso è usato precedentemente alla pubblicazione dei contenuti).

Conflitto

Un conflitto si presenta quando diversi soggetti fanno modifiche allo stesso documento. Non essendo il software abbastanza intelligente da decidere quale tra le modifiche è quella 'corretta', si richiede ad un utente di **risolvere** il conflitto.

Risolvere

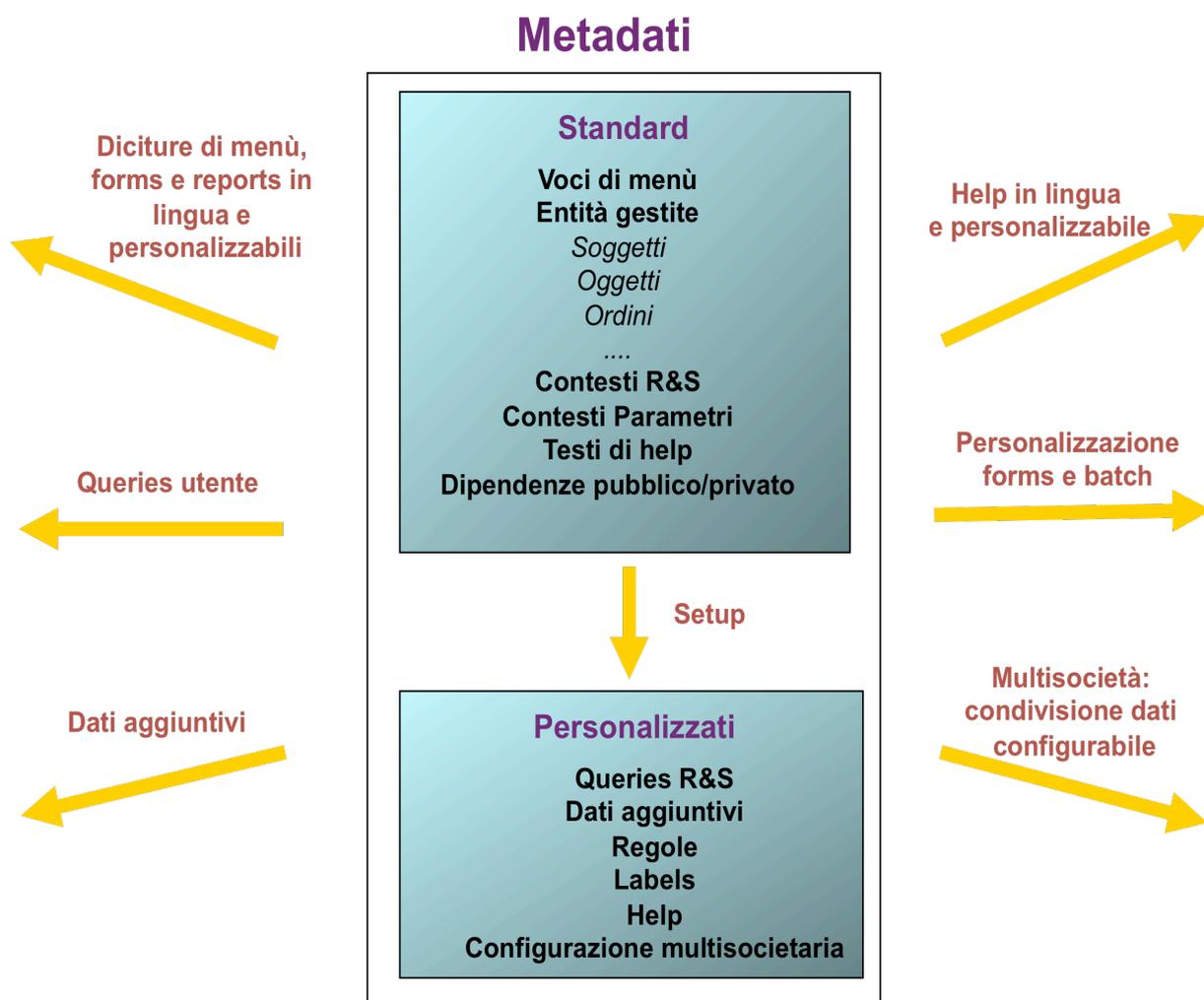
L'intervento di un utente per la risoluzione di un conflitto tra modifiche differenti di uno stesso documento.

3.8 Personalizzazione

La parte della personalizzazione è il fulcro dell'attività attuale nella Business Unit. Questa attività è sostenuta e facilitata dalla tecnologia con la quale Stealth 3000 è stato prodotto.

- Prima di tutto la tecnologia web, Interconnessione via protocollo TCP/IP su: Lan, Wan Intranet e Extranet, Internet. Permette di accedere all'applicativo anche tramite browser (Internet Explorer) favorendo la delocalizzazione, elemento non di poco importante nella gestione Intercompany.
- La Scelta di un Database relazionale che garantisce l'integrità dei dati, la referenzialità, e accesso completo tramite SQL. Con il modello relazionale abbiamo un spostamento della complessità dai programmi al DBMS, e un livello di prestazioni e sicurezza ottime rispetto ad altri sistemi.
- Stealth 3k è multilingua, permette di adattare il menù, l'help online e tutta l'interfaccia alla lingua del cliente.
- La Gestione Multisocietaria è implementata a livello Data Base con l'inserimento del Codice Società tra gli elementi della Chiave Primaria in ogni tabella. Più società sono gestibili all'interno dello stesso Database fisico con una gestione di accesso di utente a livello di: società di appartenenza, gruppi di appartenenza, funzioni autorizzate e operazioni ammesse per singolo campo garantendo la sicurezza e la riservatezza dei dati.
- I Metadati, Repository centralizzato dei componenti di sistema favorizza una gestione a alto livello della nomenclatura dei campi e dell'Help.
- I Dati aggiuntivi offrono la Possibilità di definire in tutti gli ambienti di gestione campi aggiuntivi personalizzati ad una tabella. In qualunque momento utilizzabili come i campi standard.
- Regole è Linguaggio dichiarativo "user oriented" per personalizzare senza scrivere programmi: i controlli attributi a video, gli impostazioni di default, i calcoli e esecuzioni di funzioni. Linguaggio prodotto internamente.
- R&S Modulo software progettato e realizzato nell'ambito di Stealth_3000 con lo scopo di: definire queries comunque complesse sul data base e definire le modalità di presentazione dei risultati della ricerca.

Con tutta questa tecnologia, Stealth 3K, offre un scenario abbastanza ampia di sviluppo e di adattamento alle esigenze del cliente, riducendo il tempo necessario per attuazione di nuove funzionalità.



3.9 Stealth nel Contesto del SI Aziendale

Stealth, essendo per definizione una delle componenti del sistema informativo aziendale, deve interfacciarsi con le altre applicazioni eventualmente presenti in azienda. L'interfacciamento riguarda sia la ricezione o l'invio di flussi di dati da e verso altre applicazioni (si pensi ad esempio al flusso degli ordini provenienti da un sistema di acquisizione remoto o al flusso delle fatture attive da inviare alla contabilità); sia le modalità di gestione e allineamento di stesse tipologie di informazione che tipicamente vengono gestite all'interno di più sistemi (si pensi ad esempio alle anagrafiche clienti e fornitori che sono gestite sia da Stealth che dal sistema informativo contabile). Tipicamente, l'analisi degli interfacciamenti con le altre applicazioni aziendali fa parte delle attività di progetto specifiche di una installazione. Stealth ha delle predisposizioni che agevolano la generazione o l'acquisizione di specifici flussi informativi e rende disponibile la struttura e la composizione degli archivi per rendere agevole la mappatura dei dati e l'analisi di specifici interfacciamenti come quelli, ad esempio, relativi alle anagrafiche dei soggetti o ad alcune tabelle tipicamente gestite in più applicazioni.

In particolare le predisposizioni riguardano:

per il Ciclo Attivo

- La generazione del flusso relativo alla fatturazione attiva
- La gestione del flusso relativo agli ordini Clienti

Per il Ciclo Passivo

- La generazione del flusso relativo alla contabilizzazione delle fatture passive.

Stealth 3000 dispone poi di un ambiente generalizzato di import/export di dati.

L'integrazione di Stealth con Oracle Finance prevede un "pacchetto" preconfigurato di software e approcci applicativi già finalizzati alla coesistenza dei due ambienti applicativi. Tale "pacchetto" seguirà un piano di rilasci coerente con i rilasci di Stealth 3000 per le componenti legate alla gestione dei dati e dei processi gestionali, seguendo comunque una logica di releases. Le integrazioni con gli ambienti di controllo e decisionali avverranno successivamente al secondo rilascio (orientativamente I trimestre 2003). In particolare, gli elementi su cui si articola il "pacchetto" di integrazione sono:

- L'identificazione dei dati che necessitano di una definizione compatibile tra i due ambienti:
 - Tabella Nazioni
 - Tabella Province
 - Tabelle di Classificazione
 - Tabella Aliquote IVA
 - Tabella Condizioni di Pagamento
 - Tabella Divise
 - Piano dei Conti
 - Centri di Costo
 - Anagrafica Banche
 - Multisocietà
- L'identificazione di processi gestionali che sono presenti in ambedue gli ambienti
 - Anagrafica Clienti
 - Anagrafica Fornitori
 - Contabilizzazione Ciclo Attivo
 - Controllo fatture fornitori
 - Provvigioni
 - Cambi
- La definizione delle modalità di alimentazione degli ambienti collegati ai processi di controllo e decisionali
 - Contabilità Analitica
 - Budget
 - Business Intelligence
- La predisposizione dei flussi di interfacciamento contabile
- Un modello di installazione e set up di entrambi gli ambienti che assicuri configurazioni compatibili sia a livello tecnico che funzionale

- La predisposizione di componenti software da utilizzare per la realizzazione di procedure di interscambio dei dati relativi a processi e archivi comuni
- L'identificazione, nell'ambito dei processi comuni ai due ambienti, della modalità con cui tali processi vanno ad interagire (es. processo master e processo slave).

Metodologia di Implementazione

La metodologia di implementazione è basata su CSC Catalyst:

- ***Discovery***
Analisi del Modello Aziendale, Mappatura della soluzione, Definizione dei requirements, Definizione del Diagramma di Contesto, Modalità di Start up.
- ***Design***
Prototyping, Configurazione e set up (Regole, Dati Aggiuntivi...), Individuazione delle personalizzazioni.
- ***Develop***
Sviluppo Personalizzazioni, Interfacce, Conversione Dati, Test.
- ***Deploy***
Training, Avviamento, Supporto.

Servizi di Supporto:

Help Desk, Corsi (utilizzatori e personale tecnico), Teleassistenza, e-business on demand, Hosting/outsourcing.

Possiamo dire che nel vedere e nell'usare questi strumenti in questi mesi, ho potuto considerare la loro importanza, utilità ed efficacia nell'implementare programmi complessi in poco tempo, direttamente integrabili col sistema.

Nel capitolo successivo presenteremo il Progetto GESTION del gruppo FASHION e la sua struttura, gruppo cliente della CSC s.r.l. Questo rappresenta il contesto specifico nel quale abbiamo svolto le nostre attività durante il tirocinio. Ho scelto tra le attività svolte, le due di maggior interesse: La generazione di codice a barre e la gestione di flussi di ordini Intercompany in FASHION. Inizieremo partendo dalla presentazione del contesto dei codici a barre e della gestione degli ordini prima della ristrutturazione societaria, poi seguiremo con lo stato dell'arte per i due argomenti e in fine la nostra scelta d'implementazione.

Capitolo 4: Il Progetto GESTION del Gruppo FASHION

L'industria della moda Italiana è conosciuta in tutto il mondo per la creatività, l'innovazione e lo stile. FASHION è una grande firma che ha estimatori in tutto il mondo, ed è sempre più sinonimi di qualità e convenienza.

Con la notorietà e il prestigio ottenuto a livello mondiale, negli ultimi anni, il Marchio oggi vede nell'aumentare dell'efficienza produttiva e nell'abbattimento dei costi le proprie priorità strategiche.

La sfida fondamentale, dunque, è rappresentata dalla capacità di tenere sotto costante controllo le singole fasi del processo manifatturiero, della scelta delle materie prime fino alla distribuzione dei prodotti finiti, per riuscire così a garantire, in tutte le fasi della produzione, la gestione ottimale di ogni singolo step.

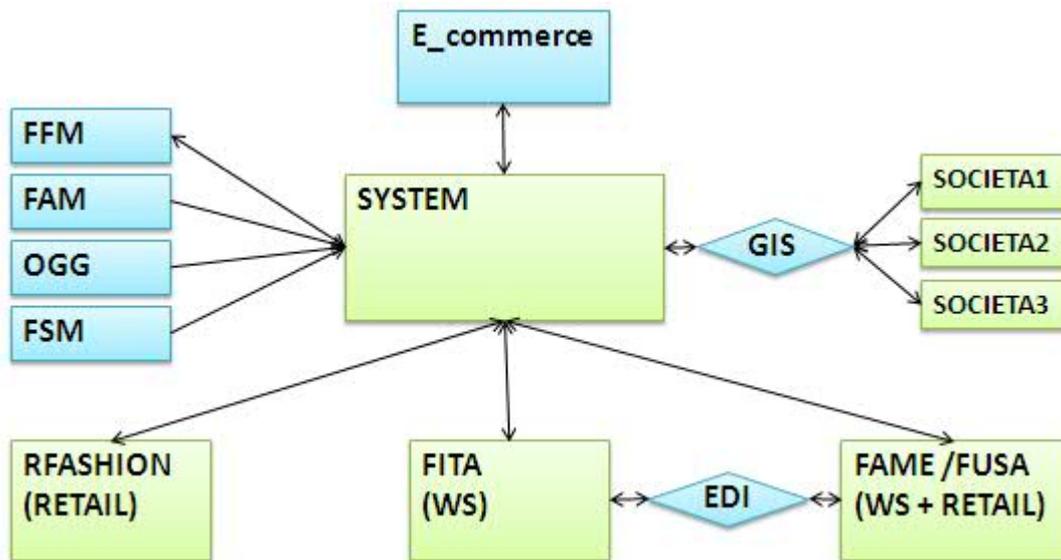
Se è vero, che le grandi firme tendono ancora a mantenere la produzione a livello locale per sfruttare al massimo il valore insito nel marchio "made in", è anche vero che molte altre aziende come FASHION, stanno percorrendo, con sempre maggiore frequenza, la strada della delocalizzazione produttiva nei paesi a basso costo, aumentando, di fatto, la complessità e l'estensione geografica della propria catena del valore.

In tal contesto, nel 1997, nasce il rapporto tra la CSC e FASHION con Stealth 400. All'Information Technology viene chiesto di trovare soluzioni che permettano di relazionare i dati di vendita dispersi geograficamente con processi a monte della supply chain. La disponibilità in tempo reale d'informazioni puntuali, anticipare le richieste (domande) e pianificare in modo efficace l'intero ciclo produttivo (offerta).

Nel 2000 viene lanciato Stealth 3000, sette anni dopo, anche la GF sceglie di migrare sul nuovo applicativo. Questo ultimo offre oltre alle funzionalità gestite in S400, una maggiore integrazione nel Sistema Informativo Aziendale grazie alla tecnologia Oracle, web, la gestione multisocietaria, la R&S, l'introduzione di dati aggiuntivi nelle tabelle. In somma, un sistema arricchito che risponde alle esigenze aziendali attuali e una piattaforma solida su cui continuare a costruire.

Prima della ristrutturazione societaria, il Gruppo FASHION era formato da diverse società a se stanti che riferivano al gruppo. Con l'introduzione di GESTION FASHION Total Operazione Management nel 2008 che rappresenta la gestione dell'intero sistema informativo aziendale (Stealth, Oracle finances...), sono state introdotte dei cambiamenti nei vari flussi aziendali. Noi affronteremo le modifiche che sono state fatte in particolare al flusso di Ordini Intercompany.

4.1 Architettura di GESTION



Terminologia

FITA Distribuzione Italia.

WS Wholesales.

OGG Oggetti, Articoli.

RFASHION Retail Europe.

FAME FASHION America.

EDI Electronical Data Intergange.

GIS Sistema esterno per la gestione del trasporto file dal SYSTEM verso le Industrie.

INDUST Industrie del Gruppo FASHION.

CANALI E_COMMERCE sistema esterno per E-commerce.

Stealth essendo una parte del sistema informativo del Gruppo FASHION, il numero d'istanza di database rispecchia l'organizzazione aziendale. Ci sono esattamente 4 istanze di produzione¹:

- FITA (WholeSales) sistema locale che si occupa della raccolta degli ordini. Raggruppa le società GF Swiss Branch, GF Spa, e la GFO FASHION Operations. Per il Gruppo ci sono tre canali di vendita. Uno sono gli Agenti, che contrattano direttamente i negozi e presentano campioni. Gli agenti possono presentare anche i campionari nei loro Showrooms e come sistema usano FSM FASHION Sales Management che fa da front office e di validazione di tipi standard di Stealth.

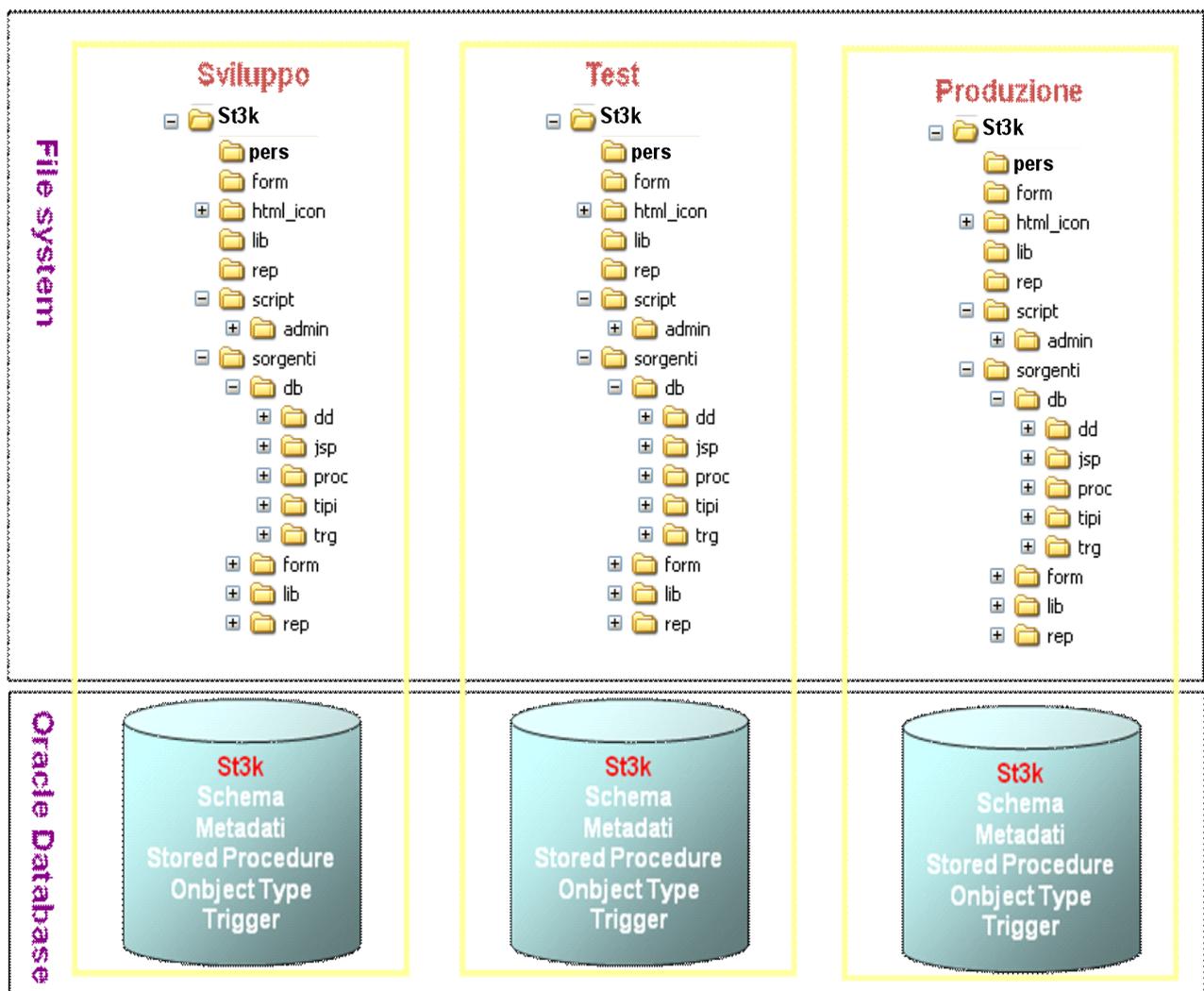
In secondo c'è la vendita diretta dai Showrooms FASHION durante le sessioni di vendita, dove ci sono più uscite e delle sfilate. A queste occasioni, il cliente può fare un'autocomposizione del suo ordine. Per ultimo abbiamo la generazione di ordini via Web, o vendita Web. Lo strumento usato da FASHION è un servizio esterno offerto dalla Canali e_commerce, che mette a disposizione siti Web di e-commerce (Commercio elettronico) e funziona come distributore.

- RFASHION è il sistema che si occupa di tutta la gestione dei negozi di proprietà e delle movimentazione di merce (Giacenza, venduto...). Tutte le altre attività di Back Office come la gestione di Pos, degli acquisti, dell'anagrafe negozi sono gestite in questo sistema.
- FAME sistema locale per la zona americana con società GF Corp, GF Brasile, GF Canada, raggruppa le funzionalità sia del Wholesales sia del Retail.
- SYSTEM è il sistema main o centrale che accorpa i vari sistemi locali. Attraverso l'uso delle code Oracle e il Datapump vengono aggiornati e allineati i sistemi. Le Code Oracle e il Datapump sono dei meccanismi messi a disposizione dalla tecnologia Oracle per portare i dati in modo automatico di un sistema a un altro. Questa propagazione è disponibile nelle due direzioni, il controllo della consistenza funzionale di quest'operazione è un dato importante nella gestione funzionale dei flussi aziendali. La gestione dei Business Object privati e pubblici e dei dati proprietari ci fornisce un'idea sulla direzione della propagazione. In modo generale ci sono gli Oggetti che sono pubblici. Quando viene creato un Business Object (Oggetto, Soggetto) viene inserito nel sistema SYSTEM e con un sistema di trigger e di listener per propagazione puntuale e datapump per propagazione massiva, viene esportato nel sistema locale di competenza rispettando la cronologia. Il database gestisce un flag di annullamento logico, inserito al livello di tabella, per fare sì che se su un sistema locale viene cancellata l'anagrafe di un soggetto o oggetto, viene attivato quel flag e il sistema non farà più riferimento a questo dato anche se presente nel database. La Propagazione dei dati tiene conto della gestione pubblica e privata dei dati. Se per gli oggetti, le anagrafiche sono pubbliche per tutti i sistemi locali, non è la stessa cosa per i dati anagrafici dei soggetti o dei magazzini che appartengono a un sistema locale. Dunque dal SYSTEM la propagazione di questi ultimi sarà solo sul sistema locale di riferimento.

¹ Istanza di produzione verrà spiegato nel paragrafo successivo.

Il flusso di propagazione permette di avere/concentrare sul SYSTEM la gestione centralizzata di tutte le attività del Gruppo multisocietario permettendo anche una certa indipendenza ai sistemi locali. La comunicazione al livello di SYSTEM non è solo con i sistemi locali, ma anche con le Industrie del Gruppo come SOCIETA1, SOCIETA6, SOCIETA2. Questa comunicazione viene attuata con il sistema GIS. Il SYSTEM è il sistema che si interfaccia con sistemi esterni per l'inserimento di nuovi soggetti con FAM FASHION Anagrafica Master, di nuovi oggetti con le anagrafiche OGG che provengono dalla produzione, con CANALI E_COMMERCE, FFM FASHION Finance Management e FSM FASHION Sales Management.

4.2 Installazione Stealth 3000 per FASHION



Rappresentazione di una installazione di St3k

In FASHION come per ogni cliente, abbiamo 3 istanze di St3k per ogni sistema. Queste istanze sono allo stesso livello di aggiornamento e sono organizzate in questo modo:

- St3k Sviluppo

Questa istanza si trova in sede, dove specialisti e tecnici IT continuano di sviluppare nuovi programmi da integrare al sistema standard e personalizzato. È il punto di partenza di ogni nuovo package, form o report. In particolare per FASHION ci sono circa 30 sviluppatori che producono funzionalità personalizzate per il Gruppo. I dati contenuto in questo sistema è poco simili ai dati reali di produzione dal cliente, il punto importante è coerenza della struttura di essi. A questo livello non sempre si riesce a capire gli impatti di tempo di esecuzione, soprattutto su della attività molto complesse. Per questi ultime, è solo durante la fase di test che si riuscirà a ottimizza al meglio il codice.

- St3k Test

Come chiamata, questa è l'istanza di test. Quest'ambiente riceve dall'ambiente di sviluppo i pacchetti Non Consegnati (.NC) tramite SmartDelivery, che devono essere testati. St3k Test è maggiormente simile al istanza del cliente al 75%. Dopo i tests, se tutto è andato a buon fine, viene fatto i pacchetti da consegnare nell'ambiente di produzione. Nel caso ci sono delle modifiche da fare o delle ottimizzazioni, si riparte al livello di sviluppo con la revisione del codice prodotto.

- St3k Produzione

Questa istanza rappresenta il database effettivo del cliente, è l'ambiente di lavoro corrente dal cliente, contiene dati aggiornati e tutte le nuove funzionalità sviluppate e testate. L'installazione e l'aggiornamento di nuovi packages in questo ambiente avviene a freddo cioè quando i sistemi sono a riposo o in funzionamento ridotto ad esempio alle 10 di sera di domenica. Nei casi di urgenza sono previste delle installazioni spots per risolve problemi puntuali.

- St3k Training

Questa istanza viene installata a fronte di necessita di formazione sia da parte del personale IT aziendale che del personale amministrativo del cliente. Questa formazione verte a fornire al personale citato, le capacità di usare St3k e una piene autonomia per quei clienti che altro al personale funzionale, ha anche un personale/squadra IT qualificato. Il Training parte all'avvio di un nuovo progetto e per altre volte all'introduzione di una serie di modifiche o aggiornamenti al sistema, o ancora a richiesta del cliente per formare nuovi utenti o per aggiornamento.

4.3 Stealth 3000 Multisocietà

ST3K permette la gestione di un Ambiente Multisocietario: un unico Data Base contiene tutti i dati (archivi, files e tabelle) relativi alle “n” Società costituenti il Gruppo o la Corporate. Gli archivi possono essere Societari o Pubblici:

- un archivio Societario, contiene informazioni appartenenti ad un’unica Società. Solo questa Società è abilitata a consultare, inserire, mantenere, annullare i dati presenti nell’archivio; le altre Società NON possono “vedere” l’archivio societario.
- un archivio Pubblico contiene informazioni disponibili a tutte le Società che, aderendo ad una Configurazione, hanno di fatto deciso di operare in modalità pubblica sullo specifico archivio; l’archivio non è visibile alle Società che hanno scelto di avere lo stesso archivio come societario. Ogni “oggetto”, fisicamente inteso come record (o occorrenza), presente nell’archivio pubblico, pur essendo disponibile alle N Società in Configurazione pubblica, appartiene alla Società che lo ha inserito: tale Società è l’unica abilitata a modificarne i dati.

Si sottolinea che un Archivio, nel Data Base, contiene i dati di tutte le Società della Corporate, sia quelli configurati “pubblici” che quelli “societari”; una Società, che ha deciso di configurare un archivio come societario, però, “vedrà” solo i propri dati come se fisicamente l’archivio contenesse solo quelli. La maggior parte degli archivi di ST3K sono per definizione solo societari (Listini di Vendita, Ordini Clienti, Ordini Fornitori, Commesse di Produzione, ...). Gli archivi che le Società possono condividere sono fondamentalmente quelli contenenti i dati statici: Oggetti, Soggetti, Distinte Base, Cicli di Lavoro, alcune Tabelle.

La scelta tra societario e pubblico non avviene di solito per singoli Archivi, ma per Gruppi predefiniti e gerarchici di Archivi: ad esempio, se l’Anagrafica Parti è definita pubblica, allora anche le Misure (delle Parti), i Colori (delle Parti), le Descrizioni in Lingua (delle Parti) ..., devono essere pubblici. In questo caso i Colori degli Oggetti pubblici devono appartenere ad un’unica Tabella Colori pubblica. Le Tabelle o gli Archivi che controllano un archivio devono essere pubblici se l’archivio è pubblico, pubblici o societari se l’archivio è societario. Ad esempio se l’Archivio degli Attributi Societari è pubblico, le Tabelle di decodifica di Divisione Produttiva e Linea di Vendita sono obbligatoriamente pubbliche; se è societario le rispettive Tabelle possono essere pubbliche o societarie. E’ possibile scegliere a livello di singola Tabella di decodifica se impostarla pubblica o societaria. Ogni Tabella pubblica, può essere gestita interamente da una sola Società (anche se i dati sono disponibili a tutte le Società); in alternativa la proprietà della Tabella può essere sezionata per occorrenza (record) tra le Società che la condividono.

Con questo vogliamo fare notare una particolarità della gestione multisocietaria di FASHION. Per il Gruppo ci sono più di una istanza del data base, questo fatto se di una parte ha degli lati positivi , ne ha anche quelli “meno positivi”. I lati positivi sono la duplicazione dei dati per una gestione del Disaster Recovery, una gestione centralizzata dei sistemi con una indipendenza al livello locale. Come punti meno positivi, il costo per l’allineamento dei dati tra i diversi sistemi.

Capitolo 5: Ristrutturazione societaria del Gruppo

La svolta epocale nella struttura del Gruppo GF è stata quella di assorbire nella GFO FASHION Operations tutte le altre società del Gruppo. Tutte le società del gruppo diventano divisioni che sono tutte servite dall'unico magazzino: la GFO magazzino del gruppo. Con la nascita di nuove società GF Corp, GF Brasile, GF Canada, GF JP Japan, GF HK Hong-Kong, la ristrutturazione a come obiettivo l'unificazione e la riduzione dei nuovi flussi aziendali.

Prima se avevamo società all'interno del Gruppo: una società distributrice italiana GF Spa, una distributrice estera GF Swiss Branch e società produttrici le Industrie, adesso sono tutte divisioni l'interno della società GFO. I cambiamenti introdotti vanno a impattare anche sulla gestione degli ordini e sulla gestione degli articoli. Se prima, un ordine poteva contenere articoli di una sola produttore, adesso un ordine può essere multi produttore e cioè di più di una divisione. Gli ordini multi produttori, sulle quali si basano i rapporti tra divisione, sono possibili perché tutte le divisioni versano la merce nell'unico magazzino del gruppo. Per gli articoli, avendo un unico magazzino, bisognerebbe rivedere la gestione dell'identificativo univoco di questi: i codici a barre. I cambiamenti sono tanti, abbiamo scelto di circoscrivere il nostro raggio di azione in questi due punti: Barcode e Ordini intercompany, che sono le attività più importante che abbiamo svolto durante il tirocinio.

5.1 Codice a barra

Esistono svariati tipi di codici a barre, con caratteristiche anche radicalmente differenti. Nel corso degli anni sono state proposte molte tecniche di codifica, alcune delle quali sono sparite con il passare del tempo mentre altre si sono evolute; attualmente solo una decina sono ampiamente utilizzate. Tra queste alcune sono presente nella realtà quotidiana di chiunque, altre sono destinate ad applicazioni specialistiche.

I tipi di codici a barre usati dal Gruppo sono IND Barcode Read (Industries), SERVIC Pos Service Barcodes, RETEK Barcod RETEK US, GTIN13 Global Trade Item Number, UPC UPC (US) Barcode, FASHION13 Barcode FASHION13, 001 Barcode FASHION. La presenza di numerosi tipologia di codice a barre rendeva difficile la gestione degli articoli all'interno del Gruppo a causa di diversi fattori, tra cui la distribuzione su scala globale, la presenza di diversi fornitori di merce, e di differenti società di servizi e l'elevata quantità di merce. I vendor e i retailer trovano sempre più complessa gestire l'insieme degli oggetti.

5.1.1 Checkpoint Systems Spa Open EPC Number

Checkpoint Systems, leader mondiale per la gestione delle differenze inventariali, la visibilità della merce e le soluzioni di etichettatura dei capi d'abbigliamento offre delle soluzioni per migliorare la gestione dell'inventario, ridurre i casi di rotture di stock ed il capitale circolante, e accrescere le vendite, nonché la sfida dei vendor e dei retailer di gestire i singoli codici di etichetta necessari per

ciascuno degli articoli tra le centinaia di milioni presenti. “In un ambiente retail – afferma Prasad Putta, Executive Vice President della divisione Merchandise Visibility Solutions di Checkpoint Systems - ogni articolo necessita di un numero univoco, costituito dal Global Trade Item Number (GTIN), che identifica il tipo di prodotto, e dai dati di serializzazione, che lo definiscono in maniera univoca. Ma – prosegue Putta – a causa di diversi fattori, tra cui la distribuzione su scala globale, la presenza di diversi fornitori di merce, e di differenti agenzie di servizi e l’elevata quantità di merce, i vendor e i retailer trovano sempre più difficile gestire e assicurare numeri unici nelle loro operazioni. Tuttavia questo aspetto è fondamentale per garantire l’integrità dei dati.

5.1.2 GESTION24

Con l’introduzione di GESTION e la ristrutturazione societaria del Gruppo, si è pensato di introdurre un nuovo codice a barre per la gestione interna degli articoli. Questo nuovo codice a barre di nome GESTION24 di lunghezza 24 è costruito sulla base del GTIN13 Global Trade Item Number. Il suddetto codice a barre serve a tutte le industrie del Gruppo per identificare univocamente ogni Articolo prodotto. Nelle prossime righe cercheremo di presentare in dettaglio questa nuova funzionalità e il programma che mette a disposizione nuovi range di GESTION24 alla richiesta di un nuovo GTIN13.

5.1.3 GTIN Global Trade Item Number

Prima di tutto Il GTIN (Global Trade Item Number), è l’ex codice EAN, che semplifica l’automazione dei processi commerciali (in sostanza acquisto e vendita). Per questo i GTIN sono assegnati ad ogni articolo (prodotto o servizio) che possa comparire in un listino, un ordine o una fattura in qualsiasi punto di qualunque supply chain, e vengono utilizzati per risalire a informazioni pre-definite sull’articolo. Il termine indica il codice GS1 assegnato alle unità consumatore, unità imballo (o trade unit) o unità logistiche ovvero a tutte le unità commerciali sulle quali è possibile recuperare informazioni predefinite, utili per tutte le operazioni commerciali. Un GTIN può utilizzare la struttura di numerazione standard GTIN-8, GTIN-12, GTIN-13 oppure GTIN-14. In GESTION viene usato il GTIN-13, e l’GESTION24 viene costruito in seguito sulla base di un GTIN-13.

Il GTIN dell’unità consumatore è così composto:

- Prefisso Aziendale GS1: è costituito dal Prefisso Nazionale GS1 e dal Codice Proprietario del Marchio.
- Codice prodotto: ogni azienda può identificare fino a 1.000 prodotti.
- Cifra di controllo calcolata in base alle altre presenti nel codice.

Per utilizzare il sistema di codifica GS1 è necessario aderire a Indicod-Ecr.

Le industrie, mediante l'invio di un apposito file contenente gli attributi necessari e tipo operazione 1 (parametro della funzione per indicare l'operazione da eseguire), richiedono ad GESTION la generazione dei barcode GTIN13 per i loro articoli. In produzione, ogni mezz'ora, si avvia il batch PGGFKGTINTRC che verifica i sistemi abilitati all'invio delle richieste PGGFT_ESS_FLUSSI e per ciascuno di essi importa dai vari server ftp configurati nella tabella S3TW_ESS i file da elaborare.

Ogni file viene inizialmente caricato in una tabella di appoggio PGGFTI_GTIN_TRC_IEX che non viene mai cancellata perché serve da storico per determinare, incrociando i dati con la tabella PGGFT_ESS_FLUSSI_ELAB, in caso di barcode mancanti, se effettivamente le industrie hanno inviato la richiesta e/o vi sono stati errori.

Per ogni record della tabella PGGFTI_GTIN_TRC_IEX i cui campi contengono gli attributi dell'articolo (nella tabella S3T_CBA Parametri di configurazione del Barcode) viene innanzitutto verificato nella tabella S3T_BAR Barcode istanze oggetti, che non esista già un GTIN13 associato: se esiste viene restituito nel file di output che torna all'industria il barcode esistente, in caso contrario viene creato e inserito un nuovo GTIN13 (PGGFKBARUTILS Utility per la generazione/gestione/stampa Bar Code). I barcode GTIN13 e GESTION24 nella maggior parte dei casi nascono prima che esista l'istanza. Le anagrafiche che giungono dalle industrie percorrendo una strada diversa, e spesso pervengono ad GESTION successivamente all'arrivo delle richieste di creazione/registrazione dei barcode in questione.

La funzione PL/SQL che provvede a fare questa operazione è la Funzione SEARCH_BAR del nostro Package PGGFKINTGTIN13 che segue i standard di naming e di programmazione all'interno del progetto, parte non meno importante per una piena integrazione con tutto il sistema. La funzione prende come input la chiave logica della tabella S3T_BAR codice modello, combinazione, parte, colore, misure, tipo variante, variante, drop, statura, griglia e taglia. Ha come output se presente il codice a barre GTIN13.

NB: Come Standard nel l'ambito di sviluppo delle funzioni pl/sql per ST3K, abbiamo scelto come return code, un PIs_integer che può essere 0 per indicare un esito positivo della funzione oppure 1 per un esito negativo.

```
Function search_bar(p_bar_cod_gtin13 out s3t_bar.bar_cod%type,
                  p_mod_cod          in s3t_bar.bar_mod_cod%type,
                  p_cmb_cod          in s3t_bar.bar_cmb_cod%type,
                  p_par_cod          in s3t_bar.bar_par_cod%type,
                  p_col_cod          in s3t_bar.bar_col_cod%type,
                  p_mis_cod          in s3t_bar.bar_mis_cod%type,
                  p_tpv_cod          in s3t_bar.bar_tpv_cod%type,
                  p_var_cod          in s3t_bar.bar_var_cod%type,
                  p_drp_cod          in s3t_bar.bar_drp_cod%type,
                  p_stt_cod          in s3t_bar.bar_stt_cod%type,
                  p_grt_cod          In s3t_tgl.tgl_grt_cod%Type,
                  p_tgl_cod          in s3t_bar.bar_tgl_cod%type)
return PIs_Integer;
```

;

Il barcode GTIN13 viene generato estraendo un progressivo dal primo range disponibile (PGGFT_GTIN_RNG) al quale viene concatenata il check digit calcolato.

Prima di toccare i range che sono a pagamento la funzione atomica che crea il GTIN va a verificare se vi sono dei barcode disponibili nella PGGFT_GTIN_REC: la tabella dei recuperati: se c'è un barcode viene usato quello, altrimenti lo crea. Per la creazione del nuovo barcode, si verifica che siamo sul sistema SYSTEM, in seguito l'esistenza della referenza a parità di griglia, taglia e drop del articolo di cui si vuole generare il codice a barre.

Essendo che ci sono più di una industria e momenti diversi di richiesta di barcorde, abbiamo implementato il meccanismo di semaforo sulla tabella S3T_BAR per potere leggere in mutua esclusione durante la ricerca del barcode se già inserito o l'inserimento del nuovo.

/*-----

Autore : CKamdem

Scopo : creazione barcode GTIN13

Parametri : p_bar_cod_gtin13 => parametro di output: codice del GTIN13 generato

p_mod_cod

p_cmb_cod

p_par_cod

p_col_cod

p_mis_cod

p_tpv_cod

p_var_cod

p_drp_cod

p_stt_cod

p_grt_cod

p_tgl_cod

p_rng_top => tipo di range da utilizzare: passare 'I' o 'M'

p_soc_cod => è possibile omettere di valorizzarli in chiamata

p_utente => è possibile omettere di valorizzarli in chiamata

Valori di ritorno : 0 => gtin creato correttamente (p_bar_cod_gtin13 valorizzato)

- 1 => gtin già presente nella s3t_bar (p_bar_cod_gtin13 valorizzato)
- 2 => si è tentato di chiamare la funzione da un local
- 1 => referenza non trovata
- 2 => griglia/taglia non trovate
- 3 => drop non trovata
- 4 => errore inserimento s3t_bar
- 5 => range GTIN esaurito

-----*/

```
Function get_gtin13(p_bar_cod_gtin13 out s3t_bar.bar_cod%type,
    p_mod_cod      in s3t_bar.bar_mod_cod%type,
    p_cmb_cod      in s3t_bar.bar_cmb_cod%type,
    p_par_cod      in s3t_bar.bar_par_cod%type,
    p_col_cod      in s3t_bar.bar_col_cod%type,
    p_mis_cod      in s3t_bar.bar_mis_cod%type,
    p_tpv_cod      in s3t_bar.bar_tpv_cod%type,
    p_var_cod      in s3t_bar.bar_var_cod%type,
    p_drp_cod      in s3t_bar.bar_drp_cod%type,
    p_stt_cod      in s3t_bar.bar_stt_cod%type,
    p_grt_cod      In s3t_tgl.tgl_grt_cod%Type,
    p_tgl_cod      in s3t_bar.bar_tgl_cod%type,
    p_rng_top      in pggft_gtin_rng.gtin_rng_top%type,
    p_soc_cod      in varchar2 default '001',
    p_utente       in varchar2 default 'GET_GTIN13')
return Pls_Integer ;
```

In concomitanza con la creazione del GTIN13 avviene anche la creazione del barcode GESTION24 di DEFAULT costituito dalle 13 cifre del gtin alle quali vengono concatenata una stringa fissa di 11 caratteri: '10000000000'.

L'GESTION24 di DEFAULT è la matrice di partenza dalla quale le industrie all'interno del Gruppo inizieranno a costruire i barcode GESTION24 per i loro prodotti.

Le richieste di registrazione degli GESTION24 vengono fatte pervenire ad GESTION da parte delle industrie mediante il solito file contenente record con tipo operazione '3'.

Se l'industria invia più di una richiesta di registrazione dello stesso GESTION24 o per la stessa combinazione modello, combinazione, parte, colore, misure, tipo variante, variante, drop, statura, griglia e taglia viene segnalato un errore nel file di output.

Di seguito la funzione per l'inserimento del GTIN13 e la creazione dell'GESTION24.

```
Function insert_bar(p_row_barcode_gtin13 in out nocopy s3t_bar%rowtype)
```

```
return Pls_Integer;
```

Per ogni articolo, GESTION24 viene generato estraendo il progressivo dell'ultimo codice usato all'interno dell'industria. Questa funzionalità rende completa e efficiente, la gestione dell'univocità degli articoli tra i vari fornitori e distributori del Gruppo al livello globale.

5.2 Flusso Ordini Intercompany

In Stealth 3000, il modulo degli Ordini Clienti permette principalmente di gestire l'inserimento di ordini clienti, di diverse tipologie e mediante varie modalità di acquisizione ordine, e di seguirne l'evoluzione a fronte di variazioni, storni, assegnazioni e consegne effettuati a livello di singola riga d'ordine.

E' possibile caricare ordini quadro o di impegno che definiscono in modo più o meno specifico la richiesta di un cliente in termini di prodotti e relative quantità, da cui andranno a scalare successivi ordini di rilascio.

Il modulo Ordini Clienti contiene:

- La Gestione dei singoli Ordini Clienti;
- Una serie di Funzioni di Consultazione On-Line e di Statistiche Batch dei dati relativi agli Ordini Clienti presenti nel sistema;
- Una serie di Funzioni di Utilità che permettono di eseguire su di un gruppo di righe ordini clienti storni massivi, con gli eventuali ripristini, variazioni massive di dati evitando di dover intervenire manualmente riga per riga.

Gli Ordini Cliente possono essere immessi in modo manuale, tramite la normale funzione on-line di Gestione Ordini Clienti. Nel caso siano ordini di rilascio faranno riferimento ad altri Ordini Clienti di tipo previsionale dai quali andranno a scalare prodotti e relative quantità.

E' possibile acquisire gli ordini direttamente da clienti, show-room ed agenti, permettendo loro il caricamento degli ordini raccolti mediante un sistema dipartimentale su PC o con collegamenti internet (extranet). In questo caso si definisce la visibilità dei dati di sistema concessa al profilo cliente/agente e si stabilisce i controlli che deve superare un ordine cliente per divenire effettivo. Il caricamento degli ordini può essere interattivo (on-line) oppure differito (in modalità batch).

Infine, è possibile emettere ordini da pronto, nei quali i prodotti e le quantità riportati sono ottenuti consultando la disponibilità effettiva di magazzino.

5.2.1 La struttura dell'ordine

Gli ordini cliente sono costituiti da una testata (tabella societaria S3T_OCT) e da N righe (tabella societaria S3T_OCR). La numerazione di ogni ordine è caratterizzata da stagione, marchio e numero progressivo. All'interno di uno stesso ordine possono essere inseriti prodotti finiti di linee di vendita diverse purché appartenenti al marchio definito per l'ordine. Tutti gli altri oggetti, privi di linea di vendita in anagrafica, possono essere inseriti in un ordine di un qualsiasi marchio. La chiave dell'ordine di una società è obbligatoriamente costituita da **Stagione, Marchio/ Segmento e Numero d'Ordine**. La Stagione è un dato statistico; è utilizzata per i Fabbisogni, per l'accesso alle Condizioni Commerciali, e per la ricerca del Listino ecc ...

5.2.2 Testata ordine, S3T_OCT

Un ordine può contenere solo prodotti finiti che appartengono al Marchio indicato. In particolare in GF il Marchio è indicato come il Segmento. Il Numero Ordine è il numero progressivo raggiunto dalla numerazione definita per Stagione e Marchio.

E' possibile effettuare ordini per Evento; in questo caso l'ordine potrà essere intestato solo ad un Cliente appartenente ad una delle Classi Cliente abilitate all'Evento (se indicate) e potrà contenere solo Oggetti (prodotti finiti, componenti, ...) abilitati all'Evento specifico. Se sull'anagrafica Eventi non è specificata nessuna Classe Clienti per l'Evento, allora l'ordine ad Evento potrà essere intestato ad un qualsiasi cliente abilitato ad essere intestatario d'ordine. Si ricorda che la Classe Clienti per Evento è un dato che viene assegnato al Cliente sulle condizione commerciali del Cliente. L'Evento non rientra nella chiave di numerazione dell'ordine, perciò un ordine ad evento ha come numero il progressivo raggiunto dalla combinazione Stagione e Marchio presenti sull'ordine.

Sull'ordine è obbligatorio inserire la Data dell'Ordine. La Data dell'Ordine è la data di emissione o di competenza dell'ordine non è la data di inserimento che è registrata nei file di log. E' possibile tramite regola valorizzarla con la data odierna. La data di inserimento deve essere compresa nei periodi di accettazione ordine associati alla stagione o all'evento (se l'ordine è ad evento) se indicati.

E' inoltre presente l'indicazione di un Ente di Inserimento Ordine che individua un gruppo di utenti o un ufficio. L'ente è un dato obbligatorio, ma non è valorizzato automaticamente da programma. Può essere tramite regola associato al ruolo dell'utente di inserimento.

In testata ordine è indicata una Tipologia di Vendita dell'ordine, che consente una suddivisione commerciale dell'offerta. E' un dato obbligatorio e non è presente sulle righe dell'ordine.

E' possibile inoltre attribuire all'ordine una Tipologia di Pianificazione, in modo da considerare gli effetti dell'ordine sul bilanciamento del prodotto ordinato e sui fabbisogni dei componenti di tale prodotto; potrebbe infatti essere richiesto che un ordine cliente non sia inserito per attivare la produzione del prodotto ordinato, ma solo per attivare le procedure di approvvigionamento dei materiali componenti il prodotto. La Tipologia di Pianificazione consente un'analisi trasversale, poiché può essere indicata anche sugli ordini fornitori, sulle commesse di produzione e sulle giacenze di magazzino; può essere utilizzata ad esempio per distinguere la produzione del campionario da quella normale. La Tipologia di Pianificazione non è un dato obbligatorio; non è presente sulle righe.

Un ordine può essere intestato ad un solo Cliente, Cliente Intestatario, che nella rispettiva anagrafica deve essere abilitato a questa funzione. Oltre al cliente intestatario deve essere indicato il Cliente di Fatturazione e l'Indirizzo di Fatturazione, scelto tra gli indirizzi del Soggetto corrispondente al Cliente di Fatturazione. Il Cliente Intestatario può non coincidere con quello di Fatturazione: è il caso del negozio che effettua l'ordine, ma la fatturazione ed i pagamenti sono gestiti da un ente centrale, ad esempio dalla casa madre. Questi dati sono presenti solo in testata perciò sono gli stessi per tutte le righe contenute nell'ordine.

Il Tipo Ordine può assumere i seguenti valori: normale, previsionale e di rilascio, ed abilita le gestioni specifiche di ogni Tipologia di ordine.

Il Flag Ordine Importatore (o Distributore) se attivato, predispose la struttura parallela delle righe ordine che deve ospitare i dati dei clienti dell'importatore ed i relativi prezzi per gestire la fatturazione in service.

Ogni Ordine può essere emesso in una sola Valuta con relativo Cambio. Il Cambio proposto è prelevato, seconda la priorità stabilita a livello societario, dai cambi per Area Aziendale (in questo caso Vendite), Stagione, Marchio o per una combinazione parziale dei precedenti campi o dai cambi assoluti, anche se può essere inserito e/o variato manualmente.

E' obbligatorio indicare un Mercato per l'Ordine; il Mercato insieme a Valuta, Marchio, Stagione e Codice Listino (dato di Riga ordine) costituisce l'identificativo del listino a cui accedere per reperire i prezzi da riportare sulle righe.

In testata, inoltre, è possibile specificare un Canale di Vendita, un Codice Esenzione IVA, una Condizione di Pagamento., a cui è collegato uno Sconto Finanziario riportato sull'ordine in sola visualizzazione. Il Canale di Vendita ha valenza puramente statistica; costituisce un criterio di raggruppamento dei clienti rispetto al processo di distribuzione del prodotto (Dettaglio, Grande Distribuzione, ...) e solitamente viene impostato con il valore dell'anagrafica cliente, anche se

modificabile. Il Codice Esenzione IVA indica se l'ordine è o meno esente IVA specificando una causale; sulle righe invece sarà possibile indicare il Codice IVA, corrispondente ad un'aliquota, da applicare all'articolo.

La Condizione di Pagamento è un dato obbligatorio ed indica la modalità di pagamento con cui il cliente effettua il pagamento dell'ordine (ad esempio rimessa diretta, bonifico bancario a 30 gg, bonifico bancario a 60 gg, ecc...). Sull'ordine viene riportato, in sola visualizzazione, lo Sconto Finanziario collegato alla Condizione di Pagamento indicata. Lo Sconto Finanziario rappresenta lo sconto che l'azienda intende concedere al cliente sull'imponibile dell'ordine in base a motivazioni riconducibili alla tipologia e dalla tempestività nell'esecuzione del pagamento ed è da applicare a piede fattura indipendente dallo sconto commerciale di testata e dagli sconti commerciali presenti a livello di riga. Lo sconto finanziario non rientra nella valorizzazione delle statistiche commerciali degli ordini. In testata ordine è possibile indicare anche uno Sconto Commerciale; questo sconto viene applicato al valore dell'ordine dopo che sono stati applicati gli eventuali sconti commerciali indicati sulle righe.

In Testata è indicato un Agente d'Ordine con relativo Agente di Liquidazione a cui devono essere liquidate le provvigioni dell'ordine se presenti. Le provvigioni non sono riportate in testata, ma solo a livello di riga. L'Agente di Liquidazione è prelevato dall'Anagrafica dell'Agente d'Ordine anche se modificabile. Se l'ordine fa riferimento a più agenti è possibile elencare in un archivio di dettaglio collegato alla testata dell'ordine gli agenti concorrenti all'ordine con i rispettivi agenti di liquidazione. In questo archivio può anche non essere contenuto l'agente d'ordine, che ha valenza puramente statistica. La stessa struttura sarà riportata a livello di singola riga ordine (sulla riga non è modificabile) per poter specificare le liste di provvigioni di ogni agente, che verranno applicate in cascata all'importo della riga d'ordine.

In testata ordine è possibile sospendere a vari livelli il processo di evasione dell'ordine, motivando il blocco con causali. Le sospensioni d'ordine previste sono le seguenti:

- Fabbisogno (l'ordine non è considerato nel calcolo dei fabbisogni, perciò non genera richiesta di materiali).
- Produzione (l'ordine non è considerato nel bilanciamento MPS e non è possibile lanciare una commessa di produzione collegata all'ordine).
- Assegnazione (il processo di evasione dell'ordine è sospeso in fase di assegnazione).
- Formazione Colli (il processo di evasione dell'ordine è sospeso in fase di formazione colli)
- Spedizione (il processo di evasione dell'ordine è sospeso in fase di spedizione).
- Prebollatura.

E' possibile inizializzare questi blocchi causalizzati con i corrispondenti valori dei blocchi indicati nelle condizioni di vendita del cliente, da cui sono stati reperiti i dati commerciali dell'ordine secondo le priorità e le regole stabilite.

Si ricorda che nelle Condizioni Commerciali del Cliente è presente anche la sospensione causalizzata dell'acquisizione ordini, che impedisce di inserire a sistema ordini intestati a quel cliente per la combinazione Stagione/Marchio/Tipologia di Vendita (completa o parziale) definita sull'ordine.

E' obbligatorio dichiarare se l'ordine deve essere considerato o meno nelle proiezioni di vendita degli agenti mediante il flag Proiettabile, e nel caso sia proiettabile è obbligatorio specificare una Classe di Proiezione di appartenenza. I coefficienti di proiezione possono essere relativi a classi di proiezione ordini combinate con classi di proiezioni di Modello e di Parte, validi per una data Stagione, Linea di Vendita, Tipologia di Vendita.

Alla testata dell'ordine è associata una Lista di Made In inclusi o esclusi che viene inizializzata con la lista dei Made In (inclusi o esclusi) prelevata dall'anagrafica del cliente intestatario. La lista di Made In può essere comunque modificata. E' utilizzata in fase di assegnazione e/o di spedizione dell'ordine: infatti è possibile assegnare (per spedire) all'ordine solo prodotti (in giacenza di magazzino, in commesse di produzione, in ordini a fornitore) che hanno un Made In compatibile con la lista di Made In indicata sull'ordine. Se il Made In del prodotto in giacenza non soddisfa la richiesta di Made In dell'ordine l'assegnazione (e la spedizione) si blocca.

Infine per ogni Testata Ordine sono gestiti gli Archivi delle Note in Lingua e degli Attributi Aggiuntivi, contenenti un numero finito di attributi (tra campi e flag) liberamente personalizzabili dall'azienda per denominazione e significato, ciascuno collegabile ad una tabella di decodifica valori. Sono utilizzabili nella Ricerca e Sintesi, pertanto su di essi è possibile effettuare delle estrazioni e delle elaborazioni statistiche.

5.2.3 Riga ordine, S3T_OCR

Sulla riga ordine l'oggetto deve essere indicato puntualmente come Modello-Parte se gestito a modello, come Parte se gestito con solo la parte. Non è possibile utilizzare classi di modelli e/o di parti sull'ordine cliente. Se l'Oggetto è costituito da un Modello con combinazioni è necessario specificare una Combinazione; se l'oggetto è gestito a colore deve essere specificato un Colore; se l'oggetto è gestito a misura deve essere dichiarata la Misura. E' possibile specificare ulteriormente l'oggetto da ordinare valorizzando tutti o solo alcuni degli altri componenti la chiave di un oggetto: drop, statura, tipo e codice variante, etichetta e cartellino.

Se l'Oggetto è gestito a taglie è proposta come default la prima griglia abilitata all'oggetto e viene esplosa la configurazione delle taglie abilitate con anche l'indicazione della fascia prezzo di appartenenza: deve essere ordinata una Quantità dell'oggetto almeno in una delle taglie in cui è gestito. Le taglie che compongono una griglia taglie possono essere espresse anche tramite la descrizione alternativa degli Alias. Su una riga ordine è possibile memorizzare il codice alias utilizzato nell'imputazione delle quantità per taglia, in modo che rientrando sulla riga ordine in modifica o visualizzazione, le taglie saranno già visibili anche secondo la descrizione alternativa scelta. Per gli Oggetti non gestiti a taglie si indica un'unica Quantità: come unità di misura è proposta di default l'unità di misura di vendita definita in anagrafica anche se può essere modificata con altra unità di misura avente definito un fattore di conversione (per oggetto o assoluto) con l'unità di misura interna dell'oggetto.

Sulla riga può essere indicato il Codice Listino, che insieme agli altri dati che identificano un listino di vendita (Stagione, Marchio, Valuta, Mercato) presi dalla testata dell'ordine, permette di individuare il listino a cui accedere per reperire i prezzi di vendita (assoluti o per griglia/fascia) dei prodotti. In alternativa si imputa il Prezzo in modo manuale per ogni fascia prezzo presente. Se l'oggetto è gestito a misura o non ha gestione dimensionale si imputa un Prezzo unico. Il prezzo è obbligatorio per tutte le righe la cui Tipologia Contabile lo richieda; se una riga ha Tipologia Contabile che non richiede il prezzo, ad esempio 'omaggio', deve comunque essere imputato un valore 'zero' al prezzo. Sulla riga d'ordine è presente il flag Prezzo Manuale con il quale l'utente dichiara che da quel momento in poi i prezzi di quella riga sono svincolati dal listino da cui sono stati recuperati. Se si esegue un riallineamento dei prezzi ai listini le righe che sono state svincolate (di solito perché i prezzi sono stati modificati) non vengono toccati dalla routine di riallineamento. E' comunque possibile richiedere alla funzione di riallineamento di ignorare il flag ed intervenire anche sui prezzi ritoccati.

Alla riga d'ordine (e dunque all'oggetto) è possibile anche attribuire una Divisione Produttiva utile in caso di ordine cliente relativo a materiali che si vuole vendere (privi di Divisione Produttiva propria perché non prodotti finiti). In caso di giacenze di magazzino differenziate per Divisione Produttiva, valorizzando questo attributo, si dichiara da quale giacenza di magazzino si vuole prelevare il materiale.

Sulla riga Ordine sono presenti le seguenti Date:

Inizio e Fine Consegna (o Periodo di Consegna) richieste dal Cliente.

Inizio e Fine Consegna (o Periodo di Consegna) fissate dall'Azienda.

Data Consegna Pianificata (MRP).

I periodi di consegna ordine o le date di inizio e fine consegna devono essere comprese nei periodi di consegna associati alla stagione o all'evento dell'ordine (se l'ordine è ad evento) se sono indicati nelle rispettive tabelle. La Data Pianificata può essere digitata dall'utente o valorizzata mediante regole (ad esempio può essere posta uguale ad una delle date standard inserite: data di Fine Consegna dell'Azienda o del Cliente). Altre eventuali Date o Periodi possono essere memorizzati direttamente nei Dati Aggiuntivi della riga ordine, ad esempio Data di Inizio e Fine Consegna (o Periodo di Consegna) concordate tra Azienda e Cliente.

5.2.4 Conferma ordine

E' prevista la Procedura di Conferma d'Ordine: quando un ordine viene confermato tutti i dati relativi alla situazione confermata vengono memorizzati in un apposito Archivio Conferme D'Ordine. La Conferma d'Ordine non è altro che la fotocopia dell'ordine nel momento in cui viene confermato con l'indicazione della data (Data Conferma d'Ordine) in cui questi dati sono stati memorizzati. Le conferme d'ordine solitamente sono effettuate tutte insieme a chiusura campagna

vendite. E' possibile modificare ancora l'ordine togliendo la conferma e poi riconfermandolo di nuovo.

Zona di Spedizione è un attributo che può essere utilizzato per determinare delle condizioni di spedizione per fini statistici. Si consiglia di definire una regola che prelevi il dato dalle condizioni commerciali del cliente di spedizione dell'ordine.

5.2.5 Provvigione agente su riga

Sulla riga d'ordine è riportata la struttura agenti definita in testata. Ad ogni agente può essere associata una propria Lista di Provvigioni. L'indicazione sulla riga d'ordine di una provvigione, o di una lista di provvigioni, è obbligatoria se nell'anagrafica agente il relativo flag lo richiede. La lista di provvigioni (può essere un'unica percentuale o un elenco di percentuali da applicare in cascata ed in unica soluzione) può derivare dall'anagrafica del prodotto, dalle condizioni commerciali del cliente intestatario, dalle condizioni commerciali dell'agente (è la lista di provvigioni dell'agente, non del rispettivo agente di liquidazione) seguendo la rigida priorità di provvigioni di prodotto, a seguire di cliente intestatario ed infine di agente. La lista di provvigioni derivante dal prodotto o dal cliente intestatario viene attribuita (a livello di riga) in toto all'agente d'ordine e se è stata definita una struttura di più agenti in testata, viene imputata al primo agente della struttura e potrà essere distribuita tra i vari agenti concorrenti: il totale provvigione dopo l'assegnazione potrà essere anche modificato e risultare maggiore o minore rispetto a quello ereditato. Se le provvigioni non sono indicate né a livello di prodotto, né a livello di condizione commerciale del cliente intestatario, allora sulla riga d'ordine vengono riportate le liste di provvigioni corrispondenti ciascuna ad ogni agente indicato nella struttura agenti (definita a livello di testata). Si ricorda che la struttura agenti di testata può anche non contenere l'agente d'ordine.

Sulla riga ordine è possibile specificare una Lista di Sconti (elenco di percentuali) da applicare in cascata ed in unica soluzione all'importo di riga. Gli Sconti possono derivare dall'anagrafica del prodotto o dalle condizioni commerciali del cliente di fatturazione seguendo la rigida priorità di sconti di prodotto e a seguire di cliente di fatturazione; gli sconti proposti come default possono comunque essere modificati.

5.2.6 Avanzamento della Quantità per riga

A livello di riga ordine sono disponibili dei dati di sintesi sullo stato di avanzamento della quantità ordinata della riga:

- Quantità netta: ordinato iniziale più tutte le modifiche (positive e negative).
- Quantità modificata: comprende solo le modifiche (storni) negative di qualsiasi tipo.
- Quantità assegnata: rappresenta il totale dell'assegnato a giacenza, a commessa di produzione e a ordine fornitore.
- Quantità spedita: è la somma delle quantità spedite.
- Quantità resa: comprende tutti i resi (normali e quelli che hanno riaperto l'ordine).

- **Quantità inevasa:** rappresenta la quantità che deve essere ancora evasa e si ottiene sottraendo alla quantità netta la quantità spedita e sommando, se presenti, solo i resi che hanno riaperto l'ordine.

Il residuo operativo: si ottiene sottraendo la quantità assegnata alla quantità inevasa.

Se l'oggetto è gestito a taglie, le quantità definite precedentemente sono dettagliate a livello di singola taglia. L'informazione principale della quantità della riga ordine è la **Quantità Netta** (al netto di tutte le modifiche); la **Quantità Lorda** si ricava riaccorpando tutte le modifiche dovute a causali che concorrono alla ricostruzione dell'ordinato lordo.

5.2.7 Abbinamento righe

E' possibile attribuire a due o più righe, appartenenti ad uno stesso ordine o a ordini diversi di uno stesso Cliente Intestatario, uno stesso Codice Numerico di Abbinamento, in modo che possano essere raggruppate e trattate insieme dal sistema. Gli abbinamenti vengono usati nelle applicazioni, ad esempio, per spedire tutte insieme un gruppo di righe, e pertanto un gruppo di prodotti in determinate quantità, per soddisfare esigenze di vetrina di una catena di negozi; la procedura automatica di assegnazione può richiedere il superamento di controlli specifici sulle righe abbinate per consentirne l'assegnazione.

Il fatto di poter attribuire uno stesso Codice Numerico di Abbinamento a righe appartenenti a ordini diversi di uno stesso cliente, permette di gestire abbinamenti tra prodotti appartenenti a marchi diversi e tra ordini inseriti in tempi successivi.

Infine per ogni Riga Ordine sono gestiti gli Archivi delle Note in Lingua e degli Attributi Aggiuntivi, contenenti un numero finito di attributi (tra campi e flag) liberamente personalizzabili dall'azienda per denominazione e significato, ciascuno collegabile ad una tabella di decodifica valori. Sono utilizzabili nella Ricerca e Sintesi, pertanto su di essi è possibile effettuare delle estrazioni e delle elaborazioni statistiche.

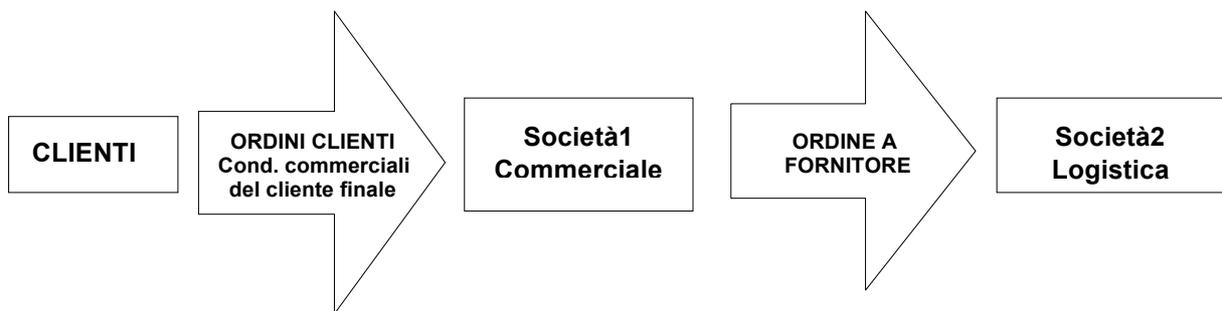
5.2.8 Storni su righe

E' possibile variare (aumentare o diminuire) la quantità ordinata. La modifica viene effettuata dichiarando una causale che giustifica la variazione e viene sempre registrata per data. La causale di storno non ha segno, perciò il segno della variazione è ricavato direttamente dal confronto tra la nuova quantità digitata rispetto a quella precedente. In ogni momento è possibile risalire alle quantità di un ordine ad una certa data e tracciarne le variazioni subite.

Le causali esplicitano il motivo per cui viene modificata la quantità ordinata e possono o meno concorrere alla ricostruzione dell'ordinato lordo: se lo storno è dovuto ad errata digitazione la quantità non deve essere considerata nell'ordinato lordo, mentre se lo storno è dovuto a motivazioni di cui è necessario tenere traccia (disdetta cliente) può essere utile considerare le quantità stornate nell'ordinato lordo. controlli sulle abilitazioni tra cliente importatore e tipo ordine importatore; tra clienti e prodotti abilitati all'ordine per evento.

5.3 Configurazione di Flussi Intercompany in Stealth

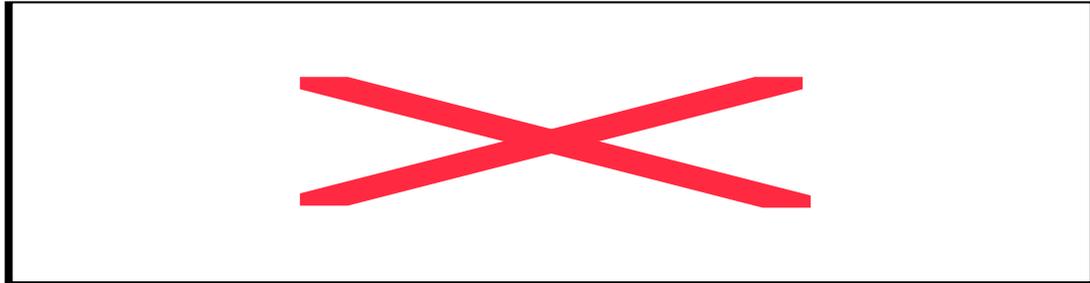
A fronte di un ordine cliente la società commerciale emette un ordine a fornitore alla società di logistica (per la società di logistica ciò corrisponde all'acquisizione di un ordine cliente). Una procedura avverte ST3K del rapporto intercorrente tra Società Commerciale e Società Logistica in modo che ci sia la generazione automatica di ordini fornitore a fronte di ordini clienti (replicare gli ordini clienti acquisiti ed accorparli). Le anagrafiche dei modelli e dei soggetti in caso di triangolazione devono essere di gruppo (pubbliche) altrimenti quando si trasmette un ordine cliente bisogna inviare anche le anagrafiche. La società di Logistica deve poter accedere ai dati del cliente e dell'ordine inseriti dalla società Commerciale, in modo da ricevere gli aggiornamenti dei dati anagrafici, condizioni commerciali, quantità dell'ordine.



La società di logistica ragiona sul proprio portafoglio ordini che non include quelli della distributiva; la distributiva se vuole può comunicare alcuni ordini che ha acquisito e che poi verranno girati alla Logistica, ma non è automatico che la logistica veda quelli della distributiva.

La logistica produce capi (appartenenti a commesse) che sono versati nel suo magazzino. La complicazione nasce se la spedizione della logistica deve essere preparata per soddisfare l'ordine cliente originario e non a fronte di un ordine fornitore corrispondente all'ordine cliente della distributiva. Si può preporre un ente (o un utente) della Società distributiva che ha visibilità sul magazzino della logistica e che decide come assegnare e spedire i capi del magazzino della logistica. Esiste comunque il problema di rendere visibile i dati commerciali del cliente finale alla società di logistica. La situazione più semplice si verifica se la spedizione avviene a fronte dell'ordine a fornitore emesso dalla distributiva e solo quando i capi arrivano nel magazzino della distributiva vengono preparati per essere spediti al cliente finale a fronte del suo ordine. La società di Logistica fattura alla Distributiva con un Listino Intercompany.

Intercompany: modalità possibili di flusso fisico della merce spedita



E' necessario stabilire chi svolge le diverse attività: la produzione (Società di Produzione), l'assegnazione, la spedizione. la merce può essere spedita direttamente al cliente finale o inviata prima ad un magazzino di transito o spedita alla Società di Commerciale. Se la merce non è spedita direttamente al cliente finale, quando la merce arriva presso la Società Commerciale (con le bolle) può essere impaccata con altra merce ed inviata al cliente finale in un'unica spedizione cumulativa con un'unica fattura.

Il "transit point" è un magazzino doganale "blindato" in cui però la merce è ancora di proprietà del gruppo, cioè non è ancora stata emessa la fattura al cliente finale; esiste solo la fattura della società di produzione verso quella di distribuzione (è una specie di fattura accompagnatoria). Quando la merce arriva in un transit-point ha un costo e quando esce ha un costo più alto da fatturare al cliente finale.

Possibilità di pianificare le spedizioni ignorando l'ordinato e basandosi sulla situazione della giacenza (scorta) di ciascun articolo presso il negozio. Viene effettuato un ordine macro, ad esempio, dal Retail Europa che cumula gli ordini dei singoli negozi, i quali vengono soddisfatti considerando il reale andamento delle vendite effettuate dal negozio, o meglio controllando le scorte di magazzino del negozio (se il negozio ha ancora a scorta una quantità rilevante di un articolo presente nell'ordine effettuato in precedenza, l'articolo non viene spedito al negozio).

Capitolo 6: Nuovi flussi di ordini intercompany per FASHION

La configurazione del flusso ordini in FASHION ha subito qualche cambiamento sia dal punto funzionale che tecnico, con l'introduzione di nuovi processi intercompany.

Le divisioni distributive sono GF Spa che si occupa del mercato italiano e la GF SW Branch che si occupa del mercato estero. Gli ordini sono raccolti dalle divisioni commerciali ad esempio la GF Corp America, e poi vengono mandati alla GF SW Branch che li consolida e li mista alle divisioni produttive che sono le Industrie. Con il passaggio da società del Gruppo ad divisioni, la GFO FASHION Operation ha rilocato le industrie in questo modo:

- SOCIETA6 I -> G.F.Operation Polo produttivo
- SOCIETA1 -> G.F.Operation Polo produttivo
- SOCIETA2 -> G.F.Operation Polo produttivo sede di Fossò
- SOCIETA3 -> G.F.Operation Polo produttivo

Poi più avanti anche queste:

- SOCIETA5 -> SOCIETA5
- SOCIETA4 -> Confezioni di Matellica.

6.1 Definizioni

Polo logistico è un soggetto in Stealth, rappresenta l'ente che si occupa della spedizione dell'ordine. Nella tabella degli ordini S3T_OCT, è codificato nel dato aggiuntivo DAGG_CAR78.

Ubicazione è il codice che rappresenta una suddivisione logica del magazzino. Viaggia insieme al polo logistico e serve ad indicare all'industria dove versare la merce, e agli operatori dove prelevare la merce.

Stato ordine codice che ci identifica lo stato dell'ordine. È cablato nella tabella S3T_OCT nel dato aggiuntivo DAGG_CAR1. Un esempio: stato S3 sta per Chiuso, S4 per Confermato, S5 Ordine confermato al cliente, SL send to production ecc...

Replica è un ordini OC inserito dal cliente che viene subito mandato all'industria per la produzione. Questo specifico ordine OC è sincronizzato e allineato con il suo Ordine fornitore OF.

Bulk è un ordine che può essere accorpato con altri ordine bulk per produrre una unica o più ordine fornitore. L'accorpamento delle righe degli ordine è sulla base di varie elementi ad esempio la stagione, la divisione produttiva, la Business Unit, il marchio o il segmento ecc... Gli ordini

accorpati sono consolidati e mandati alle industrie che producono la merce e li mettono in magazzino. La merce poi sarà impacchettata secondo i singoli ordini d'origini (non esiste un rapporto di univocità tra l'ordine cliente e l'ordine fornitore). Esiste uno sotto tipo di Bulk che è il Bulk Dettaglio o Bulk uno a uno.

Il **Bulk Dettaglio** si comporta esattamente allo stesso modo di un ordine Replica. Gli ordini 1 a 1 servono a prendere in considerazione la gestione di articoli particolare ad esempio Articoli Cites (fatti con materiali particolari che devono rispettare delle regole internazionali) o come articoli di Su Misura. All'inserimento dell'ordine, so solo che è un ordine Bulk, solo le specifiche come l'imballo mi danno informazione sul dettaglio.

Intercompany si intende quell'ordine al fornitore che è indirizzato a una altra società/divisione del gruppo. Di questi ordini ne esistono di due tipologie :

- Ordini intercompany GESTION (OIA)
- Ordini intercompany extra GESTION (OIEA)

La differenza principale fra i due ordini sta nel fatto che l'ordine intercompany GESTION (OIA) darà origine a un ordine cliente GESTION di una società (divisione) già in GESTION, mentre Ordini intercompany extra GESTION (OIEA) verranno esportati verso altre società del gruppo non ancora in GESTION oppure arrivano da altre società del gruppo non ancora in GESTION (ad esempio OC su OC da GF Honk Kong). Nel seguito ci soffermeremo sugli ordini intercompany GESTION.

6.2 Descrizione dei processi introdotti

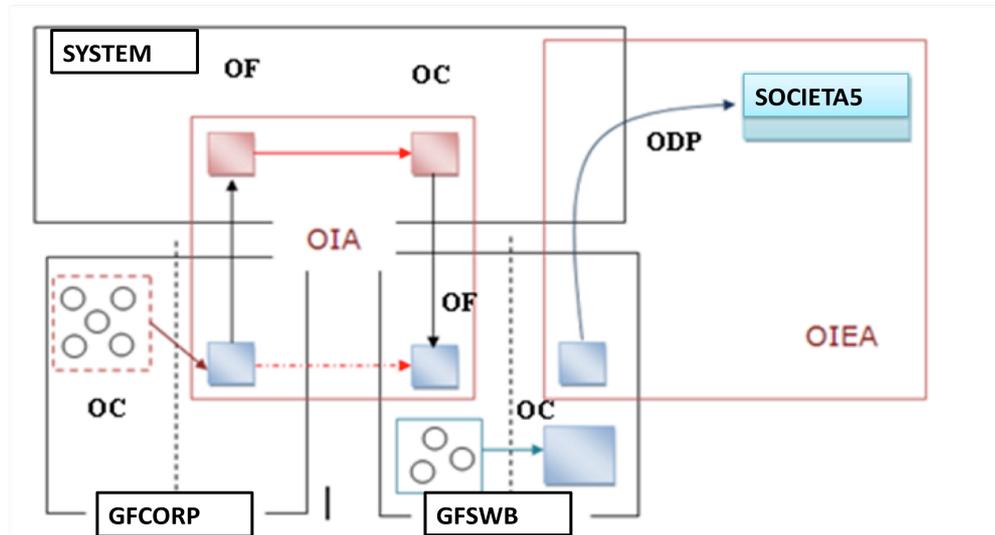
Ci poniamo l'obiettivo di presentare le novità introdotte con i nuovi processi intercompany. In questa sessione ci soffermeremo sul punto funzionale del flusso degli OIA all'interno di una catena distributiva su più livelli, nel nostro caso 3: GFCORP divisione commerciale, GFSWB divisione distributiva, GFO divisione logistica magazzino del gruppo e infine le industrie (SOCIETA5, SOCIETA6, SOCIETA1) divisioni produttive (con ODP Ordine Di Produzione).

L'introduzione dell'Intercompagny per una istanza di Stealth 3000, secondo lo Standard, comporta pochi accorgimenti sulla gestione delle tabelle di Stealth impostate per i flussi ordini in particolare la tabella degli ordini clienti S3T_OCT, tabella delle righe ordini S3T_OCR, tabella degli ordini fornitori S3T_OFT, tabella delle righe fornitori S3T_OFR e le tabelle di collegamento ordine cliente ordine fornitore PGGFT_OCOF e PGGFT_OIR. Come spiegato nel secondo capitolo, il fatto che in FASHION ci sia 4 istanze (SYSTEM, FITA, RFASHION, FAME) di Stealth 3000, ha moltiplicato i punti di intervento dovuto alla ristrutturazione.

Partendo dal punto fermo in questa fase di gestione Intercompany, che gli ordini clienti nascono sul SYSTEM e sono di sua competenza, gli ordini fornitori nascono sui sistemi locali e sono di loro competenza. Competenza nel senso di modifica e di propagazione e sincronizzazione delle informazioni relative all'ordine. I processi introdotti prendono anche in considerazione le particolarità di flussi presenti da e verso le industrie. Cioè Ordini in replica

caso SOCIETA1, la gestione ordini clienti senza magazzino caso GFHK oc su oc e EA Divisione produttiva SOCIETA6 ma che dovranno essere spediti in SOCIETA1 Jeans.

Nello specifico un Ordine Intercompany GESTION OIA è composto da un OF ed un OC collegati fra di loro e sono 2 differenti rappresentazioni della stessa informazioni, detto questo gli ordini dovranno essere sempre allineati fra di loro.



Esempio di Ordini OIA e OIEA

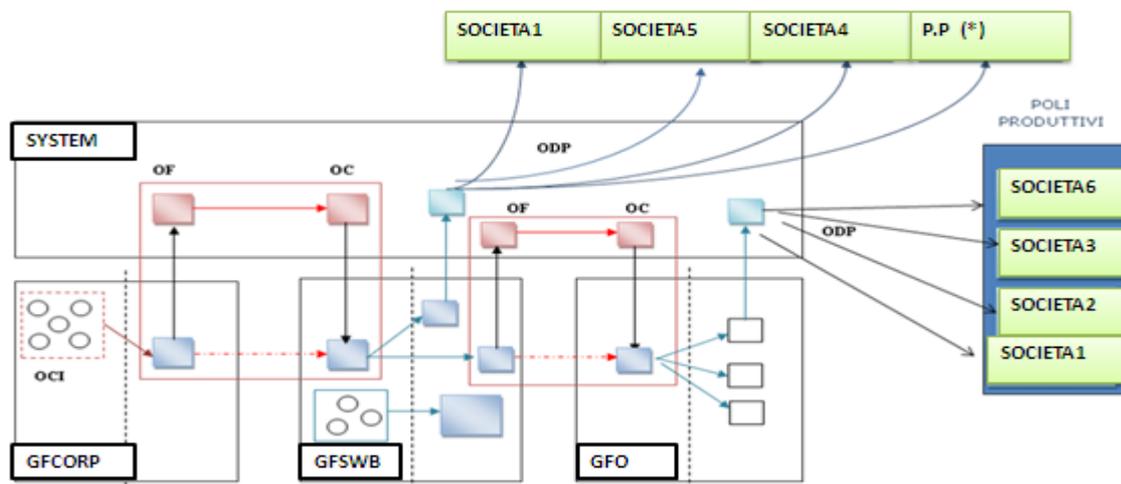
6.3 Punti d'intervento

Per poter raggiungere gli obiettivi del cliente, siamo riusciti ad individuare questi punti d'intervento sul programma che stava girando:

- A ogni riga d'ordine cliente deve essere assegnato il codice della divisione produttiva di riferimento. Questa informazione ci permetterà di gestire gli ordini che passeranno per la GFO prima di arrivare alle industrie.
- Gli ordini bulk hanno un nuovo livello di raggruppamento: La lista delle divisioni produttive. Con l'introduzione di ordini multi produttori, le righe degli ordini bulk potranno essere raggruppate a seconda della divisione produttiva. Ogni divisione produttiva produce articoli diversi, questa rende maggiormente la gestione degli ordini multi produttori.
- Gli OIA devono essere automaticamente esclusi dalla procedure di Generazione OF replica, Generazione OF bulk, Generazione previsioni. Per poter trattare questa specificità per gli Ordini Intercompany GESTION, viene cablato nella testata dell'ordine la specifica Ordini Intercompany OIA. I riferimenti del primo OIA viaggeranno all'interno di tutta la catena fino all'ordine produttivo
- Gli aggiornamenti vanno sempre nella direzione OC → OF. Si tratterà ogni singola coppia OF – OC slegandolo dal passaggio successivo. Ad esempio, se modifico OC della GFO non si riuscirà a modificare automaticamente OF di GFCORP

- Al momento non è prevista l'inversione del flusso.
- Il fornitore è reperito tramite i canali di distribuzione.
- E' indicato sul OC se l'ordine è un ordine di 1° livello, OC verso il cliente finale.

6.4 Presentazione dei nuovi flussi



(*) Poli produttivi per ordini diretti che non transitano da GFO GESTION

Flusso generale degli Ordini Intercompany

Gli OIA arrivano a GFO senza subire accorpamenti o divisioni. L'eventuale divisione per produttore avviene su GFO con la generazione degli ODP;

La generazione di un OIA sul sistema del distributore GFSWB, scatena una serie di passaggi automatici che porta ad avere direttamente l'ODP su GFO;

Esistono le procedure che da menu ci permettono di elaborare un unico OIA; Gli ordini OIA bulk non possono subire cambi codice, i replica si;

Le industrie invieranno gli storni produttivi associandoli alla chiave del primo ordine fornitore, informazione passata con gli ODP. Gli storni saranno automaticamente distribuiti e applicati sul sistema più opportuno, ribaltando le modifiche sull'ordine seguente OC → OF;

Per il momento gli OIA in replica non arriveranno in GFO.

Un ordine al fornitore generato da GFCORP se passerà attraverso a degli intermediari GFSWB non verrà accorpato con altri ordini, al massimo potrà essere diviso.

6.5 Descrizione dei dettagli tecnici

6.5.1 Modifiche alla struttura dati

Lista Licenziatari PGGFT_LIC_LIT/PGGFT_LIC_LIR.

Sono due tabelle societarie, inserita sul SYSTEM e propagate sui sistemi locali. Servono a raggruppare insieme più divisioni produttive.

Tabella configurazione OIA PGGFT_OIAC.

Societaria, inserita sul SYSTEM e non propagata. Serve per determinare, a fronte di una società Seller/società Buyer/divisione produttiva, se:

- si deve produrre un OIA di primo o secondo livello o se può essere inviato all'industria.
- se l'OC OIA deve essere forzato in replica ad esempio del caso SOCIETA1.
- Il codice del cliente intestatario dell'OC OIA.
- Il gruppo di consegna dell OC OIA.
- Eventuali forzature sul codice fornitore.
- L'inclusione degli ordini in replica nella generazione degli OC OIA.

La tabella delle condizioni di rottura PGGFT_ROC, servirà durante l'elaborazione degli ordini per capire per un determinato ordine come accorpare le sue righe, cioè come spaccare l'ordine in ingresso.

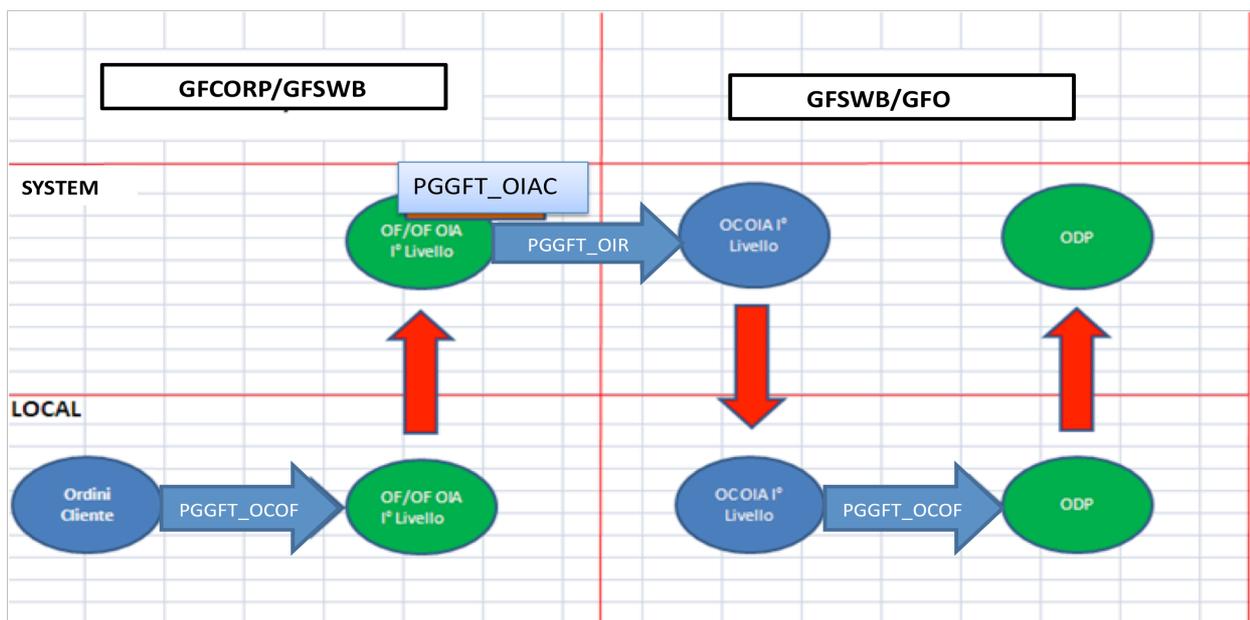
Tabella legami ordini intercompany PGGFT_OIR.

E' già presente nel database una tabella che contiene i legami fra gli OF e gli OC ed è strutturata anche per poter contenere le informazioni dei legami fra OC e OF. La tabella è presente e popolata solo nel SYSTEM e contiene:

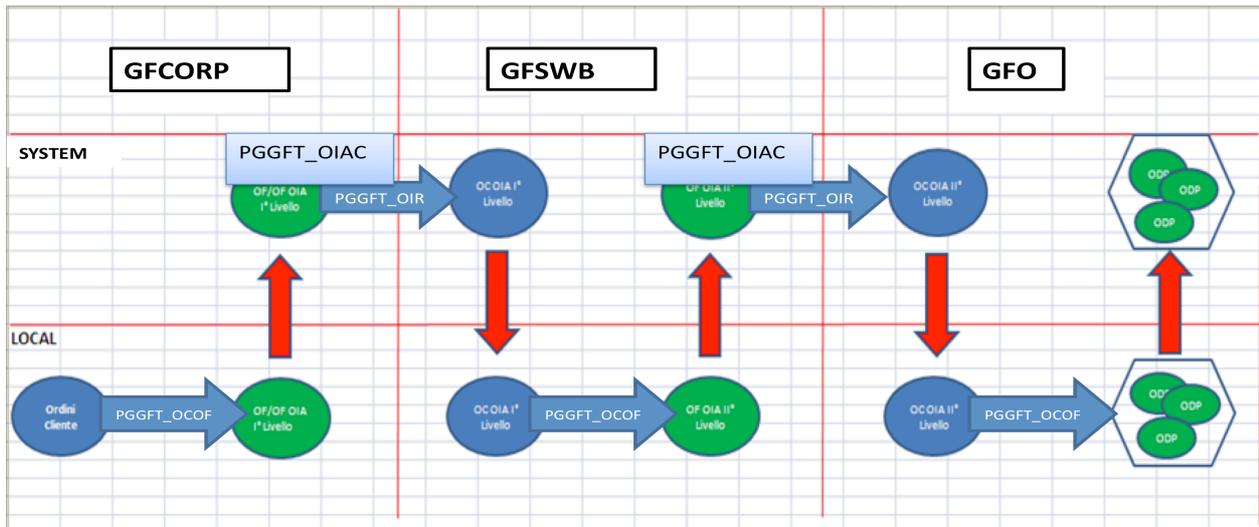
- Chiave logica OC OIA.
- Chiave fisica OC OIA.
- Chiave logica OF OIA.
- Chiave fisica OF OIA.
- Il lancio dell'elaborazione in caso di errore, per facilitare il recupero dei log e l'individuazione degli ordini in errore.

Dato aggiuntivo testata OC che rappresenta il livello. Nuovo dato aggiuntivo testata OF Tipo Competenza simile al campo nella S3T_OCT. Sempre nella S3T_OFT nuovi dati aggiuntivi canale di fornitura e categoria prodotto. E in fine nuovi dati aggiuntivi su S3T_OFT e S3T_OCT per riportare i riferimenti al primo Ordine Intercompany.

Dalla fase di inserimento dell'ordine, alla emissione dell'ordine di produzione verso le divisioni produttive di competenza, queste tabelle ci aiuteranno ad tenere traccia delle informazioni utili per le varie fasi del flussi Intercompany. I grafici in seguito mettono in evidenza l'uso di queste tabelle e dei vari legami per ogni livello.



Schema dei legami di 1 livello



Schema dei legami di 2 livello

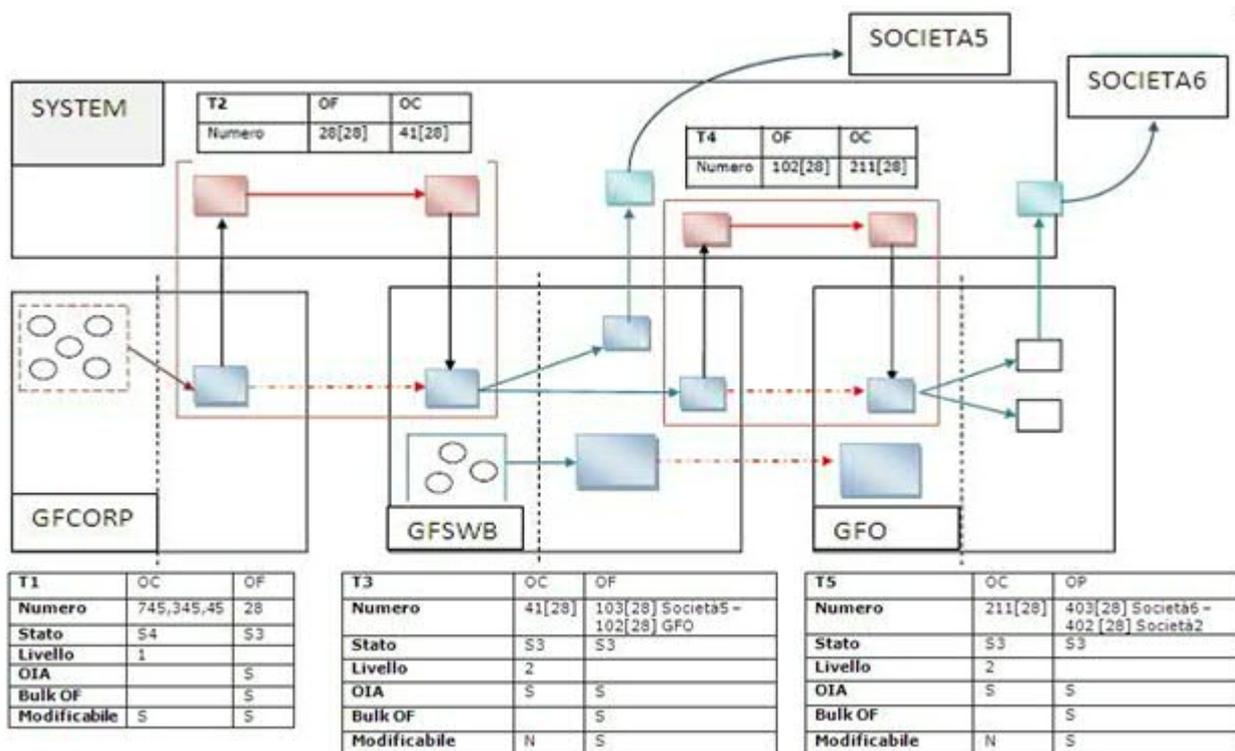
6.5.2 Personalizzazione dei packages di gestione dei flussi intercompany

- **Procedura di generazione ordini fornitori bulk (OFBULK)**

La procedura personalizzata al momento presente genera degli ordini al fornitore di tipo bulk , quest'ultimo identificato con un flag sull'ordine fornitore. La procedura parte selezionando una serie di ordini clienti con le seguenti caratteristiche: Ordini clienti in stato S4 e che non sono in replica. Gli ordini sono raggruppati per:

- BU, la Business Unit.
- Fornitore, elemento presente sull'ordine cliente determinato dai canali di fornitura.
- Segmento, o il marchio è presente sulla testata dell'ordine cliente .
- Collezione.
- Uscita, il giorno durante la campagna, ordinato per giorno.
- Categoria
- Articoli cites (opzionale).

E spacca gli ordini se impostata in tabella per etichetta/tipologia/nazione/OC/tipo cliente. Gli ordini generati sono legati all'ordine cliente attraverso una tabella ma non crea assegnazioni



- Generazione degli OF su GFCORP con la procedura OFBULK (T1 come riassunto delle caratteristiche) e lo propaga sul SYSTEM
- Il sistema crea l'ordine OC a partire dall'OF (T2)

Per la natura degli ordini in replica caso SOCIETA1 dove chi comanda è sempre OC non è possibile accorpate nello stesso bulk articoli con gestione in replica con articoli con gestione a bulk, questo determina che dovrà già essere accorpato in fase iniziale un ordine bulk solo per SOCIETA1, la procedura OFBULK viene modificata per renderla sensibile alla divisione produttiva.

- **Procedura di conferma ordini fornitori bulk (OFCONF)**

La GFSWB ha un portafoglio OC composto da una serie di ordini di 1° livello in S4, una serie di ordini di 2° livello di tipo OIA in stato S3 bulk, una serie di ordini di 2° livello di tipo OIA in stato S3 replica. Il portafoglio della GFSWB contiene anche una serie di ordini di 2° livello NON OIA, una serie di ordini GF CORP di 1° livello (che in fase 2 daranno origine a OF sul sistema GF corp).

Un ordine al fornitore generato da GFCORP se passerà attraverso a degli intermediari come GFSWB distributore estero, non verrà accorpato con altri ordini, al massimo verrà diviso. Gli ordini in replica viaggiano sempre in modo separato.

Si procederà a

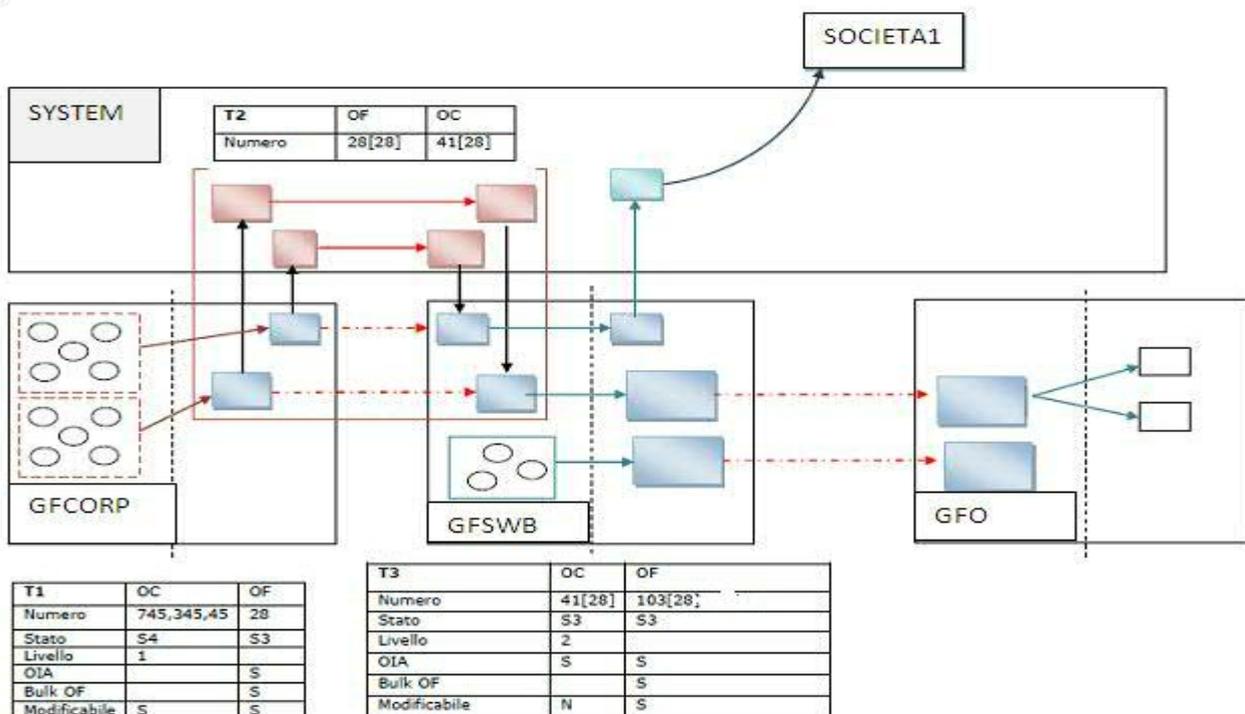
- Accorpamento degli altri ordini clienti ed esclusione degli OIA.
- Manualmente si effettuerà la spaccatura degli OIA su differenti fornitori/produttore/divisione produttiva.
- Si trasmettono gli ordini non diretti alla GFO attraverso export di files.
- Il sistema crea l'ordine OC a partire dall'OF (T4).

Le informazioni di stato, livello e flag bulk non sono sufficienti per identificare un OF/OC di tipo OIA, si pensa quindi di utilizzare i riferimenti (già presenti su OC), e usare dati aggiuntivi sulla OF per memorizzare i riferimenti all'ordine originale. Premesso che gli ordini OIA non possono essere raggruppati con gli altri, una eventuale generazione di OF bulk dovrà escludere tutti gli ordini OIA cioè nella selezione di ordini da trattare, escludere gli ordini con riferimenti OIA.

Gli OIA verranno trattati separatamente con una procedura che provvederà se necessario a spaccarli o a mantenerli inalterati. Per la spaccatura degli OIA bisogna reperire il fornitore degli articoli presenti nel bulk GFCORP e questa informazione potrà essere reperita in due modi o attraverso i canali di fornitura o attraverso i gli oggetti societari.

La differenza fra le due soluzioni sta che per i canali di fornitura si ha una configurazione dinamica che permette maggiore flessibilità, se viene riportata sugli oggetti societari dovrà per ogni oggetto/società definire il suo fornitore, che è utilizzato per la generazione degli OF.

CASO SOCIETA1



Il fornitore GFO non avrà bisogno di spaccare per divisione produttiva gli ordini SOCIETA1, essendo questi ordini già separati in origine. A causa di questo, già in questa fase di export verso una divisione produttiva e non verso un fornitore (GFO), dovranno essere modificate le logiche e le

configurazioni di trasmissione per includere nella chiave non solo il fornitore ma anche la divisione produttiva.

Fornitore GFO

- GFO ha un portafoglio OC composto da solo ordini di tipo OIA
- Una serie di ordini di 2° livello di tipo OIA in stato S3
- Un ordine al fornitore generato da GFCORP, se passerà attraverso a degli intermediari (GFSWB) non verrà accorpato con altri ordini, al massimo diviso.

Si procederà automaticamente a spaccare gli ordini OIA in ordini produttivi (OP) che avranno come elemento di spaccatura la divisione produttiva.

Questa informazione potrà essere recuperata dallo standard nella tabella oggetti societari, questo implica che un oggetto societario dovrà appartenere ad una sola divisione produttiva, e non potrà essere legato all'anno stagione di vendita, questa informazione è presente sull'ordine al fornitore sia a livello di riga che di testata. A livello di riga deve essere specificata solamente per oggetti di tipo componente cioè non prodotti finiti, mentre sulla testata è facoltativa e può essere utilizzata a piacimento anche se sugli oggetti non è specificata. Al momento, questa informazione è rappresentato dal produttore che invia l'anagrafica dell'oggetto, e cablato nel campo aggiuntivo del modello, e sui canali di fornitura.

Se ci fosse necessita di creare un ordine in replica partendo già dalla GF Corp non si evidenziano particolari problematiche sulla raccolta ordini, nemmeno su chi modifica l'ordine in quanto per la natura della replica anche se l'ordine cliente diventa S4 quelli successivi rimarranno in stato S3 e per cui non modificabili, ma questo non è bloccante in nessun passaggio essendo anche gli ordini in S3 trasmessi ai poli produttivi.

Il programma di spaccatura da un OC a più OF è unico e considerare due casi:

GFSWB vs SOCIETA1, SOCIETA5, GFO;

GFO e GFO vs divisione produttiva.

In altri termini deve spaccare gli OC in più OF per fornitore e divisione/polo produttore.

NOTA: per gestire il caso SOCIETA1, nella creazione degli OC di SOCIETA1 si deve passare da OIA OF bulk a OIA OC replica.

- **Aggiornamento OC/OF**

Per la gestione degli OIA, il programma di creazione degli OC da OF e l'aggiornamento in entrambi i versi permette la gestione della date di consegna. Quest'ultima è una procedura batch schedulata che aggiorna OC/OF. Per gli ordini bulk l'aggiornamento è a mano, invece per gli ordini replica l'aggiornamento è in modalità Batch e asincrono. Per questa procedura ci servirà la tabella PGGFT_OCOF già presente nel database.

La creazione OC da OF è gestita con la propagazione tramite coda. Un OF ha un fornitore che rappresenta una società per cui esiste una coda di propagazione attiva. Esempio OF con codice fornitore = 'GFSWB' che ha per società 079 per la quale esiste una coda di propagazione.

6.5.3 Integrazione con funzionalità esistenti

- **Cambio Codice**

E' l'unica procedura che crea ordini nel distributore locale. Per quanto riguarda gli OIA bulk il cambio codice non sarà possibile effettuarlo, salvo richieste e/o casi particolari che dovranno essere realizzate manualmente. L'abilitare della funzione deve essere a singoli utenti particolari.

Ad esempio: spostamento di merce GF Corp vs altro cliente, oppure GF corp su negozi di proprietà GF corp che dovranno dare origine ad OF su SYSTEM in modo manuale mediante cambio stato.

Per quanto riguarda gli OIA in replica è possibile effettuarlo con la procedura attuale, ciò comporta l'annullamento dell'OC originale con successiva propagazione nella catena, e creazione di un OIA nuovo.

6.5.4 Nuovi flussi intercompany: codifica

Dati Aggiuntivi della tabella S3T_OFT

	DES BLOCCO	DESCRIZIONE	NUM	TIPO	LUNG
1	Testata ordini (BOFT)	Tipologia Ordine	1	ALFA	1
2	Testata ordini (BOFT)	Stato ordine	2	ALFA	2
23	Testata ordini (BOFT)	Ordine Bulk	23	ALFA	1
24	Testata ordini (BOFT)	Generato da cambio codice	24	ALFA	1
25	Testata ordini (BOFT)	Codice Magazzino	25	ALFA	50
26	Testata ordini (BOFT)	Rif Esterno Primo Ordine	26	ALFA	50
27	Testata ordini (BOFT)	Flag Ordine Origine	27	ALFA	1
28	Testata ordini (BOFT)	Abilita Traccia Generazione O.F.	28	ALFA	1
31	Testata ordini (BOFT)	Rif. O.C. Generazione O.F.	31	ALFA	30
37	Testata ordini (BOFT)	Rif. IC Primo Ordine Fornitore	37	ALFA	50
39	Testata ordini (BOFT)	Polo Logistico	39	ALFA	10

Dati Aggiuntivi della tabella S3T_OFR

	DES BLOCCO	DESCRIZIONE	NUM	TIPO	LUNG
14	Riga Fornitore (BOFR)	Cambio codice: rif. riga OF di partenza	14	ALFA	50
22	Riga Fornitore (BOFR)	Divisione produttiva/produttore	22	ALFA	4
26	Riga Fornitore (BOFR)	Componente Prezzo Wholesale	26	NUM	18
27	Riga Fornitore (BOFR)	Componente prezzo I fornitore	27	NUM	18
28	Riga Fornitore (BOFR)	Componente prezzo II fornitore	28	NUM	18
35	Riga Fornitore (BOFR)	Polo Logistico	35	ALFA	10

Dati Aggiuntivi della tabella S3T_OCT

	DES BLOCCO	DESCRIZIONE	NUM	TIPO	LUNG
1	Ordine Cliente (BOCT)	Stato ordine	1	ALFA	2
10	Ordine Cliente (BOCT)	Rif. Societa OI	10	ALFA	3
11	Ordine Cliente (BOCT)	Rif. Anno OI	11	NUM	4
12	Ordine Cliente (BOCT)	Rif. Ente OI	12	ALFA	6
13	Ordine Cliente (BOCT)	Rif. NumOrd OI	13	NUM	7
30	Ordine Cliente (BOCT)	Produttore della merce	30	ALFA	10
31	Ordine Cliente (BOCT)	Fornitore	31	ALFA	10
56	Ordine Cliente (BOCT)	Canale di Fornitura	56	ALFA	18
57	Ordine Cliente (BOCT)	Ordine Bulk	57	ALFA	1
76	Ordine Cliente (BOCT)	Rif. IC Primo Ordine Fornitore	76	ALFA	50
78	Ordine Cliente (BOCT)	Polo Logistico	78	ALFA	10

Dati Aggiuntivi della tabella S3T_OCR

	DES BLOCCO	DESCRIZIONE	NUM	TIPO	LUNG
5	Righe ord. (BOCR)	Rif. Riga OI	5	NUM	5
19	Righe ord. (BOCR)	Made In	19	ALFA	6
28	Righe ord. (BOCR)	Divisione produttiva/produttore	28	ALFA	4
29	Righe ord. (BOCR)	Componente prezzo Wholesale	29	NUM	18

30	Righe ord. (BOCR)	Componente prezzo primo fornitore	30	NUM	18
31	Righe ord. (BOCR)	Componente prezzo secondo fornitore	31	NUM	18
39	Righe ord. (BOCR)	Escludi da gen. OF Bulk	39	ALFA	1
44	Righe ord. (BOCR)	Polo Logistico	44	ALFA	10

La procedura ORCOFBULK ***Conferma OC se in stato S4***

```
--Query di selezione con filtri fissi

-- Divisione produttiva
--Ubicazione
--Polo Logistico
--Tipo Preimballo
-- flag_di
-- Escludi da Gen. OF Bulk
-- Non esiste già alcun legame ad OF
-- Escludi OC Bloccati (recupero il flag di esclusione per capire
se elaborarlo o stamparlo tra i bloccati)
--Filtro su lista tipologia
--Filtro su Nazione
--Filtro Cliente intestatario
--Filtro su Numero ordine
-- Codice consegna
--Ubicazione
--Polo Logistico
--Tipo Preimballo
-- flag_di
-- Divisione produttiva
  commit;
End;
```

Elaborazione

Durante la fase di elaborazione si produce gli ordini fornitore nella tabella S3T_OFT e le rispettive righe nella tabella S3T_OFR.

Prima di tutto --Dati di Raggruppamento ordini

```
Procedure Leggi_OFT(pLancio varchar2,
                   pRoc_id pggft_roc.roc_id%type,
                   pRt      varchar2,
                   p_cod_consegna
PGGFTT_ORFOCR.ORFOCR_COD_CONSEGNA%TYPE,
                   P_MGU_COD
PGGFTT_ORFOCR.ORFOCR_MGU_COD%TYPE,
                   P_CLI_COD_PL
PGGFTT_ORFOCR.ORFOCR_CLI_COD_PL%TYPE,
```

```

                P_TP_PRM
PGGFTT_ORFOCR.ORFOCR_TP_PRM%TYPE,
                p_flag_di      Varchar2,
                p_lic_cod      Varchar2,
                pRis          IN OUT trcRisOft)

```

```

--Rottura in Raggruppamento OC 1:1

```

```

/* fornitura */
/* priorità */
/* codice Tipizzazione */
/* Ubicazione
/* Polo Logistico */
/* Tipo Preimballo */
--NON richiesta rottura ordini 1:1
/* priorità */
/* codice Tipizzazione */
/* Ubicazione */
Polo Logistico */
/* Tipo Preimballo

```

```

--Riferimenti di raggruppamento Ordini

```

```

--Condizioni Commerciali Cliente (solo se in Raggruppamento OC è richiesta rottura 1:1)

```

```

/* fornitura */
--Verifica Esenzione IVA
/* Stato dell'ordine */
/* Ordinamento merceologico */
/* Uscita */
/* Codice cliente */
/* Data OC */
/* Priorità */
/* Priorità */
/* Codice agente */
/* Codice destinatario */
/* Causale sospensione ordine */
/* Data sospensione ordine */
/* Causale sospensione spedizione */
/* Data sospensione spedizione*/
/* Canale di vendita */
/* Numero ordine cliente iniziale */
/* Importanza commerciale */
/* Importanza per consegna */
/* Importanza per pagamento */

/* Listino di vendita */
/* Flag di ordine generato in replica */
/* Campi per ASIS Web */
/* Canale di fornitura */
/* Categoria Prodotto */
/* Codice Tipizzazione */
/* Cliente di Spedizione */
/* Polo Logistico */

```

```

/* Tipo Preimballo */
/* Flusso Dati */
--Inserisci Testata
--Inserisci Righe
-- *** Scrittura Primo Numero Ordine IC
/* Verificare se tutte le righe of per quella testata hanno codice
produttore omogeneo.
In caso, si aggiorna il codice produttore di testata of.*/

```

Righe degli Ordini Fornitori

```

Function Inserisci_OFR(pRoc_id pggat_roc.roc_id%type,
                      pRt      varchar2,
                      p_cond_comm_frn
PGGFKSOGCCO.T_COND_PAG_REC,
                      p_cod_consegna
PGGFTT_ORFOCR.ORFOCR_COD_CONSEGNA%TYPE,
                      P_MGU_COD
PGGFTT_ORFOCR.ORFOCR_MGU_COD%TYPE,
                      P_CLI_COD_PL
PGGFTT_ORFOCR.ORFOCR_CLI_COD_PL%TYPE,
                      P_TP_PRM
PGGFTT_ORFOCR.ORFOCR_TP_PRM%TYPE,
                      p_flag_di      Varchar2,
                      p_lic_cod      Varchar2,
                      pN_Ins out number)
                      return boolean

End;

Procedure Leggi_OFR (pLancio          varchar2,
                    pRoc_id          pggat_roc.roc_id%type,
                    pRt              varchar2,
                    porfocr_dett_rt_bulk_nmax number,
                    p_cod_consegna
PGGFTT_ORFOCR.ORFOCR_COD_CONSEGNA%TYPE,
                    P_MGU_COD
PGGFTT_ORFOCR.ORFOCR_MGU_COD%TYPE,
                    P_CLI_COD_PL
PGGFTT_ORFOCR.ORFOCR_CLI_COD_PL%TYPE,
                    P_TP_PRM
PGGFTT_ORFOCR.ORFOCR_TP_PRM%TYPE,
                    p_flag_di          Varchar2,
                    p_lic_cod          Varchar2,
                    pRis              IN OUT trcRisOfr)

End Leggi_OFR;

```

La procedura Leggi_OFR viene richiamata all'interno della procedura Inserisci_OFR che a sua volta è richiamata dalla procedura di creazione e inserimento del nuovo ordine fornitore e delle righe dell'ordine.

La Procedura OFBULK *** Aggiungere esclusione OIA , Aggiungere raggruppamento per ordini di proprietà e generazione OF bulk GF Corp in stato s3 di tipo bulk **

```
-- non considero gli OIA
```

Per gli Ordini Bulk, nel package PGGFKORCUTILS la procedura seguente reperisce gli ordini per la produzione ODP:

```
Function possible_SL_transition(p_oct_soc_cod
s3t_soc.soc_cod%type,
                                p_oct_id
s3t_oct.oct_id%type)
    Return Number;
```

La gestione della replica ORCOFREPLICA

```
FUNCTION ELABORA(p_oct_soc_cod    VARCHAR2 DEFAULT NULL,
                p_oct_id          VARCHAR2 DEFAULT NULL,
                p_f_commit        BOOLEAN DEFAULT TRUE,
                p_ocr_riga        VARCHAR2 DEFAULT NULL,
                p_f_cambio_stato  BOOLEAN DEFAULT TRUE,
                p_storno_of       NUMBER DEFAULT 0)
    RETURN PLS_INTEGER
    /* Elaborazione ordine cliente $1. */

    --Gestione Ordini in Replica
    /* Codice fornitore non valorizzato per ordine $1. */
    * Condizione di acquisto non presente per il fornitore $1 e la
    stagione $2. */
-- *** chiamata get_listino con dati prima riga OC
-- se valorizzato lo sovrascrivo A quello delle condizioni
commerciali
/* Il codice listino non valorizzato nella condizione commerciale
per il fornitore $1
e la stagione $2. */
/*      Se viene passata solamente la testata dell'ordine,
allora elaboro testata e tutte le
sue righe; altrimenti salto l'elaborazione della testata e
passo direttamente alla riga
di quell'ordine. */
/* DATI AGGIUNTIVI PER ASIS WEB */
/* Etichetta 1 , 2, 3 */
```

```

/* Tipo Servizio */
/* Etichetta Alt. */
/* Divisione produttiva/Produttore */
/* Direct Import */
/* Codice Pacchetto */
/* Numero Pacchetto */
/* polo Logistico */
/* Codice Magazzino */
/* Prezzo mancante per il listino $1, modello/parte $2. */
--RAISE_APPLICATION_ERROR(NULL, -20200);
--ord. su misura
--2° livello fornitore
--1° livello fornitore
--Errore in inserimento/modifica nota testata ordine.
/* Inserisco OFR */
ord. su misura
--2° livello fornitore
--1° livello fornitore
/* Verificare se tutte le righe of per quella testata hanno codice
produttore omogeneo.
    In caso, si aggiorna il codice produttore di testata of.*/
    --ricaviamo il codice fornitore dalla tabella PGGFT_LIC (
    Licenziatario)
END ELABORA;

```

La gestione della creazione di un nuovo ordine fornitore replica è stato gestito in modo particolare introducendo il numero ordine fornitore NUOF fittizio per ovviare alla perdita di numeratore durante la fase di inserimento dell'ordine dovuto agli errori sulle anagrafiche dell'oggetto la mancanza di un dato obbligatorio per la creazione dell'ordine. La procedura è la seguente:

```

PROCEDURE aggiorna_of_t_nuof(p_of_t_soc_cod    in
s3t_soc.soc_cod%type,
                                p_of_t_eof_cod    in
s3t_of_t.of_t_eof_cod%type,
                                p_of_t_id        in
s3t_of_t.of_t_id%type ,
                                po_of_t_nuof     out
s3t_of_t.of_t_nuof%type) is

    n_nuof_eff  s3t_of_t.of_t_nuof%type;

BEGIN

    S3KORFUTILS.ins_num_ord(n_nuof_eff,
                            p_of_t_soc_cod,

```

```

        p_offt_eof_cod,
        g_anno_nuof,
        NULL,
        NULL);
if n_nuof_eff is null then
    S3KSYSMESS.batch_messaggi(758, sysdate);
    return;
end if;

update s3t_offt offt
    set offt.offt_nuof          = n_nuof_eff,
        offt.dagg_car37       = offt.offt_soc_cod || '-' ||
                                offt.offt_eof_cod  || '-' ||
                                offt.offt_anno     || '-' ||
                                TO_CHAR(n_nuof_eff)
where offt.offt_soc_cod      = p_offt_soc_cod
    and offt.offt_id        = p_offt_id;

update pggft_ocof ocof
    set ocof.ocof_offt_nuof    = n_nuof_eff
    where ocof.ocof_offt_soc_cod = p_offt_soc_cod
        and ocof.ocof_offt_id   = p_offt_id;

po_offt_nuof := n_nuof_eff;

EXCEPTION
    when others then
        raise;
END aggiorna_offt_nuof;

```

Conclusione

Valutazione

Per l'introduzione dell'accorpamento delle varie società del gruppo creando la possibilità di ammettere ordini multi produttori, e ampliando il flusso degli ordini nel sistema informativo GESTION, abbiamo fatto delle modifiche sulle strutture esistenti come le tabelle, i programmi e introdotto nuovi moduli funzionali. La soluzione innovativa per la gestione dei codici a barre ha comportato la stesura di un programma con cinque mille righe di codice. Questa soluzione permette una gestione al livello mondiale degli articoli prodotti dal gruppo. E non solo, favorisce una individuazione rapida e meno costosa dei prodotti presenti in un'ordine di più polo produttivo novità introdotta con la ristrutturazione.

Durante il periodo di deployment del nuovo flusso degli ordini intercompany, la documentazione prodotta per il software è stata distribuita per servire di linea guida agli operatori finale. La durata prevista di quattro mesi complessivo di lavoro, è risultato di tre mesi, con l'entrata al regime delle nuove funzionalità.

Per ridurre il rischio di fallimento dello sviluppo del software, abbiamo combinato il metodo agile e il metodo cascata nelle varie fasi del progetto: analisi, progettazione, implementazione, test e documentazione. La metodologia tradizionale a cascata, ci ha permesso di dare struttura e di porci degli obiettivi chiari in partenza. I suoi difetti, sono stati sormontati con i pregi della metodologia agile. I metodi agili preferiscono la comunicazione in tempo reale, preferibilmente faccia a faccia col cliente "Comunicazione stretta - Secondo Alistar Cockburn, probabilmente il primo teorico delle metodologie leggere, questo è l'unico vero aspetto nodale che renda leggera una metodologia", a quella scritta (documentazione) e la semplicità, direttamente mutuato dalla programmazione Object-Oriented, semplicità nel codice, semplicità nella documentazione, semplicità nella progettazione, semplicità nella modellazione; i risultati così ottenuti sono una migliore leggibilità dell'intero progetto ed una conseguente facilitazione nelle fasi di correzione e modifica. L'obiettivo è la piena soddisfazione del cliente e non solo l'adempimento di un contratto, rispondendo ai cambiamenti o modifiche al progetto, e abbattendo i costi di sviluppo del software.

In conclusione, i parametri di qualità esterni, che si riferiscono alla qualità del software così come percepita dai gli utenti raggiunti sono: La correttezza che Nelle metodologie di sviluppo e progettazione agili, si parla piuttosto di "soddisfazione del cliente", intendendo con questo che i requisiti iniziali potrebbero essere stati modificati, ma con il consenso e a piena soddisfazione del committente. L'utilizzazione degli ultimi meccanismi offerti dalla tecnologia Oracle, ha contribuito alla realizzazione di moduli con alte prestazione in unità di transazione al secondo (DataPump, Semaforo ecc...) aumentando l'efficienza, e la scalabilità dell'elaborazione generale degli ordini. La gestione delle varie casistiche di errori incontrate, ha maggiormente alzato la robustezza l'affidabilità e l'usabilità di tutto il modulo di gestione degli ordini. Per i parametri di qualità interni, la metodologia agile favorisce la produzione di software facilmente verificabile, mantenibile, riparabile e riusabile a causa del codice semplice derivato dalla programmazione Object-Oriented. Per la tipologia del nostro intervento, la alta integrabilità

delle procedure introdotte, hanno permesso di assicurare una maggiore coesione con il resto dell'applicativo e di comunicare con gli applicativi delle altre società fuori dal gruppo.

Difficoltà incontrate

Tra le difficoltà incontrate, la prima di tutte è stata la variazione durante l'evoluzione del progetto dei requisiti delle nuove implementazioni durante gli incontri col cliente. Seguito dal poco rispetto dello standard di sviluppo nei programmi esistenti, aumentando così il tempo di analisi per favorire una maggiore integrabilità tra i vari moduli.

Proposte future

Sul lavoro svolto sugli ordini, l'introduzione di un nuovo stato ordine aiuterebbe a ridimensionare la quantità di ordini selezionati per i vari aggiornamenti. Questo nuovo stato dell'ordine: S7 servirebbe dunque ad indicare gli ordini evasi e chiusi, aumentando le prestazioni e assicurando allineamenti dei dati.

La corporazione delle società del gruppo, se da una parte aiuta la gestione intercompany, dall'altra rende la struttura di GESTION molto macchinosa. La gestione dei sistemi System e Local aumenta il flusso di dati e favorisce il disallineamento dei dati molto sensibili.

Proponiamo una maggiore documentazione dei programmi e rispetto dello standard di sviluppo di codice per ridurre i tempi di analisi al livello tecnico e permettendo una rapida indipendenza negli aggiornamenti e nelle modifiche da apportare ai programmi.

Bibliografia

www.csc.com/it , CSC Itatia S.r.l . (Ultima visita 26/09/2012).

www.oracle.com, Oracle Report, Forms, PL/SQL Developer, ADF. (Ultima visita 26/09/2012).

GESTION Ordini Intercompany documentazione aziendale della nuova gestione (Giugno 2011).

Grant R.M. , Contemporary Strategy Analysis , Wiley-Blackwell 2007, ISBN-13: 978-1405163095

D. Smith, Exploring innovation, McGraw-Hill, 2006.

www.indicod-ecr.it , Soluzioni, standard e servizi per le imprese. (Ultima visita 02/09/2012).

www.gs1.org/barcodes/technical/bar_code_types, The global language of business. (Ultima visita 26/09/2012).

www.checkpointsystems.com , Checkpoint Systems Srl. (Ultima visita 03/09/2012).