POLITECNICO DI MILANO

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

MONITORAGGIO DELLA QUALITA' DEI DATI IN PROCESSI AZIENDALI

Relatore: Prof. Barbara PERNICI

Tesi di Laurea di:
Laura VILLANI
Matr. 739299

Anno Accademico 2010 / 2011

Ad Alessandro, Lucia, Luigi e Sofia.

Ringraziamenti

Desidero innanzitutto ringraziare la Prof. Pernici per i preziosi insegnamenti durante quest'anno di lavoro insieme e per le numerose ore dedicate alla mia tesi. Inoltre, ringrazio sentitamente l'Ing. Cappiello sempre disponibile a dirimere i miei dubbi durante la stesura di questa tesi. Intendo poi ringraziare il Dr. Fakir Hossain della Dublin City University per avermi fornito nuovi articoli e dati per la realizzazione della tesi, ed il Dr. Antonio Pintus per la notevole attenzione dedicatami.

Inoltre, vorrei esprimere la mia sincera gratitudine a tutti i miei compagni di corso per i numerosi consigli durante questi anni di studio. Infine, ho desiderio di ringraziare con affetto i miei genitori per il sostegno ed il grande aiuto che mi hanno dato.

Prefazione

Molte imprese hanno ormai compreso i benefici a cui porta una buona gestione dei dati. La qualità dei dati è fondamentale per le aziende che offrono ai loro clienti servizi e prodotti con un elevato livello di personalizzazione. La competitività delle imprese nell'era dell'Informazione dipende dalla loro abilità di offrire servizi personalizzati basati su una precisa segmentazione dei clienti. La rilevanza della qualità dei dati può essere capita se si osservano gli effetti della sua mancanza. Infatti la scarsa qualità dei dati può sfociare in predite sia economiche che sociali.

In questo lavoro abbiamo mostrato una tecnica per analizzare un processo e confrontare diverse configurazioni di verifica della qualità dei dati, che possono essere applicate al processo per misurarne le performance di qualità. Questa metodologia di analisi è composta da vari formalismi usati per investigare diversi aspetti che influenzano la qualità. La prima analisi riguarda le collezioni di dati aziendali, mentre la seconda fase studia il processo alla ricerca delle dipendenze che causano i guasti. Una volta spiegato come analizzare il processo, abbiamo mostrato alcune possibili configurazioni per effettuare le verifiche di qualità, specificando un metodo di scelta che permetta di preferire uno scenario agli altri. Durante lo svolgimento di questa tesi abbiamo deciso di disegnare il Quality Block, cioè il monitor utilizzato per le verifiche di qualità che è uno dei fulcri di questa tesi e che è stato da noi concepito secondo due prospettive traversali. La prima prevede che ci sia un blocco dedicato al calcolo di ogni dimensione. Mentre l'altra prevede che ci siano blocchi differenti per misurare le prestazioni di qualità per ciascuno dei database. Queste scomposizioni riducono la complessità del processo complessivo ed il tempo di esecuzione, perchè evitano la ripetizione di alcune verifiche.

Abstract

Many companies have just realized about the increasing importance of data quality for enterprises which offer a high degree of customization. The competitiveness of enterprises in the Information era depends on their ability to offer personalized services based on an increasingly fine segmentation of their customer base. The relevance of Data Quality could be ensured by checking the effects of its lack. Poor quality results in social and economic losses.

In this work we are addressing a technique to analyze the business process and to compare different checking configurations, that could be applied to processes in order to reduce the time's interval between the source of a fault and its notification by the quality monitor to the process manager and to measure the data quality of the process. This interval represents a critical aspect that influences the success of repair strategies, because reducing it could make more easy to find the causes of the fault. The methodology for the analysis is composed of different formalisms used to investigate different quality aspects. The first analysis involves the data collection. The second phase of this methodology studies the process flow, researching the dependences of data that causes failure.

After the analysis, we provide a set of possible quality monitor configurations. Each one of this configurations is suitable for a specific scenario of process requirements. Finally we propose a method to choose the best configuration. To do this we applied some changes in the implementation of the Quality Block, that is the monitor dedicated to check if the quality requirements are satisfied. First of all, we divided it according to two different criteria. On one hand, we splited this block creating a sub-block for each quality dimension. On the other hand, we implemented a block that measures the quality dimension on every database used that the process uses. This simplification lets the process designer to save process

execution time because its allows he or she to fix only the needed sub-blocks in a certain point of the process flow, avoiding the repetition of the same monitor in a sequential point.

Lista degli acronimi

A – API :Application Programming Interfaces

B – BPEL: Business Process Execution Language

– BPML: Business Process Modeling and Notation

C – CIS: Cooperative Information System

D – DB: Data Base

– DFD: Data Flow Diagram

– DPML: Data Product Markup Language

– DQ: Data Quality

– DQM: Data Quality Monitor

– DQMF: Data Quality Monitoring Framework

E – ER: Entity-Relation (Diagram)

F

G

H – HTTP: HyperText Transport Protocol

I – IMS: Information Manufactoring System

– IP: Information Product

– IQML: Information Quality Markup Language

J

K

L

M – MAIS: Multi-channel Adaptive Information Systems
– MIS: InformationManufacturing System

N

O

P – PLI: Programmazione Lineare Intera

Q – QoS: Quality of Service

R

S – SOA: Service Oriented Architecture

T – TCP: Transmission Control Protocol

U – UML: Unified Modeling Language

V

W – W3C: World Wide Web Consortium
– WS: Web Service
– WSCI: Web Service Choreography Interface
– WSDL: Web Services Description Language

X – XML: eXtensible Markup Language

Y

Z

Indice

Capitolo 1

Introduzione

1.1 Introduzione al concetto di Data Quality

La qualità dei dati rappresenta oggi un problema di vitale importanza per tutte le imprese. E' fuori di dubbio che il successo delle aziende sia sempre più legato alla raccolta e all'utilizzo di grandi quantità di informazioni. Le decisioni a tutti i livelli sono guidate inevitabilmente dai dati acquisiti da un numero sempre maggiore di fonti e fruiti attraverso svariate tipologie di sistemi per i quali gli investimenti rappresentano una quota significativa del budget aziendale. Infatti gli investimenti in sistemi esperti di data warehouse, CRM ed ERP dimostrano l'attenzione crescente che un numero sempre maggiore di aziende rivolge a questo tipo di problematiche. Viviamo in un mondo dinamico dove i dati cambiano continuamente ed esiste la necessità di ritrovarli velocemente e con sicurezza e di tenerli aggiornati in modo che siano corrispondenti al valore attuale dell'entità che rappresentano.

Il meccanismo di Total Quality Management e altre filosofie di gestione hanno focalizzato l'attenzione delle aziende sulla convenienza per gli utenti dei prodotti e servizi finali di una buona gestione della qualità dei dati, hanno accentuato il bisogno di costruire la qualità intorno ai processi di produzione e di distribuzione e hanno sottolineato l'importanza della complessità richiesta in tali processi al fine del miglioramento del prodotto e del servizio finale. Negli anni in cui viviamo la competitività delle imprese è fortemente legata ai dati che riesce a trattare in maniera efficiente ed efficace per trarne un vantaggio competitivo. Infatti la

crescente offerta e richiesta di servizi e prodotti con un livello di personalizzazione molto elevato rende i dati degli utenti materiale prezioso da utilizzare e controllare con la dovuta cura. E' per questo che la qualità dei dati sta assumendo un ruolo sempre più importante nelle voci di costo delle aziende che vogliono mantenersi competitive.

Oggi abbiamo a nostra disposizione molte raccolte di informazioni in diversi formati: audio, record, disegni, mappe, immagini, blueprint, metadati, dati dettagliati, dai estrapolati, ect... Queste informazioni possono essere collezionate in basi di dati o in archivi disponibili online. Le aziende di oggi conservano una grande quantità di dati ma non necessariamente la qualità di questi è adeguata.

Esistono tre concetti fondamentali, tra loro dipendenti, che bisogna conoscere per parlare di dell'Infomation Quality. I dati vengono solitamente identificati con i fatti. Quando i dati vengono inseriti in un contesto e combinati tra loro con una struttura diventano informazione. Infine quando l'informazione assume un significato a seconda dell'interpretazione si parla di conoscenza. La motivazione che spinge le organizzazioni a conoscere e migliorare la qualità dei dati e dell'informazione sta diventando sempre più pressante. Le aziende non mantengono più un rapporto diretto con i loro clienti, fornitori e dipendenti e neanche con le istituzioni, il legame tra questi e l'azienda passa attraverso i dati su di loro che l'impresa riesce a collezionare. Molte aziende hanno capito l'importanza della Data Governance e si sono impegnate per migliorare. Sono nate così compagnie e servizi per la gestione e la risoluzione delle problematiche legate alla qualità dei dati, però tali soluzioni risultano spesso costose e difficili da attuare per svariate ragioni [2].

La qualità dei dati è un fattore critico per il successo delle iniziative di Business Intelligence aziendali. I dati di scarsa qualità possono facilmente e rapidamente propagarsi all'interno degli altri sistemi. Se le informazioni condivise dalle molteplici istanze applicative aziendali risultano essere contraddittorie, inconsistenti o imprecise allora anche le relazioni con clienti, fornitori e partner si baseranno su informazioni inesatte ed approssimative, innalzando i costi, riducendo la credibilità aziendale e causando perdite sensibili in termini di giro di affari.

Secondo Thomas C. Redman [32] per capire perchè la Data Quality sia così importante per un'impresa bisogna pensare alle conseguenze che può avere la sua

mancanza, perchè la scarsa qualità è, secondo Redman, pervasiva e costosa. Per capire quanto questo problema sia rilevante per un'impresa bisogna quindi osservare la portata dei suoi effetti. La scarsa qualità dei dati influenza il successo e l'immagine della organizzazione in vari modi. Primo tra tutti introduce costi superflui, alcuni studi hanno dimostrato che il costo diretto che un'organizzazione deve sostenere per correggere errori nei dati ricevuti da una seconda organizzazione è il 6 % del suo budget totale e che il costo stimato che una compagnia deve sostenere per la scarsa qualità dei dati relativi alla clientela è circa il 6-14 % dei ricavi. Inoltre diminuisce la customer satisfaction e crea alcune incongruenze nei processi decisionali perchè implementare sistemi di data warehouse o data mining su dati di scarsa qualità è molto rischioso. Infine ostacola il successo delle attività di re-engineering dei processi poichè in molti progetti di re-engineering la cosa fondamentale è individuare i dati giusti da collocare nel posto giusto al momento giusto per soddisfare le esigenze di un cliente, invece se i dati individuati sono sbagliati non possono essere di alcuna utilità al cliente. Un'altra ricerca in merito alla data quality del Data Warehousing Institute dimostra che la qualità dei dati gioca un ruolo fondamentale in termini di efficienza ed efficacia. La ricerca condotta su un campione di 647 intervistati ha stimato che, per le imprese americane, i costi relativi a problemi di qualità dei dati sono pari a più di 600 miliardi di dollari annui [7].

Le conseguenze della scarsa qualità dei dati si possono osservare nell'esperienza quotidiana, ma spesso è difficile risalire alla loro cause. Le cause più frequenti possono essere dovute a cambiamenti storici in cui l'importanza di un dato può cambiare nel tempo, a cambiamenti nei processi di business infatti l'importanza dei dati dipende dal processo in cui vengono utilizzati, oppure a fusioni societarie, dove l'integrazione dei dati può provocare non poche difficoltà, o anche alle leggi sulla privacy.

Dal punto di vista della ricerca, la qualità dei dati è stata affrontata in diverse aree: in ambito statistico, management e IT. Gli statistici furono i primi ad interessarsi alla qualità dei dati, proponendo una teoria matematica che analizzasse la duplicazione dei dati, già negli anni sessanta. All'inizio degli anni ottanta seguirono gli studi in management, che avevano come elemento chiave il controllo del sistema manifatturiero dei dati al fine di identificare ed eliminare eventuali

problemi durante la gestione dei dati. Infine all'inizio degli anni novanta, con l'avvento delle nuove tecnologie, i ricercatori informatici hanno iniziato a definire, misurare e migliorare la qualità dei dati elettronici salvati nei database, nei data warehouse, e nei sistemi legacy. La qualità dei dati nel campo IT può essere ritenuta un'area di ricerca piuttosto nuova. Molte altre aree informatiche hanno trattato temi simili quelli della Data Quality in passato e nel corso di questi anni sono stati sviluppati molti paradigmi, modelli e metodologie per affrontare il problema. Le aree legate alla qualità dei dati sono diverse. Si parla genericamente di Data Quality, per intendere una serie di azioni, strumenti, attività, processi e metodi, atti a migliorare la qualità dei dati a supporto del business in campo statistico, in quello della rappresentazione della conoscenza, come nei sistemi informativi gestionali, e nei sistemi di data mining e data integration. Spesso il concetto di data quality viene ridotto a quello di accuratezza, cioè la differenza tra il valore rilevato e quello reale. Infatti, i dati sono spesso considerati di scarsa qualità se contengono errori di digitazione o valori errati. Tuttavia la qualità dei dati è molto più che accuratezza dei dati, ma contempla altre dimensioni significative come la completezza, la consistenza o l'attualità. Gli studi hanno confermato che la qualità dei dati è un concetto multi-dimensionale. Le aziende devono affrontare sia le percezioni soggettive delle persone coinvolte con i dati sia le misurazioni oggettive sulla base dei dati di cui trattasi [7]. Definire il concetto di qualità non è un'impresa semplice. Per molte dimensioni è difficile trovare un'unica definizione anche se considerando le diverse definizioni presenti in letteratura si può comprendere come ciascuna di esse si riferisca ad un ambito ben preciso. Le diverse dimensioni hanno lo scopo di catturare il comportamento del sistema da un particolare punto di vista, ci sono dimensioni che analizzano i dati dal punto di vista del loro valore, altre che si riferiscono all'interpretazione dei dati e alla loro presentazione infine alcune dimensioni riguardano il contesto generale [7].

Spesso la qualità dei dati viene identificata con la definizione di fitness for use che potremmo tradurre come l'idoneità dei dati allo scopo per cui sono stati raccolti. Questa definizione sottolinea la dipendenza della valutazione della correttezza dei dati dal loro utilizzatore, infatti dati ritenuti opportuni per un dato scopo ed utilizzati da un certo utente possono rivelarsi inadatti per un altro.

Questa dipendenza della qualità dal contesto porta a considerare la qualità dei dati sia per quel che riguarda l'accuratezza sia per l'interpretazione semantica del dato. I dati di una società, conservati in basi di dati diverse, possono essere idonei per alcune divisioni ma non adeguati per altre della stessa società. Per quel che riguarda la correttezza semantica il problema diventa la diversa interpretazione dei dati a seconda dell'utilizzatore perchè alcuni possono essere pienamente consapevoli delle sfumature del significato dei vari elementi, che possono essere diverse per un altro utente. Inoltre con l'aumento del volume dei dati, la questione della consistenza interna dei dati diventa fondamentale, indipendentemente dalla idoneità all'uso per scopi esterni, ad esempio, l'età e la data di nascita di una persona possono entrare in conflitto all'interno di diverse parti della stessa base di dati.

L'analisi della qualità dei dati va condotta tramite quattro fasi principali: la definizione delle dimensioni di qualità, l'analisi dei dati, la misurazione delle dimensioni di qualità ed infine il miglioramento della qualità dei dati. Per quanto riguarda la prima fase, cioè la definizione delle dimensioni, rimandiamo alla lettura del prossimo capitolo. Nell'analisi dei dati il processo di memorizzazione è visto come attività principale in un sistema informativo. I sistemi possono essere classificati in relazione al ruolo dato alla memorizzazione dei dati. Se la memorizzazione dei dati è la fase finale, il sistema si dice di acquisizione, se l'accesso a dati memorizzati è l'attività iniziale, il sistema si dice di utilizzo e invece il sistema si dice di tipo combinato se i processi effettuano entrambe queste attività cioè utilizzano e acquisiscono i dati. L'analisi delle dimensioni di qualità dei dati è vincolata all'analisi del processo in cui i dati vengono utilizzati, per individuare le attività che introducono errori o influenzano la qualità dei dati. Purtroppo non ci sono algoritmi precisi per il calcolo delle singole dimensioni ma esistono algoritmi consolidati solo per alcune dimensioni come la completezza, l'accuratezza e l'attualità. Infine per quel che riguarda le strategie per il miglioramento della qualità dei dati esistono tre approcci. Il primo è l'ispezione e correzione in cui i dati sono controllati e confrontati con standard di qualità, gli elementi che non sono ritenuti idonei vengono scartati o corretti fino a quando non passano il controllo. Il secondo è il miglioramento e controllo dei processi dove l'obiettivo è identificare e eliminare le cause di errori. Mentre il terzo consiste nella progetta-

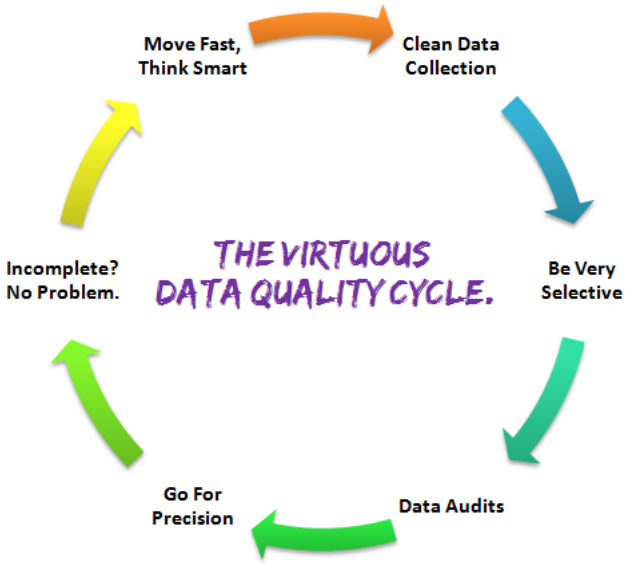

Figura 1.1: Ciclo di Data Quality che un'azienda dovrebbe seguire

zione dei processi tenendo conto dei requisiti di qualità. Le azioni di riparazione si applicano una volta studiate le cause della scarsa qualità i monitor di qualità hanno segnato la presenza di errori o il verificarsi di condizioni di scarsa qualità. Queste azioni possono coinvolgere l'intero processo o solo alcune delle attività che ne fanno parte. Per decidere quali siano le strategie di riparazione per la qualità dai dati più adatte il progettista deve analizzare il processo e scegliere quali azioni di riparazione adottare per garantire la riparabilità del processo. Per far ciò è necessario che conduca un'analisi del contesto aziendale e che stabilisca i requisiti di business, gli obiettivi ed i processi da monitorare. Analizzato il processo dettagliando le attività, gli attori, i ruoli, e i vincoli, segue la fase di progettazione dei processi, considerando di includere sia meccanismi di diagnosi che di riparazione. Infine il progettista stabilisce le azioni di riparazione da eseguire a run time.

Sul mercato sono presenti molte aziende che si occupano di fornire soluzioni per migliorare la qualità dei dati. Queste aziende si occupano soprattutto di effettuare data cleaning e di fornire linee guida per la gestione di questo problema. Sfogliando i consigli di queste aziende di consulenza si possono ricavare alcuni suggerimenti utili per la creazione ed il mantenimento di un buon livello di qualità come ad esempio la descrizione di un ciclo della qualità dei dati. Come prima cosa bisogna prestare molta attenzione alla fase in cui i dati vengono acquisiti e

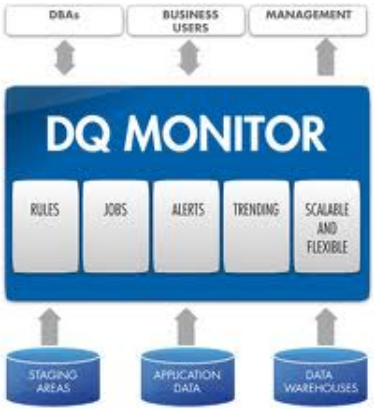

Figura 1.2: Monitor di qualità

collezionati, il salvataggio deve essere eseguito seguendo le best pratice in materia di data collection. Successivamente bisogna effettuare audit periodici sui dati, per controllare che la raccolta sia il più completa possibile. E' necessario raccogliere solo i dati di cui si ha bisogno, non esiste di fatto alcun limite alla quantità di dati che si possono raccogliere e conservare sul web, ma spesso la ridondanza e la sovrabbondanza dei dati ne causano la scarsa qualità. Senza dimenticare che la Privacy è un obbligo legale ma anche una dimensione di qualità e non è lecito conservare dati che siano eccedenti rispetto alle finalità della raccolta [1].

1.2 Obiettivo del nostro lavoro

La prima fase di questa tesi è stata quella di dedicarsi allo studio dello stato dell'arte riguardo la Data Quality. Tra i risultati che ci hanno particolarmente interessato dobbiamo citare le metodologie di misurazione della qualità per le principali dimensioni, seguite da alcune simulazioni di inserimento di guasti all'interno del sistema per studiarne gli effetti e da una metodologia di analisi delle informazioni e dei processi nuova rispetto alle competenze precedentemente acqusite. Questa metodologia di analisi permette di descrivere il processo di crezione delle informazioni attraverso alcune analogie tra questo processo e quelli manifatturieri. Una volta identificati gli sforzi fatti e i risultati raggiunti dal lavoro dei ricercatori informatici, abbiamo analizzato più approfonditamente uno dei componenti utilizzati per le analisi di qualità cioè il Quality Block. L'obiettivo

principale del nostro lavoro è quello di fornire la configurazione, cioè la modalità di inserimento di tale componente all'interno del processo, più adatta alle esigenze del processo di business, considerando la tempestività della notifica di eventuali anomalie senza trascurare eccessivamente il tempo di esecuzione dell'intero processo. Prima di fare questo è necessario tracciare una metodologia per analizzare il processo catturando i comportamenti e le informazioni che possano rilevarsi essenziali per stabilire quale configurazione scegliere. Questa metodologia affianca all'analisi della fonti dei dati uno studio dettagliato del processo. Individuare caratteristiche e strutture del processo possono rilevarsi azioni chiave per migliorare la qualità poichè permettono di identificare punti critici per la propagazione degli errori di qualità. La qualità dei dati è influenzata sia dalle attività che compongono il processo che dal modo in cui queste si susseguono. Abbiamo proposto un'analisi che permette al manager di delineare la struttura dei database e dei dati che il processo elabora, perchè il primo obiettivo è quello di rilevare se esistono dipendenze tra i dati che possono causare un peggioramento a cascata degli errori o errori in dati risultanti da fasi apparentemente non corrotte. Successivamente all'analisi dei dati collezionati, abbiamo descritto come sia possibile individuare due punti che determinano una distanza molto importante cioè quella tra il verificarsi di un errore ed il suo riscontro. Come mostrato anche in letteratura, ridurre questa distanza può aumentare la probabilità di successo delle strategie di miglioramento perchè facilita l'identificazione della cause del guasto, e conoscendo il motivo di un guasto è possibile progettare strategie di riparazione più adatta.

Per testare la metodologia proposta è stata applicata ad un caso di studio. Come processo test abbiamo scelto quello di gestione degli ordini, un processo semplice e soprattutto presente, se pur con qualche variabile, in tutte le aziende che producono e vendono beni. A questo processo abbiamo applicato l'analisi per la ricerca delle fonti dei dati e delle loro dipendenze per poi caratterizzare la struttura del processo.

Una volta conclusa la definizione dell'analisi abbiamo provato a disegnare delle configurazioni per il monitoraggio da applicare ai processi aziendali per misurarne le prestazioni in termini di qualità dei dati e per migliorare la tempestività di identificazione della cause di casi di scarsa qualità. Abbiamo anche stabili-

to dei criteri per scegliere quale sia la configurazione più adatta al processo da analizzare. Per concludere abbiamo studiato gli standard esistenti in termini di rappresentazione dei processi per implementare un processo ed i monitor che ne valutassero la qualità dei dati relativamente alle quattro dimensioni principali e verificare quale configurazione adottare sotto determinate condizioni. Una volta implementato il processo e i suoi monitor di qualità abbiamo applicato le configurazioni da noi proposte e sulla base dei risultati abbiamo tratto alcune considerazioni.

1.3 Struttura della tesi

A questo capitolo introduttivo, in cui abbiamo presentato i concetti generali che stanno alla base della qualità dei dati, seguiranno altri cinque capitoli in cui analizzeremo il problema della qualità dei dati sotto diversi aspetti. Uno dei punti centrali di questa tesi è il blocco di monitoraggio di qualità, cioè il Quality Block, vale a dire il monitor che si occupa di verificare il livello di prestazioni del processo in conformità ai requisiti di qualità dei dati.

Nel secondo capitolo viene descritto lo stato dell'arte, che disegna un quadro ad oggi del problema della Data Quality. Abbiamo scelto di descrivere i principali sforzi compiuti dai ricercatori, soffermandoci in particolare sul Quality Block. Abbiamo anche approfondito lo studio degli algoritmi per il calcolo di alcune dimensioni di qualità che il blocco deve verificare proponendo un'integrazione per uno di questi con quanto appreso nel corso di calcolo numerico in merito alla propagazione dell'errore in una funzione matematica.
Considerando che il periodo in cui viviamo è un periodo che molti definirebbero a cavallo tra l'era dell'informazione (in cui l'informazione viaggia molto velocemente soprattutto grazie alla rete e che nei primi anni ha portato a rappresentare l'informazione e la sua corretta gestione un vantaggio competitivo per le imprese che ne hanno saputo cogliere il valore) e un periodo caratterizzato dall'economia della conoscenza (che evidenzia i legami esistente tra i processi di apprendimento, l'innovazione e la competitività e fa sì che la conoscenza sia vista come una forma di capitale) abbiamo anche cercato di sottolineare l'enorme e crescente importanza di una buona gestione della qualità dei dati nell'industria proprio in questo

periodo, presentando da una parte i vantaggi di una buona gestione dei dati e dall'altra quali sono i costi in cui si incorre per colpa della scarsa qualità. La parte restante del capito relativo allo stato dell'arte si sviluppa in una presentazione dei maggiori risaluti di ricerca per il monitoraggio, la verifica e la simulazione della scarsa qualità dei dati.

Nel terzo capitolo abbiamo mostrato una metodologia di analisi del processo, utilizzando strumenti di analisi consolidati ed universalmente accettati. Abbiamo proposto la nostra analisi definendo e motivando le scelte fatte. Gli strumenti di analisi, come i diagrammi ER, i DFD, le rappresentazioni dei process flow, ci hanno permesso di caratterizzare come prima cosa la struttura delle collezioni di dati, alla ricerca di dipendenze, poi ci hanno aiutato nello studio del processo e delle attività per l'identificazione di dipendenze tra i dati e per individuare i vari punti in cui i guasti si verificano e vengono osservati. Abbiamo poi mostrato come applicare la nostra metodologia di analisi ad un processo in particolare.

Nel capitolo successivo vengono presentate le modalità di inserimento dei monitor di qualità nel processo di business. Prima dell'inserimento abbiamo apportato alcune modifiche ai concetti tradizionali dei monitor cioè la suddivisione del blocco di qualità in tanti sotto-blocchi quante sono le dimensioni da considerare e la scoposizione in blocchi dedicati uno per ciascun database che il processo utilizza. In seguito abbiamo ricavato tre diverse configurazioni possibili che sono state confrontate tra loro e con i possibili requisiti del processo, per ricavare il criterio che permette di scegliere l'alternativa più adatta al processo e alla richieste di qualità e tempi di servizio.

Nel quinto capitolo mostriamo gli strumenti utilizzati facendo una breve panoramica del paradigma SOA e dei due possibili modi di composizione dei processi, cioè orchestrazione e coreografia. Abbiamo poi implementato sia il processo che i blocchi di qualità ad esso associati e definito i risultati sperimentali ottenuti per effettuare la scelta più adatta alle esigenze di qualità del processo.

Infine abbiamo raccolto le nostre considerazioni e gli insegnamenti tratti da questa esperienza nel capitolo conclusivo. In cui abbiamo anche cercato di mostrare i pro e contro di questa analisi ed immaginato possibili sviluppi futuri di questo lavoro.

Capitolo 2

Stato dell'arte

2.1 La qualità dei dati e la sua importanza

2.1.1 Definizione della Data Quality

Contrariamente a ciò che si crede i dati non sono esenti da errori e devono rispettare dei requisiti. Quando si definisce la qualità bisogna definire dei vincoli, determinare come impostare tali vincoli e determinare quale sia il grado di obbedienza a questi (cioè il livello di tolleranza degli errori). La Data Quality si stabilisce quindi conoscendo se l'obiettivo è stato raggiunto o meno [20].

Nel corso del tempo le procedure e le tecniche si sono evolute per assicurarsi che i dati richiesti dai sistemi tradizionali di elaborazione delle transazioni possiedano un adeguato livello di qualità. Tuttavia, l'uso di dati legacy ha rifocalizzato l'attenzione sulla qualità dei dati e ha messo in luce problemi come la necessità di dati soft che non si riscontrano nei sistemi tradizionali. Inoltre, i dati di oggi sono visti come una risorsa organizzativa chiave e devono essere gestiti di conseguenza.

La qualità dei dati è il grado di rispondenza dei dati ai requisiti ad essi associati ed agli obiettivi per i quali vengono utilizzati. Il termine qualità dei dati può essere meglio definito come fitness for use, per sottolineare come il concetto di qualità dei dati sia relativo. Così i dati di qualità ritenuti opportuni per uno scopo potrebbero non possedere qualità sufficiente per un altro uso. Questa definizione di qualità dei dati implica una dipendenza dal contesto e permette di osservare due problemi uno legato all'accuratezza e l'altro alla semantica. Per

quanto riguarda la prima l'idoneità all'uso implica che si deve guardare oltre le preoccupazioni tradizionali. I dati che si trovano in sistemi di tipo contabilità possono essere precisi, ma inadatti per l'uso se non sufficientemente tempestivi. Le basi di dati situate in diverse divisioni di una società possono essere corretti, ma non idonee per l'uso se il desiderio è quello di combinare dati dai formati incompatibili. Un altro problema legato alla presenza di diversi utenti riguarda la semantica. I raccoglitori di dati e l'utente iniziale possono essere pienamente consapevoli delle sfumature che riguardano il significato dei vari elementi, ma che purtroppo non saranno gli stessi per tutti gli altri utenti. Così, anche se il valore può essere corretto, può essere facilmente frainteso. Un'altra tendenza che spiega la maggiore consapevolezza dell'importanza della qualità dei dati è il maggiore utilizzo dei dati soft in sistemi computer-based. (Con i dati soft si intendono dati con una qualità intrinsecamente non verificabile). In un senso molto reale i dati costituiscono la materia prima per le industrie nell'era dell'informazione. Il valore della materia prima in un'organizzazione è chiaro, almeno dal punto di vista contabile. Mentre il valore dei dati dipende quasi interamente dai suoi usi, che potrebbero anche non essere completamente noti [37].

La Total Quality Management è un modello organizzativo adottato da tutte le aziende leader mondiali e rappresenta una svolta importante nella gestione della qualità. Secondo questo approccio tutta l'impresa deve essere coinvolta nel raggiungimento della mission. Ciò comporta anche il coinvolgimento e la mobilitazione dei dipendenti e la riduzione degli sprechi in un'ottica di ottimizzazione degli sforzi. Uno dei concetti di base della qualità totale è che ogni analisi della situazione ed ogni azione di miglioramento deve essere basata su dati oggettivi, e non su sensazioni, in modo da poter comprendere e misurare il fenomeno e valutarne quindi l'effettivo miglioramento o meno.

2.1.2 Importanza della Data Quality per le aziende

I nostri giorni vengono spesso definiti con il termine di era post-industriale o, come abbiamo detto sopra, età dell'informazione in cui l'informazione è una forma di capitale molto importante se non addirittura la principale moneta di scambio. Pertanto non possiamo sottovalutare la sua qualità.

Una buona Data Governance può comportare diversi benefici per le imprese come ad esempio:

- Un miglioramento della qualità ed una velocizzazione delle procedure di decision-making.
- Il miglioramento dell'abilità di rispondere rapidamente ai cambiamenti del mercato permettendo l'adozione dei strategie di lungo termine.
- Il miglioramento della business intelligence.
- Una riduzione dei costi.
- L'effettiva osservanza del regolamento del governo e delle disposizioni in materia di privacy.
- Un aumento della customer satisfaction.
- Un miglior posizionamento sul mercato.

2.1.3 La scarsa qualità dei dati

Il primo step per risolvere un problema è quello di essere consapevoli del problema stesso e del suo impatto, tale passo è però difficoltoso da affrontare. Per facilitare il compito di creare consapevolezza si possono mostrare le conseguenze della scarsa qualità dei dati tramite effetti più conosciuti come error rate ed errori [31]. L'aspetto più efficace che evidenzia la necessità di Data Governance nelle aziende è rapppresentato proprio dalle perdite economiche risultanti dalla scarsa Data Quality.

La competitività delle imprese moderne dipende dalla loro capacità di offrire servizi personalizzati sulla base di una segmentazione sempre più fine della loro clientela. Il grado di personalizzazione che possono raggiungere dipende dalla qualità delle informazioni sui clienti che sono in grado di gestire, raccogliere, conservare ed estrarre dai loro database. La qualità delle informazioni è una questione fondamentale che consente all'impresa di ottenere un vantaggio competitivo basato sulla strategia customer-centric. Se le informazioni sui prodotti non incontrano le esigenze dei clienti i profitti diminuiscono. Con il miglioramento

delle informazioni sui prodotti un'organizzazione può migliorare la soddisfazione del cliente, aumentando l'efficacia e l'efficienza di un processo aziendale.

Gestire la qualità delle informazioni è una attività costosa, ma la prevenzione di un errore può costare ben dieci volte meno della sua risoluzione. I costi della non qualità dei dati per le aziende, compresi quelli irrecuperabili, possono variare dal 10 al 25 % delle entrate o del budget totale di una organizzazione [25]. Si rende così necessaria l'adozione di meccanismi atti a rilevare problemi di qualità dei dati ed a migliorare le prestazioni della qualità dei dati. Le fonti di costo dovute alla scarsa qualità sono fortemente context-dependent. Questo rende la valutazione delle perdite di scarsa qualità particolarmente difficile, come il valore dei dati stessi e la certificazione di qualità relativa ha conseguenze diverse a seconda del destinatario. Ad esempio, informazioni obsolete su uno stock possono far prendere ad un operatore decisioni di investimento sbagliate con notevoli conseguenze in termini di perdite economiche. Al contrario, se le stesse informazioni obsolete vengono utilizzate da un giornale mensile questo potrebbe non subire alcuna perdita economica [10].

Costi delle non qualità

I costi associati alla scarsa qualità dei dati sono spesso difficili da quantificare poichè, come accade anche per la maggior parte dei benefici IT, interessano sia componenti tangibili che intangibili. Senza una stima precisa le organizzazioni non sono in grado di comprendere l'esatto impatto della scarsa qualità sulla loro performance e quindi non sono spinte ad implementare le soluzioni corrette. Alcuni studi (Redman - 2003) hanno stimato il costo delle scarca qualità dei dati, qualora non si intervenga per migliorarla, pari a circa il 20 % del revenue di un'impresa [2].

La scarsa qualità aumenta i costi operativi perché il tempo ed altre risorse vengono consumate per identificare e correggere gli errori. Stimare il costo totale della scarsa qualità è un'operazione molto complessa. Alcuni studi sono riusciti a stimarla dall'8 % al 12 % del revenue, altri studi affermano che circa il 50 % delle spese di organizzazioni basate sui servizi possono essere consumate come effetto negativo della poor data quality [31].

Il costo totale della qualità dei dati è determinato dalla somma dei costi di valutazione della qualità dei dati e delle attività di miglioramento sia di prevenzione che di rettifica. Il costo delle scarsa qualità può essere ridotto implementando programmi efficaci di data quality, che sono però molto costosi. Tale costo può essere considerato un costo di prevenzione per evitare errori sui dati. I costi della scarsa qualità dei data possono essere classificati come costi di processo, che sono costi dovuti alla riesecuzione dell'intero processo a causa dell'occorrenza di errori sui dati, o come costi sulle opportunità, cioè quelli causati dalle perdite o mancate entrate [6].

Per misurare il costo della scarsa qualità dei dati devono essere considerati diversi componenti. In letteratura non esiste un quadro di riferimento generale per calcolare tale costo. Ci sono diverse interpretazioni, ma nessuno di loro considera tutti gli aspetti dei costi della non qualità. In [10] viene definito un modello generale in tale ambito. I costi sono classificabili in base all'impatto della scarsa qualità sulle attività di business. I principali problemi sono causati dalla perdita di performance nelle quattro principali dimensioni di qualità dei dati che sono relativi al valore dei dati: *accuratezza, completezza, consistenza e tempestività.* Per quanto riguarda l'accuratezza, i dati inesatti tipicamente causano comunicazioni errate, decisioning imperfetto, perdita di fiducia interna sui dati e la perdita di fiducia da parte dei consumatori. La completezza è influenzata da dati mancanti o incompleti. I dati mancanti in genere implicano occasioni mancate e decisioni sbagliate. Invece quando i dati derivano da fonti diverse, possono essere inconsistenti, e quindi causare la scelta di una decisione inappropriata o una perdita di fiducia sia all'interno che da parte del cliente.

Ogni problema di qualità dei dati ha costi diretti e indiretti. I primi sono connessi con le attività necessarie per compensare o correggere qualità inadeguata e possono essere espressi in termini monetari, mentre i secondi possono essere solo stimati e sono conseguenze della scarsa qualità dei dati.

I fattori che influenzano la qualità sono classificati in diverse categorie:

- **Tempo**: La scarsa qualità dei dati implica interventi extra sia di personale IT che di business, che devono analizzare i processi che manipolano i dati, per individuare le cause della scarsa qualità dei dati. Una volta individuate le cause il personale IT migliorerà i processi e correggerà i dati.

- **Fattore economico**: Se scarsa qualità dei dati implica l'insoddisfazione dei clienti e la perdita di clienti si possono avere perdite dei futuri revenue. Quindi le aziende che offrono servizi personalizzati potrebbero essere fortemente penalizzate dalla cattiva qualità poichè queste offrono un servizio basato su un buon trattamento delle informazioni.

- **Computing resource**: Il costo del computing può essere calcolato in base alle risorse coinvolte nella scarsa qualità del trattamento dei dati a cui corrisponde il tempo perso in prestazioni inutili [10].

La scarsa qualità può quindi far crescere i costi di un'organizzazione in diversi modi [2]. Il costo totale si determina combinando i costi di prevenzione e quelli di rettifica. Le attività di prevenzione sono tutte quelle che mirano al miglioramento del processo, ed i costi ed esse legati sono i costi IT che l'impresa deve sostenere per implementare un'architettura IT capace di percepire e correggere gli errori della data quality. Sono da includere in questa categoria anche costi derivanti da attività di controllo statistico del processo, programmi di miglioramento, operazioni di defect detection, testing, ispezioni ed audit. I costi relativi alle attività di rettifica sono invece imputabili ad internal failure, rework, investigazioni, azioni correttive e rettifiche dei dati anche dopo che questi sono stati comunicati all'esterno [10].
Oltre che sui costi, la scarsa qualità può avere effetti anche sul revenue, riducendolo a causa dell'insoddisfazione dei clienti o di una cattiva pubblicità, che spinge consumatori e partner a rivolgersi altrove. Inoltre, secondo alcuni studi, l'aver a che fare con dati di scarsa qualità, può influire negativamente anche sul lavoro dei dipendenti [2].

2.2 Misure di qualità

In letteratura è ormai accettata la definizione di data quality come costrutto multi-dimensionale. Tipicamente le definizioni delle dimensioni di qualità e le loro metriche si basano su una comprensione intuitiva e sull'esperienza industriale. Un ottimo modo per caratterizzare le dimensioni e le metriche è quello di distinguere

tra le misure quantitative e le valutazioni soggettive.
Il primo gruppo viene chiamato oggettivo e diviso in due categorie:

- La prima categoria è formata da metriche application independent: le loro definizioni si basano su una teoria esistente in un dominio del problema, ad esempio, metriche che misurano il grado di aderenza ai vincoli d'integrità in un modello relazionale. Questo tipo di metrica è universale e quindi applicabile in ogni data base.

- La seconda categoria contiene metriche appliction dependent: un esempio di questo tipo di applicazione è la frazione di record di una tabella contenenti dati ospedalieri esatti rispetto al numero totale di record della tabella stessa. Tale metrica si può applicare ad una specifica situazione come nell'esempio precedente in cui ci si riferisce ad un ospedale in particolare, ma non a tutte le tabelle in generale.

Le metriche di valutazione soggettiva sono quelle che misurano il giudizio personale (circa la qualità) dell'individuo che fornisce, manipola, usa o gestisce i dati. La diagnosi richiede una serie di misurazioni, purtroppo la letteratura non fornisce una scala per tali misurazioni e ciò può portare ad una errata interpretazione della misura. Nella maggior parte dei casi le metriche di valutazione della qualità dei dati sono espresse in forma percentuale. Stabilire quale scala applicare ad una dimensione non è affatto un compito banale e solo una volta determinata la scala completa si potranno confrontare i vari risultati [2].

2.2.1 Dimensioni di qualità

In tutte le metodologie per la valutazione ed il miglioramento della qualità dei dati la definizione delle dimensioni e delle metriche è un'attività controversa. Non esiste un'unica definizione per ciascuna di esse e diverse metriche possono essere associate alla stessa dimensione [6].

Ciascuna dimensione descrive un particolare aspetto della qualità dei dati. Sono importanti sia le dimensioni dei dati che degli schemi infatti, come la scarsa qualità dei dati influenza negativamente la qualità dei processi di business, così la scarsa qualità degli schemi può provocare ridondanze ed anomalie nel ciclo di

vita dei dati. Spesso però la data quality è considerata più importante perchè interessa le applicazioni della vita reale. Le dimensioni di qualità possono riferirsi sia ai valori che i dati assumono che alla loro intension (ad esempio al loro schema). Le dimensioni sia dei dati che degli schemi vengono espresse in maniera quantitativa ma non esiste un modo univoco di tradurle che sia condiviso nelle diverse metodologie presenti in letteratura. Per ciascuna metrica esistono diverse metodologie proposte per la sua misurazione, queste differiscono in molti aspetti:

- Dove avviene la misurazione.
- Quali dati sono coinvolti.
- Quale strumento viene utilizzato per tale misurazione.
- Su quale scala vengono rappresenti i risultati.

Le dimensioni di qualità sono diverse. Noi ci occuperemo delle quattro principali che sono associate al valore dei dati, ma ne esisteno altre. Le dimensioni oltre che associate al valore che assumono i dati possono essere relative alla data view, tra queste ci sono la granularità, la rilevanza e il livello di dettaglio, oppure possono essere associate alla presentazione dei dati come il formato e la facilità di interpretazione ed infine esistono anche dimensioni generali che sono la privacy, la sicurezza e l'ownership [11].

2.2.2 Le quattro dimensioni principali

Queste quattro dimensioni vengono considerate dimensioni privilegiate ai fini dell'analisi della qualità poiché sono misure oggettive [23].

Accuratezza

E' l'ampiezza con cui i dati si considerano corretti, affidabili e certificati. Viene definita come conformità con il mondo reale [36] e si calcola come la vicinanza tra un valore v ed un valore v', dove v' viene considerato come valore corretto nel mondo reale. Ad esempio se v=Jhn e v'=John il valore non è corretto. E' facile determinare il livello di accuratezza del valore di un dato quando il dato rappresenta una caratteristica di un'entità del mondo reale che può essere usata

come fonte corretta dell'informazione. Se invece i dati sono l'output di un processo di trasformazione complesso, l'accuratezza potrebbe risultare difficile da valutare, in quanto comporta la valutazione di un processo dinamico, in contrapposizione al confronto statico di valori. Molto spesso si calcola come rapporto tra il numero dei valori corretti ed il totale dei valori disponibili da una determinata fonte [11]. Esistono due tipi di accuratezza la prima sintattica e la seconda semantica:

L'**Accuratezza sintattica** è la vicinanza tra v e l'elemento corrispondente definito nel dominio D. In questo tipo di accuratezza ciò che interessa non è di quanto v si discosti da v' ma verificare se v appartiene al dominio D contenente i valori attesi. Ad esempio se v=Jack, v'=John e il dominio D rappresenta i nomi di persone allora v verrà considerato corretto perché Jack è un nome ammissibile. Questo tipo di accuratezza si misura con funzioni chiamate comparison function che valutano la distanza tra v e i valori del domini. Come la distanza di edit che considera il numero di operazioni di edit necessarie per riportare il valore attuale a quello ad esso più vicino presente nel dominio D. Questo è solo un esempio di funzioni di comparazione ma esistono anche funzioni più complicate.

L'**Accuratezza Semantica** considera la vicinanza di v al valore vero v'. Questo tipo di accuratezza è calcolato con valori di tipo [corretto, incorretto]. Questa accuratezza è più difficile da quantificare rispetto alla precedente perché per calcolarla è necessario sapere il vero valore di v'. Quando la frequenza degli errori è bassa e gli errori sono principalmente di digitazione, tali da produrre valori vicini a quelli reali, le due accuratezze coincidono. Di conseguenza l'accuratezza semantica può essere risolta sostituendo il valore inaccurato con il valore del dominio ad esso più vicino. In un contesto più generale, una tecnica per verificare questo tipo di accuratezza consiste nel risalire alle altre fonti dello stesso dato e cercare il dato corretto confrontando i valori. Quest'ultimo approccio richiede la risoluzione di un problema chiamato object identification problem, cioè quello di capire se due tuple si riferiscono o meno alla stessa entità del mondo reale. Per risolvere tale problema bisogna scomporlo in due questioni che riguardano l'identification e la decision strategy. La prima questione richiede di applicare una chiave di matching tra le tuple che nelle diverse fonti di dati hanno identificativi diversi, per metterle in corrispondenza, mentre la seconda si occupa di prendere una decisione in merito alla corrispondenza una volta effettuato il matching. E'

inoltre possibile calcolare l'accuratezza di un attributo o di una relazione ma anche dell'intero database. Se analizziamo l'accuratezza di un set di dati, invece che di un unico dato, dobbiamo considerare anche la duplicazione dei dati. Questo fattore diventa rilevante in strutture dati che non sono soggette a vincoli della chiave e può provocare costi aggiuntivi.

Sia R un schema di relazione consistente di K attributi e r un tabella di relazione consistente di N tuple. Sia $q_{i,j}$ (i=1..N, j=1..K) un booleano definito per ogni cella $y_{i,j}$ t.c. $q_{i,j} = 0$ se $y_{i,j}$ è sintatticamente accurato, $q_{i,j}$ =1 altrimenti.

1. Weak accuracy error:

   $\sum_{i=0}^{N} \frac{\beta((q_i > 0) \wedge (s_i = 0))}{N}$

   $\beta()$ è un funzione che restituisce una variabile booleana pari ad 1 se l'argomento è vero e 0 altrimenti,

   q_i>0 se si verificano errori di accuratezza,

   s_i=0 se non si è verificato il fenomeno dell'identification.

2. Strong accuracy error:

   $\sum_{i=0}^{N} \frac{\beta((q_i > 0) \wedge (s_i = 1))}{N}$

   in cui s_i=1 se si è verificato il fenomeno dell'identification.

3. $\sum_{i=0}^{N} \frac{\beta((q_i = 0) \wedge (s_i = 0))}{N}$

Completezza

La completezza si riferisce alla disponibilità di tutti i dati rilevanti per soddisfare i requisiti dell'utente [36] e coincide con la misura in cui i dati sono di sufficiente ampiezza, profondità e portata per quel compito. Viene definita in diversi modi: come grado con cui una certa collezione di dati include la descrizione del dato e l'insieme delle corrispondenze con gli oggetti del mondo reale, come abilità di un sistema informativo di rappresentare ogni stato significativo di un sistema nel mondo reale, oppure come insieme dei valori inclusi nella data collection, o infine anche come percentuale di informazioni del mondo reale presenti nel

data warehouse. Per verificare la completezza è necessario identificare se i dati mancanti esistono o meno e le cause di tale.

Esistono tre diversi tipi di completezza, caratterizzata dal soggetto a cui si riferisce:

- Schema completeness: è definita come il grado con cui i concetti e le loro proprietà non sono missing dallo schema.
- Column completeness: è la misura dei valori missing per una specifica proprietè o colonna della tabella.
- Population completeness: valuta i dati mancanti in relazione alla popolazione.

La completezza può inoltre riferirsi a diversi sistemi:

- **Completezza nei modelli relazionali** La completezza nei modelli relazionali può essere caratterizzata secondo due ipotesi chiave:

  1. Presenza/assenza di valori (NULL value).
  2. Assunzione di mondo aperto (OWA) o chiuso (CWA).

  In un modello che ammette il valore NULL questo sta a significare l'assenza di tale valore. Per descrivere la completezza è importante capire perché manca un valore, infatti un dato può esistere ed essere sconosciuto, può non essere noto se esso esista o meno o infine può non esistere.

  Nel caso in cui i vincoli siano l'assenza del valore NULL e l'ipotesi di mondo aperto la completezza di una relazione r si misura come una frazione delle tuple veramente rappresentate nella relazione r rispetto al numero totale di tuple in ref(r):

  $$C(r) = \frac{|r|}{|ref(r)|}$$

  dove ref(r) rappresenta la relazione che contiene tutte le tuple che soddisfano lo schema relazionale r e si chiama reference relation della relazione r.

  Invece nel caso in cui NULL sia un valore ammesso e vale l'ipotesi di mondo chiuso la completezza si calcola considerando la granularità del modello degli elementi come valori, tuple, attributi o relazioni. Per ciascuna di queste

granularità è possibile definire una completezza per catturare la presenza di valori null rispetto ai singoli valori/campi di una tupla, tutti i valori di una tupla, attributi di una relazione o all'intera relazione.

- **Completezza dei Web Data** La dimensione di completezza tradizionale prova solo una descrizione statica della completezza. Per considerare anche una componente dinamica, necessaria nei Web Information System, è stata introdotta la nozione di completabilità .

  Sia la funzione C(t) la completezza all'istante t con t ∈[t-pub,t-max] dove t-pub è lo stato iniziale di pubblicazione del dato e t.max corrisponde al tempo massimo in cui la serie dei diversi update schedulati viene completata.

  La completabilità sarà:

  $\int_{t-curr}^{t-mac} C(t)\, dt$

  dove t-curr è il tempo in cui la completabilità viene valutata con t-curr minore t-max.
  La completabilità può essere rappresentata da un'area Cb che rappresenta l'evolversi di C(t) nel tempo tra l'istante t-curr e t-max. Confrontando Cb con A è possibile attribuire a Cb i valori nel range [High, Medium, Low].

$$A = \frac{(t - max\text{–}t - curr) * (c - max\text{–}c - pub)}{2}$$

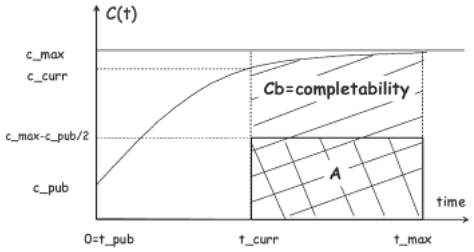

Figura 2.1: Rappresentazione grafica del calcolo della completabilità [7]

Dimensioni legate al tempo: attualità, volatilità e tempestività

Queste tre dimensioni relative al tempo sono forse più note tramite i termini rispettivamente Currency, Volatility e Timeliness:

Attualità Questa dimensione riguarda quanto prontamente o frequentemente i dati sono aggiornati. Non esiste una definizione standard per questa dimensione, essa viene definita come istante in cui i dati vengono salvati sulla base di dati [39], come intervallo di tempo che intercorre tra il momento in cui i dati vengono salvati e quello in cui vengono utilizzati [11] oppure come grado in cui i dati sono aggiornati [32]. Viene solitamente calcolata come segue:
Currency=Age+(DEliveryTime – InputTime)
Dove Age si riferisce all'età dei dati al momento della ricezione, DeliveryTime indica il tempo in cui l'IP è stato consegnato al cliente ed infine InputTime denota il tempo in sui si ottiene il dato.

Volatilità Si riferisce alla frequenza con cui dati cambiano nel tempo.
La Volatility è pari a 0 se i dati sono stabili, cioè non variano. Si esprime quantitativamente come l'intervallo di tempo in cui il dato rimane valido.

Tempestività Indica quanto i dati sono aggiornati per quel compito. Il suo valore è ricavabile dalla seguente equazione:
Timeliness=max{0,1-Currency/Volatility} dove 0 indica una cattiva Timeliness e 1 buona.

Consistenza

Si riferisce ad un eventuale violazione delle regole semantiche (vincoli di integrità) definite per un insieme di dati. Cattura la violazione delle regole semantiche definite su un set di dati. In relation theory queste regole sono istanziate come integrity constraints, mentre in statistica sono rappresentate dai data edits. Le prime sono proprietà che devono essere soddisfatte da tutte le istanze dello schema di un database e sono tipicamente definite sugli schemi. Esistono principalmente due categorie di tali vincoli: intrarelation constraints e interrelation constraints. Il secondo tipo di regole si riferisce a dati non soggetti a relazioni, consiste nel ricostruire relazioni su dati all'interno di un db (ad esempio se esistono due campi come "stato civile" ed "età" questo dovranno avere valori in contrasto tra loro)

2.3 L'approccio Information Product

Negli ultimi anni si è molto diffuso un nuovo modo di vedere l'informazione da parte sia dei professionisti che dei ricercatori in ambito di Data Quality che tratta le informazioni come prodotti manifatturieri cioè Information Product (IP).

Per definire in modo molto semplice cosa sia un IP si può descriverlo come un pezzo di conoscenza che è stato registrato in una certa maniera, ad esempio in formato audio o video oppure in formato di stampa, in modo tale che questo possa essere trasferito e condiviso con altri. Nel contesto di internet gli IP sono definiti come prodotti Knowledge-based o digital goods. Esistono davvero molti modi di confezionare e vendere informazioni. I prodotti più diffusi sono:

- Libri, giornali, opuscoli e report.
- Audio musicali.
- Video e film.
- Telecassi.

Queste sono tutte informazioni intangibili che sono state tradotte in formati trasferibili ad altri utenti.

Nel contesto aziendale invece un IP è utilizzato per comprendere meglio le varie fasi di un processo e misurare le performance dell'impresa in termine di Data Quality.

I prodotti di questo tipo posso assumere varie forme:

1. **Standard:** comprende informazioni come fatture, report, contratti o ricette mediche che hanno un formato predefinito e vengono utilizzati da moltissimi consumatori.
2. **Ad hoc:** un formato definito per una specifica applicazione.
3. **Storage:** costituiscono collezioni i record come file, cataloghi o database.
4. **Free format:** in questa categoria troviamo vari IP come libri, articoli ma anche radio broadcast. Questi sono i dati piú difficili da trattare perché spesso si tratta di informazioni create dagli utenti e sono anche quelli

piú soggetti a errori di qualità come sorgenti inaffidabili, ritardi o scarsa presentazione [29].

2.3.1 IMS e DataQuality Block

Da un'indagine condotta su un campione di 112 CIO di aziende che, nel 2008, hanno fatturato più di 1 miliardo di dollari, è emerso che solo per il 15 % di questi la loro organizzazione offriva un livello di qualità dei dati "alto o molto alto", però tale livello era stato raggiunto con uno sforzo economico notevole da parte dell'azinenda (nell'ordine di qualche decina di milioni). Questo è un risultato persino peggiore di quello fornito da un'analisi di 1996, in cui solo il 60 % delle compagnie lamentava scarsa qualità dei dati. La scarsa qualità dei dati in un'impresa può avere impatti rilevanti in ambito sia sociale che economico. Avendo a che fare con la crescita del problema della qualità dei dati molti ricercatori hanno adottato vari approcci per contrastare tale problema. La crescita dell'area di ricerca della data quality ha portato allo sviluppo dell'Information Manufactoring System. L'IMS è composto di vari blocchi tra cui uno è si focalizza in modo particolare su vari aspetti di qualità dei dati. Una parte dell'IMS è responsabile di garantire la qualità dei dati e viene quindi chiamato Data Quality Block. Il Data Quality Block è tradizionalmente integrato con l'IMS ma nonostante gli sforzi degli sviluppatori e ingegneri dell'IMS questo blocco è spesso soggetto ad errori che possono causare scarsa qualità dei dati. Per assicurarsi una valutazione indipendente della qualità dei dati è necessario un monitor indipendente che però richiede tempi e costi aggiuntivi. Un'altra sfida che caratterizza tale blocco è che spesso i sistemi vengono disegnati senza un adeguato modello che permetta di definire la qualità o che definisca la conformità degli output. È quindi difficile monitorare la qualità.

Si possono adoperare vari modi per legare i modelli dei dati al processo di business, ma la maggior parte di questi non supporta un ambiente integrato per la modellizzazione delle regole di business in relazione al modello dei dati. Alcuni ricercatori hanno suggerito una modello per la configurazione delle regole di qualità in modo che gli stessi dati nel IMS provino quali siano i requisiti di qualità nel loro ciclo di vita. Il modello proposto è il DQMF (Data Quality

Monitoring Framework). La modellizzazione del IMS assume un ruolo chiave e deve descrivere in maniera accurata e consistente tutte le informazioni in relazione con l'IMS [22].

2.3.2 Approccio IP per monitorare le Data Quality

I ricercatori hanno condotto molti studi per conoscere il crescente problema della qualità dei dati e hanno adottato vari approcci per contrastare tale problema. Come abbiamo visto la crescita dell'area di ricerca della data quality ha portato allo sviluppo dell'IMS.

Tra i modelli proposti uno ci sembra di particolare interesse: il DQMF (Data Quality Monitoring Framework). Il suo obiettivo è quello di completare l'IMS, sviluppando un sistema di monitoraggio indipendente dal processo che tenga continuamente sotto controllo i vari aspetti della data quality. Il DQMF è composto da tre componenti fondamentali.

Il primo è il DQM (Data Quality Monitor). Esso è un'applicazione che accetta come input le regole sulla qualità dei dati visti come prodotti e monitora continuamente i dati per assicurarsi che il grado di qualità sia soddisfacente. L'obiettivo del DQM non è intervenire nel processo ma è esclusivamente monitorare il processo per vedere se i dati soddisfano i vincoli di qualità. Se il prodotto non incontra i requisiti richiesti verrà inviato un segnale che notifichi l'inconsistenza. I vincoli possono essere espressi secondo diverse matrici che descrivono le misure di correzione.

Il secondo componente è il DPML (Data Product Markup Language) il quale è l'elemento fondamentale del DQMF. Per controllare effettivamente la qualità i modelli dei sistemi informativi devono descrivere in maniera sufficientemente accurata gli aspetti dell'IMS. Questo framework usa l'approccio IP che prevede di trattare i dati come output di un processo manifatturiero di dati grezzi. Durante la fase di design devono essere definiti i criteri di qualità che i dati devono soddisfare ai vari livelli della produzione, a tale scopo si utilizza il formalismo IP-UML. Usando il linguaggio UML si può creare un mapping visuale dei data process. UML è universalmente accettato perché facilmente esportabile.

Infine l'ultimo componente è il IQML (Information Quality Markup Language) un linguaggio basato sul XML che descrive il dato visto come prodotto. Lo scopo del IQML è lo stesso del DPML, l'IQML può essere auto-generato dal DPML o generato indipendentemente da questo. Dopo aver modellato il processo con DPML bisogna tradurlo in un eseguibile che possa essere processato da un software automatico [21].

L'approccio DQM non si limita a trovare le deficienze del dato visto come prodotto ma identifica anche le responsabilità. Può essere usato separatamente come meccanismo di test indipendentemente dal processo di manifattura dell'informazione. Però è difficile da applicare a sistemi già esistenti perché potrebbero non soddisfare i requisiti del framework.

2.4 IP-MAP

Le organizzazioni hanno finalmente riconosciuto l'importanza di un'alta qualità dell'informazione e a tale proposito molti ricercatosi si sono prodigati per proporre nuovi metodi per misurare e migliorare l'Infomation Quality. Tra i metodi proposti uno è appunto quello di trattare l'informazione come un prodotto e di descrivere i processi che la creano come se si trattasse di processi manifatturieri. Per applicare questo approccio si ricorre a uno standard di mappatura IP-MAP, che è un'estensione di IMS, un metodo di modellizzazione sistematico della manifattura di un IP. IP-MAP è un modello grafico disegnato per aiutare le persone a comprendere, valutare e descrivere come un IP venga assemblato [35].

Un utile estensione per rappresentare i processi è l'UML (Unified Modeling Language), questo può essere usato in moltissime applicazioni e costituisce uno standard de facto come linguaggio per l'analisi e il design nel mondo dell'ingegnaria del software, l'UML è un linguaggio molto generale e proprio per questo risulta molto versatile. Alcune estensioni dello standard permettono inoltre di adoperare tag, valori e stereotipi, ma purtroppo questo modello è soggetto a restrizioni semantiche [34].

I principali vantaggi dello standard IP-MAP sono:

- Facilità di visualizzare il processo.

- Visione complessiva che permette di implementare miglioramenti.
- Possibilità di misurare la qualità di un IP secondo le dimensioni descritte.

Oltre a questi obiettivi principali una mappatura IP viene impiegata per i motivi sottoelencati:

- Consente di provare un insieme di costrutti che facilitano la rappresentazione degli step coinvolti nel processo di creazione dell'IP.
- Permette al manager di esaminare in maniera critica gli step del processo, di implemantare l'Information Quality alla sorgente, di assegnare responsabilità e ownership relative alle attività e quindi anche alla qualità risultante.
- Costituisce una rappresentazione formale che può essere usata per valutare la qualità di un IP basata sulle dimensioni di qualità selezionate. La possibilità di utilizzare metadati permette anche di tracciare e gestire informazioni associate agli IP.
- Infine permette al manager di visualizzare le fasi più importanti ed identificare quelle critiche, individuando eventuali colli di bottiglia del processo.

Il formalismo IP-MAP prevede la scomposizione del processo in otto diversi blocchi e la rappresentazione del flusso di informazioni tra questi.

2.4.1 Blocchi

Il formalisimo IP-MAP scompone il processo in diversi blocchi. Ciascun blocco corrisponde a una fase della manipolazione dell'informazione ed è caratterizzato da una serie di attributi: [35]

- **Name** Nome univoco per identificare il blocco, non nullo.
- **Department/Role** Unità di business responsabile di processare i dati in esame.
- **Location** Luogo fisico (edificio, piano, stanza, area, etc.) in cui ha luogo l'attività.

- **Business Process** Descrizione dell'insieme di regole e procedure usate per il checking.
- **Composition** Descrive il tipo di dato che entra nel blocco.
- **Base System** Identifica il sistema che compie il movimento degli elementi di dati coinvolti.
- **DQ Metrics** Descrive requisiti delle dimensioni di qualità ad esempio: costi, tempo, descrizione di quanto bene il Data Quality Block rileva il difetti sui dati [29].

Source block

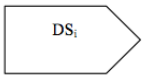

Figura 2.2: Rappresentazione di un Data Source block

Questo blocco è utilizzato per rappresentare la sorgente di dati grezzi che devono essere disponibili per produrre l'IP atteso dal cliente. A tale blocco vengono associate unità di business responsabili del dato grezzo, il processo che li utilizza e i database in cui reperirli. Questo blocco racchiude le informazioni messe a disposizione da parte dei clienti.

Customer block

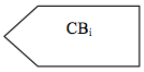

Figura 2.3: Rappresentazione di un Customer block

Descrive il consumatore dell'IP, che in questo blocco specifica gli elementi che costituiscono l'IP finito. I consumatori possono essere i clienti finali ma anche manager e dipendenti dell'organizzazione.

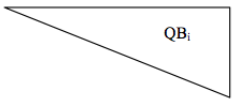

Figura 2.4: Rappresentazione di un Quality block

Data Quality block

Il Data Quality Block viene chiamato anche Quality Block, Evaluation Block o Check Block [29]. Quando si analizza la qualità delle informazioni in un processo è indispensabile definire il Data Quality Block, esso è un meccanismo di monitoraggio che viene collocato all'interno del process flow per valutare le dimensioni di qualità dei dati che interessano una determinata fase del processo. Tale blocco verifica quindi la conformità della qualità del servizio in rapporto ai parametri desiderati dall'utente. Ogni Data Quality Block si basa su un modello del sistema per interpretare un insieme di misurazioni, scoprire la presenza di errori ed identificare le cause specifiche di questo. Quando la qualità è al di sotto di alcune soglie il manager verrà avvisato con opportuni sistemi di allarme. Per poter applicare le azioni di repair più adatte è necessario identificare le cause dell'errore cui devono ovviare [13].

Una tecnica di analisi che è molto diffusa nell'ambito dell'analisi della qualità dei dati si basa sulla visione dei dati come Information Product. Il Quality block effettua le verifiche per la qualità dei dati su cui sono definiti requisiti per produrre un IP privo di difetti. Perciò, a tale blocco è associata una lista di check sulla qualità dei dati che vengono effettuati su specifici dati.

Questo blocco riceve in input dati grezzi (raw data items) mentre produce due diversi output: lo stream "corretto" (con una probabilità P) e uno "incorretto" (con probabilità 1-P). Le verifiche possono essere effettuate in vari modi ad esempio con verifiche manuali, tecniche di matching e verification automatiche o riconciliazioni elettroniche.

Purtroppo questo blocco non ha equivalenti in BPMN (Business Process Modeling Notation). Questa è la mancanza maggiore del BPMN che può essere oviata grazie al fatto che il BPMN consente la composizione delle marcature e la creazione di nuovi simboli, questo elemento può comunque essere rappresentato estendendo BPMN con l'aggiunta di un nuovo simbolo "DQDim" che permetta il

controllo dei dati valutando una serie di dimensioni di qualità [33].

Processing block

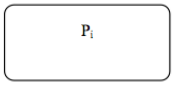

Figura 2.5: Rappresentazione di un Processing block

Tale blocco viene impiegato per descrivere ogni manipolazione, calcolo e combinazione di alcuni o dell'intero insieme di dati grezzi o Component Data per la produzione del IP. Sono esempi di processing block le operazioni di update, modifica, upload e la creazione di report e file.

Data Storage block

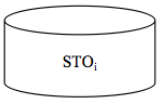

Figura 2.6: Rappresentazione di un Storage block

Questo blocco rappresenta i database e i file system in cui sono presenti le collezioni di dati in attesa di essere processati.

Decision block

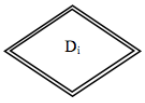

Figura 2.7: Rappresentazione di un Decision block

In alcuni processi è necessario, a seconda del valore di alcuni dati, dirigere gli stessi verso un certo insieme di blocchi. Questo blocco serve per descrivere le diverse condizioni da valutare e le corrispondenti procedure da applicare.

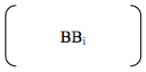

Figura 2.8: Rappresentazione di un Business Boundary block

Business Boundary block

Questo blocco rappresenta i casi in cui l'elemento di dati grezzi o i Component Data sono consegnati da una business unit ad un'altra, come il trasferimento di dati da un dipartimento all'altro.

Information System Boundary block

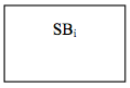

Figura 2.9: Rappresentazione di un Information System Boundary block

Riflette i cambiamenti dei dati grezzi o dei Component Data, cioè il loro movimento da un sistema informativo ad un'altro, come l'attività di data entry dal formato cartaceo a quello digitale. Possono avvenire sia all'interno di uno specifico ramo aziendale che all'esterno.

2.4.2 Flussi dati

Tra i componenti di IP-MAP, oltre ai blocchi, sono presenti anche alcuni strumenti per descrivere il flusso dati tra i diversi blocchi.

Figura 2.10: Rappresentazione di un Flusso dati

I metadati associati a questo tipo di componenti sono di due:

- **Description** Descrizione del dato.
- **Quality Metrics** Metriche di qualità per quel dato [35].

Information Product - IP

Rappresenta una collezione di dati prodotti in maniera manuale, meccanica o elettronicamente come risultato delle lavorazione dei dati grezzi. Alcuni esempi di questo tipo di informazioni sono certificati, fatture e ricevute.

Raw Data - RD

Sono i dati grezzi, un insieme predefinito di dati che vengono usati come materie prime in un processo che produrrà come output l'IP , ad esempio un numero, un record, un file o un'immagine.

Component Data - CD

Sono informazioni temporanee e semi-processate, necessare per la manifattura dell'IP, costituiscono, cioè, uno stadio intermedio tra dati grezzi e informazione finale. I CD sono i dati che dovrebbero essere generati durante l'IP-MAP e usati durante la creazione dell'IP finale. Esempi di questo tipo di dato sono i file estratti, i report intermedi e i dati semi-elaborati [29].

2.5 Analisi della qualità di servizi composti tramite inserimento di errori e ritardi

La composizione di servizi Web può essere adottata per sviluppare i sistemi informativi. Spesso si definiscono i processi con il termine servizi composti poiché questi possono essere visti come un unico servizio composto da tanti altri. Mentre le interfacce di servizi sono note al momento della composizione, la qualità del processo composto può dipendere dalla capacità dei servizi componenti di reagire a situazioni impreviste, come problemi di qualità dei dati o di coordinamento. La qualità di un Cooperative Information System dipende sia dalla qualità della composizione del processo che dalla qualità dei singoli servizi che lo compongono. Uno dei principali ostacoli all'adozione del paradigma dei Web Service in un CIS riguarda il problema di stabilire la qualità. Infatti i servizi sono distribuiti ed eterogenei, e spesso invocati senza una completa conoscenza della loro stessa natura

e qualità, in particolare nelle composizioni context-aware o in caso di processi flessibili e adattivi. In tali applicazioni, la natura dinamica della composizione di servizi esclude la possibilità di prevedere il comportamento delle applicazioni di fronte a guasti e situazioni inaspettate. Una delle abilità desiderate in un processo service-base composto è la self-healing, cioè la capacità del processo di reagire autonomamente al verificarsi di eventi anomali, causati da guasto al servizio o sui dati.

Alcuni ricercatori si sono mossi nella direzione di creare un metodo di testing sistematico per processi CIS service-based per studiare gli effetti di guasti sulle performance di qualità. Tale metodo si basa sull'inserimento di data fault e ritardi. I guasti possono essere di vario tipo a seconda quanto stabilito dalla definizione delle dimensioni di Data Quality [19].

Per le valutazioni quantitative è solitamente più comodo utilizzare le quattro dimensioni principali (accuratezza, completezza, consistenza e dimensioni legate al tempo) che sono dimensioni obiettive. I fault di queste dimensioni possono essere guasti riguardanti il valore del dato, come errori di digitazione, di formato, conflitti semantici o ritardi nell'aggiornamento, o possono riferirsi alla mancanza di un dato per indisponibilità o per ritardo nell'aggiornamento. Per valutare i guasti sui dati una delle metriche proposte riguarda la misurazione della distanza tra il punto in cui si inserisce il guasto e quello in cui se ne percepiscono gli effetti. I guasti possono essere inseriti durante lo scambio dei messaggi o direttamente all'interno dei database usati dai servizi, per effettuare il secondo tipo di inserimento è necessario accedere al database e conoscere il codice del Web Service. L'inserimento di data fault crea dati corrotti all'interno del servizio e tale guasto si può manifestare attraverso messaggi di notifica, arresto del processo, riparazione interna del fault oppure senza alcuna reazione. Inoltre gli effetti dei guasti non sono immediatamente riscontrabili durante il flusso di esecuzione del processo ma spesso capita che si manifestino dopo alcune interazioni. Riconoscere il fault il prima possibile, cioè il più vicino possibile alla sua nascita, è davvero importante per poter arginare gli effetti dell'errore e effettuare azioni di recovery in modo efficiente. Il tempo trascorso non è un buon indicatore della qualità perché è strettamente legato alla durata del servizio, al contrario si è rivelata soddisfacente l'adozione della metrica basata sul numero di messaggi scambiati

dalla sorgente del guasto sui dati fino alla rilevazione dell'errore, in cui un numero basso significa che il guasto è stato catturato vicino alla sua fonte.

Per alcuni guasti il processo può fornire autoriparazioni, secondo l'approccio WS-Diamond, tramite attività di recovery [19]. Come abbiamo visto le risposte del processo ad un fault possono esser diverse. In alcuni casi il guasto può non essere rilevato e quindi non si verifica alcuna reazione del processo, che conclude normalmente la sua esecuzione senza gestire i dati errati che il fault ha provocato. In altri casi si può verificare un arresto del processo. Infine il processo può reagire con azioni di recovery. Quando il numero di messaggi scambiati per ovviare a una situazione di errore è elevato si può provvedere a migliorare la qualità avvicinando la verifica al punto in cui il processo riceve i dati errati.

Un altro esperimento di analisi ha cercato di valutare gli effetti dell'inserimento di ritardo in servizi composti [19]. Il ritardo artificialmente inserito poteva essere fisso, quindi con un approccio sistematico, o definito seguendo una distribuzione di probabilità.

Dai dati sperimentali riportati si possono distinguere due situazioni possibili. Il primo caso riguarda l'invocazione di servizi in parallelo in cui il servizio con le performance peggiori influenza negativamente il ritardo di tutto il processo e si potrebbe implementare un miglioramento qualora si invocassero in parallelo diverse istanze dello stesso servizio o si sostituisse il servizio più lento con uno più veloce. Nel secondo caso l'errore, identificato dall'attivazione di un timeout, causa la terminazione incorretta del processi. In questo caso si potrebbe intervenire aggiungendo la possibilità di una re-invoke dell'attività che ha subito il guasto.

Altre analisi hanno investigato sugli effetti dei fault inseriti a vari livelli. Infatti, considerando le dipendenze all'interno di un'applicazione SOA complessa, diventa chiaro che ciascun componente, ciascun host, ciascuno scambio di messaggi è una potenziale fonte di guasti e che il suo impatto interessa l'intero sistema. Inoltre i guasti si possono verificare a diversi livelli dello stack di comunicazione. A livello di messaggi si parla di dati corrotti, a livello di esecuzione del servizi i guasti riguardano la QoS ed infine a livello di rete gli errori possono intralciare il flusso dei pacchetti tra gli host. I Web Service usano documenti WSDL per descrivere le loro interfacce e per definire la sintassi dei loro messaggi. I fault sui messaggi possono violare lo schema XML dei messaggi stessi o rendere un

documento XML incorretto. A seconda del grado di corruzione dei messaggi i meccanismi di correzione che si applicano possono essere a livello puramente semantico o sulla sintassi sia ad alto che a basso livello. I fault che si verificano durante l'esecuzione si possono contrastare intercettando le invocazioni delle istanze dei Web Service per simulare QoS. Infine a livello di rete i guasti in un architettura orientata ai servizi sono già gestiti del protocollo TCP-IP. Si può intervenire su guasti di questo livello intercettando i pacchetti ed effettuando operazioni di lancio, duplicazione, riordino, rallentando, etc...[23]

2.6 Integrazione per migliorare la qualità dei dati in sistemi multi-canale

Le cause della scarsa qualità possono essere diverse: dati sporchi all'interno del database, procedure di gestione dei dati inadeguate, errori software o incertezza contestuale. Un'altra causa di scarsa qualità che impatta le principali dimensioni è costituita dalla mancanza di integrazione tra i diversi canali e funzionalità di un sistema necessarie per fornire servizi ai clienti. [2]

Le organizzazioni moderne offrono servizi attraverso diversi canali. Diversi moduli funzionali condividono gli stessi dati che vengono salvati in diversi database locali. Solitamente non c'è integrazione tra moduli funzionali e canali, in quanto i canali vengono implementati in un momento diverso con progetti software indipendenti e spesso subiscono variazioni dei requisiti di disponibilità e performance. Tale mancanza di integrazione tra canali e funzionalità fa sorgere problemi di Data Quality che influiscono sulla qualità del servizio o del prodotto offerto al consumatore. Questa mancanza di integrazione inficia principalmente sulle dimensioni legate al tempo e sul periodo di esecuzione del processo. Le scelte architetturali possono quindi avere effetti sulla qualità.

In strutture multi-canale l'integrazione dei dati si ottiene con allineamenti periodici supportati da tecnologie di integrazione come datawarehouse e tool per database distribuiti. A differenza dei datawarehouse classici, le architetture multi-canale possono avere periodi di refresh diversi per ciascun database. Di conseguenza i ritardi nell'aggiornamento dei dati nel database possono essere diversi

nei canali corrispondenti. I clienti possono ottenere valori diversi a seconda del canale, del servizio e del tempo in cui richiedono i dati. Purtroppo solo una parte di questi dati è corretta. Per quanto riguarda l'accuratezza in sistemi informativi multi-canale una delle sue maggiori cause è proprio il ritardo nell'aggiornamento dei dati. Per capirlo basta pensare allo stesso dato che, se salvato in database con refresh diversi, assume valori diversi a causa del loro disallineamento. Per quanto concerne invece la completezza, affinché sia rispettata, bisognerebbe che al verificarsi di un nuovo evento questo sia simultaneamente registrato su tutti i database. Solitamente invece avviene che l'applicazione che genera il nuovo evento aggiorni solo il suo database locale affidando alle procedure di riallineamento il compito di propagare tale cambiamento. E' tra questi due eventi che si hanno problemi legati alla completezza. Ovviamente anche le dimensioni relative al tempo subiscono gli effetti di una cattiva integrazione tra canali, poiché queste dipendono direttamente dal numero di cambiamenti e dal periodo di riallineamento e di refresh dei database.

L'obiettivo a cui le imprese devono tendere è quello di un'integrazione totale perché questo tipo di architettura non presenta problemi legati al tempo. Gli utenti che accederanno ad ogni funzionalità tramite ogni canale leggeranno sicuramente le stesse informazioni come se si trattasse di un comune database. Partendo da un livello con basso gradi di integrazione, in cui ogni applicazione serve ciascuna combinazione canale-funzionalità, si può procedere andando verso un'integrazione totale ma anche fermarsi ad un grado intermedio. Il grado di integrazione può essere migliorato integrando tutte le funzionalità fornite dallo stesso canale (strategia channel-integration) o integrando le stesse funzionalità disponibili sui diversi canali (strategia functional-integration). Gli utenti di un'architettura del primo tipo possono ricevere dati obsoleti solo se cambiano canale. Gli utenti di un sistema che ha adottato la seconda strategia di integrazione possono realizzare che stanno ricevendo dati obsoleti se accedono ad altre funzionalità.

2.7 Strategie di repair

Una volta investigate le cause della scarsa qualità si procede applicando azioni di riparazione opportunamente disegnate volte al miglioramento delle misure di qualità. Queste azioni possono riguardare l'intero processo o una parte delle attività che lo costituiscono. La questione della riparabilità nasce nel momento in cui il monitor segnala un'anomalia.

2.7.1 Fasi della definizione di repair strategy adatte

Le fasi di analisi e di design delle azioni di repair sono fasi molto delicate: il disegnatore deve analizzare il processo e decidere quali azioni di riparazione adottare per garantire la riparabilità del processo. L'analisi del contesto aziendale è un metodo utile per analizzare l'ambiente, sia interno che esterno, in cui l'azienda opera e stabilire i requisiti di business, gli obiettivi ed i processi da monitorare. Le fasi dell'analisi e di design delle repair strategy sono tre. La prima è un'analisi del processo in cui si descrivono dettagliatamente i processi, le attività, gli attori, i loro ruoli, e i requisiti da garantire. La seconda fase consiste nel disegno dei processi in modo tale da garantire la self-healability di ciascuno di essi, in questa fase bisogna considerare anche meccanismi di diagnosi e di riparazione. Infine si disegnano le azioni a run time, che sono quelle azioni di repair che servono a contrastare errori che sorgono durante l'esecuzione del processo.

2.7.2 Classificazione dei tipi di repair

Le strategie di riparazione si possono classificare a seconda del tipo di riparazione che apportano, del livello su cui agiscono ed infine secondo il momento in cui sono disegnate.

Funzionali o non funzionali

Le strategie di riparazione, sulla base del tipo di riparazione che comportano possono essere distinte in *funzionale* o *non funzionali*.

Le prime azioni si applicano in tutte le situazioni eccezionali in cui, all'interno della logica di business, il workflow delle attività risulta in qualche modo corrotto

e l'attività produce errori e risultati anormali. Mentre fanno parte della classe di difetti non funzionali tutti gli errori che non sono legati alla logica di business interna. Questi difetti sono legati alle proprietà del workflow del processo, non a qualche attività o ai loro scopi. Il gruppo principale di tali difetti sono i QoS faults.

Livelli

Le strategie di riparazione si possono distinguere anche sulla base del livello della loro applicazione: *livello di istanza, a livello di classe, livello infrastrutturale.*

Le prime correggono la singola istanza errata in un certo istante di tempo. Le strategie di riparazione a livello di classe estendono la loro azione a istanze diverse di uno stesso processo. Infine i sistemi di gestione del flusso di lavoro e dei servizi web hanno i loro propri parametri e requisiti e quando questi limiti vengono violati diciamo che l'azione di riparazione è a livello infrastrutturale. Motivi di tali violazioni possono essere trovati nella progettazione del workflow: i parametri delle attività dipendono dal workflow concreto in cui vengono utilizzati, così in un flusso di lavoro possono errese adeguate, ma in un altro possono violare alcuni requisiti e vincoli. I difetti infrastrutturali sono per lo più guasti riguardanti la QoS e possono essere causati da valori errati dei parametri di processo del flusso di lavoro e sono le conseguenze di guasti di classe o di istanza e di riparazione devono andare al corrispondente di livello.

Tipologie di Repair Strategy

Talvolta le applicazioni possono incorrere in situazioni che non sono state previste nei modelli sviluppati. Quindi una prima distinzione da fare è quella tra le *eccezioni previste* che sono presenti nei modelli, e le *eccezioni inaspettate* che non possono essere anticipate a design time o per le quali il costo aggiuntivo per affrontarle a design time sarebbe troppo elevato e quindi non giustificato. Di conseguenza, è possibile distinguere tra strategie di riparazione a *run time* o a *design time* che spesso vengono chiamate rispettivamente correttive e preventive [12].

Strategia	Tipo di cambiamento	Livello
Inserimento di Monitor Funzionali	Funzionale	Di istanza
Exception Handler	Funzionale	Di istanza
Serivizi Ridondant	Funzionale	Di classe
Monitor sui Vincoli QoS	Non funzionale	Di classe e infrastrutturale

Tabella 2.1: Riassunto delle strategie di riparazione definite a design-time

Le strategie preventive sono i metodi di riparazione predefiniti che il progettista fornisce insieme al modello di flusso di lavoro, per aumentare l'affidabilità del processo. Questi meccanismi sono molto importanti e il designer li fornisce al fine di permettere ad un sistema di reagire in alcune situazioni eccezionali che il progettista può prevedere. Ogni metodo copre un'attività o un certo gruppo di attività e viene richiamato in alcune condizioni note a priori. In realtà, le strategie di recovery possono essere considerate una parte del modello. Poichè una strategia di riparazione non è altro che un insieme di attività che vengono richiamate in un determinato momento se una certa condizione è soddisfatta [4].

Le tecniche usate per realizzare questo tipo di recovery sono fortemente legate al modello usato per descrivere i Web Service, che può essere quello che segue dell'orchestrazione dei servizi o la coreografia di questi. Le repair action a design time si riferiscono solo al primo modello [12].

Le strategie definite a design time, per migliorare l'auto-riparazione possono prevedere:

- **Inserimento di Monitor Funzionali**: Il Monitor Funzionale è un monitor che analizza i messaggi scambiati tra i diversi moduli per rilevare la presenza di anomalie ed attiva azioni per prevenire i difetti o rimediarvi. Questi moduli vengono anche chiamati *data quality monitor* e vengono inseriti nel process flow per valutare le dimensioni di qualità dei dati scambiati. Quando i valori di qualità sono al di sotto di determinate soglie vengono inviati opportuni allarmi al manager del sistema. Le azioni di riparazione a design time richiedono l'identificazione delle cause degli errori. Un errore negli output può essere dovuto ad un errore generato dalle attività

che precedono quella analizzata oppure ad un errore generato dall'attività analizzata (self-generated error) . La rilevazione degli errori può essere effettuata con diverse tecniche: attraverso una tecnica chiamata data cleaning manuale ,cioè una comparazione tra il valore salvato e quello reale, oppure la data bashing, che consiste nella comparazione dei valori salvati nei diversi database, o infine tramite una procedura, che verifica che il nuovo dato inserito soddisfi i requisiti specificati cioè data cleaning utilizzando le modifiche dei dati.

- **Exception Handler**: Rileva e corregge difetti presenti in una singola istanza. Per far ciò utilizza diversi handler base come: Fault handler, Compensation handler, Event handler e Termination handler opportunamente combinati tra loro.

- **Serivizi Ridondanti**: Quando si disegna un processo è possibile inserire elementi ridondanti per ridurre la probabilità di failure durante l'esecuzione del processo e per migliorare la disponibilità del processo. Le azioni possono essere collegate da AND se la loro esecuzione avviene in parallelo o da XOR se sono alternative.

- **Monitor sui Vincoli QoS**: Per evitare failure si possono definire vincoli QoS che devono essere soddisfatti durante l'esecuzione del servizio. Questi monitor verificano se i vincoli vengono violati e invocano le relative azioni di recovery [4].

Lo svantaggio principale della repair strategy a design time è che, purtroppo, non copre tutti i casi possibili né tantomeno tutte le eccezioni.

D'altra parte lo sforzo di un generatore automatico di riparazioni a run time deve essere in grado di portare lo stato del sistema in modalità normale mediante l'esecuzione di una serie di azioni semplici di riparazione in un dato ordine [4].

Le azioni correttive, o a runtime, non modificano il process flow ma sono procedure che necessitano dell'aggiunta di componenti nel sistema di computing per supportare procedure di recovery quando si verifica un failure. Le azioni correttive per migliorare l'affidabilità del sistema possono essere implementate come procedure semiautomatiche e possono prevedere:

Strategia	Tipo di cambiamento	Livello
Redo/retry invocazione servizio	Funzionale	Di istanza
Sostituzione del servizio	Funzionale	Di istanza
Riconfigurazione architetturale	Non Ffunzionale	Di classe e infrastrutturale

Tabella 2.2: Riassunto delle strategie di riparazione definite a run-time

- **Redo dell'invocazione del servizio**: Riesecuzione del servizio con nuovi input.
- **Retry dell'invocazione del servizio**: Si applica quando si verifica una temporanea indisponibilità di uno o più servizi invocati durante l'esecuzione del processo. Si procede sospendendo il processo poi si ripete l'invocazione con gli stessi parametri.
- **Sostituzione del servizio**: E' una situazione più complessa in cui uno o più servizi sono considerati definitivamente non disponibili. E' quindi necessario sostituire tali servizi con altri che siano in grado di offrire le stesse operazioni.
- **Riconfigurazione architetturale**: E' particolarmente utile in casi di violazione di vincoli QoS, dovuti a mancanza di risorse hardware e software. In questo caso si procede riallocando ed eseguendo il servizio su una nuova macchina o su un altro application server [13].

La tabella 2.3 mostra la corrispondenza tra le dimensioni di qualità e le repair strategy, in tale tabella il simbolo + indica che quella determinata dimensione migliora se adottiamo la corrispondente strategia, il simbolo - che peggiora e = che la dimensione non subisce cambiamenti [4].

I valori in figura sono trend generali considerati validi per la maggior parte dei processi e dei contesti. Questa matrice fornisce al disegnatore del processo un utile supporto nella selezione della strategia di miglioramento da applicare. Il programmatore può scegliere di cambiare questi pesi a seconda del contesto. Per valutare il grado di influenza di una strategia su una dimensione è necessario

	Accuracy	Completeness	Availability	Timeliness	Execution Time	Reputation	Fidelity
Functional Monitors	+	+	= or -	-	-	+	=
Exception Handlers	+	+	+	-	-	+	=
Service Redundancy	+	+	+	-	-	=	=
QoS constraints Monitors	+	+	+	-	-	+	=
Redo/retry service invocation	+	+	+	-	-	=	=
Service substitution	+ or -	+ or -	+	-	-	+ or -	-
Architectural Reconfiguration	=	=	+	=	-	+	=

Tabella 2.3: Tabella che illustra la correlazione tra dimensioni e repair stratergy

definire un valore numerico. Spesso capita però che sia difficile per i designer attribuire un valore ai vari impatti, allora si ricorre all'utilizzo di fuzzy set, grazie alla logica fuzzy si può combinare l'espressività del linguaggio naturale ai vantaggi della rappresentazione numerica ed algebrica.

2.8 Ottimizzazione di processi per migliorare la qualità dei servizi

La qualità dei dati ha un ruolo importante in tutti i tipi di processi anche in quelli cosiddetti flessibili. Un progetto molto importante nell'ambito dell'ottimizzazione di processi flessibili con Web Service adattivi introduce un nuovo approccio di modellizzazione per affrontare il problema di selezione dei Web Service, problema rilevante se si tratta di grandi processi con vincoli di QoS severi. Anche le tecniche di negoziazione sono state prese in considerazione per poter dare una soluzione a tale problema. Un processo flessibile è composto da Web Service astratti, i Web Service vengono selezionati da un insieme di servizi con funzionalità equivalenti. L'obiettivo è quello di selezionare i servizi in modo ottimale tenendo in considerazione i vincoli dei processi, le preferenze dell'utente e il contesto. Quello della selezione dei Web Service è un problema studiato sia nell'ambito dei processi di business che in campo di e-science. La selezione dinamica dei Web Service si focalizza in particolare sulla consapevolezza del contesto del processo di business.

La consapevolezza del contesto può essere necessaria per la personalizzazione dei Web Service e quando il servizio viene personalizzato secondo le esigenze dei cliente. Le selezione dei Web Service è un problema di ottimizzazione che è stato studiato in diversi ambienti, sia in ambito grid che nell'area di ricerca dei servizi orientati al computing per processi business. In letteratura sono state presentati due tipi di soluzione. La prima soluzione propone un approccio locale che seleziona un Web Service alla volta associando le attività astratte a run-time con il servizio che rappresenta il candidato migliore [41]. Questo approccio garantisce solo un soddisfacimento locale di vincoli di QoS. La seconda soluzione risulta un po' più complicata e si occupa di ricercare, prima dell'esecuzione, l'insieme di servizi che soddisfano i vincoli del processo e le preferenze dell'utente per l'intero processo [5]. Quindi questo approccio è quello globale, così facendo i vincoli di QoS possono agire a livello globale, ad esempio imponendo regole sull'intera esecuzione del servizio composto, inoltre questo secondo tipo di soluzione considera il caso pessimo come scenario di esecuzione del servizio complesso. Una limitazione di questo approccio è che, nei processi con cicli, i loop vengono srotolati in un numero finito di iterazione, ad esempio nel numero massimo. Inoltre l'approccio globale introduce maggior complessità rispetto al locale. Talvolta in un processo di business di lunga durata l'insieme dei servizi identificati dall'ottimizzazione può cambiare proprietà di QoS durante l'esecuzione oppure alcuni servizi possono diventare indisponibili. Per garantire vincoli globali la selezione dei Web Service e l'esecuzione devono intersecarsi: l'ottimizzazione avviene quando il processo di business è inizializzato ed è cominciata la sua esecuzione, e viene iterata quando si esegue la riottimizzazione a run-time [14]. Per ridurre la complessità di ottimizzazione e riottimizzazione sono state proposte diverse soluzioni che garantiscono i vincoli globali solo per il percorso critico o che riducono i cicli ad una singola attività. Queste soddisfano i vincoli globali solo staticamente e applicano un metodo stocastico di riduzione del flusso di lavoro che consiste nell'imposizione di un insieme di regole di riduzione del flusso di lavoro fino a quando non esiste un solo compito atomico. Ogni volta che una regola di riduzione viene applicata, il flusso di lavoro della struttura cambia e dopo diverse iterazioni rimarrà solo un task. Quando questo stato finale è raggiunto il task rimanente contiene i parametri di qualità del servizio corrispondente al flusso di lavoro sotto analisi [15]. Un altro

svantaggio di soluzioni di seconda generazione è che, se l'utente finale introduce gravi vincoli di QoS per Web Service composti cioè limita le risorse (ad esempio, il budget o il tempo di esecuzione), nessuna soluzione può essere identificata e l'esecuzione del servizio composto fallisce.

É stata trovata la mappatura ottimale tra ciascun servizio Web astratto di un processo flessibile e un servizio Web concreto che implementa la descrizione astratta, in modo che la QoS globale percepita dall'utente sia massimizzata in presenza di severi vincoli di QoS. Per fare questo è stato introdotto un nuovo modello che prevede il peeling dei loop, la negoziazione (in caso non si riesca a trovar una soluzione) e una nuova classe di vincoli globali, che permettano di eseguire Web Service stateful. Per questo studio sull'ottimizzazione è stato utilizzato il framework MAIS in cui l'invocazione del servizio Web è basato sulla selezione dinamica dei servizi concreti in fase di runt-ime. (Il compito di selezionare i servizi concreti migliori in tale architettura è affidato al Concretizator.) Diversi criteri di qualità possono essere associati all'esecuzione dei servizi Web. Il registro di servizio MAIS comprende ben 150 dimensioni di qualità rilevanti. Per ogni dimensione vengono proposti una definizione, una metrica, e un sistema di misurazione. L'ottimizzazione è calcolata quando inizia l'esecuzione di servizi composti mentre la re-ottimizzazione viene eseguita periodicamente in fase di run-time, con l'intervallo di tempo che varia in base ai cambiamenti dell'ambiente e del comportamento degli utenti. In questo studio il problema di Web Service Concretization viene trattato come un problema di programmazione lineare intera. Il valore aggregato della QoS si ottiene applicando la tecnica Simple Additive Weighting, una delle più usate per ottenere un punteggio partendo da una lista di dimensioni. A run-time si applica la riottimizzazione quando l'invocazione di un servizio fallisce anche se, teoricamente, si potrebbe applicare dopo l'esecuzione di ogni task finché nuovi Web Service con miglior caratteristiche sono disponibili. La riottimizzazione però introduce un sovraccarico del sistema, dal momento che richiede tempo e che utilizza il registro del servizio della MAIS, accessibile al fine di recuperare l'insieme di servizi Web candidati ed i loro valori di qualità corrispondenti. Come in altri approcci, l'idea di base è quella di monitorare la qualità del servizio delle chiamate di servizio e rivalutare i valori di qualità attesi per il servizio composto. Dalle analisi dei risultati si nota che essi variano

a seconda del valore globale vincoli. Quando i vincoli globali non sono rigorosi, il peeling loop e lo srotolamento danno gli stessi risultati. Viceversa quando i vincoli globali sono più gravi il ciclo di peeling dà risultati migliori. Quando il vincolo di bilancio è minore o la probabilità di eseguire il numero massimo di iterazioni è alta, gli approcci di srotolamento e peeling determinano la stessa soluzione. Al contrario, quando la probabilità di eseguire il massimo numero di iterazioni è bassa, e il vincolo di bilancio è ridotto, l'approccio peeling migliora il tempo medio di esecuzione fino al 45%. La riottimizzazione viene attivata solo se il numero attuale del ciclo di iterazione è superiore al suo valore atteso. Le analisi di questo studio hanno dimostrato che questo comportamento è del tutto indipendente dalla distribuzione di probabilità dei loop. In conclusione questo studio dimostra la possibilità di garantire il rispetto dei vincoli a livello globale in condizioni più severe e di individuare la soluzione ottimale globale, invece di ottime locali o subottime [14].

L'analisi della letteratura ci è stata utile per vedere quali fossero gli studi fatti ed i risultati raggiunti in termini sia di analisi che di strategie per il miglioramento della qualità dei dati.
In particolate lo spaccato di stato dell'arte che abbiamo scelto di mostrare in questo capitolo si sofferma sulla figura del monitor di qualità che permette di valutare le prestazioni in qualità misurando se il livello di precisione per le diverse dimensioni di qualità incontra in maniera soddisfacente i vincoli cui è sottoposto il processo. Partendo da quello che è stato fatto in ora abbiamo controllato l'impatto delle verifiche di qualità sui processi di business e delineato una strategia di scelta per il monitoraggio più efficace.

Capitolo 3

Analisi dei processi per l'inserimento di Quality Block

3.1 Come analizzare un processo

L'obiettivo di questo capitolo è quello di delineare una metodologia per affrontare l'inserimento dei Quality Block all'interno del flusso di un processo, al fine di monitorarne la qualità. Per inserire in maniera corretta i blocchi è necessario conoscere il processo da controllare, osservandolo da diverse prospettive.

La nostra metodologia di analisi propone di tre fasi principali, che sono:

- Descrizione del processo.
- Studio delle collezioni dei dati.
- Analisi e semplificazione del flusso del processo

Queste tre fasi sono formate da analisi che ci permettono di cogliere diversi aspetti del processo e dei dati che si vogliono considerare.

3.1.1 Descrizione del Processo

Una descrizione del sistema in linguaggio naturale è fondamentale perchè la sua stesura aiuta a conoscere il processo. Per essere in grado di descrivere il processo, è necessario conoscere le varie fasi di cui è costituito. Per raccogliere informazioni

sul processo si possono effettuare osservazioni dirette o interviste ai manager. Contrariamente a quanto di pensa conoscere il processo oltre ai dati può essere fondamentale per poter migliorare la qualità dei dati che utilizza.

3.1.2 Definizione della struttura dei dati

Gli strumenti di questa analisi sono quelli utilizzati più comunemente nell'ingegneria del software durante la fase progettuale. Per conoscere un'informazione dobbiamo sapere sulla base di quali dati è stata generata, possiamo farlo con un'analisi del dominio.

Un'analisi del dominio solitamente si effettua tramite due tipi di diagrammi che fotografano la struttura dei dati da diverse angolazioni. Il diagramma Entity-Relation è uno schema concettuale che ci aiuta a comprendere le relazioni tra gli oggetti della realtà aziendale e la struttura dei dati, mentre un diagramma del flusso dei dati descrive, attraverso schemi funzionali, la realtà aziendale dal punto di vista delle operazione che il sistema deve svolgere sui dati. A questo punto conosciamo le risorse che il processo utilizza e ricordando che queste fotografie sono statiche andiamo ad approfondire l'evoluzione del processo e dei dati nel tempo.

Successivamente bisogna identificare le interazioni dell'utente con il sistema e quindi con il processo. Questo si fa tramite un'analisi dei requisiti con vari diagrammi UML. Il primo diagramma UML che ci sembra utile suggerire ed includere nella nostra analisi è quello dei casi d'uso che cattura il comportamento del sistema in termini di interazione con utenti sia interni che esterni al sistema. Per risalire alle competenze delle singole attività del processo possiamo usare un diagramma delle attività che rappresenta una possibile evoluzione del sistema nel tempo. Infine per controllare quali utenti utilizzino, creino o eliminino i dati ed in che ordine le vari attività avvengano si può utilizzare un diagramma di sequenza. Si può rappresentare nuovamente la struttura dei dati come sistema software secondo un modello ad oggetti e per farlo si usa solitamente un diagramma delle classi. Inoltre possiamo dettagliare la nostra analisi e verificare quali siano le informazioni necessarie all'esecuzione del processo con un diagramma di struttura del processo.

3.1.3 Analisi del process flow

Dopo aver analizzato il processo dal punto di vista dei dati e delle sue interazioni è il momento di analizzarlo in maniera più approfondita, non più come unico oggetto, ma guardare all'interno del processo e descriverlo come sequenza di attività. Tracciando il process flow si identificano le attività ed il modo con cui queste vengono eseguite. Le attività, come abbiamo visto, possono essere strutturate in parallelo, in alternativa oppure in cicli. Come già mostrato in letteratura una volta identificate le strutture per migliorare l'analisi di qualità si effettua un unrolling dei loop [14] e si calcolano le probabilità dei branch [15].

Giunti a questo punto si devono identificare le possibili cause di errore e ricavare il punto del processo in cui tali errori nascono e quello in qui vengono osservati. Questo porta alla scoperta delle dipendenze tra i dati che possono causare la propagazione dei fault. E' difficile e costoso prevedere ogni possibile eccezione e disegnare gli exception handler per ciascuna di esse. Diversi exception manager possono essere necessari per il medesimo failure, a seconda dello stato del processo e dell'origine dell'errore. D'ora in poi per semplicità ci riferiremo alla fonte della scarsa qualità con il termine errore e alla sua manifestazione con il termine failure. Evidenziare le dipendenze tra i dati è un utile strumento per effettuare forward repairability e per disegnare gli exception handler. Le dipendenze tra i dati servono per decidere quali attività devono essere compensate, ri-eseguite o quali servizi possano essere sostituiti, usando diverse strategia a seconda dell'origine del fault. La sostituzione si applica al processo per sostituire un'istanza di processo con un'altra, la compensazione viene solitamente usata per ripristinare stati corretti degli oggetti affetti da fault ed infine la retry si può applicare sia ai processi che ai singoli servizi e consiste nella ripetizione dell'esecuzione ed ha un esito positivo solo se i dati in input sono corretti [17]. Una fase importante dell'analisi di un processo è proprio l'identificazione delle dipendenze tra i dati per l'identificazione delle concatenazioni dei guasti.

Infine verificheremo come le informazioni vengano scambiate ed elaborate all'interno ed all'esterno del processo. Controlleremo più da vicino quali siano le informazioni, attraverso quali unità di business passino, da chi vengano create, da quali altri dipartimenti vendano utilizzati. Per far ciò è necessario descrivere

il processo secondo le specifiche del formalismo IP-MAP.

3.2 Descrizione del caso di studio

In questo paragrafo e nei seguenti applicheremo la metodologia di analisi proposta ad un caso di studio in particolare. Il processo adoperato come caso di studio è quello di gestione degli ordini di un negozio online. Il processo preso in esame deve prima di tutto riconoscere i clienti, una volta identificato il cliente il sistema gli consente di selezionare il prodotto che desidera e, compatibilmente con le disponibilità dei prodotti nel magazzino, il prodotto scelto viene inserito nell'ordine. Terminata la scelta il cliente effettua il pagamento ed il processo termina.

3.2.1 Analisi del dominio

In questa fase rappresentiamo in forma diagrammatica i principali concetti del dominio e le relazioni tra essi e gli attori coinvolti.

Diagramma ER

Su questo modello ER è stata costruita la base di dati del processo in analisi.

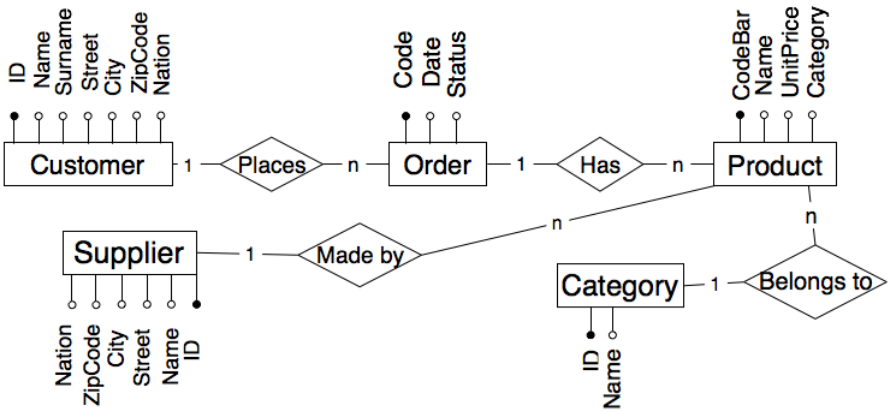

Figura 3.1: Diagramma ER

Glossario

E' necessario chiarire i termini utilizzati per comprendere i ruoli delle entità e delle relazioni utilizzate.

Per quanto riguarda le entità i termini utilizzati sono da interpretare come segue:

- Customer Scheda del cliente
  Attributi
  - ID: Identificativo univoco del cliente.
  - Name, Surname: Stringhe che indicano il nome del cliente.
  - Street, City, Nation, ZipCode: Indirizzo del cliente a cui spedire la merce acquistata.
- Order Ordine di acquisto
  Attributi
  - Code: Identificativo univoco dell'ordine.
  - Date: Data in cui il cliente ha effettuato tale ordine.
  - Status: Booleano che specifica se l'ordine è stato consegnato o meno.
- Product Prodotto in vendita
  Attributi
  - CodeBar: Codice a barre che identifica ogni singolo prodotto.
  - Name: nome del prodotto.
  - UnitPrice: Prezzo unitario di quel prodotto.
  - Category: Categoria a cui tale prodotto appartiene.
- Category Tipologia di prodotto
  Attributi
  - ID: Identificativo univoco della categoria.
  - Name: Nome della categoria.

- Supplier Fornitore che produce i diversi prodotti
  Attributi
  - ID: Indentificativo univoco del fornitore.
  - Name: Nome del fornitore.
  - Street, City, Nation, ZipCode: Indirizzo del fornitore.

Invece per le relazioni i termini assumono i seguenti significati:

- Places Associa il cliente agli ordini da lui effettuati.
- Has Specifica per ciascuna linea dell'ordine effettuato il prodotto corrispondente acquistato dal cliente.
- Belongs to Indica la categoria di appartenenza dei prodotti del magazzino.
- Make by Associa ciascun prodotto al fornitore che lo produce.

Diagramma di flusso dati

Il diagramma ci sarà utile per sottolineare le dipendenze tra i dati, perchè il sistema informativo attraverso di esso è visto come una rete di processi funzionali interconnessi da depositi di dati.

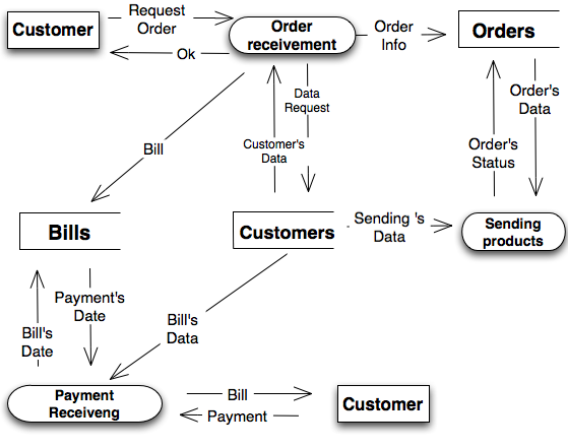

Figura 3.2: Data Flow Diagram

3.2.2 Analisi dei requisiti

Un'altra fase dell'analisi da effettuare è la descrizione del sistema tramite una serie di documenti di progetto redatti in linguaggio UML.

Use case

Il primo diagramma UML è lo use case con cui possiamo catturare il comportamento del sistema informativo in termini di interazione con l'utente o rispetto agli input esterni.

Questo diagramma dei casi d'uso descrive l'interazione tra cliente e sistema quando questo accede al sito di acquisti online per effettuare un nuovo ordine.

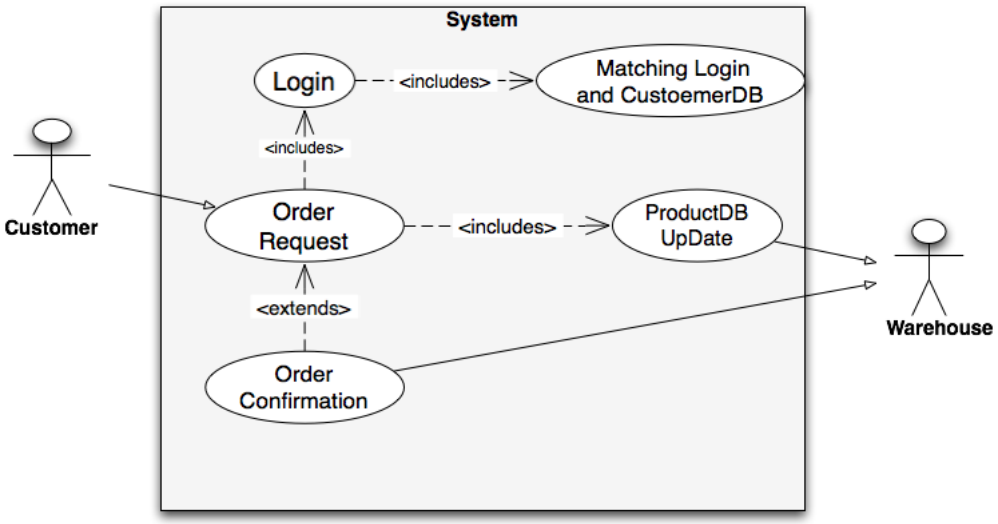

Figura 3.3: Use case diagram della generazione di un ordine

Gli attori di questo caso d'uso sono i clienti e il magazzino mentre il sistema rappresenta la nostra piattaforma di eCommerce.

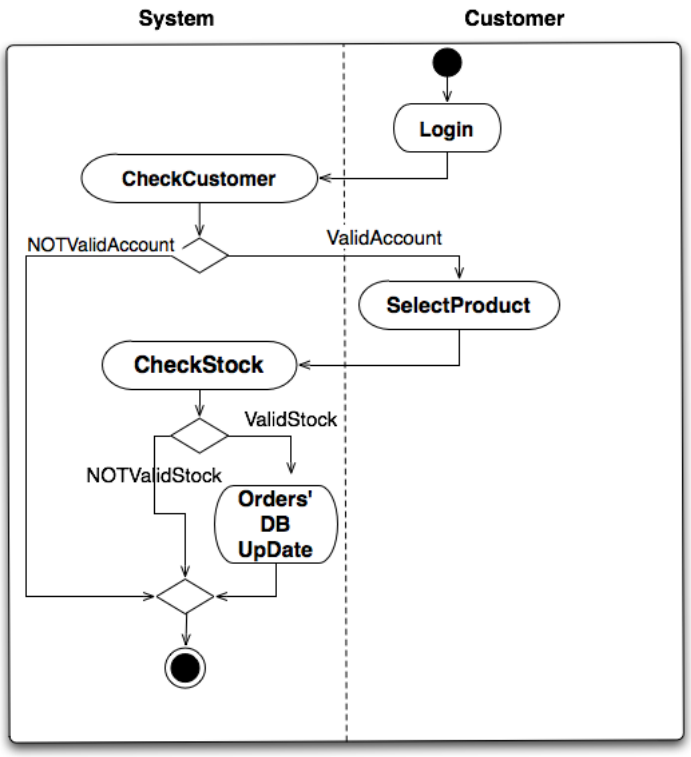

Figura 3.4: Diagramma delle attività

Diagramma delle attività

I diagrammi delle attività descrivono l'evoluzione del sistema nel tempo, sottolineando le competende delle diverse azioni. Questo diagramma descriver la richiesta di un nuovo ordine da parte del cliente.

Diagramma di sequenza

Questi diagrammi UML consentono di rappresentare le interazioni tra gli oggetti del sistema, collegandoli tramite le richieste di servizio tra loro.

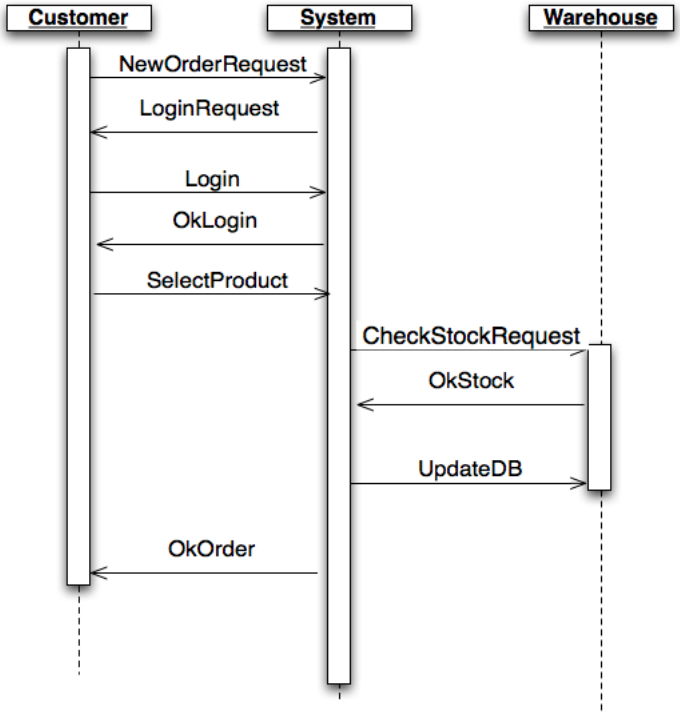

Figura 3.5: Diagramma di sequenza

Diagramma delle classi

Il diagramma delle classi rappresenta il sistema software ponendo l'accento sulla struttura degli oggetti specificando quindi classi di appartenenza, relazioni, attributi ed operazioni.

Oltre ai metodi standard *insert()* e *get()* è presente il metodo *getPrice()* per conoscere il prezzo unitario di un prodotto.

Struttura del processo

Il diagramma di struttura del processo aiuta a comprendere le risorse e le informazioni necessarie per il suo svolgimento.

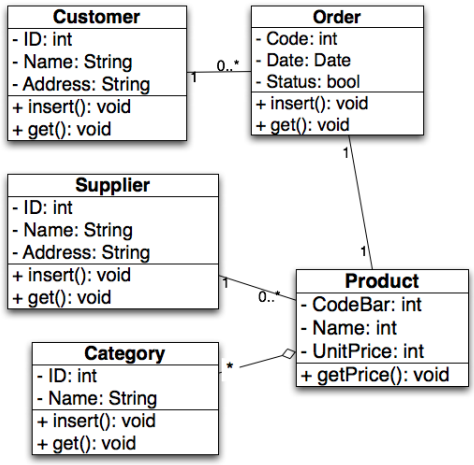

Figura 3.6: Diagramma delle classi

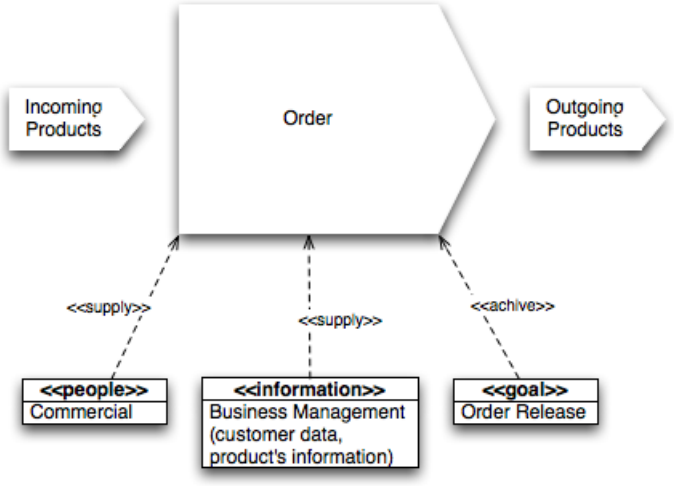

Figura 3.7: Structured process

3.2.3 Analisi del processo

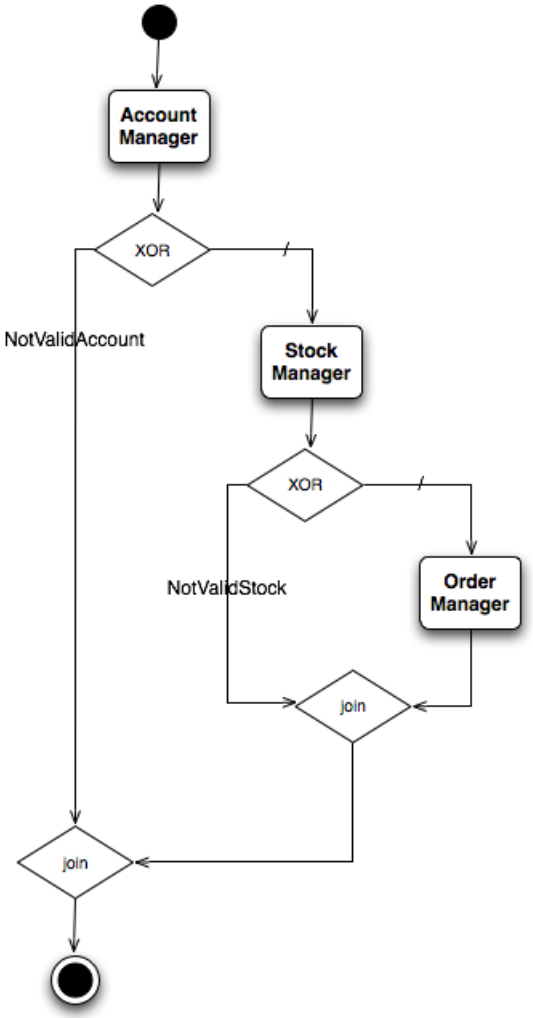

Figura 3.8: Process flow

In questa sezione analizzeremo il processo di gestione degli ordini di un'azienda che vende online i suoi prodotti sotto il profilo delle informazioni che utilizza e degli effetti risultanti dai guasti sui dati. Il processo di gestione degli ordini si compone di tre diverse fasi. La prima fase riguarda la verifica dell'identità del cliente che vuole accedere al sito. La seconda fase consiste nell'accertamento delle disponibilità del magazzino. Ed infine il processo si conclude con la terza fase che è la formulazione del nuovo ordine.

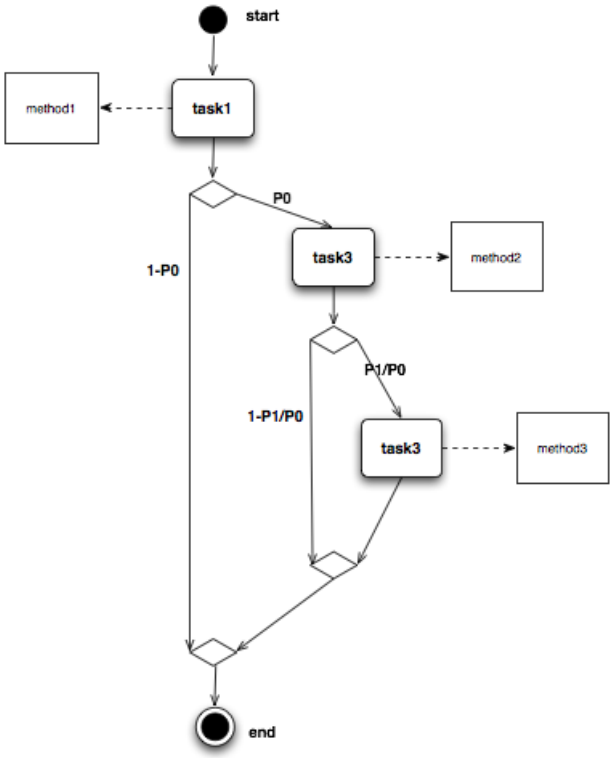

Figura 3.9: Analisi delle probabilità di branch

Probabilità di branch

In un'analisi di questo tipo è opportuno identificare le probabilità di ciascun percorso.

Come abbiamo visto può essere utile srotolare i cicli presenti e identificare le probabilità delle diverse alternative [15].

Il nostro processo non presenta cicli e le probabilità di branch sono presentate nella figura 3.9.

Analisi delle dipendenze dei dati per identificare le fonti di un guasto

Come abbiamo visto la conoscenza di informazioni riguardo il fault all'origine di Failure1 è cruciale per riparare il processo e per minimizzare il bisogno di ri-eseguire parti del processo. Queste informazioni possono essere ottenute con tool di diagnosi o ispezioni manuali [17].

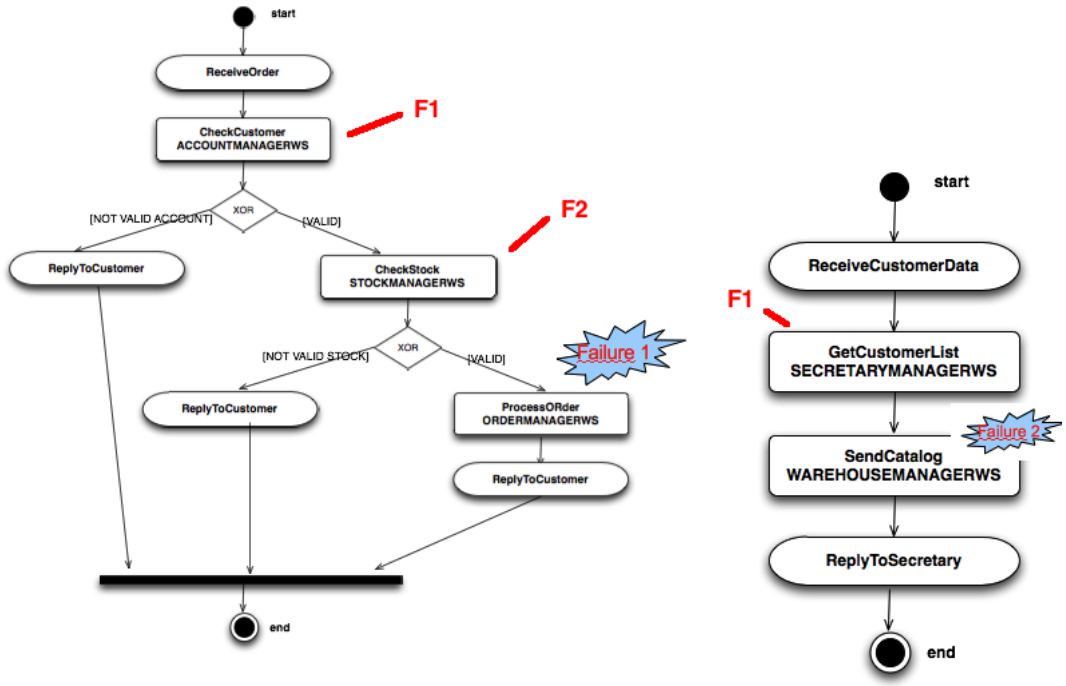

Figura 3.10: Identificazione delle fonti e delle osservazioni degli errori dovuti a scarsa qualità nel processo considerato e in quello di invio cataloghi

Le cause di tale errore possono essere F1 che rappresenta l'invio di dati errati o non aggiornati da parte del cliente e F2 che rappresenta un errata rendicontazione delle disponibilità dei prodotti. F1 e F2 possono dar luogo al Failure1 nel processo di gestione degli ordiini.

F1 potrebbe anche causare un secondo failure, il Failure2 ,in un altro processo, come quello relativo all'invio dei cataloghi dei prodotti ai clienti.

La descrizione del processo richiede non solo la definizione del control flow e della semantica dell'esecuzione ma anche delle dipendenze dei dati dati tra le attività.

Questa figura mostra un'istanza di processo con F1 in cui WS1 rappresenta il servizio che svolge le funzionalità della prima fase, WS2 rappresenta il servizio che si occupa delle seconda dare e WS3 dell'ultima, invece le attività (indicate come a0,a1,a2) corrispondono alle operazioni dei servizi. Gli oggetti o1, o2, o3 ed o4 rappresentano rispettivamente la scheda cliente, la disponibilità delle merci, gli ordini ed infine la singola riga di un ordine, cioè gli oggetti sui quali opera il

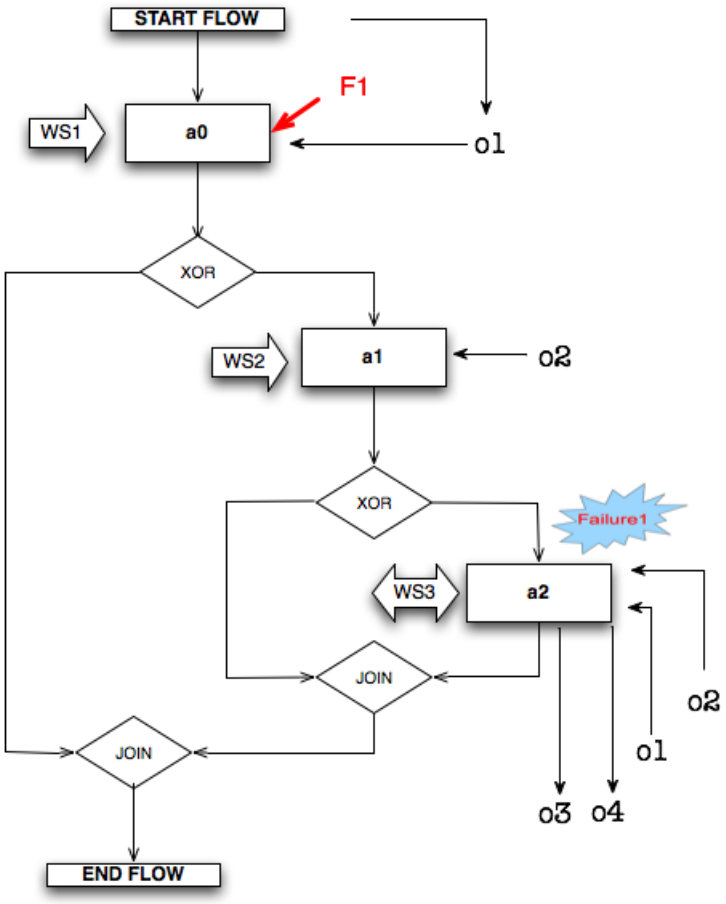

Figura 3.11: Istanza di processo in cui si verifica F1

processo.

3.2.4 Modellizzazione IP-MAP del processo di gestione degli ordini

Avendo presenti le categorie di IP precedentemente citate procediamo con la modellizzazione del processo di gestione degli ordini. Questa modellizzazione si compone di cinque fasi:

1. Catalogazione degli IP.
2. Identificazione degli IP critici.
3. Definizione dei requisiti di qualità dell'IP.

4. Costruzione dell'IP-MAP e del repository dei metadati.

5. Valutazione e miglioramento della qualità.

La modellizzazione IP-MAP ha come obiettivo principale quello di permettere ai manager di conoscere e visualizzare le più importanti fasi del processo manifatturiero che tratta un IP ed identificare quali sono le fasi critiche per quanto riguarada la qualità; serve inoltre ad individuare i colli di bottiglia del processo e di stimarne i tempi ed infine viene utilizzata per comprendere meglio la struttura aziendale e le responsabilità dei diversi processi.

Fase 1: Catalogazione degli IP

In questa fase della modellizzazione bisogna identificare le istanze di IP. Per il processo analizzato le istanze identifcate possono essere le seguenti:

1. Scheda dei clienti (che chiameremo Customer form).

2. Scheda dei prodotti (Product form).

3. Scheda dei fornitori (Supplier form).

4. Fattura degli ordini (Bill).

5. Database dei clienti (Customers DB).

6. Database dei prodotti disponibili nel magazzino (Products DB).

7. Database dei fornitori (Suppliers DB).

8. Report mensile di fatturazione e consuntivazione (Report).

Una volta identificati gli IP è necessario porsi alcune domande che ci permettono di descriverli al meglio, come ad esempio:

- Qual è il formato di tale IP?

- Chi utilizza tale IP?

- Come, dove, quando e perché viene utilizzato tale IP?

- Da quale processo proviene?
- Quali sono le sorgenti dei dati che lo costituiscono?
- Quali sono i requisiti di qualità e quali sono le sue criticità?
- Qual è la soddisfazione con l'IP?
- Esistono possibili sostituti?

Per ciascuno degli IP citati il reparto commerciale che si occupa della gestione degli ordini deve descriverne le caratteristiche come nella tabella seguente.

Scheda Cliente	Composta dal nome, dall'id e dall'indirizzo del cliente
Formato	Standard
Utilizzatore	Addetto del reparto commerciale che gestisce le fatture e magazziniere che invia gli oridini
Descrizione dell'utilizzo	Durante la fase di fatturazione e di spedizione del prodotto
Data Source	Customers' DB
Creazione	La scheda viene creata dei clienti stessi
Livello soddisfazione	Il livello è medio
Criticità	Non si può spedire la merce se non si conoscono nome ed indirizzo completo
Sostituti	Non esistono

Tabella 3.1: Scheda cliente

Fase 2: Identificazione delgli IP critici

In questa fase si cerca di individuare quali tra gli IP abbiano maggior bisogno di monitoraggio, per la criticità delle loro performance di qualità. Alcuni IP possono essere sufficientemente critici da giustificare un intervento sull'organizzazione se causano la crescita di costi di produzione, l'introduzione di nuove apparecchiature e sistemi o mancate vendite. Per questo processo abbiamo individuato come IP

critici, dal punto di vista della data quality, la scheda cliente e di conseguenza l'ordine. Da qui in poi identificheremo l'ordine con IP1 e la scheda come IP2.

Fase 3: Definizione dei requisiti di qualità dell'IP

Le metriche di qualità sono statistiche che misurano alcuni aspetti della qualità dei dati per uno specifico IP. Esempi di queste metriche sono l'accessibilità, l'interoperabilità, l'usabilità e la credibilità, di cui mostreremo alcuni esempi relativi al caso di studio.
Applicheremo il modello di mappatura proposto in letteratura [29] al caso di studio.

Requisiti di qualità per l'IP1: Ordine

Accessibilità:
Questa metrica aiuta a valutare quanto sia facile per le persone accedere all'IP

Obiettivo	Criterio
Costi effettivi	Il costo dello storing dei dati non deve eccedere rispetto al budget a disposizione, gli eventuali errori nello storing non devono causare mancati guadagni
Sicurezza	Leggi sulla privacy e integrità dei dati
Disponibilità	Dati disponibili solo agli addetti dell'ufficio commerciale e al cliente cui le informazioni si riferiscono (previa autenticazione)
Tempo di accesso	L'ordine deve essere gestito entro una settimana
Data source	Oders DB

Tabella 3.2: Criteri di valutazione dell'accessibilità per l'IP1

Interoperabilità:

Per conoscere quanto sia facile capire il significato di un IP è opportuno definire un formato standard per ogni IP critico.

Obiettivo	Criterio		
Formato	Cliente	Data	
		Prodotto	Prezzo
	StatoOrdine	Tot	

Tabella 3.3: Criteri di valutazione dell'interpretabilità dell'IP1

Usabilità:

Obiettivo	**Criterio**
Dichiarazione di intenti	L'ordine viene usato per addebito fatture invio merci

Tabella 3.4: Criteri di valutazione dell'usabilità dell'IP1

Credibilità:

Con questa metrica si valuta il grado di affidabilità richiesto.

Obiettivo	**Criterio**
Attualità	L' attualità evita mancati invii o doppie spedizioni
Accuratezza	Gli errori relativi al cliente dovrebbero essere meno del 1%
Esistenza	Non dovrebbero esistere ordini duplicati
Completezza	100% degli ordini dovrebbe essere completo

Tabella 3.5: Criteri di valutazione della credibilità per l'IP1

Requisiti di qualità per l'IP2: Scheda cliente

Accessibilità:

Obiettivo	Criterio
Costi effettivi	Il costo dello storing dei dati non deve eccedere rispetto al budget a disposizione
Sicurezza	Leggi sulla privacy
Disponibilità	Dati disponibili solo agli addetti dell'ufficio commerciale e al cliente cui le informazioni si riferiscono, previa autenticazione
Tempo di accesso	L'invio del catalogo è previsto ogni 6 mesi
Data source	CustomersDB

Tabella 3.6: Criteri di valutazione dell'accessibilità di IP2

Interoperabilità:

Obiettivo	Criterio
Formato	Nome, Cognome Via Città, Stato, CAP

Tabella 3.7: Criteri di valutazione dell'interpretabilità dell'IP2

Usabilità:

Obiettivo	Criterio
Dichiarazione di intenti	La scheda viene utilizzata per l'addebito delle fatture e l'invio delle merci e del catalogo

Tabella 3.8: Criteri di valutazione dell'usabilità dell'IP2

Credibilità:

Obiettivo	Criterio
Attualità	Almeno il 97% delle schede dovrebbe essere aggiornate al valore attuale
Accuratezza	Errori di digitazione nel nome e nell'indirizzo dovrebbero accadere in meno del 1% dei casi
Esistenza	Dovrebbero non esistere schede duplicate
Completezza	100% delle schede dovrebbe essere completo

Tabella 3.9: Criteri di valutazione della credibilità dell'IP2

Fase 4: Costruzione dell'IP-MAP e del repository dei metadati

Dopo aver individuato l'IP ed i suoi requisiti di qualità si può procedere alla rappresentazione IP-MAP secondo gli 8 tipi di blocchi descritti nel capitolo precedente.

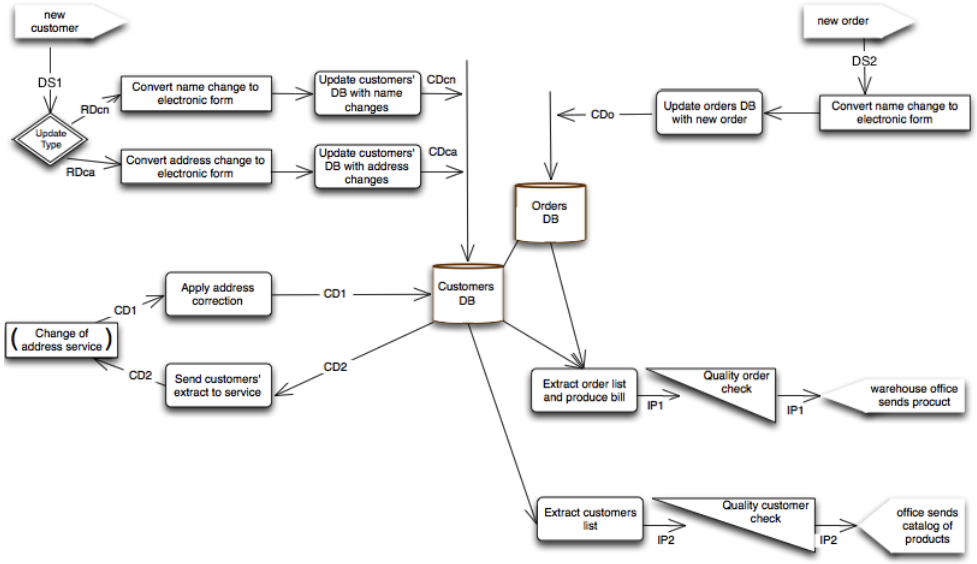

Figura 3.12: Rappresentazione di un processo di gestione ordine realizzato tramite una mappatura IP

Il processo di gestione degli ordini è stato modellato in modo tale che i prodotti siano sempre disponibili per semplificare il caso e concentrare l'attenzione sulla qualità di questi due IP. Quindi non è sono stati rappresentati nella mappatura IP-MAP il database del magazzino e le operazioni di approvvigionamento dei prodotti.

Blocchi

Qui di seguito elenchiamo, suddivisi per categoria, tutti i blocchi individuati durante la mappatura del processo, specificando per ciascuno di essi:

- Nome.
- Ruolo.
- Dove si trovano.
- Descrizione.
- Componenti.
- Sistemi su cui si basano.
- Problematiche relative alla qualità.

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
New customer (DS1)	Cliente	Dipende dal cliente	Il cliente segnala che i suoi dati sono variati	Nome, Indirizzo in Free Format	E-mail o telefono	Typos
New order (DS2)	Cliente	Dipende dal cliente	Il cliente effettua un nuovo ordine	Selezione dei prodotti	Sito web dell'azienda	

Tabella 3.10: Definizione dei Source block del processo in analisi

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Send Product (CB1)	Addetto magazzino	Via Rossi	Il responsabile invia la merce	Le bolle/fatture vengono usate per inviare le merci	Operazione manuale	Scambio merci
Send catalog of products (CB2)	Impiegato reparto commerciale	Via Verdi	La segreteria invia un catalogo ad ogni cliente	Le schede cliente vengono usare per inviare i cataloghi	Sito web dell'azienda	

Tabella 3.11: Definizione dei Customer block del processo in analisi

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Quality orders check (Q1)	Magazzino	Via Rossi	L'addetto controlla la bolla e l'ordine prima dell'invio	Input: IP1	Manuale o automatico	
Quality customers check (Q2)	Reparto commerciale	Via Verdi	La segretaria controlla la scheda prima di inviare il catalogo	Input IP2	Manuale o automatico	

Tabella 3.12: Definizione dei Quality block del processo in analisi

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Customers DB (STO1)	Registro delle schede clienti	Via Verdi	Conservazione dati cliente	Tutte le schede clienti	Oracle DB	Dati obsoleti
Orders' DB (STO2)	Registro ordini	Via Rossi	Conservazione degli ordini	Tutti gli ordini inevasi	Oracle DB	Eliminare i dati ogni 10 anni

Tabella 3.13: Definizione dei Storage block del processo in analisi

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Update type (D1)	Reparto commerciale	Via Verdi	La segretaria seleziona quale dato aggiornare	Input: RD1	LAN aziendale e scheda cartacea	

Tabella 3.14: Definizione dei Decision block del processo in analisi

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Convert name change to elctronic form (SB1)	Segreteria del reparto commerciale	Via Verdi	L'adetto registra la richiesta di aggiornamento in un file	Input: RDcn	Applicazione Visual Basic	Typos
Convert name change to elctronic form (SB2)	Segreteria del reparto commerciale	Via Verdi	L'addetto registra la richiesta di aggiornamento in un file	Input: RDca	Applicazione Visual Basic	Typos

Tabella 3.15: Definizione dei Information System Boundary block del processo in analisi

Name-Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Change of address service (SBS1)	Reparto esterno	Via Neri	Il servizio Address Service fa un confronto con i proprio DB	Input: CD2	Sw ad hoc	

Tabella 3.16: Definizione del blocco risultante dalla combinazione di Information System block e Business Boundary

Name - Type	Dept - Role	Location	Business Process	Composed Of	Base System	Quality Issues
Apply adrress correction (P1)	Reparto commerciale	Via Verdi	Update degli indirizzi	Input: CD1	Sw per la ricezione	Solo il 98% dei cambiamenti viene trovato
Send customers extract to service (P2)	Reparto commerciale	Via Verdi	Preparazione dei file da analizzare (CD2)	Inpurt: dati estratti da STO1	Sw per l'estrazione e la notifica	
Update customers' DB with name changes (P3n)	Reparto commerciale	Via Verdi	La segretaria carica CDcn in STO1	Input: CDcn	MySQL + Visual Basic	Solo il 10% dei clienti notifica le modifiche
Update customers' DB with address changes (P3a)	Reparto commerciale	Via Verdi	La segretaria carica CDca in STO1	Input: CDca	MySQL + Visual Basic	Solo il 10% dei clienti notifica le modifiche
Update orders DB with new order (P4)	Magazzino	Via Rossi	La segretaria carica CDo in STO2	Input: CDo	MySQL + Visual Basic	Solo il 10% dei clienti notifica le modifiche
Extract orders list and produce bill (P5)	Reparto commerciale	Via Verdi	La segretaria cerca i dati sugli ordini	Input: dati da STO2	MySQL	
Extract customers list (P6)	Reparto commerciale	Via Verdi	La segretaria cerca i dati dei clienti	Input: dati da STO1	MySQL	

Tabella 3.17: Definizione dei Processing block del processo in analisi

Dati

Per ogni dato scabato durante il processo abbiamo specificato il nome, una breve descrizione e le problematiche di qualità che lo riguardano.

Name - Type	Data Elements	Quality Issues
RD1	Richiesta di aggiornamento in Free Format	Typos
RDcn	Informazione per aggiornare il nome in Free Format	
RDca	Informazione per aggiornare il indirizzo in Free Format	
RD2	Richiesta di ordine	
CD2	Dati estratti in formato standard	Solo il 98% dei cambiamenti viene trovato
CD1	Dati aggiornati in formato standard	Solo il 98% dei cambiamenti viene trovato
CDcn	Informazione per aggiornare il nome in formato standard	Solo il 10% dei clienti notifica i cambiamenti
CDca	Informazione per aggiornare il indirizzo in formato standard	Solo il 10% dei clienti notifica i cambiamenti
CDo	Ordine in formato std	
IP1	Ordine in formato ad hoc aziendale	
IP2	Scheda cliente in formato ad hoc aziendale	

Tabella 3.18: Definizione dei flussi di dati del processo in esame

Data Manufacturing Analysis Matrix Questa tabella riassume le relazioni tra blocchi e flussi di dati in termini di input e output.

	RD1	RDcn	RDca	RD2	CD2	CD1	CDcn	CDca	CDo	IP1	IP2
DS1	x										
DS2				x							
CB1											
CB2											
Q1										x	
Q2											x
D1	x										
SB1		x									
SB2			x								
BSB1					x	x					
STO1					x	x	-	-		x	x
STO2									-		x
P1						x					
P2					x						
P3n							x				
P3a								x			
P4									x		
P5										x	
P6											x

Tabella 3.19: Riassunto dei legami tra i blocchi e i flussi di dati

Le X all'interno della matrice possono essere rimpiazzate con vettori contenenti parametri di qualità, costi o tempi di disponibilità dei dati [29].

Fase 5: Valutazione e miglioramento della qualità

Per completare l'analisi di qualità dei due IP in questione è necessario costruire le loro Information Product Control Matrix, da cui è anche possibile ricavare i costi e le frequenze degli IP.

Information Product Control Matrix

Source Data Error per IP1:

IP1	Dupicate data created in P5	Obsolete data in STO1	Obsolete data in STO2	Typos in DS1	Mistake in DS2	Missing data in DS1	Missing data in DS2	Bad Data from DS1	Bad Data from DS2
Estimated Frequency of Error	1%	3%	2%	5%	0%	10%	5%	6%	6%
Estimated Cost of Error per IP	€ 40,00	€ 40,00	€ 40,00	€ 2,00	€ 10,00	€ 40,00	€ 10,00	€ 2,00	€ 2,00
Reliability Rating of IP-MAP constructs									
Q2	99,00%	98,00%	99,00%	85,00%	//	98,00%	97,00%	88,00%	88,00%
Overall Quality	0,01%	0,06%	0,02%	0,75%	//	0,20%	0,15%	0,72%	0,72%

Tabella 3.20: Tabella riassuntiva delle fonti di errori per il primo IP

hp: costo invio catalogo =5€ costo re-invio = 2€
Numero di IP1 corretti = $\prod_{i=1}^{n}(1 - ErrorRate_i) = 99,94\%$
Costo stimato per 1000 IP1= $\sum_{i=1}^{n}(Cost_i * 1000 * ErrorRate_i) = 174,80$€

Matrice IP-MAP di controllo per IP1:

	Date	Status	Customer
Error Frequency	0,00%	1,33%	0,67%
Avg. Cost of Error per IP1	€ 0,00	€ 5,00	€ 5,00
IP-MAP constructs that control for that data error			
New Order	10,00%	10,00%	10,00%
Status changes by commercial department	99,00%	99,00%	
Receive updates from address service			98,00%

Tabella 3.21: Matrice di controllo del primo IP

Source Data Error per IP2 :

IP2	Dupicate data created in P6	Obsolete data in STO1	Typos in DS1	Missing data in DS1	Bad Data from DS1	Wrong format used in P6
Estimated Frequency of Error	2%	3%	5%	10%	6%	4%
Estimated Cost of Error per IP	€ 5,00	€ 5,00	€ 2,00	€ 5,00	€ 2,00	€ 2,00
Reliability Rating of IP-MAP constructs						
Q1	98,00%	98,00%	85,00%	98,00%	88,00%	97,00%
Overall Quality	0,04%	0,06%	0,75%	0,20%	0,72%	0,12%

Tabella 3.22: Tabella riassuntiva delle fonti di errori di IP2

hp: acquisto medio = 40€ , costo o mancato guadagno medio per ogni pezzo con inviato = 10€ Numero di IP2 corretti = $\prod_{i=1}^{n}(1 - ErrorRate_i) = 99,94\%$
Costo stimato per 1000 IP2= $\sum_{i=1}^{n}(Cost_i * 1000 * ErrorRate_i) = 46,80$€

Matrice IP-MAP di controllo per IP2:

			Address			
	Name	**Surname**	**Street**	**City**	**Nation**	**Zip code**
Error Frequency	0,50%	0,50%	1,33%	0,67%	0,33%	1,00%
Avg. Cost of Error per IP	€ 1,00	€ 1,00	€ 5,00	€ 5,00	€ 5,00	€ 5,00
IP-MAP constructs that control for that data error						
New Custromer	10,00%	10,00%	10,00%	10,00%	10,00%	10,00%
Receive updates from address service	98,00%	98,00%	98,00%	98,00%	98,00%	98,00%

Tabella 3.23: Matrice di controllo del secondo IP

Reliability di IP2:

	Time Period											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Name Changes												
% of customers who change name each month	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
% of name changes self-reported & corrected each month	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05
% of name changes caught by Address Service & corrected per month	0,00	0,00	1,35	0,00	0,00	1,35	0,00	0,00	1,35	0,00	0,00	1,35
Cumulative % of forms with undetected name changes each month	0,93	1,38	0,48	0,93	1,38	0,48	0,93	1,38	0,48	0,93	1,38	0,48

Tabella 3.24: Reliability del secondo IP per i cambiameni di nominativo

hp per la reliability del secondo IP in meritoa ai cambiameni di nominativo: il periodo 1 inizia con in Error Rate del 0,48%

hp per la reliability del secondo IP in meritoa ai cambiameni di indirizzo: il periodo 1 inizia con in Error Rate del 1,27%

	Time Period											
	1	2	3	4	5	6	7	8	9	10	11	12
Address Changes												
% of customers who change address each month	1,33	1,33	1,33	1,33	1,33	1,33	1,33	1,33	1,33	1,33	1,33	1,33
% of address changes self-reported & corrected each month	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13
% of address changes caught by Address Service & corrected per month	0,00	0,00	3,60	0,00	0,00	3,60	0,00	0,00	3,60	0,00	0,00	3,60
Cumulative % of forms with undetected name changes each month	2,47	3,67	1,27	2,47	3,67	1,27	2,47	3,67	1,27	2,47	3,67	1,27

Tabella 3.25: Reliability del secondo IP per i cambiameni di indirizzo

	Time Period											
	1	2	3	4	5	6	7	8	9	10	11	12
Tot												
Cumulative % of forms with undetected name changes each month	0,93	1,38	0,48	0,93	1,38	0,48	0,93	1,38	0,48	0,93	1,38	0,48
Cumulative % of forms with undetected name changes each month	2,47	3,67	1,27	2,47	3,67	1,27	2,47	3,67	1,27	2,47	3,67	1,27
Adjustment for 60% Dual Name & Address Changes	-0,56	-0,83	-0,29	-0,56	-0,83	-0,29	-0,56	-0,83	-0,29	-0,56	-0,83	-0,29
Total % of forms with undetected errors each month	2,84	4,22	1,46	2,84	4,22	1,46	2,84	4,22	1,46	2,84	4,22	1,46

Tabella 3.26: Reliability totale del secondo IP

Percentuali di schede con problemi di qualità dei dati nei prossimi 12 periodi:

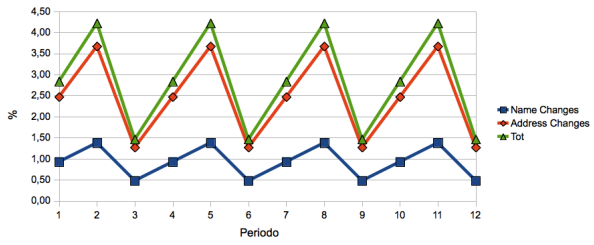

Figura 3.13: Grafico delle percentuali degli IP2 di scarsa qualità

Capitolo 4

Configurazioni di monitoraggio

4.1 Valutazioni circa l'inserimento dei blocchi di qualità

Uno dei soggetti principali di questo lavoro è il Quality Block cioè il blocco posizionato all'interno del processo per valutarne le prestazioni in termini di qualità dei dati. Per poterlo disegnare è necessario conoscere sia le dimensioni che interessano al manager del processo che la struttura delle basi di dati e del processo stesso. Le principali domande che ci siamo posti nel corso di questo lavoro, legate all'utilizzo dei blocchi, sono:

- Come implementare i blocchi.
- Dove posizionarli. Prima o dopo le attività.
- Quanti utilizzarne. Uno per ogni attività o uno per tutto il process flow.

4.2 Implementazione dei blocchi

La risposta alla prima domanda si può soddisfare studiando, come abbiamo fatto nello stato dell'arte, gli algoritmi di valutazione presenti in letteratura e proponendo alcune modifiche o integrazioni, che abbiamo riportato nella descrizione più approfondita dei singoli blocchi. Rispondendo a questo primo quesito abbiamo ritenuto opportuno applicare due scomposizioni:

- In base alle dimensioni di qualità dei dati di interesse.
- Secondo le basi di dati da verificare.

Tali scomposizioni ci permettono di analizzare le singole attività oltra al processo intero.

4.2.1 I sotto-blocchi

Prima di studiare i possibili scenari delle strategie alternative che descriveremo in questo capitolo abbiamo ritenuto vantaggioso dividere il Quality Block in quattro sotto-blocchi, uno per ogni dimensione da monitorare.

Oltre a scomporre il blocco in sottoblocchi secondo le dimensioni, i blocchi si possono anche dedicare all'analisi delle diverse fonti di dati, in modo da combinare le diverse configurazioni che vedremo sia secondo le dimensioni che per i diversi database in gioco. I quattro sotto-blocchi da noi utilizzati per i controlli delle dimensioni sono i seguenti:

1. **Accuratezza:** Questo blocco permette di effettuare verifiche di accuratezza. Abbiamo scelto di dividerlo in due parti.
    - **Valutazione dell'accuratezza di stringhe** che serve per calcolare l'accuratezza dei dati che verranno utilizzati durante il processo secondo uno degli algoritmi presenti in letteratura, quello della weak accuracy. Per effettuare questa verifica i valori delle tabelle utilizzate sono stati confrontati con altri valori contenuti in tabelle di riferimento, appositamente create per contenere i valori esatti.
    - **Valutazione dell'accuratezza di valori numerici** che non segue un algoritmo presente in letteratura ma cerca di sfruttare alcune nozioni della teoria della propagazione dell'errore studiate nel calcolo numerico, questo perchè gli errori e la scarsa qualità non dipendono solo dai dati presenti nei database ma anche dalle operazioni con cui tali dati vengono elaborati (e quindi dipendono dai metodi all'interno delle attività). La teoria della propagazione dell'errore stabilisce che l'errore relativo sui dati si propaga in maniera diversa al cambiare

delle operazioni sui dati e fornisce tecniche per calcolarlo. Quindi conoscendo l'errore sui dati all'ingresso e la tipologia di operazioni svolte dell'attività del processo si può conoscere l'errore sull'output. Questo ci aiuterà a rispondere anche alla seconda domanda poichè ponendo il monitor prima di un'attività si è comunque in grado di valutare l'errore sull'output.

In relazione ad una tra le molteplici definizioni di accuratezza esistenti possiamo considerare l'accuratezza come la distanza tra il dato ed il valore corretto del dato. Questa definizione coincide con la definizione matematica di errore assoluto che si esprime nel seguente modo:

$$e_x = \bar{x} - x$$

dove $\bar{x}$ è il valore misurato del dato e x quello "vero". Da questo si può definire l'errore relativo:

$$\xi_x = \frac{\bar{x} - x}{x}$$

da cui seguono

$$\bar{x} = x(1 + \xi_x)$$

e

$$f(\bar{x}) = f(x(1 + \xi_x)) \simeq f(x) + f'(x)x\xi_x$$

quindi

$$\xi_f = \frac{f(\bar{x}) - f(x)}{f(x)} \simeq \frac{f'(x)x\xi_x}{f(x)}$$

Per le funzioni in più incognite risulta

$$f(\bar{x}, \bar{y}) = f(x(1 + \xi_x), y(1 + \xi_y)) \simeq f(x, y) + f_x(x, y)x\xi_x + f_y(x, y)y\xi_y$$

e

$$\xi_f = \frac{f(\bar{x}, \bar{y}) - f(x, y)}{f(x, y)} \simeq \frac{f_x(x, y)x\xi_x + f_y(x, y)y\xi_y}{f(x, y)}$$

Dallo studio della propagazione delle errore sulle funzioni più utilizzate possiamo ricavare le seguenti formule [18].

– Media e somma tra due varibili:

$$\xi_{x+y} = \frac{x}{x+y}\xi_x + \frac{y}{x+y}\xi_y$$

$$\xi_{\frac{x+y}{2}} = \frac{x}{x+y}\xi_x + \frac{y}{x+y}\xi_y$$

Sommatoria:

$$\xi_{\sum_{i=1}^{n} x_i} = \sum_{i=1}^{n} \frac{x_i}{\sum_{i=1}^{n} x_i}\xi_{x_i}$$

– Sottrazione tra due valori:

$$\xi_{x-y} = \frac{x}{x+y}\xi_x - \frac{y}{x+y}\xi_y$$

– Moltiplicazione tra due valori:

$$\xi_{x*y} = \xi_x + \xi_y$$

Produttoria:

$$\xi_{\prod_{i=1}^{n} x_i} = \sum_{i=1}^{n} \xi_{x_i}$$

– Divisione tra due valori:

$$\xi_{x/y} = \xi_x - \xi_y$$

Queste formule all'interno del monitor valutano l'errore sul dato dopo l'esecuzione di un'attività.

2. **Completezza:** L'analisi che effettua questo blocco riguarda la ricerca dei valori nulli all'interno delle tabelle utilizzate dalle attività che vogliamo monitorare e restituisce una percentuale, caratterizzata dal rapporto tra dati non nulli e il totale, che identifica il grado di completezza del database in esame.

3. **Consistenza:** La consistenza è una dimensione di qualità fortemente dipendente dalle attività del processo. Per esempio, nel caso di studio in esame abbiamo applicato la verifica di questa dimensione al legame tra le

città e le nazioni cui appartengono e tra le città ed i relativi codice postale. Questo controllo avviene effettuando la ricerca sulla tabella che contiene le triplette corrette di città, codici e nazioni, per verificare che le coppie in esame appartengano alla stessa tupla. Per calcolare questa dimensione sono richieste ancora una volta tabelle aggiuntive, ripetto a quelle che usa il processo, che contengano i dati corretti delle corrispondenze utili per verificare la consistenza.

4. **Tempo:** Il controllo delle dimensioni relative al tempo è stato diviso in tre operazioni, una per ciascuna dimensione. Tutte queste dimensioni si possono calcolare a design time, sulla base di informazioni messe a disposizione dal parte del programmatore, perché descrivono il database, la stabilità dei suoi dati e le frequenze di aggiornamento.

La scomposizione del Quality Block ci permette non solo di utilizzare il componente necessario solo quando serve ma anche di combinare le tre diverse alternative in base alle necessità legate alle dimensioni di qualità e dei database da monitorare.

4.3 Strategie di monitoraggio

Questi blocchi sono stati concepiti come attività che effettuano check che andranno inserite all'interno del processo originario.
Tornando alle domande che ci siamo posti all'inizio di questo capitolo, è arrivato il momento di rispondere alle ultime due, quella riguardante il punto in cui posizionare i blocchi e quella circa il loro numero. Le semplificazioni sopra descritte aiutano a rispondere a questi quesiti perchè, come abbiamo visto, applicando ogni blocco alle attività che usano i dati che questo controlla e se un blocco effettua un check su una dimensione o un DB già monitorate è possibile, anzi utile, evitare di ripetere tale verifica posizionando i blocchi in maniera più efficace.

Quindi una volta analizzato il processo e costruiti i blocchi di qualità si possono identificare i punti più adatti per l'inserimento dei blocchi di verifica. Nel nostro percorso abbiamo suddiviso le configurazioni cercando prima di privilegiare la qualità e poi verificando di non gravare troppo sul tempo del processo.

Abbiamo così identificato tre alternative. Le semplificazioni descritte sopra consentono di evitare la ripetizione di parti del blocco se una specifica dimensione o un database sono già stati monitorati in un punto precendente del processo. Ad esempio se due attività in sequenza utilizzano lo stesso database, se la consistenza di questo è già stata controllata per la prima attività, si potrà evitare un'altra verifica per la seconda attività. Inoltre se un processo lavora su due database con probabilità di fault diverse è possibile combinare tra loro le diverse soluzioni. Abbiamo identificato tre diverse configurazioni, la prima è la più precisa per quanto riguarda il monitoraggio ma non tiene conto del tempo di esecuzione del processo, le altre due eseguono analisi su tutto il processo considerando il tempo ma hanno performance minori in termini di tempestività della notifice rispetto alla prima alternativa.

- **Prima configurazione: Verifica locale**
  Come prima cosa abbiamo cercato di disegnare un processo localmente ottimo dal punto di vista del monitoraggio della qualità. Considerando che le attività di monitoraggio solitamente si limitano a inviare segnali al manager del processo, sospendendolo, abbiamo ritenuto preferibile per le attività che effettuano aggiornamenti su database, che le verifiche avvengano prima dell'attività stessa, in modo tale da bloccare subito il processo nei casi di qualità insoddisfacente. Questo evita che si scrivano dati errati, obsoleti o di scarsa qualità ricavati da dati incorretti. Mentre, per quanto riguarda le attività di lettura da database, il controllo risulta più efficiente se eseguito in parallelo all'attività perchè il parallelismo consuma tempi minori e la scarsa qualità comporta la riesecuzione solo di quell'attività. Questo consente una facile e tempestiva localizzazione della scarsa qualità e della sua fonte, favorendo il successo delle azioni delle repair action che, come abbiamo visto nello studio dello stato dell'arte, è fortemente legato alla tempestività. Questo scenario è adatto per un'azienda con processi molto semplici in quanto aumenta notevolmente la complessità del process flow e dilata i tempi di esecuzione del processo. Purtroppo per la maggior parte delle aziende il tempo di risposta è un KPI molto importante che può influenzare negativamente il livello soddisfazione del cliente finale e diminuire il revenue.

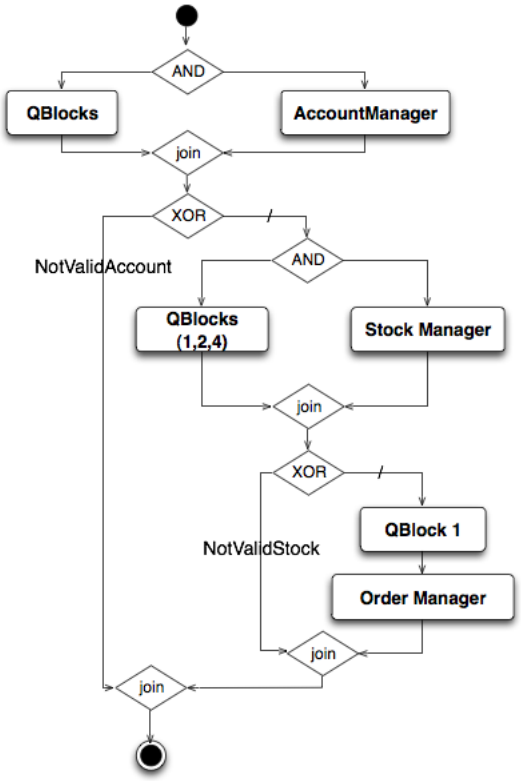

Figura 4.1: Inserimento dei blocchi secondo la prima modalità di monitoraggio

Per monitorare la qualità dei processi in tempi minori si potrebbe scegliere di rinunciare ad un'analisi "ravvicinata" ed adottare una soluzione che controlli il processo nel suo insieme. In questo caso si possono effettuare altre due scelte. Queste soluzioni diventano scelte obbligate nel caso in cui si trattino i processi come scatole nere, cioè se non si conoscono nel dettaglio le attività che li compongono o la loro sequenza.

- **Seconda configurazione: Verifica prelimiinare**
  La seconda alternativa proposta tiene conto dell'impatto dei Quality Block sul tempo di esecuzione del processo, ma vuole comunque evitare che sui database vengano inseriti dati di scarsa qualità. Quindi questa alternativa prevede l'inserimento del Quality Block prima che abbia inizio l'esecuzione del processo il cui inizio, in caso di anomalie, verrà ritardato fino a che non siano state apportare le opportune repair action. Questa soluzione è particolarmente adatta per quei processi di business che si trovano spesso a

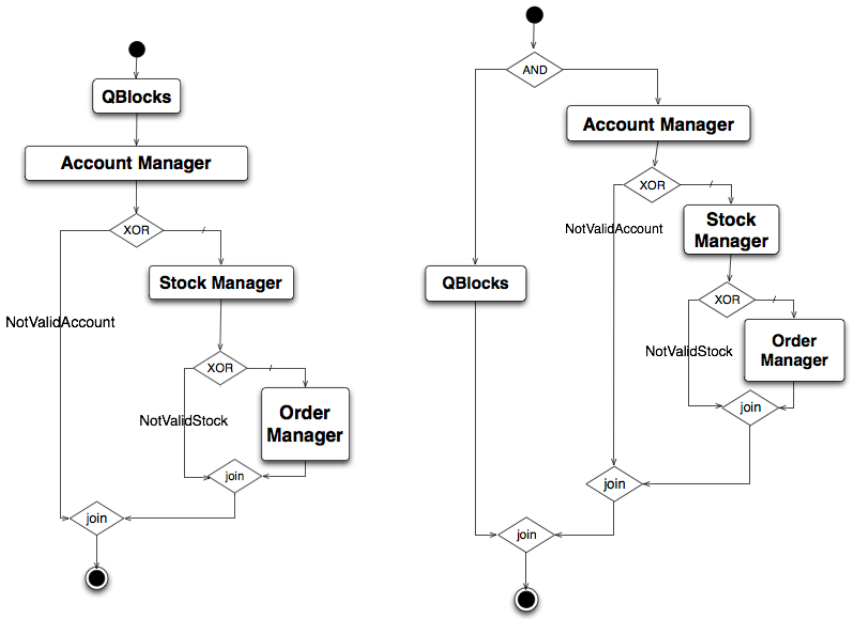

Figura 4.2: Configurazioni 2 e 3

confrontarsi con dati di scarsa qualità poichè la qualità viene monitorata e la sua mancanza viene segnalata prima dell'esecuzione. In questo modo si può rinviare il processo fino a che non venga eseguita un'opportuna riparazione.

- **Terza configurazione: Verifica in parallelo**
  L'ultima alternativa prevede che il controllo di qualità sia effettuato in parallelo al processo, riducendo i tempi di esecuzione. Questa soluzione notifica l'eventuale presenza di errori solo al termine dell'esecuzione del processo, rischiando di compromettere il successo delle azioni di repair, ma si rivela la soluzione più adatta se i dati rispettano i vincoli di qualità imposti. Ricordiamo che il parallelismo risulta molto vantaggioso se le diverse attività del processo vengono eseguita da diverse unità. Questo va tenuto in considerazione quando si sceglie la soluzione da adottare.

La prima configurazione è quindi da preferire quando il tempo di esecuzione del processo non costituisce un fattore critico, mentre le ultime due, che considerano anche il tempo, sono da preferire se questo è un aspetto importante del servizio che si vuole offrire. Un altro cirterio da tenere in considerazione per scegliere tra le ultime due alternative è quale sia il livello di qualità atteso del

processo. Per sapere quale delle ultime due soluzione preferire bisogna chiedersi quale sia la probabilità di scarsa qualità e confrontare i tempi di esecuzione e riparazione al variare di tale probabilità. Trovando il punto di incontro delle due rette che rappresentano i tempi rispetto alla probabilità di guasto possiamo decidere quale alternativa preferire.

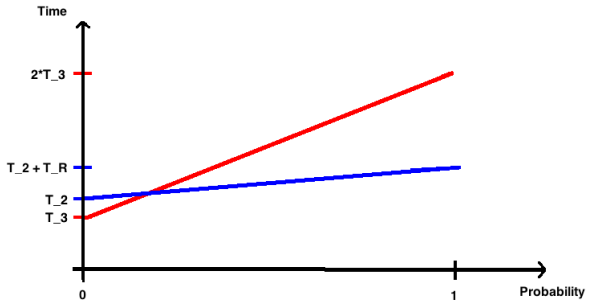

Figura 4.3: Possibile andamento dei tempi di esecuzione delle alternative 2 e 3

In questo grafico sono mostrati i possibili andamenti delle alternative 2 e 3, la seconda alternativa è indicata in rosso con una durata pari a T_2 mentre la terza è in blu e dura T_3, infine con T_R indichiamo la durata delle azioni di repair.

Si potrebbero anche combianare le tre alternative in modo da monitorare i diversi database in modo diverso.

La figura mostra la combinazione tra le altrenative 2 e 3, preferibile nel caso in cui due database abbiano una probabilità d'errore molto diverse, che portino a preferire configurazioni di verifica diverse per i due database, allora il controllo di quello con probabilità di scarsa qualità avverà prima del processo come stabilito per la seconda alternative mentre per il database per cui ci aspettiamo che i dati soddisfino i vincoli di qualità il check avverrà in parallelo al processo con le modalità della terza configurazione. Un'altra soluzione potrebbe essere eseguire le verifiche di alcune dimensioni che interessano l'intero processo come quelle legate al tempo in parallelo adottando invece una strategia più precisa per le altre dimensioni, combinando la terza e la seconda configurazione.

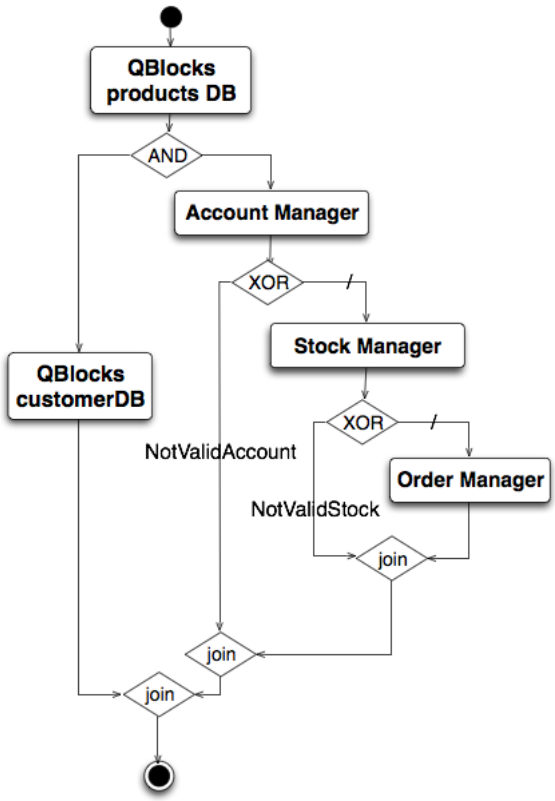

Figura 4.4: Esempio di combinazione delle alternative 2 e 3

4.4 Risultati dell'analisi e confronto delle alternative

Per poter constatare la veridicità delle conclusioni al capitolo precedente è stato necessario realizzare il processo e i monitor di qualità ed anche misurare le prestazioni ottenute.

Per prima cosa abbiamo realizzato il processo del caso di studio e successivamente abbiamo disegnato i blocchi di qualità da posizionare secondo quanto stabilito nella descrizione delle iniziative.

4.4.1 Implementazione del processo

Come abbiamo visto il processo si può semplificare suddividendolo in tre fasi o attività. Per ciascuna attività abbiamo realizzato un Web Service java contenente

diverse operazioni. Le operazioni sono i metodi con cui i servizi forniscono il loro contributo al processo.

Prima di implementare il processo in java abbiamo costruito il database in MySQL, realizzando le tabelle indicate nel diagramma ER del capitolo precedente.

La prima fase è quella di login, svolta dal WS da noi chiamato AccountManager. Questo servizio richiede all'utente di inserire il proprio identificativo e controlla, tramite una query sql, che tale id sia presente nel database dei clienti registrati. Se l'identificazione non da riscontro positivo il processo termina. Se invece l'utente è registrato nel nostro database gli sarà permesso, tramite il servizio StockManager, di specificare quale prodotto vuole acquistare (tale servizio, nel caso di studio, da sempre riscontro positivo). Una volta completata la ricerca viene richiesta la quantità di prodotto che si vuole acquistare tramite le operazione all'interno del WS OrderManager, che, sulla base nelle informazioni raccolte, aggiorna il database degli ordini.

4.4.2 Implementazione dei monitor di qualità

Durante la fase di implementazione abbiamo realizzato i sottoblocchi descritti nel capitolo 3. Per far questo abbiamo implementato diverse operazioni all'interno di quattro Web Service.

Il primo Web Service realizzato si occupa di valutare l'accuratezza. Come abbiamo visto durante la descrizione teorica questa dimensione viene spesso definita come distanza tra il valore contenuto nel DB e quello reale. Per la nostra simulazione i valori reali sono stati salvati in un DB con la stessa struttura di quello utilizzato dal processo, abbiamo inoltre creato appositamente degli errori nel database del processo in modo che, i dati processati si discostassero da quelli di riferimento, e per poter calcolar la qualità.
Per verificare la consistenza è necessario conoscere altre informazioni oltre a quelle presenti nel DB dell'azienda. Nel caso studiato la consistenza riguarda le coppie che vedono abbinate le città con i codici postali corrispondenti e le città con le nazioni cui appartengono. Per valutare questa dimensione abbiamo costruito una tabella che contenesse una tupla per ogni città che ne specificasse il nome,

la nazione e lo zipcode. Per quanto concerne la completezza la verifica è stata effettuata tramite la ricerca delle occorrenze del valore null nella base di dati di interesse. Infine per la valutazione delle dimensioni time-related, come abbiamo visto, si calcolano i valori dell'attualità, della volatilità e della tempestività sulla base di informazioni che descrivono il DB note al suo disegnatore.

4.4.3 Simulazioni

Abbiamo visto che i blocchi di qualità possono essere inseriti nel process flow in punti diversi e a seconda del modo in cui questi sono stati inseriti diverso è l'impatto che possono avere sul processo. Questi diversi modi caratterizzano le tre diverse alternative che ora applicheremo al caso di studio.

Prima alternativa: Come prima cosa abbiamo applicato la prima soluzione al processo in esame. Per applicare la prima alternativa è necessario conoscere cosa fa ciascuna attività in modo da posizionare i blocchi per il monitoraggio di determinate dimensioni per determinati dati in corrispondenza di attività che usano questi dati e potrebbero risentire della non conformità dei requisiti.

Per processi semplici come il nostro la prima alternativa non risulta particolarmente complessa nè dispendiosa sotto il profilo temporale. Confermando che la scelta di questa alternativa è da preferirsi in situazioni di processi snelli quando si vuole porre un forte accento sulla qualità e sulla sua tempestiva verifica.

Seconda alternativa: La seconda alternativa prevede l'esecuzione in parallelo di tutti i monitor di qualità necessari prima del processo. Il controllo iniziale evita la scrittura di dati insoddisfacenti risultanti dall'elaborazione di dati di scarsa qualità. All'inizio del nostro studio avevamo ipotizzato di posizionare il controllo alla fine del processo ma questo avrebbe portato ad un risultato equivalente alla terza alternativa in termini di tempestività della rilevazione, introducendo un ritardo sul tempo di esecuzione del processo.

Terza alternativa: Infine i quattro sotto-blocchi vengono eseguiti in parallelo tra loro e in parallelo al processo per studiare la terza alternativa. Que-

st'ultima alternativa permette di conosce la qualità dei dati alla fine dell'esecuzione, ammortizzando il tempo necessario per le verifiche.

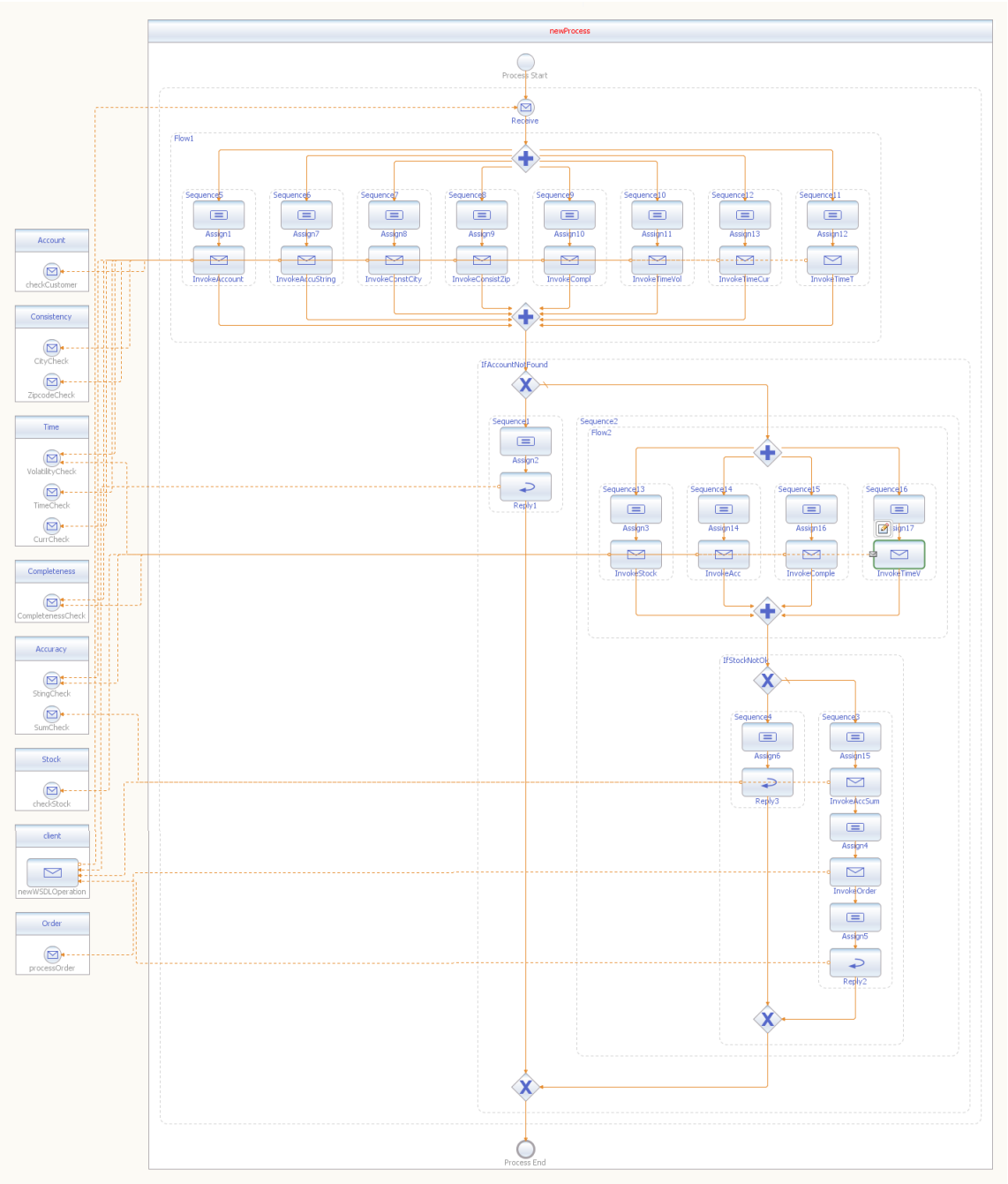

Figura 4.5: Realizzazione della prima alternativa

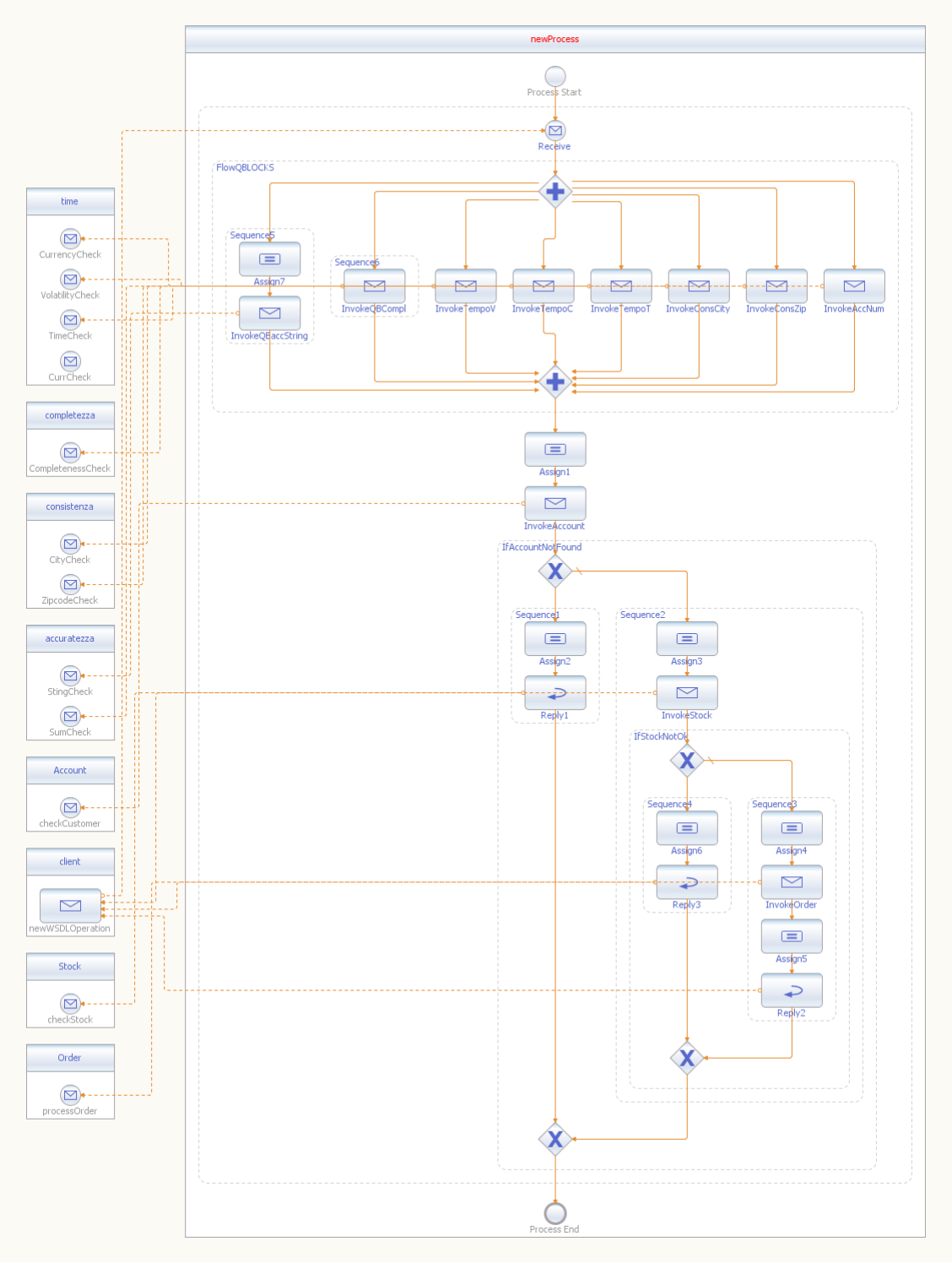

Figura 4.6: Realizzazione della seconda alternativa

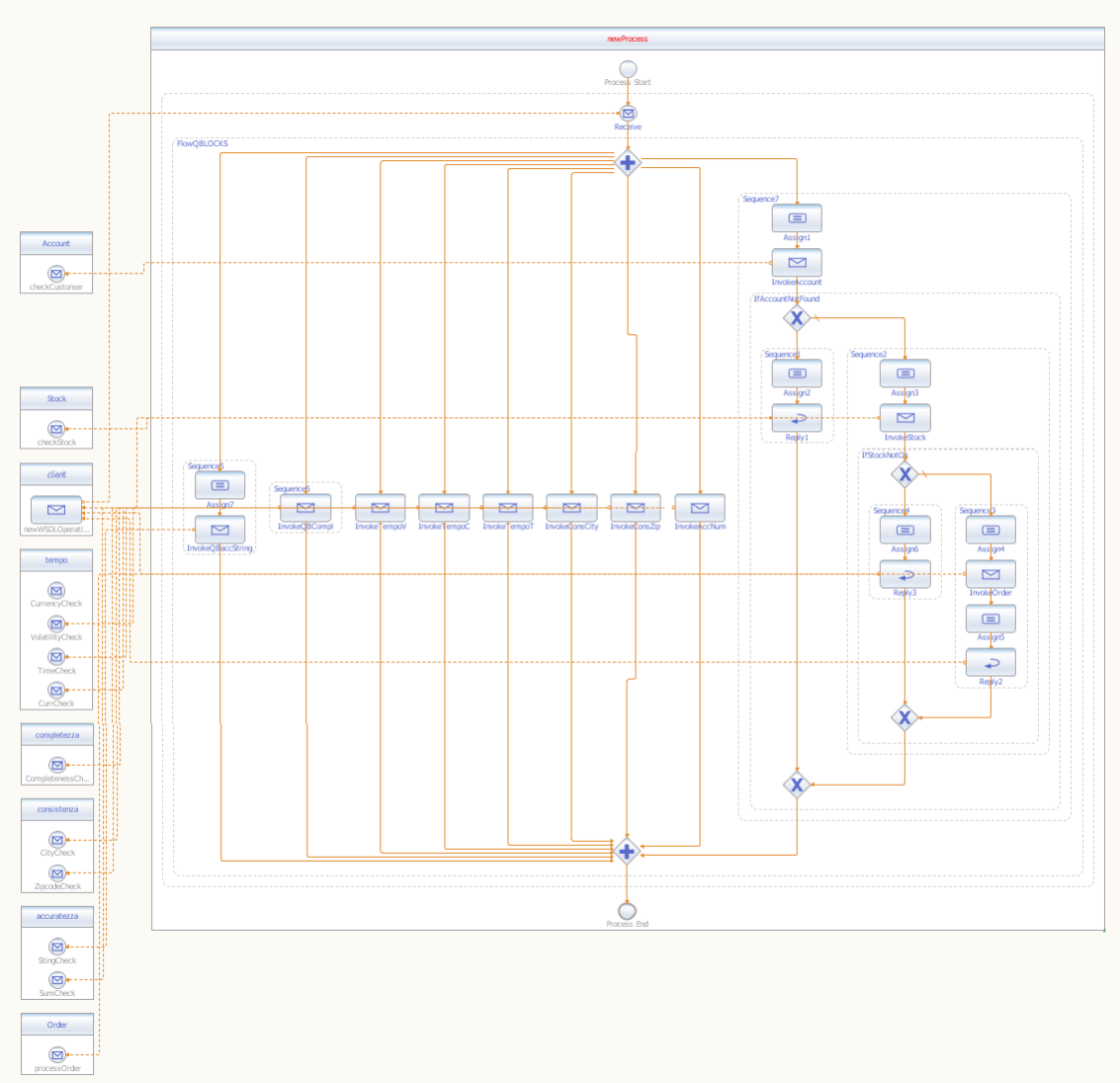

Figura 4.7: Realizzazione della terza alternativa

I risultati più interessanti che vi vogliamo mostrare sono quelli che riguardano le altre due alternative, cioè le alternative che monitorano la qualità, non dimenticando che anche il tempo di esecuzione è un fattore molto critico per le imprese moderne. Vogliamo quindi considerare anche il tempo oltre alla qualità e ci chiediamo quale delle ultime due soluzione preferire all'altra. Per farlo calcoliamo la probabilità di scarsa qualità e confrontiamo i tempi di esecuzione e riparazione al variare di tale probabilità. La terza alternativa avrà un tempo complessivo di esecuzione, in caso di scarsa qualità, pari al doppio del tempo di esecuzione del processo monitorato, dovuto alla riesecuzione del processo, mentre la seconda in caso di errore, avvenuta la correzione, riprenderà l'esecuzione del processo con un ritardo pari al tempo delle azioni di repair.

Ipotizzando che il tempo di riparazione sia nullo le due rette dei tempi di esecuzione sono quelle mostrare nel grafico sottostante.

Da questo grafico si ricava una probabilità di soglia pari al 12,53% , risultato dell'intersezione tra le rette t=0,7612 per la seconda alternativa e la retta t=0,67644(x+1) per la terza. Quindi, nel nostro caso, se vogliamo considerare anche il tempo come fattore di scelta e optare per le ultime due alternative, sceglieremo la seconda quando la probabilità di avere scarsa qualità è superiore al 12,53 % mentre preferiremo la terza in caso contrario.

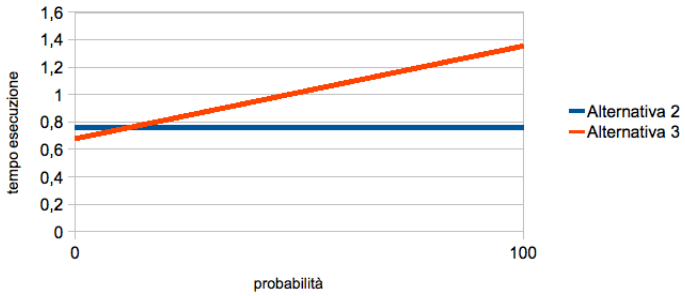

Figura 4.8: Grafico rette ricavate dai dati sperimentali delle alternative 2 e 3

Capitolo 5

Il sistema per la valutazione della Qualità

5.1 Ambiente di sviluppo

Per lo sviluppo del progetto software è stata utilizzata una piattaforma Open Source: la piattaforma NetBeans, Open Source dal mese di giugno 2000 grazie a Sun Microsystems che è rimasto lo sponsor del progetto. La piattaforma NetBeans permette l'implementazione dei Web Service tramite il linguaggio java ed include strumenti visuali per la composizione di applicazioni SOA, fornendo agli sviluppatori Orchestration Designer, UML modeling e tool per XML. Il progetto NetBeans fornisce un Sun Java System Application Server PE 9 (Glassfish) e può essere integrato con l'ulitizzo di un Enterprise Pack. Questo pack contiene un Orchestration Designer che fornisce un ambiente completo per orchestrare in modo intuitivo e veloce processi BPEL.

Per la realizzazione della base di dati abbiamo utilizzato MySQL, prodotto Open Source della società virtuale MySQL AB. MySQL è classificabile come un "database relazionale" permettendo la conservazione dei dati in tabelle separate piuttosto che in una unica grande entità. Questo, oltre ad aggiungere flessibilità e velocità nell'accesso ai dati, permette una più efficace modellazione delle basi di dati. Il sistema si presenta con una struttura client/server, dove il server SQL, scritto in C e C++, utilizza una architettura multithread e offre una completa raccolta di API utilizzabili sia tramite i client forniti nella distribuzione sia da

applicazioni scritte in una notevole varietà di linguaggi: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby e TCL.

L'implementazione e le simulazioni sono state eseguite su una macchina con processore Intel Core 2 Duo ed una memoria ram di 2,5 GB.

5.2 Strumenti

Abbiamo simulato il processo di gestione degli ordini utilizzando un'architettura orientata ai servizi, in particolare abbiamo realizzato le diverse attività che compongono il process flow tramite l'implementazione di Web Service, utilizzando il linguaggio java, ad abbiamo assemblato il processo come un unico flusso utilizzando un linguaggio di orchestrazione.

Per una descrizione più dettagliata degli strumenti utilizzati rimandiamo alla lettura dei seguenti paragrafi, in cui sono state analizzate le tecnologie e le loro potenzialità.

5.2.1 Web Service

La tendenza che si osserva in questi ultimi anni è quella di trasformare sempre più le applicazioni in servizi disponibili online. Questa è una pratica che si è dimostrata molto utile, sia per quanto riguarda i servizi interni ma soprattutto per quelli esterni rivolti ai clienti. Le aziende offrono e utilizzano servizi. Per fa ciò è necessario che si accordino su un linguaggio comune di descrizione dei servizi in modo tale da poter ricavare cosa un sistema mette a disposizione [28]. Un Web Service è un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori su una stessa rete. La caratteristica fondamentale di un Web Service è quella di offrire un'interfaccia software descritta in linguaggio WSDL (Web Services Description Language) per mezzo del quale cui altri sistemi possano interagire con il Web Service stesso, attivando le operazioni descritte nell'interfaccia, tramite appositi "messaggi". Tali messaggi sono, solitamente, trasportati tramite il protocollo HTTP e formattati secondo lo standard XML [40]. Il compito di un Web Service non è solo questo, essi infatti consentono di comporre diversi servizi in rete, combinando tra loro singoli servizi, per of-

frire nuove funzionalità [30]. Tale integrazione è possibile grazie all'utilizzo di standard basati su XML. Tramite un'architettura basata sui Web Service (SOA), applicazioni software, scritte in diversi linguaggi di programmazione e implementate su diverse piattaforme hardware, possono cooperare, grazie alle interfacce che "espongono" pubblicamente e lo scambio di informazioni, per effettuare operazioni complesse (quali, ad esempio, la realizzazione di processi di business che coinvolgono più aree di una medesima azienda) sia su reti aziendali che su Internet. L'interoperabilità fra diversi linguaggi di programmazione e diversi sistemi operativi è resa possibile dall'uso di standard "aperti" [40]. Per poter meglio comprendere cosa sia un Web Service è necessario conoscere il paradigma SOA in cui si riscontra la stessa interazione del paradigma client-server.

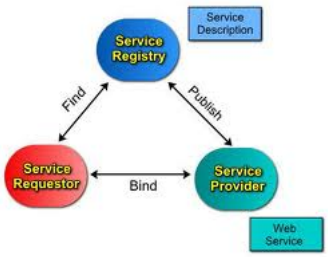

Figura 5.1: Paradigma SOA

Attraverso la SOA questa interazione viene arricchita con un ulteriore attore detto Service Directory o Service Broker che, come mostrato in figura 1, si inserisce all'interno della comunicazione tra fornitore e fruitore del servizio [30].
La ragione principale per spinge all'utilizzo di Web Service è il disaccoppiamento, che l'interfaccia standard esposta dal Web Service rende possibile, fra il sistema utente ed il Web Service stesso. Infatti la separazione tra la logica applicativa e quella di presentazione è netta: il Web Service si fa carico solo della prima lasciando al fruitore del servizio la seconda. Tale flessibilità consente la creazione di sistemi software complessi costituiti da componenti svincolati l'uno dall'altro e consente una forte riusabilità di codice ed applicazioni già sviluppate. Un'altra ragione della forte diffusione dei Web Service è l'utilizzo della porta 80 e del protocollo di trasporto HTTP "over" TCP. La porta 80 è, normalmente, una delle poche lasciate aperte dai sistemi firewall al traffico di entrata ed uscita dall'esterno verso i sistemi aziendali e ciò consente l'utilizzo dei Web Service senza

modifiche sulle configurazioni di sicurezza dell'azienda (anche se questo aspetto talvolta solleva preoccupazioni concernenti la sicurezza) [40]. Il WSDL però propone solo una fotografia del servizio, dandone una visione statica. Ciò che manca, quindi, é la dinamica del servizio chiamata anche comportamento. Attraverso il comportamento é possibile sapere come un servizio funziona e quali sono le operazioni ammissibili in accordo con il suo stato interno.

5.2.2 Composizione di servizi

L'affermarsi dei Web Service rende naturale l'estensione dei concetti alla base dei sistemi di gestione di workflow anche per coordinare servizi in rete forniti da diverse organizzazioni. In questo ambito, l'obiettivo principale è quello di comporre più servizi forniti da diversi fornitori al fine di creare nuovi servizi a valore aggiunto. Tale composizione richiede però la definizione di standard per modellare le interazioni tra i servizi, e a tali standard stanno lavorando numerosi vendor e ricercatori. La letteratura propone due principali approcci al coordinamento dei servizi in rete, che vengono designati sotto le denominazioni di "orchestrazione" e "coreografia" [30]. I termini orchestrazione e coreografia si riferiscono a due aspetti riguardanti la creazione di processi di business tramite la composizione di Web Service. Gli standard proposti per entrambi gli approcci devono soddisfare diversi vincoli e richieste tecniche per il design di processi di business che coinvolgano Web Service. Questi requisiti riguardano sia il linguaggio, con cui descrivere il workflow, sia l'infrastruttura sottostante. L'architettura del processo deve anche permettere di gestire le eccezioni e preservare l'integrità transazionale.

Orchestrazione

L'orchestrazione si riferisce a un processo di business eseguibile che può interagire con Web Service sia interni che esterni. Tale interazione avviene a livello message. L'orchestrazione prevede che il controllo sia gestito da una sola unità. L'orchestrazione di Web Service deve essere dinamica, flessibile e adattabile per poter andare incontro ai cambiamenti che il business richiede. Un punto importante che quest'approccio deve garantire è la netta separazione tra logica di processo e di presentazione, caratteristica chiave dei Web Service che ne permette la forte

flessibilità.
Il linguaggio di orchestrazione che modella il comportamento dei Web Service in un processo di business integration è il Business Process Execution Language. É un linguaggio XML-based per descrivere i requisiti della logica di controllo per coordinare i Web Service che fanno parte dei process flow. BPEL è un livello al di sopra del WSDL. L'interfaccia WSDL definisce le operazioni consentite, e il BPEL definisce con quale sequenza queste debbano essere svolte. Il WSDL descrive entrate pubbliche ed exit point per ogni processo BPEL, e i data type WSDL descrivono le informazioni che si scambiano i processi. Il WSDL può anche riferirsi a servizi esterni di cui necessita il processo BPEL.

Il formalismo BPEL supporta processi di business sia eseguibili che astratti. Un processo eseguibile modella precisamente il comportamento dei partecipanti in una specifica interazione di business, modellando un workflow per ciascuno di essi. Mentre un processo astratto, chiamato anche business protocol, specifica gli scambi di messaggi pubblici tra le parti. I business protocol non sono eseguibili e non forniscono dettagli interni sul flusso di processo.

La specifica di BPEL4WS include il supporto per attività di base e per quelle strutturate. Un'attività di base è un'istruzione che interagisce con qualcosa all'esterno del processo stesso. In uno scenario tipico, un processo BPEL eseguibile riceve un messaggio, dopo il quale può invocare una serie di servizi per raccogliere ulteriori dati e rispondere al richiedente. Le attività strutturate gestiscono il flusso complessivo del processo, specificando la sequenza di riferimento Web Service. Tali attività supportano anche loop e branching dinamici. Uno degli aspetti chiave per implementar un processo BPEL è la specifica delle variabili e dei partnerLink. Le variabili identificano il dato specifico scambiato in un messaggio. Quando un processo BPEL riceve un messaggio, assegna le opportune variabili in modo tale che le richieste successive possano accedere ai dati. Invece un partnerLink può essere un qualsiasi servizio che il processo intende invocare o un servizio che invoca il processo.
E' inoltre possibile gestire le eccezioni e le transazioni, sulla base delle specifiche WS-Coordination e WS-Transaction.

Coreografia

La coreografia, diversamente dall'orchestrazione, è collaborativa e permette ad ognuna delle parti coinvolte di partecipare all'interazione attraverso lo scambio di messaggi pubblici.
L'interfaccia Web Service Choreography Interface definisce un estensione collaborativa per il WSDL, specificando i messaggi scambiati tra i Web Service. L'interfaccia, diversamente dal BPEL, non supporta la definizione di processi di business eseguibili. Una singola interfaccia WSCI descrive solo la partecipazione di un partner nello scambio di un messaggio. Ogni azione WSCI rappresenta un'unità di lavoro che è mappata su una specifica operazione WSDL. WSCI estende il linguaggio WSDL descrivendo come coreografare le operazioni disponibili in un WSDL (cioè Il WSDL descrive entry point per ciascun servizio disponibile e la WSCI descrive l'interfaccia tra le operazione del WSDL). Anche la WSCI supporta sia attività di base che strutturate, e consente la gestione di eccezioni e transazioni di business.
Infine l'ultimo linguaggio che è opportuno conoscere quando si vuole trattare la composizione di servizi è BPML. Il Business Process Management Language è un linguaggio basato sull'XML che si utilizza per descrivere processi di business. E' stato disegnato per supportare processi che un sistema di gestione di processi di business sia in grado di eseguire. BPML supporta l'interfaccia WSCI, con cui ha in comune lo stesso modello di esecuzione del processo, quindi gli sviluppatori possono usare la WSCI per descrivere interazioni pubbliche tra i processi di business e BPML per le implementazioni private. Il linguaggio BPML fornisce costrutti per il process flow e attività simili a quelle di BPEL, l'a attività di base per l'invio, la ricezione e l'invocazione dei servizi disponibili e le attività strutturate per le esecuzioni di costrutti in parallelo, di cicli e di join. BPML permette inoltre la composizione ricorsiva per aggregare processi di business partendo da sottoprocessi [28].

5.3 Architettura

Per poter effettuare questa analisi abbiamo realizzato il processo tramite composizione di Web Service, uno per ciascuna attività del process flow. Il primo servizio effettua il login eseguendo una query che ricerca il codice utente inserito dal cliente all'interno del database dei clienti registrati. Il secondo WS verifica la disponibilità dei prodotti nel magazzino. Infine l'ultimo servizio compone l'ordine con la quantità richiesta dell'utente, il suo identificativo e il codice del prodotto desiderato.
Per quanto riguarda l'implementazione del Quality Block dopo aver svolto lo studio delle quattro dimensioni principali, riassunte nei capitoli precendenti, abbiamo suddiviso il blocco di qualità in quattro sotto-blocchi dedicati alle singole dimensioni. Questo rende possibile un'analisi di qualità più precisa. Anche i quattro sotto-blocchi sono stati implementati come quattro WS, e sono stati posizionati nel process flow descritto dal BPEL rispettando gli scenari ipotizzati nella presentazione delle tre soluzioni di qualità alternative.

Capitolo 6

Conclusioni e sviluppi futuri

Abbiamo visto che la qualità dei dati ha effetti significativi nelle imprese moderne che lavorano con grandi quantità di dati e che necessitano di dati qualitativamente adatti ai loro processi. Migliorare la metodologia di valutazione della qualità può aiutare a prendere provvedimenti e a seguire strategie di riparazioni adatte alle proprie esigenze. In questo lavoro abbiamo descritto un modello di analisi preliminare alla verifica di qualità dei dati che investiga nella struttura dei dati e del processo che li utilizza cercando possibili fonti di errori. La nostra analisi si basa su un approccio chiave della verifica di qualità dei dati, l'approccio Information Product. Questo approccio descrive il processo che porta alla creazione delle informazioni con alcune analogie con un processo manifatturiero, consentendoci di poterlo analizzare similmente. La metodologia di analisi proposta è supportata dall'esempio di un caso di studio, uno tra i processi più comuni in ambito aziendale, cioè le attività di gestione degli ordini. Una volta analizzato il processo ed i dati abbiamo proposto delle linee guida alternative per disegnare il processo monitorato. Le configurazioni proposte si possono catalogare in due diversi modi, la prima categoria privilegia la qualità e la tempestività di individuarne la mancanza, la seconda monitora il processo nel suo insieme. Per ora abbiamo individuato tre alternative. La prima proposta, appartienente alla prima categoria, la possiamo definire localmente ottima sotto il profilo della qualità dei dati poiché associa ad ogni attività il monitoraggio dei dati che questa usa. Per realizzare questa configurazione occorre implementare attività di verifica da affiancare o anteporre a quelle del processo. Della seconda categoria fanno parte le altre due alternative.

Queste due modalità di configurazione verificano i risultati di qualità dei dati, senza trascurare un altro fattore fondamentale per un processo di business, cioè il tempo di servizio. All'interno della seconda categoria di configurazioni la scelta tra le due alternative si basa sulla probabilità che i dati non soddisfino i requisiti imposti, poiché l'eventuale insoddisfazione dei vincoli può far variare il tempo di esecuzione, causando un breve ritardo nella seconda configurazione dovuto alle azioni di riparazione e la riesecuzione del l'intero processo nella terza che deve essere rieseguito su istanze corrette dei dati.

Uno dei possibili sviluppi futuri di questo studio potrebbe riguardare l'ampliamento delle soluzioni di monitoraggio, tali sviluppi potrebbero essere rivolti a migliorare le configurazioni creando altre alternative che siano in grado di tenere conto anche di altri aspetti oltre al tempo di esecuzione. Si potrebbero definire altre configurazioni che monitorino la qualità considerando altri KPI o cercare di adattare le alternative in base al variare dell'importanza di questi KPI.

Bibliografia

[1] Web data quality: A 6 step process to evolve your mental model.

[2] *Information Quality.* Elizabeth M. Pierce, AMIS, 2006.

[3] *Proceedings of IEEE International Conference on Communications, ICC 2008, Beijing, China, 19-23 May 2008.* IEEE, 2008.

[4] Danilo Ardagna and Barbara Pernici. Global and local qos guarantee in web service selection. In Bussler and Haller [9], pages 32–46.

[5] Danilo Ardagna and Barbara Pernici. Adaptive service composition in flexible processes. *IEEE Trans. Software Eng.*, 33(6):369–384, 2007.

[6] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41(3), 2009.

[7] Carlo Batini and Monica Scannapieco. *Data Quality: Concepts, Methodologies and Techniques.* Data-Centric Systems and Applications. Springer, 2006.

[8] Paul L. Bowen, Ahmed K. Elmagarmid, Hubert Österle, and Kai-Uwe Sattler, editors. *Proceedings of the 14th International Conference on Information Quality, ICIQ 2009, Hasso Plattner Institute, University of Potsdam, Germany, November 7-8 2009.* HPI/MIT, 2009.

[9] Christoph Bussler and Armin Haller, editors. *Business Process Management Workshops, BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM,*

WSCOBPM, BPS, Nancy, France, September 5, 2005, Revised Selected Papers, volume 3812, 2006.

[10] Cinzia Cappiello and Chiara Francalanci. A model of non-quality cost. *Metodologie e Strumenti per la Qualità dei Dati in Sistemi Informativi Cooperativi*, 2002.

[11] Cinzia Cappiello, Chiara Francalanci, and Barbara Pernici. Time- related factors of data quality in multichannel information systems. *J. of Management Information Systems*, 20(3):71–91, 2004.

[12] Cinzia Cappiello and Barbara Pernici. Quality-aware design of repairable processes. In Neely et al. [27], pages 382–396.

[13] Cinzia Cappiello and Barbara Pernici. Quads: Quality-aware design of dependable service-based process. *IEEE Trans. Software Eng.*, 2010.

[14] Antonio J. Silva Cardoso. *Quality of Service and Semantic Composition of Workflows*. PhD thesis, University of Georgia, Athens, GA, 2002.

[15] Luca Console, Claudia Picardi, and Daniele Theseider Dupré. A framework for decentralized qualitative model-based diagnosis. In Veloso [38], pages 286–291.

[16] Craig Fisher and Bruce N. Davidson, editors. *Seventh International Conference on Information Quality (IQ 2002)*. MIT, 2002.

[17] Gerhard Friedrich, Mariagrazia Fugini, Enrico Mussi, Barbara Pernici, and Gaston Tagni. Exception handling for repair in service-based processes. *IEEE Trans. Software Eng.*, 36(2):198–215, 2010.

[18] Marco Frontini. *Fondamenti di Calcolo Numerico*. Libreria Clup, 2003.

[19] Maria Grazia Fugini, Barbara Pernici, and Filippo Ramoni. Quality analysis of composed services through fault injection. *Information Systems Frontiers*, 11(3):227–239, 2009.

[20] J. G. Geiger. Data quality management -the most critical initiative you can implement. *Intelligent Solutions, Inc., Boulder, CO*, 2010.

[21] Dr. Markus Helfert and Fakir Mohammad Zakir Hossain. An approach to monitoring data qualtiy - product oriented approach -. *AMCIS 2010 Proceedings*, 2010.

[22] Dr. Markus Helfert and Fakir Mohammad Zakir Hossain. Certifyng data quality conformance. In *CompSysTech'10*, 2010.

[23] Lukasz Juszczyk and Schahram Dustdar. Programmable fault injection testbeds for complex soa. In Maglio et al. [26], pages 411–425.

[24] Barbara D. Klein and Donald F. Rossin, editors. *Fifth Conference on Information Quality (IQ 2000)*. MIT, 2000.

[25] Andreja Kovacic. Business renovation: business rules (still) the missing link. *Business Process Management*, pages 158–170, 2004.

[26] Paul P. Maglio, Mathias Weske, Jian Yang, and Marcelo Fantinato, editors. *Service-Oriented Computing - 8th International Conference, ICSOC 2010, San Francisco, CA, USA, December 7-10, 2010. Proceedings*, volume 6470 of *Lecture Notes in Computer Science*, 2010.

[27] M. Pamela Neely, Leo Pipino, and John P. Slone, editors. *Proceedings of the 13th International Conference on Information Quality, MIT, Cambridge, MA, USA, 2008*. MIT, 2008.

[28] Chris Peltz. Web services orchestration and choreography. *IEEE Computer*, 36(10):46–52, 2003.

[29] Elizabeth Pierce. Ip-map standards and giudelines. *IQ*, 2002.

[30] Pierluigi Plebani and Barbara Pernici. Un'introduzione ragionata al mondo dei web service. *Mondo Digitale*, 2004.

[31] Thomas Redman. the impact of poor data quality on the typical enterprise. *Commun. ACM*, 41(2):79–82, 1998.

[32] Thomas C. Redman. *Data quality for the information age*. Artech House, 1996.

[33] Noelia Sánchez-Serrano, Ismael Caballero, and Félix García. Extending bpmn to support the modeling of data quality issues. In Bowen et al. [8], pages 46–60.

[34] Monica Scannapieco, Barbara Pernici, and Elizabeth M. Pierce. Ip-uml: Towards a methodology for quality improvement based on the ip-map framework. In Fisher and Davidson [16], pages 279–291.

[35] Ganesan Shankaranarayanan, Richard Y. Wang, and Mostapha Ziad. Ip-map: Representing the manufacture of an information product. In Klein and Rossin [24], pages 1–16.

[36] Ying Su and Zhanming Jin. A methodology for information quality assessment in data warehousing. In *ICC* [3], pages 5521–5525.

[37] Giri Kumar Tayi and Donald P. Ballou. Examining data quality - introduction. *Commun. ACM*, 41(2):54–57, 1998.

[38] Manuela M. Veloso, editor. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2007.

[39] Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *J. of Management Information Systems*, 12(4):5–33, 1996.

[40] World Wide Web Consortium. *W3C*.

[41] Liangzhao Zeng, Boualem Benatallah, Anne H. H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. Qos-aware middleware for web services composition. *IEEE Trans. Software Eng.*, 30(5):311–327, 2004.

Elenco delle figure

Elenco delle tabelle