

# Teoria organizzazione

Il sistema informativo è la componente (o sotto-sistema) di un'organizzazione, che gestisce le informazioni di interesse. In particolare, ci focalizziamo sui **sistemi informativi aziendali**, ovvero sistemi informativi che possono essere utilizzati in azienda per gestire (dove per gestire intendiamo: Analizzare, Elaborare ed Estrapolare) le informazioni di interesse, dove tali **informazioni di interesse** (all'interno di un'azienda) sono quelle informazioni che permettono all'organizzazione stessa di raggiungere l'obiettivo che si è prefissata. Capiamo, allora immediatamente, che le informazioni di interesse cambiano a seconda del contesto dell'azienda (nel senso che le informazioni di interesse di una banca sono diverse rispetto ad un'azienda di vendite) e sono quelle informazioni che vengono utilizzate e prodotte durante l'esecuzione dei processi aziendali → quindi, le informazioni di interesse "accompagnano" l'azienda in tutte le sue attività e quindi sono quelle informazioni su cui si appoggiano (e producono) i processi aziendali. Con il termine "**processo aziendale**" intendiamo un insieme di attività, che devono essere eseguite, per raggiungere un determinato risultato/obiettivo.

Essendo che abbiamo detto che parleremo di sistemi informativi aziendali, è buona cosa capire cosa intendiamo con il termine "**organizzazione**": il termine organizzazione ha due significati che ci interessano:

1. intendiamo un **insieme di persone e di tecnologie**, le quali (ovvero persone e tecnologie) vengono utilizzate per gestire le risorse e raggiungere l'obiettivo aziendale. Dato l'insieme di persone, che con il loro lavoro partecipano allo svolgimento dell'attività dell'azienda, l'organizzazione è il processo attraverso il quale tale insieme di persone viene strutturato secondo i principi di divisione del lavoro e coordinamento. Grazie all'organizzazione, tale insieme acquisisce una struttura e diventa un sistema;
2. intendiamo il risultato del **processo di divisione del lavoro e coordinamento delle risorse umane**. In questo senso, quindi, il termine organizzazione è sinonimo di azienda.



Il sistema informativo aziendale è di fondamentale importanza, perchè partendo dai **dati** (memorizzati nelle nostre basi di dati relazionali), esso (ovvero il SIA) ci permette di **gestire, elaborare e produrre informazioni**. Quindi, se abbiamo memorizzato nella nostra base di dati che, ad esempio, in un certo periodo dell'anno le vendite diminuiscono e da questi dati riesco ad estrapolare un'informazione. Rispetto a tale informazione (che riusciamo ad estrapolare dai dati), prendiamo una decisione e mettiamo in atto una determinata attività.

Ai fini dello studio della sua organizzazione, l'azienda può essere considerata un sistema socio-tecnico costituito da:

- **Persone** → risorse umane che costituiscono l'organismo personale dell'azienda;
- **Tecnologie** → mezzi strumentale + know how (il quale identifica le conoscenze e le abilità operative necessarie per svolgere una determinata attività lavorativa).

Un'azienda, nell'economia aziendale, è un'organizzazione di uomini e mezzi finalizzata alla soddisfazione di bisogni umani attraverso la produzione, la distribuzione o il consumo di beni economici. Dall'interazione tra risorse umane e tecnologie deriva il **comportamento aziendale**, rivolto al raggiungimento degli obiettivi aziendali → alla base di tutto ciò, vi sono i dati e le informazioni, in quanto nessun processo aziendale può essere portato avanti senza l'uso e/o la produzione di informazioni. Quindi, qualsiasi attività all'interno dell'azienda utilizza e/o produce informazioni.

Parliamo, a questo punto, della

**teoria dell'organizzazione:** La teoria dell'organizzazione è lo studio scientifico e progettazione scientifica dei compiti, per il miglioramento delle prestazioni. Dal punto di vista economico, quindi, la teoria dell'organizzazione ci permette di dire in che modo possiamo organizzare le attività aziendali (le quali verranno modellate con dei processi aziendali), al fine di migliorare le prestazioni.

L'organizzazione aziendale:

- **prevede ruoli e funzioni diversificate** → quindi a partire da un insieme di persone, che abbiamo a disposizione all'interno dell'azienda, è intuitivo

pensare che ognuno abbia il proprio ruolo gerarchico e ha funzioni diverse a seconda del ruolo che ricopre all'interno dell'azienda;

- **caratterizzata da fini** (ovvero gli obiettivi che si intende raggiungere), **metodi** (in quanto per raggiungere gli obiettivi avremo bisogno dei metodi, ovvero attività che fanno parte dei processi aziendali) **e regole** (in quanto per raggiungere gli obiettivi, oltre che attraverso i metodi, dobbiamo rispettare determinate regole interne e/o esterne all'azienda);
- **l'azienda opera attraverso processi** e si cerca di applicare il processo ottimo (ovverosia il miglior processo) per raggiungere un determinato obiettivo, nel miglior modo e nel minor tempo. Inoltre, sottolineiamo il fatto, che ogni azienda ha almeno i macro-processi:
  - operativi o produttivi;
  - controllo e gestione.
- **ogni azienda, inoltre, dispone di risorse, che dobbiamo gestire e coordinare per raggiungere gli obiettivi prefissati.** In particolari, le risorse che ogni azienda ha a disposizione sono:
  - **umane;**
  - **materiali;**
  - **informative** → se parliamo di un'azienda che offre servizi, allora le risorse informative vanno a sostituire le risorse materiali, ma anche nelle altre tipologie di aziende (quali ad esempio aziende manifatturiere), le risorse informative sono di fondamentale importanza. In particolare, a noi interessa come tali risorse informative possono essere utilizzate, prodotte e sfruttate in azienda, al fine di raggiungere l'obiettivo aziendale.

Quindi, tutti i sistemi di cui parleremo vengono definiti **IT (Information Technology)** → **le tecnologie informatiche sono un insieme di sistemi, strumenti e tecniche predisposti per automatizzare il trattamento delle informazioni.** In particolare, siamo interessati ai sistemi informativi aziendali, perchè sono quelle tecnologie informatiche che supportano il ciclo di vita dell'informazione come risorsa aziendale. A noi, quindi (tra le risorse che l'azienda ha a disposizione) ci interessano le risorse informative e quindi ci interessa la parte di gestione dell'informazione. Ci interessa, perchè l'informazione:

- è oggetto di tutti i processi produttivi e operativi (quindi qualsiasi cosa facciamo in azienda utilizza o produce informazione);
- è oggetto delle attività gestionali (quindi qualsiasi attività gestionale svolta in azienda, si basa sulle informazioni);
- ha un ruolo organizzativo come risorsa delle attività di coordinamento e controllo.



**Il rapporto che esiste tra il concetto di sistema informativo e quello di tecnologie informatiche è di strumentalità, ovvero: il sistema informativo utilizza le tecnologie informatiche come strumento per gestire l'informazione d'interesse.**

Sorge a questo punto spontanea la domanda: **"Che cos'è un sistema informativo aziendale?"** Un sistema informativo aziendale è un insieme di elementi interconnessi che:

- raccolgono;
- catalogano;
- ricercano;
- elaborano;
- memorizzano;
- proteggono;
- distribuiscono

dati trasformandolo in informazioni utili per supportare le **attività decisionali** di controllo di un'azienda (come per esempio: potenziare una linea produttiva; cambiare completamente l'offerta dei prodotti sul mercato; cambiare la produzione dei prodotti integrando l'aspetto ecologico) → naturalmente, queste decisioni derivano in parte dalle nuove regolamentazioni del segmento di mercato in cui opera l'azienda e in parte dal "sentire comune" (nel senso che, ad esempio al giorno d'oggi, la comunità si focalizza sul rispetto dell'ambiente e di conseguenza, l'azienda adotta queste nuove decisioni per acquisire e/o fidelizzare nuovi clienti).

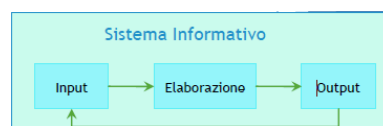
A noi quindi interessa, capire in che modo i sistemi informativi aziendali possono essere di supporto nelle attività decisionali. Inoltre, notiamo il fatto che

l'attività decisionale ha un duplice aspetto:

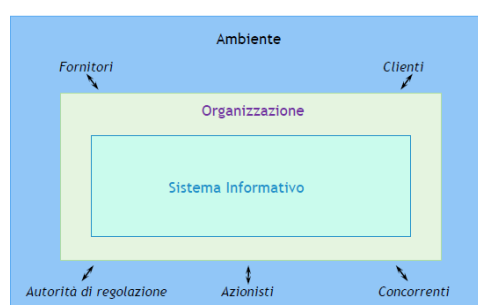
- da una parte è relativa alla gestione dell'informazione;
- dall'altra parte è relativa alla modellazione dei processi (perchè, ad esempio, produrre i prodotti in maniera più eco-sostenibile comporta la rivisitazione dei processi aziendali).

Un sistema informativo aziendale deve essere progettato per svolgere tre macro-processi fondamentali:

1. **Acquisire dei dati** (processo di input) provenienti dalle nostre basi di dati oppure provenienti dall'esterno;
2. **Trasformazione dei dati** (processo di elaborazione);
3. **Restituzione di informazioni** (processo di output) che ci permetteranno di prendere una decisione.



I sistemi informativi svolgono la funzione di automatizzare la gestione e l'elaborazione dei dati. Quindi, i dati devono essere disponibili al management per trasformarsi in informazioni utili, per prendere decisioni efficaci, tempestive, nel rispetto dell'efficienza globale dell'azienda. Possiamo quindi osservare la seguente immagine:



Vediamo che la sotto-parte di organizzazione, che ci permette di gestire le informazioni prende il nome di Sistema Informativo. Vediamo, inoltre, che abbiamo l'**Ambiente**, il quale ha una ricaduta (ha degli effetti) sull'organizzazione. Nell'ambiente ci sono:

- i fornitori;

- i clienti;
- autorità di regolazione;
- azionisti;
- concorrenti.

In generale, quindi, nell'ambiente ci sono tutti i soggetti interessati al comportamento dell'azienda e quindi nell'ambiente vi sono gli **stakeholders**. Capiamo immediatamente, allora, che al giorno d'oggi l'ambiente non è più la zona geografica limitrofa all'azienda, bensì al giorno d'oggi l'ambiente è costituito potenzialmente da tutto il mondo, in quanto l'azienda si confronta con tutto il mondo e deve "rimanere al passo" con le altre aziende e per riuscire a fare ciò, l'azienda deve utilizzare un sistema informativo aziendale. Capiamo, allora, che l'informazione assume un ruolo estremamente importante ed in particolar modo, nel livello strategico l'informazione è la risorsa fondamentale, in quanto ci permette di prendere le decisioni. Nel livello operativo, invece, vi sarà sempre l'informazione (in quanto abbiamo detto che ogni processo aziendale genera informazione), ma quest'ultima non sarà la risorsa principale. Riassumendo, quindi, possiamo dire che l'informazione:

- è la **risorsa principale**, che viene scambiata, selezionata ed elaborata nelle attività gestionali di controllo e coordinamento;
- viene **prodotta da qualunque attività** (anche quella operativa);
- è una **risorsa immateriale** (mentre le altre risorse aziendali sono materiali, come ad esempio: le risorse umane o le materie prime) ed è alla base di ogni altra risorsa immateriale e in questo senso, l'informazione può essere considerata come la conoscenza e l'esperienza individuale;
- **non è facilmente divisibile o appropriabile**;
- può essere **soggetta ad obsolescenza**;
- **non viene distrutta dall'uso**;
- **non si esaurisce, bensì si auto-rigenera** (per esempio ogni mese calcolo il fatturato mensile e alla fine dell'anno, per calcolare il fatturato annuale, utilizzo il fatturato mensile di ogni mese). Si dice, quindi, che l'informazione è self-generating e tale capacità dell'informazione, permette di creare dei cicli di:
  - generazione di altre informazioni e tali informazioni prendono il nome di **conoscenza**;

- arricchimento dell'informazione disponibile.

Le ultime due caratteristiche sono distintive dell'informazione, che la rendono diversa dalle altre risorse aziendali.

Capiamo, allora, che la quantità di informazione che abbiamo a disposizione è estremamente importante. Capiamo anche, però, che la quantità di informazione è importante finché la riusciamo a gestire, ovvero: se continuiamo a produrre informazione e tale quantità d'informazione non riusciamo a gestirla, ci troviamo nello scenario dell'**overload informativo**, ovvero:

- aumento incontrollato dell'informazione disponibile;
- disponibilità di informazioni, che eccedono le capacità di elaborazione individuale;
- rallentamento e peggioramento delle decisioni, in quanto non siamo riusciti a processare tutte le informazioni a disposizione.

Analogamente, possiamo trovarci nel caso opposto, ovvero nello scenario dell'**underload informativo**:

- disponibilità di informazioni al di sotto delle capacità di elaborazione individuale;
- semplificazione delle decisioni;
- decisioni in tempi brevi.



Da notare, che prendere le decisioni in tempi brevi e la semplificazione delle decisioni nello scenario dell'underload informativo, non è necessariamente qualcosa di positivo, in quanto è sintomo del fatto che l'azienda non ha preso in considerazione tutte le possibilità che avrebbe avuto a disposizione se le informazioni disponibili fossero state nella giusta misura delle capacità di elaborazione individuale → capiamo allora, che **lo scenario ottimale è il giusto trade-off tra l'overload e l'underload informativo, ovvero il giusto trade-off tra la produzione di informazione corretta rispetto alla capacità di elaborazione che abbiamo a disposizione.**

Continuiamo il discorso, dicendo che **l'azienda è un sistema aperto**, ovvero l'azienda deve interfacciarsi con l'ambiente (in cui essa è inserita) e questo vuol

dire, che tutte le entità che risiedono nell'ambiente, ovvero le entità con cui l'organizzazione deve interfacciarsi, quali:

- fornitori;
- clienti;
- autorità di regolazione;
- azionisti;
- stakeholder
- concorrenti

generano informazione. Un'organizzazione, quindi, è considerata un sistema aperto in quanto interagisce costantemente con l'ambiente circostante, scambiando energia, risorse e informazioni con esso. L'organizzazione ed il sistema informativo si interfacciano con azionisti, concorrenti, clienti e fornitori. Sorge allora spontanea la domanda: **"Quand'è che un'organizzazione riesce a raggiungere i suoi obiettivi, nonostante sia un sistema aperto?"**

Un'organizzazione riesce a raggiungere i suoi obiettivi, quando essa è sufficientemente **flessibile** da adattarsi a quello che gli sta intorno. Ovviamente, maggiore sarà la flessibilità dell'azienda, maggiore sarà la capacità di quest'ultima di adattarsi:

- ai cambiamenti dell'ambiente esterno;
- ai nuovi requisiti che esso impone alle organizzazioni.

**"Cosa rende un'organizzazione flessibile?"** Sicuramente un'organizzazione è resa flessibile dal suo sistema informativo aziendale, in quanto quest'ultimo è il supporto che l'azienda utilizza, per prendere decisioni in maniera veloce e corretta. Quindi, **l'informazione è la risorsa aziendale, che gioca un ruolo fondamentale nel determinare la capacità di adattamento dell'azienda.** Tutti questi input, che arrivano all'organizzazione dall'esterno, prendono il nome di **incertezza ambientale** → quindi, tutte le variabili ambientali che influenzano la progettazione del sistema informativo vengono riassunte in un unico costrutto, ovvero la incertezza ambientale. L'incertezza ambientale determina:

- requisiti di capacità elaborativa delle organizzazioni;
- adeguatezza della struttura del sistema informativo.

Quindi, abbiamo che l'organizzazione è un sistema aperto, che sta nell'ambiente. L'ambiente, a sua volta, è un posto incerto (in quanto non



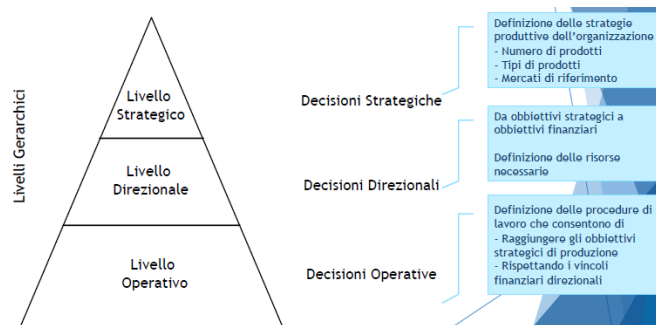
sappiamo cosa succede) e quindi dobbiamo essere in grado di interfacciarci con l'ambiente utilizzando le informazioni che abbiamo a disposizione. In particolare, l'incertezza ambientale genera **eccezioni** e a fronte di tali eccezioni, dobbiamo essere in grado di prendere una decisione ad un certo livello della piramide e quindi, dobbiamo essere in grado di adeguarci nel miglior modo possibile, andando per esempio a modificare il nostro processo aziendale per raggiungere comunque il nostro obiettivo aziendale. Riassumiamo tutti questi concetti con la seguente figura:



A livello concettuale, possiamo immaginare sempre che all'interno di un'azienda abbiamo:

- un livello **operativo** → in cui abbiamo l'esecuzione di tutti i processi operativi e quindi la catena produttiva sta nella parte bassa della piramide;
- un livello **direzionale** → coordina e gestisce il livello operativo;
- un livello **strategico** → prende le decisioni che impattano su tutta la piramide. Capiamo immediatamente, che le decisioni rispetto all'incertezza ambientale (la quale genera eccezioni) vengono prese a livello strategico.

In ognuno di questi livelli, abbiamo dei sistemi informativi diversi; in particolare, le basi di dati relazionali stanno al livello operativo, in quanto esse (ovvero le basi di dati) mi permettono di gestire le operazioni quotidiane, che quindi vengono eseguite a livello operativo. Nel momento in cui, ad esempio, dalla base dati ricaviamo dei report mensili, ci stiamo spostando nel livello direzionale e successivamente dalle valutazioni direzionali possiamo prendere delle decisioni a livello strategico. I processi e il livello in cui mi trovo sulla **piramide di Anthony** ha delle ricadute sulle decisioni che andiamo a prendere e ogni decisione ha ricadute sui livelli sottostanti:



A livello operativo prendiamo delle decisioni operative; a livello direzionale prendiamo decisioni direzionali, mentre a livello strategico prendiamo decisioni strategiche.

Naturalmente, maggiore è l'incertezza ambientale:

- **più intensa sarà l'attività di natura informativa** (nel senso che, maggiori sono gli input proveniente dall'ambiente, maggiore sarà la necessità di elaborare informazioni);
- dato che abbiamo maggiori informazioni, allora avremo bisogno di **maggiore capacità elaborativa** dell'organizzazione. In particolare, vogliamo che la capacità elaborativa di un'organizzazione non sia la mera somma delle capacità elaborative dei singoli individui che compongono l'azienda, bensì vogliamo che la capacità elaborativa dell'organizzazione sia superiore alla capacità elaborativa di ciascuno degli individui che ne fanno parte. Per riuscire a fare ciò, le capacità elaborative di ciascuno degli individuo può essere potenziata attraverso la **collaborazione**. La cooperazione fra più individui determina:
  - suddivisione del compito elaborativo;
  - suddivisione delle informazioni.

Ognuno poi necessita di un sotto-insieme di informazioni adatte allo scopo del sotto-compito di cui è responsabile. L'organizzazione per raggiungere una capacità elaborativa complessiva superiore a quella individuale deve organizzare la raccolta e lo scambio delle informazioni e quindi si deve progettare il sistema informativo. La progettazione del sistema informativo è un'attività fortemente legata alla pianificazione della struttura organizzativa. Vogliamo, quindi, avere un allineamento fra: il sistema informativo, la struttura organizzativa e il livello di incertezza.

- per avere maggiore capacità elaborativa, abbiamo bisogno di un **sistema informativo adeguato**.

Fino a questo momento, abbiamo parlato di incertezza ambientale, ovvero sia l'incertezza causata dall'esterno (essendo che l'azienda è un sistema aperto), ma dobbiamo considerare il fatto, che l'incertezza non è solamente esterna all'azienda, bensì vi è incertezza anche all'interno dell'azienda stessa. In questo senso, si parla di **incertezza comportamentale**. A livello organizzativo, quindi, dobbiamo tenere in considerazione anche l'incertezza comportamentale, ovvero sia un'incertezza legata ad un comportamento non prevedibile degli individui. Chiaramente, il comportamento di una persona è difficilmente gestibile, però quando dobbiamo prendere una decisione, dobbiamo tenere in considerazione sia l'incertezza ambientale sia l'incertezza comportamentale.



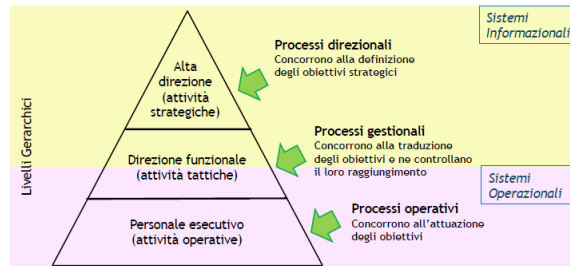
L'incertezza comportamentale genera un costo, in quanto dobbiamo costruire una sorta di "infrastruttura" che mi permette di controllare il comportamento degli individui.

## Sistemi informativi aziendali

La classificazione dei SIA può essere fatta da diversi punti di vista. In particolare, le principali due classificazioni sono:

1. Possiamo considerare la **piramide aziendale** (ovvero sia la **piramide di Anthony**) ed andare a collocare i SIA sulla piramide stessa e quindi descriviamo i SIA sulla base:
  - del **tipo di decisione** per cui viene utilizzato, ovvero in base al livello a cui il SIA viene utilizzato e quindi anche in base al tipo di informazione per cui verrà utilizzato. Per esempio, sul livello operativo (che è quello che ci permette di gestire le informazioni e le transazioni quotidiane) utilizziamo le basi di dati relazionali; mentre per quanto riguarda il livello manageriale utilizzeremo il Data Warehouse → riassumendo, quindi, possiamo dire che possiamo classificare i SIA rispetto a come vengono organizzati/disposti lungo la piramide di Anthony.

Analizziamo in maniera più approfondita questa prima classificazione dei SIA, attraverso l'immagine sotto:



Vediamo innanzitutto, che riusciamo a riconoscere la piramide di Anthony, in quanto riusciamo ad individuare:

- la parte relative alle attività operative;
- la parte relative alla attività tattiche (ovvero le attività manageriali e quindi direzionali);
- la parte relative alle attività strategiche.

In queste aree della piramide riusciamo a riconoscere un insieme di ruoli a diversi livelli, ovverosia riconosciamo un insieme di ruoli:

- del **personale esecutivo**;
- del **personale direzionale**;
- dell'**alta direzione**.

A questi ruoli corrispondono **processi** diversi (banalmente ogni ruolo fa determinate cose) e di conseguenza corrispondono **funzionalità** diverse e **decisioni** diverse. In particolare, la prima macro-suddivisione che possiamo fare è tra **Sistemi operazionali** e **Sistemi Informazionali**. Questa macro-suddivisione fa riferimento al fatto, che nella parte rosa della piramide saranno più utili i sistemi operazionali (in particolare le classiche basi di dati relazionali), che mi permettono di memorizzare le informazioni relative ai processi operativi, ovverosia i processi quotidiani (all'interno dell'azienda) che concorrono all'attuazione degli obiettivi aziendali (come per esempio: relazione con i fornitori, acquisizione del materiale) → questi sistemi operazionali, quindi, hanno lo scopo di memorizzare tutta l'informazione relativa alla gestione quotidiana dell'organizzazione.

I sistemi informazionali, invece, sono nella parte gialla della piramide ed in particolare, tali sistemi sono volti alla produzione di informazione. Quindi, nel momento in cui vogliamo prendere i dati che abbiamo a disposizione e li voglio utilizzare per produrre informazioni, allora utilizzeremo un sistema

informazionale. Capiamo, allora, che il sistema informazionale è utile nel momento in cui dobbiamo prendere una decisione.

Notiamo, inoltre, che ad ogni livello della piramide corrispondono processi aziendali diversi, in quanto a seconda del livello della piramide facciamo delle attività diverse. Quindi abbiamo che:

- al livello operativo abbiamo i **processi operativi**, i quali concorrono all'attuazione degli obiettivi aziendali;
- al livello direzionale abbiamo i **processi gestionali**, i quali concorrono alla traduzione degli obiettivi e ne controllano il loro raggiungimento;
- al livello strategico abbiamo i **processi direzionali**, i quali concorrono alla definizione degli obiettivi strategici e quindi mi permettono di prendere decisioni strategiche.

Vediamo, allora, che tutto si combina, nel senso che abbiamo diversi livelli e ogni livello ha:

- tipi di dati diversi;
- tipi di processi diversi.

Come possiamo vedere dall'immagine sotto, i sistemi operazionali ci permettono di gestire tutti i flussi informativi che riguardano le operazioni quotidiane della gestione aziendale; mentre i sistemi informazioni ci permettono di gestire i flussi informativi relativi al processo decisionale (relativo sia alla parte manageriale sia alla parte strategica) e svolgono l'attività di business intelligence.

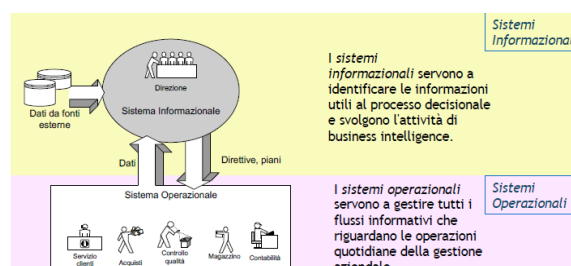


**Ovviamente tra questi due piani (ovverosia tra questi due sistemi) c'è uno scambio di informazione.** Possiamo vedere nell'immagine sotto, che ci sono diverse attività, quali:

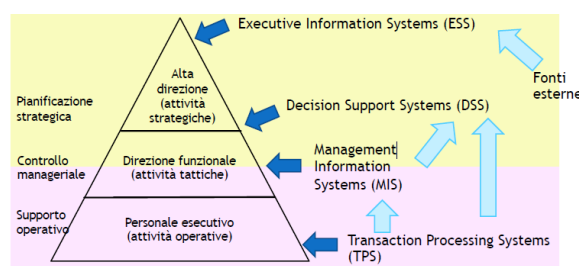
- servizio clienti;

- acquisti;
- controllo qualità;
- magazzino;
- contabilità.

Tali attività sfruttano i sistemi operazionali per svolgere effettivamente tali attività. Notiamo, innanzitutto che tali attività sono quotidiane ed inoltre, notiamo che queste attività utilizzano e producono informazione, la quale (ovvero l'informazione) viene passata al sistema informazionale (come ad esempio un Data Warehouse) e ci permette di prendere decisioni in base all'andamento dell'azienda. Ovviamente, **certe tipi di decisioni strategiche si basano anche su dati provenienti da fonti esterne** e quindi, se i sistemi operazionali considerano i dati interni all'azienda, i sistemi informazionali invece non si limitano ai dati interni, bensì considerano (in alcuni casi) anche i dati esterni. **Nel flusso di ritorno** (quindi nel flusso dai sistemi informazionali ai sistemi operazionali), infine **tornano direttive e piani, che sono effettivamente le decisioni**.



Una classificazione dei SIA lungo la piramide di Anthony è la seguente:



Al livello base abbiamo i sistemi che ci permettono di gestire le transazioni (ovverosia le nostre classiche basi di dati relazionali) chiamati **TPS (Transaction Processing Systems)**, che appunto tengono traccia di tutte le informazioni di routine delle organizzazioni (ovverosia gestiscono le transazioni di base). L'informazione di tali sistemi informativi (ovvero i TPS)

viene utilizzata dai **Management Information Systems (MIS)**, ovvero sistemi informativi che sono alla base del livello direzionale e permettono di analizzare l'informazione proveniente dalle basi di dati classiche in maniera periodica (per esempio, permettono di fare dei report a fine mese o di analizzare le entrate e le uscite di un determinato mese). Un altro sistema informativo, che utilizza i dati provenienti dai sistemi sottostanti, sono i **Decision Support Systems (DSS)**, ovvero i sistemi di supporto alle decisioni (quindi i Data Warehouse), che sono alla base del livello strategico e affiancano il management nelle decisioni non di routine e quindi, sono finalizzati al supporto delle decisioni aziendali non di routine (ovvero supportano decisioni complesse) attraverso i dati interni all'azienda. Notiamo, inoltre, che i DSS devono permettere a chi ha competenze diverse da quelle informatiche, di analizzare i dati rispetto all'idea che ha (ovvero rispetto all'idea che il soggetto ha) senza necessariamente sapere la struttura dei dati utilizzati. Infine, in questa classificazione, abbiamo gli **Executive Information Systems (EIS)**, i quali si trovano sulla punta della piramide e utilizzano principalmente fonti esterne, in quanto per riuscire a prendere una decisione strategica non possiamo guardare solamente quello che succede all'interno dell'azienda, ma devo guardare e confrontarmi anche con l'ambiente esterno. In particolare, tali sistemi sono riservati al senior manager per le decisioni strategiche non di routine (ovvero decisioni che possono cambiare il futuro dell'azienda) e lavorano in sinergia per supportare l'intero processo decisionale all'interno di un'organizzazione.



**Chi prende una decisione è probabile che utilizzi sia i DSS sia gli EIS. In particolare, utilizzerà i DSS per valutare l'andamento dell'azienda, mentre per analizzare l'andamento del mercato utilizzerà gli EIS.**

Possiamo riassumere i quattro sistemi informativi (appena sopra analizzati) con le seguenti immagini:

- ▶ Transaction Processing Systems (TPS).
  - ▶ Alla base della piramide.
  - ▶ Sistemi Informativi destinati alla gestione delle transazioni.
  - ▶ Tengono traccia di tutte le informazioni di routine nelle organizzazioni (ordini, spedizioni, ecc.)
  
- ▶ Management Information Systems (MIS).
  - ▶ A livello superiore.
  - ▶ Sistemi Informativi per il management.
  - ▶ Acquisiscono i dati dai TPS e consentono una rappresentazione strutturata periodica della situazione delle operazioni aziendali.
  - ▶ Alla base del sistema di reportistica per il management dell'organizzazione.
  - ▶ Permettono personalizzazioni (fornite ai manager) e si appoggiano ai dati di routine prodotti dall'organizzazione.
  
- ▶ Decision Support Systems (DSS).
  - ▶ Allo stesso livello dei precedenti.
  - ▶ Affiancano il management nelle decisioni di non routine.
  - ▶ Finalizzati al supporto delle decisioni aziendali.
  - ▶ Permettono di simulare ipotesi per verificare e testare la validità di una gestione.
  - ▶ Utilizzano dati provenienti dai TPS e dagli MIS
  
- ▶ Executive Information Systems (EES).
  - ▶ Al vertice della gerarchia dei Sistemi Informativi.
  - ▶ Sistemi Informativi direzionali riservati al senior manager di più alto livello dell'organizzazione.
  - ▶ A servizio delle decisioni strategiche di non routine.
  - ▶ Progettati per incorporare dati ed eventi che provengono da fonti esterne.
  - ▶ Spesso composti nel formato di un cruscotto digitale con il quale il senior manager riesce ad avere sotto controllo gli andamenti della gestione aggregando informazioni interne ed esterne e sintetizzandole con gli indicatori più importanti.

2. Possiamo considerare le **aree gestionali** dell'azienda (dove chiaramente le funzioni sono sempre quelle descritte dalla piramide di Anthony), ma non sono più rappresentate come: livello operativo, livello direzionale e livello strategico, bensì le funzioni sono descritte come aree funzionali, ovvero le funzioni vengono classificate in base alla funzionalità che svolgono e quindi alle operazioni che conducono. In questo caso, quindi, descriviamo quali sono le aree funzionali dell'azienda e andremo a mappare i SIA in base all'area a cui sono di supporto. **Andiamo, quindi, a raggruppare gli organi aziendali in unità organizzative, ognuna delle quali (ovvero ogni unità organizzativa) mette a disposizione dell'azienda un insieme di funzionalità.** Andremo, allora, a classificare i sistemi informativi in base al tipo di funzionalità che sono in grado di supportare. Le aree funzionali di un'azienda sono le seguenti:





Vediamo, quindi, che abbiamo:

- **Area direzionale** (vediamo che la rappresentazione non è una piramide, però notiamo che comunque sopra vi è l'area direzionale) → l'area direzionale ha il compito di:

- pianificare e controllare dal punto di vista strategico;
- pianificare e controllare dal punto di vista direzionale.

Capiamo, allora, che se dovessimo mappare l'area direzionale sulla piramide vista precedentemente, allora l'area direzionale verrebbe mappata sulle due aree alte della piramide (ovvero quella strategica e quella direzionale).

- **Area istituzionale** → ci permette di gestire le funzionalità relative:
  - all'amministrazione;
  - alla gestione delle risorse umane;
  - agli altri processi di supporto all'amministrazione.
- **Area operativa** (relativa ad un'azienda manifatturiera, ovvero ad un'azienda che produce qualcosa → ovviamente anche le aziende di servizio hanno un'area operativa, in cui lo scopo sarà la produzione di servizi, piuttosto che la produzione di un prodotto materiale) → l'area operativa ci permette di gestire le funzionalità relative:
  - alla progettazione e industrializzazione;
  - l'approvvigionamento;
  - la fabbricazione;
  - la distribuzione;
  - il post-vendita;
  - altri processi primari.

Capiamo, allora, che l'area operativa gestisce tutta la linea/catena di attività, che ci permettono di produrre un prodotto, fino al servizio di post-vendita.

Prende il nome di **Portafoglio applicativo** di un'organizzazione, l'insieme delle applicazioni utili in azienda e ognuna delle tre aree appena viste sopra (quindi: Area direzionale, area istituzionale e area operativa) ha un portafoglio applicativo dedicato, ovvero: a seconda dell'area funzionale che stiamo considerando, avremo degli applicativi ad hoc che ci permettono di supportare le funzionalità (per esempio, per la parte amministrativa di gestione del personale, abbiamo l'applicativo che farà parte del portafoglio applicativo istituzionale, per la gestione degli stipendi). Quindi, il portafoglio applicativo è quell'insieme di applicazioni utili, che vengono utilizzate in azienda, e che ci permettono di dare supporto alle funzionalità di quella particolare area. Come abbiamo appena detto, ogni area ha un portafoglio applicativo dedicato e di conseguenza, quest'ultimo si suddivide in tre segmenti/parti principali:

- **Portafoglio direzionale** → comprende le applicazioni informatiche a supporto dei cicli di pianificazione strategica e di pianificazione e controllo delle risorse aziendali;
- **Portafoglio istituzionale** → comprende le applicazioni informatiche per i processi di supporto all'amministrazione, alla gestione delle risorse umane alla contabilità;
- **Portafoglio operativo** → comprende le applicazioni informatiche per i processi primari dell'azienda.



Normalmente si classifica il **portafoglio direzionale e il portafoglio istituzionale come orizzontali**, nel senso che sono generali e indipendenti dalle specifiche caratteristiche dei settori. Mentre si classifica il **portafoglio operativo come verticale**, nel senso che è tipico di ogni singolo settore.

Quindi, possiamo dire che:

- il **portafoglio istituzionale** comprende tutte le applicazioni utilizzate a supporto dei processi istituzionali, dove per processi istituzionali intendiamo:
  - i processi che eseguono adempimenti di legge, quali: contabilità, retribuzione e previdenza;

- i processi che supportano l'amministrazione di infrastrutture;
- i processi che supportano l'amministrazione di fattori di produzione, quali: personale, scorte, denaro e impianti.

Tutte queste attività sono orizzontali, cioè poco variabili rispetto al settore industriale in cui opera l'azienda.

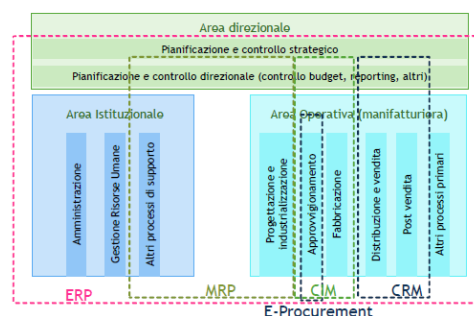
- il **portafoglio operativo** comprende tutte le applicazioni informatiche a supporto dei processi primari della catena del valore di Porter (se parliamo di un'azienda manifatturiera). **La catena del valore di Porter è una descrizione in ambito aziendale, che ci permette di descrivere le attività di un'azienda come un processo.** Quindi, attraverso la catena del valore di Porter, l'azienda viene vista come una successione di attività finalizzate a produrre valore per il cliente ed in particolare, la catena del valore per un'azienda manifatturiera è la seguente:



(questa catena del valore è specifica delle aziende manifatturiere, in quanto per le aziende che producono un servizio, la catena del valore cambia).

- il **portafoglio direzionale** comprende l'insieme di applicativi a supporto dei:
  - processi di pianificazione strategica;
  - processi di pianificazione e controllo delle risorse aziendali.

Guardiamo, infine, i diversi tipi di sistemi informativi che vanno a coprire le aree funzionali:



# Enterprise Resource Planning (ERP)

ERP indica una suite di modelli applicativi che supportano l'intera gamma dei processi di un'azienda. In particolare abbiamo che:

- **Enterprise** → con tale termine si va ad indicare il concetto di azienda;
- **Resource** → con il termine risorsa si va ad indicare varie tipologie di risorsa, quali: persone, materiali, impianti e capitale;
- **Planning** → con tale termine si va ad indicare: tempi, luoghi e quantità.

Il sistema ERP, quindi, è un sistema informatico operativo integrato, in cui si ha:

- una base dati unica;
- le procedure sono interagenti e cooperanti.

L'obiettivo primario di tale sistema è quello di **gestire in maniera ottimale tutte le risorse utilizzate nei processi di business aziendali**, l'accelerazione dei cicli operativi e la condivisione di informazione → nei sistemi ERP, quindi, l'accento è posto sulle attività comuni fondamentali dell'azienda, in quanto:

- asseconda l'azienda nei suoi processi di evoluzione o crescita;
- configurabile rispetto all'organizzazione aziendale.

Il sistema ERP è nato per le aziende manifatturiere, ma successivamente è stato applicato con successo in ambiti diversi, quali: servizi finanziari, grande e piccola distribuzione, trasporti e sanità. Da notare, che l'applicazione a settori diversi è stata resa possibile mediante:

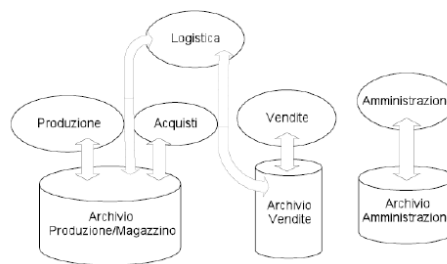
- la presenza di moduli indipendenti, ognuno dei quali copre una specifica area aziendale;
- struttura globale ERP unica, in cui cambiano i moduli di gestione del core-business aziendale.

## Caratteristiche sistemi ERP

I sistemi ERP sono sistemi che ci permettono di coprire tutte le funzionalità che abbiamo in azienda → in quanto, essendo il sistema ERP un sistema modulare a pacchetti, esso garantisce funzionalità che possono coprire tutte le esigenze

aziendali → proprio perchè è a pacchetti, il sistema ERP è estendibile e configurabile, però è un sistema che richiede un periodo di avviamento e un investimento corposo iniziale → quello, infatti, che tipicamente accade, è che un'azienda di dimensione medio-grandi vada ad acquistare il nucleo del sistema ERP, ovverosia uno o due pacchetti, e successivamente vada ad estenderlo.

Il grande vantaggio del sistema ERP è che quest'ultimo riesce a superare la situazione in cui si hanno isole informatiche autonome e specializzate, gestite con basi di dati diverse, che non ci permettono di garantire la qualità e la condivisione dell'informazione. Analizziamo la seguente immagine:

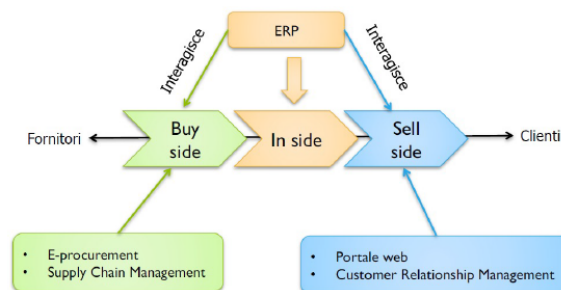


Vediamo che, con un sistema gestionale classico, le varie funzionalità che vengono utilizzate per la gestione della produzione, degli acquisti, della logistica, delle vendite e dell'amministrazione, fanno riferimento a basi di dati proprie, che in questo contesto prendono il nome di Legacy Systems → questo significa, che nel momento in cui dovessi, rispetto alle vendite, andare a cercare di gestire l'interazione con il cliente, il Legacy System mi impedisce di utilizzare il sistema che ho già e mi richiede di acquistare un nuovo sistema, che dovrà in un qualche modo combaciarsi con il sistema presente. Lo svantaggio principale di questa conformazione, è che abbiamo molteplici basi di dati diverse e quindi potremmo avere la stessa informazione memorizzata in formati diversi, schemi diversi e poi all'interno delle varie basi di dati le varie informazioni coincidono e questo naturalmente può generare un problema.

Quando, invece, si porta un sistema ERP in azienda possiamo riconoscere la catena del valore di Porter, in quanto nella catena del valore di Porter abbiamo quattro macro-processi:

- l'approvvigionamento;
- la produzione;
- la vendita;
- post-vendita.

L'affermazione appena sopra fatta, può essere meglio osservata con la seguente immagine:



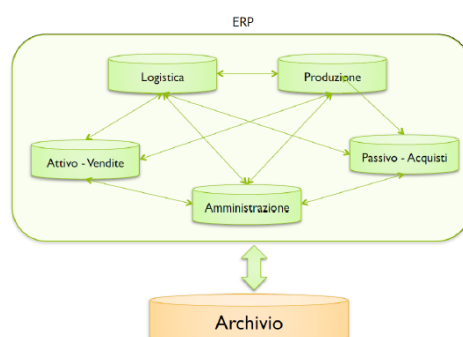
Abbiamo:

- la parte **Buy Side**, che si riferisce ai fornitori e quindi le funzionalità sono quelle relative all'acquisto delle materie prime;
- la parte **In Side**, dove il sistema ERP si focalizza sulle attività interne;
- la parte **Sell Side**, ovvero la parte di gestione dei clienti, servizi vendita e post-vendita.



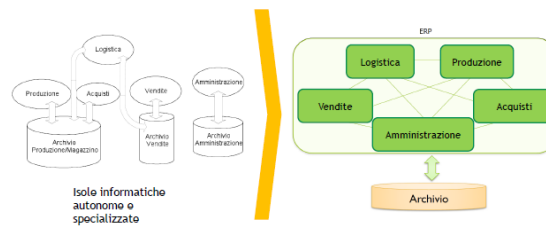
In quest'ultimo schema, si evidenzia che il sistema ERP si focalizza sulle attività interne, mentre per le attività esterne si interfaccia con altri sistemi.

Con i sistemi ERP, allora, abbiamo un'unica base dati "Archivio" (la quale va a sostituire le tre basi di dati che abbiamo visto precedentemente, quindi: Archivio Produzione/Magazzino, Archivio Vendite e Archivio Amministrazione) e possiamo notare, che abbiamo l'interazione delle varie funzionalità (ovverosia: Logistica, Produzione, Acquisti, Vendite e Amministrazione), che se notiamo sono tutte attività interne (quindi quelle che prima abbiamo identificato con la parte In Side) sono tutte gestite dal sistema ERP.



Quindi, abbiamo che prima dei sistemi ERP avevamo per la parte In Side isole informatiche autonome e specializzate, che spesso potevano non interfacciarsi tra loro, dato che l'interfacciamento non era previsto. Dopo l'avvento del sistema ERP, per la parte In Side abbiamo un'unica base di dati con la rappresentazione di tutte le funzionalità, che interagiscono tutte con l'unica base dati → questo è un notevole passo avanti, soprattutto dal punto di vista di gestione dell'informazione.

### ERP - Prima e dopo



## ERP - Moduli principali

I moduli principali del sistema ERP sono quelli che abbiamo individuato per la parte In Side, ovvero:

- **Amministrazione;**
- **Logistica;**
- **Vendite;**
- **Acquisti;**
- **Produzione.**



Da notare il fatto, che per ognuno di questi moduli si fa riferimento al processo a cui il modulo è di supporto. Quindi, ad esempio, il processo di contabilità generale è un processo amministrativo. Vediamo, allora che anche nella descrizione dei moduli del sistema informativo, si fa riferimento ai processi.

Con la frase appena sopra scritta, vogliamo dire che se io ho un sistema informativo lo uso a supporto di cosa? E siccome abbiamo detto, che il modulo decisionale (non sono sicuro sia il modulo gerarchico → registrazione 12.10 minuto 8) dell'azienda viene rappresentato con mezzo dei processi, allora io racconto qual è il processo che viene supportato dal mio sistema informativo. Il processo è quell'insieme di

attività, che poi noi utilizziamo per mandare a compimento quello che dobbiamo fare, che sia raggiungere un obiettivo o un determinato compito → questo lo si può vedere anche nelle tre parti in cui era suddivisa la piramide di Anthony, dove abbiamo detto che i tre livelli gerarchici (che poi diventavano livelli decisionali) erano anche descritti a processo. Quindi, tutti questi tre livelli (i quali ricordiamo sono stati identificati come livelli gerarchici), ovvero:

- livello operativo;
- livello direzionale;
- livello strategico.

richiedono di prendere delle decisioni, in particolar modo richiedono di prendere rispettivamente:

- decisioni operative;
- decisioni direzionali;
- decisioni strategiche.

Dietro a queste decisioni vi sono dei processi:

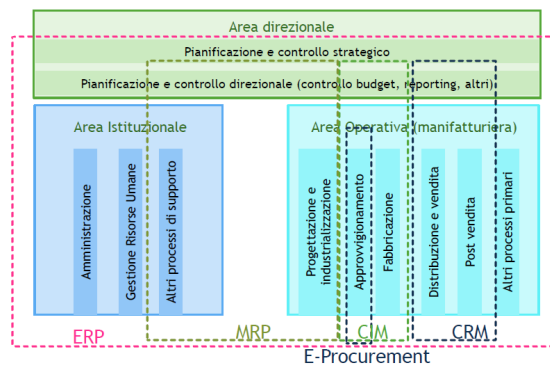
- processi operativi → finalizzati a produrre qualcosa ed eventualmente per prendere decisioni relative alla parte operativa;
- processi direzionali;
- processi strategici.

Tutto questo per dire, che analizzeremo i processi per vedere in che modo raccontare quello che abbiamo bisogno di fare e dall'altro lato, andiamo a vedere in che modo i sistemi informativi possono essere di supporto alla gestione dei processi e dell'informazione veicolata, prodotta ed utilizzata dai processi. Analizziamo brevemente, a questo punto, i moduli principali del sistema ERP:

- **Amministrazione** → il modulo relativo all'amministrazione è il modulo organizzato attorno alle procedure di contabilità generale. Quindi è il modulo che ci permette di gestire la parte di contabilità generale. Le funzioni principali di tale modulo sono essenzialmente due:
  - supporto delle attività amministrative, quindi transazioni elementari, come ad esempio: fatture, pagamenti e incassi;
  - produzione di informazioni di sintesi sull'andamento aziendale.



Da notare, che se ci ricordiamo il grafico delle aree funzionali dell'azienda, il modulo amministrazione lo possiamo collocare nell'area istituzionale (in basso a sinistra) → quindi, in questi moduli messi a disposizione dal processo ERP, andremo a riconoscere le funzionalità già citate nel portafoglio applicativo.



- **Logistica** → modulo relativo ai processi dedicati alla gestione dei materiali (quali ad esempio: materie prime, materiali di funzionamento, prodotti finiti) e in particolar modo, nei sistemi ERP l'accento è posto sulla gestione interna della logistica e non sul rapporto con i clienti e fornitori. Guardando, come prima, il grafico delle aree funzionali dell'azienda, il modulo relativo alla logistica si colloca nell'area operativa ed essendo che si tratta di un modulo relativo alla gestione dei materiali, sarà certamente un modulo utile nelle aziende manifatturiere;
- **Vendite** → modulo relativo alla gestione dei processi che permettono all'azienda di interagire con il cliente e di conseguenza anche questo modulo si colloca nell'area operativa del grafico delle aree funzionali dell'azienda ed in particolar modo nella parte relativa alla vendita e post-vendita → da notare, che per quanto riguarda la parte di vendita, non dobbiamo pensare solamente alla vendita di un prodotto, ma questo vale anche per la vendita di un servizio. Questo per dire, che questo modulo relativo alle vendite, può essere anche riferito alla vendita di un oggetto non materiale. Per quanto riguarda le vendite, le macro-attività che vengono gestite sono:
  - ricezione degli ordini → prima di accettare un ordine, si devono fare delle valutazioni sul cliente stesso, come ad esempio la solvibilità;
  - elaborazione degli ordini → si deve effettuare una valutazione della priorità e della necessità di attivare altri processi. Richiede la coordinazione con altri flussi;

- evasione degli ordini ed emissione di documenti di trasporto → tale attività interagisce con il magazzino;
- fatturazione → tale attività interagisce con la contabilità.



Vediamo che queste quattro macro-attività riguardano la parte finale dell'iterazione con il cliente e quindi non ci preoccupiamo degli aspetti prima della vendita, in quanto riguarda la parte produttiva.

- **Acquisti** → modulo relativo alla gestione dei processi per l'interazione con i fornitori per l'approvvigionamento di:
  - materiali;
  - lavorazioni o servizi esterni.

Il modulo degli acquisti è strettamente collegato con i moduli relativi alla logistica e alla produzione e le principali funzioni svolte da tale modulo sono di gestione:

- delle condizioni commerciali (listini, sconti);
- delle procedure di rapporto con i fornitori.

Anche per quanto riguarda il modulo degli acquisti abbiamo la gestione degli ordini, la quale si suddivide in:

- raccolta delle richieste, le quali possono provenire dalla produzione oppure da parte degli altri reparti;
- emissione degli ordini ai fornitori;
- ricezione dei documenti di consegna ed evasione degli ordini;
- ricezione delle fatture dai fornitori.

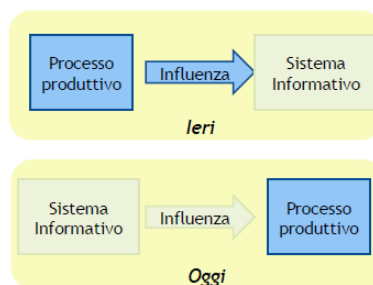


- **Produzione** → per quanto riguarda la produzione, è importante sottolineare il fatto, che nel passato il processo produttivo influenzava il sistema informativo, ovvero il sistema informativo nasceva perché il processo produttivo aveva bisogno di essere supportato. Al giorno d'oggi, abbiamo il contrario, ovvero

che il sistema informativo influenza il processo produttivo e questo è stato reso possibile grazie a:

- l'avvento della tecnologia → l'avvento della tecnologia fa sì che il processo produttivo (e qualsiasi altro processo all'interno dell'azienda) possa essere gestito in maniera diversa

È il risultato di questa idea anche l'ottimizzazione dei processi, ovverosia se io ho un sistema informativo che mi dà supporto nelle mie attività quotidiane, è chiaro che il mio sistema informativo influenza il processo. Questo discorso vale anche per l'organizzazione, ovvero prima l'organizzazione influenzava il sistema informativo, mentre adesso il sistema informativo influenza l'organizzazione → questo ci fa capire, che il sistema informativo diventa sempre più una parte importante (che ha cioè ricadute) su tutta l'azienda.



Capiamo allora, che non esiste una soluzione unica di modulo di produzione per tutte le produzioni, ma vi sono diverse possibilità, che dipendono:

- dal settore merceologico;
- dalle modalità produttive adottate.

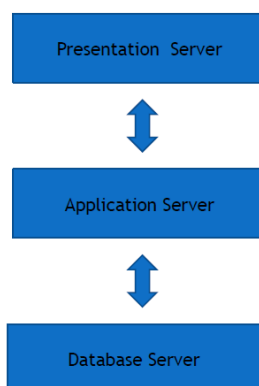
Da ricordarci anche, che quando parliamo di sistema informativo parliamo di gestione dell'informazione e la locuzione (ovverosia il termine) "sistema informativo" la utilizziamo come sinonimo di sistema informatico.

## **SAP (Sistemi, Applicazioni e Prodotti nell'elaborazione dati)**

Il leader per quanto riguarda i sistemi ERP è SAP → SAP AG è una multinazionale europea (fondata nel 1972 in Germania) per la produzione di software. È una delle principali aziende al mondo nel settore degli ERP e in generale nelle soluzioni Enterprise. Nel 1980 SAP crea il linguaggio di programmazione proprietario ABAP, la cui semantica ricalcava molto quella di altri linguaggi dell'epoca (Fortran/COBOL). Nella prima versione per mainframe e la realizzazione dell'applicazione commerciale R/2 il linguaggio rilasciato agli

utenti consentiva una veloce interrogazione dei dati, ovvero l'applicazione R/2 permetteva di ottenere rapporti scegliendo i dati finali da elencare. In seguito, con l'avvento dell'ERP e la tecnologia client-server applicata a SAP R/3 il linguaggio di programmazione si è evoluto, consentendo di interfacciarsi al linguaggio SQL dei maggiori database relazionali.

L'architettura di SAP è composta da tre layer:



Abbiamo una parte di database, ovverosia il **Database Server**, l'**Application Server** e il **Presentation Server**. L'architettura SAP, quindi, si basa sull'utilizzo di una base dati unificata per tutti i pacchetti che andiamo ad utilizzare. I pacchetti che andiamo ad utilizzare sono molteplici, come ad esempio:

- Warehouse → sistema di supporto alle decisioni;
- Sistema per la gestione di progetti e di tutte le attività finanziarie;
- Vendita e distribuzione → colloca il prodotto/servizio sul mercato e comprende la vendita, spedizione, trasporti, commercio estero e fatturazione;
- Gestione dei materiali → effettua la movimentazione dei materiali e quindi comprende la pianificazione dei consumi, acquisti, controllo delle fatture, gestione degli stock, gestione del magazzino;
- Gestione della produzione → combina i fattori produttivi e li trasforma in beni e servizi;
- Gestione delle qualità → individua le variazioni della qualità e quindi comprende la pianificazione, controllo, ispezione, avvisi e certificati;
- Gestione delle risorse umane → gestisce e amministra il personale e quindi comprende la pianificazione, reclutamento, formazione, gestione dei turni e calcolo delle retribuzioni;

- Contabilità → rileva, elabora e comunica agli stakeholders i dati economici e quantitativi dell'azienda e quindi comprende la contabilità generale, bilanci, contabilità clienti e contabilità fornitori;
- Gestione cespiti → gestisce i valori materiali e immateriali a utilità pluriennale;
- Workflow → gestisce i flussi di lavoro e di conseguenza comprende la gestione delle autorizzazioni, la gestione dei livelli di accesso e la gestione delle password.

Quindi, possiamo immaginare di avere a disposizione pacchetti per tutte le funzionalità → naturalmente questa è una cosa molto utile ma costosa, in quanto è vero che va a coprire tutte le mie esigenze, però richiede uno sforzo economico molto elevato ed inoltre richiede che all'interno dell'azienda vi siano delle professionalità in grado di utilizzare questo sistema.

---

## **Sistemi CRM (Customer Relationship Management)**

I sistemi CRM ci consentono di gestire la relazione con la clientela e da notare, che i sistemi ERP hanno pacchetti CRM, però vi sono anche applicativi indipendenti al sistema ERP, che forniscono alle aziende sistemi CRM per il supporto alla clientela. Di fatto, tutti i sistemi informativi che noi possiamo andare a pensare, che coprono le funzionalità del nostro portafoglio applicativo, hanno una corrispondenza nel pacchetto ERP. Analizziamo meglio i sistemi CRM: se ci ricordiamo la catena del valore di Porter, avevamo che la gestione della vendita, post-vendita e clientela era in fondo. Questa attività, però, diventa sempre più importante, anche perchè riesce a sfruttare nuovi canali di comunicazione → una volta, la gestione della clientela era limitata al fatto, che il cliente comprava qualcosa dal mio negozio e se il cliente era o non era soddisfatto del prodotto, ripassava in negozio o al massimo vi era il numero da contattare. Al giorno d'oggi, recarsi al negozio per riportare e/o ricomprare qualcosa non si fa più, bensì vi sono dei nuovi canali basati sulle tecnologie informatiche → ad esempio, Amazon ha un sistema CRM e si occupa in maniera molto attenta della clientela → quindi, l'obiettivo principali dei sistemi CRM è di aumentare la soddisfazione del cliente, ovvero dare al cliente la sensazione che la sua qualità, dal punto di vista di gestione, sia un aspetto fondamentale. Per fare questo, il sistema CRM utilizza tutti i canali di comunicazione a disposizione, quali:

- chat-box;
- email;
- chat Whatsapp;

- telefono → da notare, che il canale telefonico per un periodo di tempo è stato soppiantato, in quanto sembrava obsoleto, ma successivamente ci si è reso conto che è un canale di comunicazione importante.

Il sistema CRM si propone di offrire un servizio completo al cliente, dal primo contatto fino al post-vendita, in modo tale da costruire relazioni personalizzate di lunga durata con il cliente, capaci di:

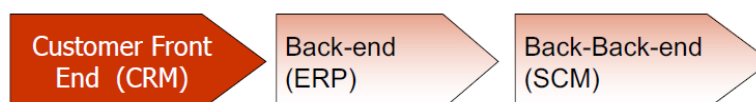
- aumentare la soddisfazione dei clienti;
- aumentare il valore dell'impresa per il cliente e del cliente per l'impresa.

**Nei sistemi CRM, quindi, il cliente rappresenta l'elemento centrale della strategia commerciale.** Una parte importante del sistema CRM è quella di raccogliere ed analizzare i reclami, le osservazioni e tutti i dati relativi ai clienti, in quanto se un sistema CRM funziona correttamente, allora esso sa:

- chi siamo;
- cosa facciamo;
- in che modo interagiamo;
- cosa ci piace e cosa non ci piace.

Tutto questo lo riesce a fare attraverso un **sistema di Data Warehouse**.

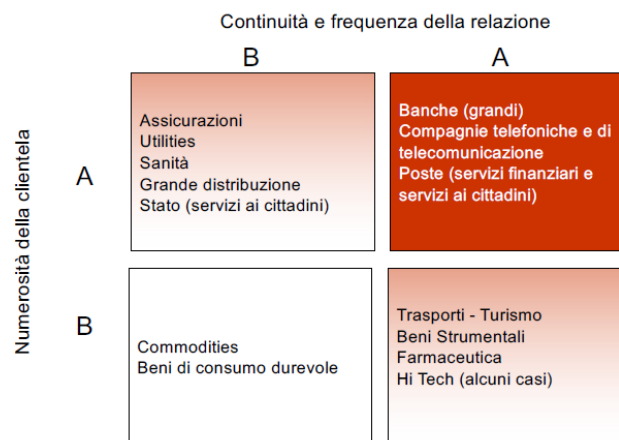
La catena di servizio dei sistemi CRM è la seguente:



Parte da un sistema di Customer Front End, il quale a sua volta si appoggia su un sistema di Back-End (nel sistema Back-End, quindi nella gestione interna, si cita nuovamente il sistema ERP) ed infine abbiamo un sistema di Back-Back-End, che è quello che ci collega con i fornitori → questa è esattamente la catena del valore di Porter girata al contrario. Capiamo, allora, che le strategie di CRM sono infattibili senza sistemi CRM e in generale senza un sistema integrato, in cui:

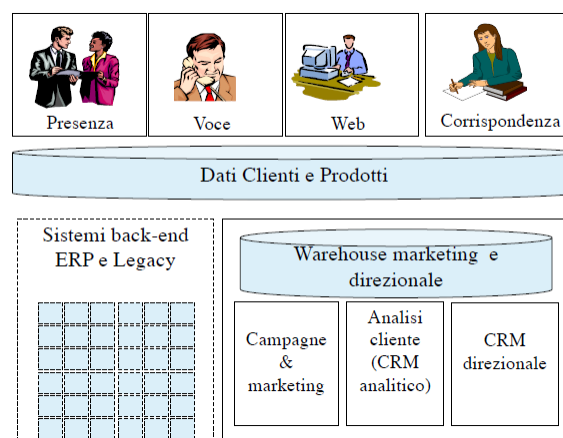
- i sistemi CRM formano il front-end delle catene di servizio;
- i sistemi ERP supportano l'esecuzione del servizio;
- i sistemi SCM (Supply Chain Management) collegano con i fornitori.

I sistemi CRM iniziano negli anni '90 e prendono sempre più piede grazie all'avvento di Internet. Essi sono molto importanti, in quanto la gestione del cliente è fondamentale per qualsiasi tipo di azienda di qualsiasi dimensione. Nella seguente griglia, vediamo un'analisi che ci permette di capire quanto più è importante avere a disposizione un sistema CRM tanto più aumenta la numerosità della clientela e tanto più aumenta la continuità e la frequenza della relazione con il cliente:



Possiamo notare, che l'insieme di aziende che possono trarre maggior vantaggio dall'avere un sistema CRM sono le aziende, che hanno una clientela molto numerosa (e quindi sono grandi aziende, come banche, poste o compagnie telefoniche) e che hanno un'alta frequenza e continuità delle relazioni con i clienti.

Lo schema dei sistemi CRM ci dà un'idea su quello che è necessario fare:



Possiamo osservare, che l'interazione con il cliente viene fatta per mezzo di canali diversi e naturalmente più canali si mettono a disposizione, più clienti diversi si riescono a raggiungere, in quanto si riescono ad accontentare le esigenze di persone diverse, ovverosia si riesce ad accontentare le diverse metodologie con cui

le persone vogliono contattare l'azienda. Per esempio, i nostri genitori preferiscono contattare l'azienda tramite il telefono o andando in presenza, mentre noi preferiamo utilizzare il Web → notiamo, che il metodo con cui le persone contattano l'azienda varia molto in base all'età e quindi entra in gioco, il fatto che l'azienda deve conoscere i clienti con cui entra in contatto e quindi vi è il discorso della **profilazione dei clienti** → per fare la profilazione e sapere con chi ho a che fare, devo raccogliere i dati. Tramite il Web è naturale che l'azienda riesca a raccogliere dati, ma vi è anche da notare, che anche tramite il telefono l'azienda riesce a raccogliere dati; basti pensare a quando chiamiamo l'assistenza di TIM, perchè non funziona il router. Quando chiamiamo per fare la segnalazione, il dipendente di TIM rileva:

- che effettivamente abbiamo chiamato;
- da dove chiamiamo;
- perchè chiamiamo;
- se sono stato soddisfatto della risposta.

Vi è, quindi, una base dati che contiene tutti i dati dei clienti, le loro iterazioni, che cosa acquistano e dei servizi che usufruiscono → questi dati dei clienti, naturalmente poi finiscono in un **Data Warehouse**, che permette di fare **analisi** → l'analisi mi permette di fare attività di marketing, campagne promozionali e per prendere decisioni strategiche.

I sistemi CRM, quindi, sono fondamentali per:

- Banche;
- Poste;
- Compagnie telefoniche e di telecomunicazioni;
- **Aziende di utility** → sono quelle aziende che forniscono gas, energia, acqua, telefono e simili, che offrono quindi un servizio → notiamo il fatto, che queste aziende forniscono un servizio duraturo e continuativo nel tempo (e non prodotti non durevoli, come ad esempio una maglietta o un mobile). In questo contesto, in cui il rapporto con i clienti è continuativo, il modo di interagire e di mettere a disposizione le informazioni diventa di fondamentale importanza ed ovviamente l'avvento e lo sviluppo di canali virtuali, come il Web e i Call center, hanno potenziato questo sistema. Quindi, i fattori che spingono le aziende di utility ad uno sviluppo massiccio di sistemi a supporto dei contatti con il cliente sono:
  - clientela molto numerosa;
  - rapporto continuativo nel tempo;



- sviluppo di canali virtuali (web, call center);
- concorrenza crescente;
- necessità di sviluppare e mantenere un'offerta personalizzata, in modo tale da non fare sentire i clienti come dei numeri, ma delle persone ben volute e "importanti" per l'azienda.

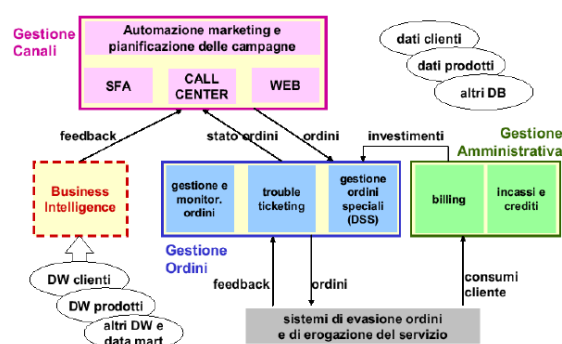
I vantaggi per le aziende di utility ad adottare questi meccanismi sono:

- abbattimento dei costi di relazione;
- grandissimo impatto sui profitti.

Le funzioni dei sistemi CRM sono le seguenti:

- inizialmente erano nati come sistemi di gestione degli ordini;
- successivamente si sono evoluti come sistemi di automazione delle forze di vendita e come sistemi di assistenza telefonica;
- infine come sistemi di assistenza al cliente con tutti i canali di comunicazione disponibili.

I moduli applicativi di un sistema CRM sono molteplici, per esempio abbiamo la parte di gestione dei canali di comunicazione, la parte di gestione degli ordini e la gestione amministrativa. Quello che però interessa a noi, è la parte di analisi relativa a tutte le informazioni del cliente e quindi parliamo di Data Warehouse dei clienti, DW dei prodotti ed altri DW e data smart.



E vediamo che parliamo di Business Intelligence, ovverosia l'analisi delle informazioni che ho raccolto e che ho a disposizione. Questa analisi serve per prendere delle decisioni, dato che abbiamo detto che il DW è un sistema di supporto alle decisioni ed informativo, che mi permette di prendere delle decisioni non di routine. Cosa significa questo? Significa, che il DW non mi serve, ad esempio, per

decidere quante matite acquistare a fine anno, bensì mi serve per decidere se cambiare linea di marketing o di pubblicità → quindi il fulcro del modulo del sistema CRM è il Data Warehouse, perchè è con quest'ultimo che riesco ad analizzare i dati.

Quindi, nel sistema CRM diventano di fondamentale importanza gli strumenti di Business Intelligence, ovverosia applicazioni che estraggono informazioni dai sistemi di supporto alla produzione per elaborarle e fornire supporto all'apparato decisionale. Si occupano di passare dai dati ad informazioni sintetiche e rispondono a richieste complesse. Inoltre, quando parliamo di sistema CRM e di Business Intelligence, dobbiamo fare una distinzione tra:

- **Sistemi OLTP (On Line Transaction Processing)** → quindi sistemi di supporto operativo, in cui si hanno molte transazioni provenienti da molti utenti e gestiti tramite un DBMS;
- **Sistemi OLAP (On Line Analytical Processing)** → quindi sistemi di supporto decisionale, in cui abbiamo scritture trascurabili, ma molte letture. Questo perchè, periodicamente prendo tutti i dati che ho a disposizione e li travaso nel DW e quindi non ho la scrittura relativa alla singola transazione → con questi travasi NON vado a fare degli aggiornamenti che vanno a sovrascrivere, bensì faccio degli append, dato che nel DW ci deve essere lo storico dei dati → è fondamentale che ci sia lo storico nel DW, perchè altrimenti non riuscirei a fare l'analisi dei dati, ovverosia non riuscirei ad aggregare i dati per ottenere delle informazioni interessanti.

---

## Portafoglio operativo delle società di servizi

Le società di servizi, che producono un servizio ed utilizzano come risorsa principale le **informazioni**, operano nel:

- settore bancario;
- settore assicurativo;
- servizi finanziari.

Esse sono caratterizzate da alta intensità informativa sia dei processi produttivi che dei prodotti-servizi → questo fa cambiare la catena del valore di Porter, perchè non ho più:

- compro le materie prime;
- produco il prodotto;
- vendo il prodotto ai clienti;

- fornisco supporto alla vendita ai clienti.

In questo caso, la catena del valore di Porter cambia, in quanto il processo produttivo diventa:

- predisposizione delle condizioni produttive → attività volte a creare la capacità produttiva per erogare il servizio, anche con la stipulazione contratti con altri operatori (acquisto e apertura nuove filiali);
- attività di back office → attività di erogazione del servizio che sono svolte in seguito di un ordine ma non in presenza del cliente (valutazione rischio polizza assicurativa). Attività molto importante in questo caso, in quanto permette di capire cosa proporre al cliente e quindi è personalizzata (es: una società di assicurazione farà una polizza assicurativa per me e una polizza assicurativa per i miei genitori, in quanto loro hanno più esperienza alla guida e hanno un'auto migliore della mia).

Il processo distributivo, invece, diventa:

- attività di front office → attività di erogazione del servizio svolte in presenza del cliente (attività di sportello banca).
- procacciamento della clientela → attività che portano alla stipulazione dei contratti di servizio o all'ordine da parte dei clienti.

La catena del valore, quindi, è la seguente:



Chiaramente avendo un processo informativo, la gestione dell'informazione diventa preponderante e di conseguenza il sistema informativo diventa ancora più importante, perchè ricordiamo che il sistema informativo è quella parte dell'organizzazione, che ci permette di gestire l'informazione → da ricordare, che abbiamo detto che l'informazione porta conoscenza e la capacità elaborativa dipende anche dalla conoscenza che abbiamo ed in questo contesto, il **knowledge**

**worker**, ovverosia l'impiegato che svolge lavoro intellettuale di natura decisionale (l'esempio classico di knowledge worker è quello che fa la valutazione dei rischi in assicurazione), dove usa sicuramente il sistema di supporto alle decisioni e il sistema informativo, ma mette in campo anche le sue competenze personali. Tipico delle società di servizi con elevato grado di personalizzazione o di trattative con il cliente e di conseguenza, il knowledge worker fornisce servizi con elevato grado di personalizzazione, fornendo:

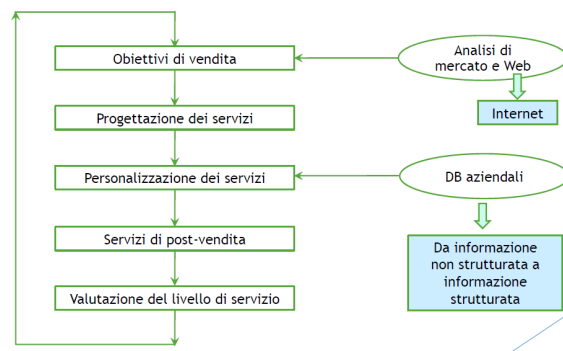
- supporto dell'informatica all'interscambio di consulenza;
- supporto alla comunicazione fra collaboratori.

La gestione della conoscenza è vitale nelle società di servizi, ovverosia la gestione della conoscenza diventa un vantaggio competitivo. Ovviamente gestire la conoscenza non è banale, in quanto gestire la conoscenza significa anche riuscire a produrre conoscenza dalle informazioni che si hanno a disposizione → la valutazione dei rischi fa riferimento proprio a questa cosa, ovvero produrre conoscenza rispetto alle informazioni che si hanno e non sbagliare a proporre un determinato servizio. In questo contesto, l'informazione non è strutturata, perchè è vero che posso fare riferimento alle informazioni che ho nei miei DB e DW, ma tanta di questa informazione è la competenza dell'individuo che fa la valutazione dei rischi, il quale gode di competenze e conoscenze → questo ci fa capire che l'individuo, quando arriva in azienda, grazie alle sue conoscenze porta un vantaggio competitivo all'azienda e quando se ne va, potrebbe portare via con se anche il vantaggio competitivo, dato che potrebbe essere l'unico (o uno dei pochi) individui ad avere quelle conoscenze.

Infine ci chiediamo: Cosa vuol dire gestire la conoscenza? Gestire la conoscenza va di pari passo con la gestione delle informazioni, ovvero la gestione della conoscenza è parte della gestione delle informazioni. Quindi esse sono complementari e molto spesso si sovrappongono. Quando noi ci diamo degli obiettivi di vendita (es: voglio acquisire una certa fetta di mercato) valuto di progettare dei servizi, in modo tale da allargare la "platea" dei miei clienti. Ovviamente, dopo la progettazione dei servizi, se io sono un'azienda che fornisce servizi, devo anche pensare alla personalizzazione dei servizi, in quanto come abbiamo detto è una parte molto importante, dato che:

- aumenta la percezione di qualità da parte del cliente, soddisfacendo le esigenze del cliente;
- garantisce il bilanciamento tra la soddisfazione del cliente e il mio margine dei rischi (es: per noi giovani la polizza assicurativa sarà più alta rispetto ad una persona che guida da molto più tempo di noi).

Una volta personalizzati i servizi, dobbiamo implementare i servizi di post-vendita, anche in ottima di migliorare la valutazione del livello di servizio da parte dei clienti.



Per quanto riguarda il sistema informativo, abbiamo che per le decisioni strategiche e gli obiettivi di vendita ci rifacciamo alle analisi di mercato e al Web (quindi ci rifacciamo a dati esterni), mentre per quanto riguarda la personalizzazione dei servizi (quindi stiamo interagendo con clienti che già conosciamo), ci rifacciamo ai DB aziendali

# Ingegneria dei processi gestionali

Innanzitutto diciamo che cos'è un processo: **Un processo è qualcosa che ci permette di rappresentare il modo di operare di un'azienda.** Nel momento in cui (come effettivamente accade nella realtà), tecnologia e il modo di operare di un'azienda diventano un tutt'uno e quindi la parte tecnologica impatta sul modo di operare dell'azienda, abbiamo bisogno di progettare, gestire e re-ingegnerizzare queste due componenti in maniera integrata → la progettazione delle attività, quindi, deve tenere conto della tecnologia e viceversa.

I processi si classificano in tre macro-categorie:

- **Materiali** → essi sono flussi di materiali e attività, quindi sono tutti quei processi in cui abbiamo un input fisico o materiale e un output fisico o materiale → intendiamo, quindi, tutti i processi di produzione di un qualche materiale;
- **Informativi** → essi sono flussi di informazioni, quindi sia l'input sia l'output sono informazioni → tutti i processi, in cui nell'azienda si prendono scelte strategiche e/o decisionali sono processi informativi. Il processo, quindi, prevede di prendere delle decisioni basandoci sull'analisi delle informazioni;
- **Business Process (processo aziendale)** → è un insieme **coordinato** di attività, finalizzato alla realizzazione di un ben definito risultato di interesse per l'organizzazione → da notare, che questa definizione di processo aziendale è più generale rispetto alle definizioni delle altre due macro-categorie di processo. Questo perchè, un processo materiale può essere un business process, se il risultato d'interesse è produrre un materiale, ma il processo aziendale, inoltre, può essere informativo se lo scopo è prendere una decisione basata sull'analisi delle informazioni → **il processo aziendale, quindi, racchiude le altre due macro-categorie di processo** e per questo motivo, da adesso parleremo in generale di processi aziendali e a seconda dell'input che hanno, sappiamo che si tratta di processi materiali oppure di processi informativi.



Quindi, non dobbiamo pensare che il processo sia una sequenza di attività (o meglio dire, nel caso più semplice è una sequenza), bensì è un insieme coordinato di attività → È bene che, il processo aziendale, non sia sempre una sequenza, così è possibile eseguire molteplici attività in parallelo e di conseguenza si riesce a minimizzare il tempo (se, invece, il processo fosse una sequenza, il tempo necessario per il processo è la somma dei vari tempi necessari per svolgere ogni singola attività).

Abbiamo detto, che in generale il processo aziendale è formato da **attività**, realizzate come processi materiali oppure processi informativi. Quindi, in un processo aziendale le attività possono avere in input materiali oppure informazioni, a seconda dell'obiettivo che si siamo posti → in realtà, anche quando le attività sono materiali, esse portano con sé informazioni e di conseguenza, in un processo aziendale avremo sempre delle informazioni da portarci dietro.

Quando parliamo dei processi, quindi, abbiamo:

- un **input** → il quale può essere materiale oppure informativo;
- un **output** → il quale può essere un prodotto, un servizio oppure una decisione;
- **qualcuno** (un beneficiario) che usufruisce dell'output → parliamo, allora, di **clienti**. Da notare, che non indichiamo solamente il cliente finale, bensì pensiamo, ad esempio, al caso di un processo interno che produce un semi-lavorato, il quale poi viene passato ad un altro processo per continuare la produzione, allora possiamo avere che il cliente può essere un cliente interno → la parola cliente, quindi, sta solo a significare chi beneficia del risultato (ovverosia dell'output) del processo.

Se volessimo rappresentarlo in maniera più formale, possiamo dire che il processo aziendale è una tupla:

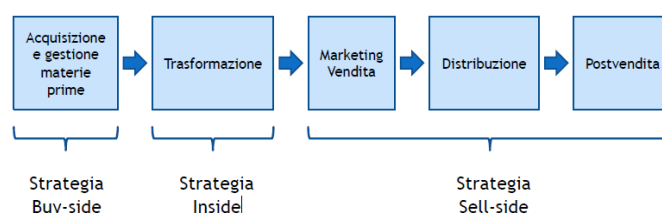
$$BP = (A, I, O, C)$$

dove:

- A = attività → serie di operazioni fisiche o di decisioni manageriali eseguite in modo coordinato;
- I = input → materie prime o risorse aziendali (informazioni se è un processo decisionale);

- O = output → oggetti fisici, beni immateriali, servizi;
- C = clienti → destinatari dell'output del processo.

Ritornando alla Catena del valore di Porter, a questo punto diventa chiaro del perchè, abbiamo detto che essa rappresenta il modo di operare di un'azienda, in cui l'azienda viene vista secondo una prospettiva di processo, ovvero: l'azienda è vista come una successione di attività finalizzate a produrre valore per il cliente ed il valore è misurato dal prezzo, che il cliente è disposto a pagare per il prodotto o servizio ricevuto. La forma canonica della Catena del valore di Porter (per le aziende manifatturiere) è:



Acquisizione, trasformazione, marketing, distribuzione e post-vendita le posso considerare le mie macro-attività (in quanto posso ulteriormente dettagliarle in altre attività) ed in particolare, possiamo pensare che ognuna di queste macro-attività, che difatti descrivono tutto ciò che viene fatto in un'azienda manifatturiera per produrre qualcosa, è a sua volta rappresentabile come un processo. Quindi:

- la macro-attività di Acquisizione e gestione delle materie prime, sarà un processo aziendale con una serie di attività coordinate e organizzate, che ci permettono di raggiungere l'obiettivo di acquisire e gestire le materie prime;
- la macro-attività di Trasformazione sarà formata da uno o più processi aziendali, che prenderanno in input le materie prime e produrranno in output il prodotto desiderato;
- la macro-attività di Marketing e vendita sarà formata da uno o più processi aziendali per la gestione della vendita, che prendono in input il prodotto e producono in output le tattiche di collocamento del prodotto sul mercato cliente e vendita del prodotto → in questo caso, il cliente è effettivamente il cliente finale.

(Stessa cosa poi per quanto riguarda la Distribuzione e il Post-vendita).

Vediamo, inoltre, che a livello di modellazione di processi e poi reingegnerizzazione, vediamo che possiamo avere una strategia di reingegnerizzazione orientata alla parte:



- **Buy-side** per quanto riguarda l'Acquisizione e la gestione delle materie prime → ricordiamo, che la strategia Buy-side è mirata all'interazione con il mondo dei fornitori e quindi ha lo scopo di trasformare il processo nelle varie fasi di ricerca del bene e del fornitore, definizione di prezzi e condizioni, ordinazione, ricezione dei beni e servizi ordinati. I benefici potenziali sono la riduzione del costo di transizione (in quanto si riduce il tempo necessario per svolgere la transizione) e del costo del materiale stesso. La trasformazione dei processi, per quanto riguarda la strategia Buy-side, si appoggia a:
  - sistemi di e-procurement;
  - sistemi che supportano i mercati elettronici;
  - reti internazionali di impresa (Business to Business);
  - infrastrutture Internet.
- **Inside** per quando riguarda la Trasformazione → la strategia In-side è mirata alla trasformazione dei processi interni all'impresa e i benefici potenziali (legati alla competitività generale e alla sopravvivenza dell'impresa) sono:
  - riduzione dei costi di funzionamento e della durata dei processi;
  - miglioramento della qualità e del servizio al cliente.

La trasformazione dei processi si appoggia a:

- sistemi ERP;
- infrastrutture Internet.
- **Sell-side** per quanto riguarda il Marketing e vendita, Distribuzione e Post-vendita → la strategia Sell-side è orientata ai processi di:
  - Vendita;
  - Marketing;
  - Distribuzione dei prodotti;
  - Servizio post-vendita e assistenza al cliente.

I benefici potenziali sono:

- maggiore valore del prodotto percepito dal cliente;
- abbattimenti dei costi di transazione.

Per avere tali benefici, la trasformazione dei processi si appoggia a:

- sistemi CRM (Customer Relationship Management);
- infrastrutture Internet.

---

Vediamo, adesso, un'altra classificazione dei processi aziendali (ricordando, sempre, che il processo è un insieme coordinato di attività che ci permettono di raggiungere un determinato obiettivo):

- Processi **intersettoriali** → processi generici che descrivono le pratiche di molteplici settori. Alcuni esempi sono: gestione delle materie, acquisto materie;
- Processi **settoriali** → esattamente l'opposto dei processi intersettoriali, ovvero distinguono i vari settori, come ad esempio: settore sanitario, settore manifatturiero;
- Processi **aziendali** → accezione differente rispetto a quella che abbiamo visto precedentemente, dato che in questo caso ci riferiamo a processi di una specifica azienda o di una sua parte. Tali processi aziendali, quindi, sono una specializzazione dei processi settoriali;
- Processi **normativi e best practice** → i processi normativi sono i processi di riferimento e le best practice derivano dal confronto tra processi diversi e descrivono come sono o dovrebbero essere nelle migliori aziende del settore.

Inoltre, se vogliamo prendere un processo e dettagliarlo, quindi andare dalla rappresentazione più astratta alla singola operazione (chiaramente il livello di dettaglio dipende dalle mie esigenze di progetto), i livelli che abbiamo a disposizione sono:

- **Macro-processo** → ovverosia descrivo un processo ad altissimo livello e successivamente lo posso andare a dettagliare. Un esempio di macro-processo sono le macro-attività della Catena del valore di Porter;
- **Processo** → livello in cui vado a descrivere le attività. Ad esempio, quindi, nella prima macro-attività della Catena del valore di Porter (Acquisizione e gestione delle materie prime) abbiamo le attività di: richiesta preventivo, valutazione preventivo, effettuamento ordine, ricezione fattura e pagamento;
- **Fase** → il processo può ulteriormente essere scomposto in fasi, le quali illustrano il modo in cui il processo è implementato e di conseguenza una fase è una tappa di un progetto;
- **Attività** → livello minimo di analisi normalmente adottato nello studio dei processi.. Le attività sono determinate scomponendo i processi secondo una logica sequenziale e sono operazioni fatte da singoli o da pochi;

- **Operazione** → passi elementari per eseguire una certa attività e quasi mai usate.

Ciascuno di questi livelli di processo può essere descritto tramite diagrammi di flusso, diagrammi gerarchici e tabelle di proprietà.



Se voglio modellare un processo che sia eseguibile (ovvero che qualcuno lo prenda in mano e lo riesca ad eseguire) devo adottare generalmente un livello di dettaglio pari a quello delle operazioni (in questo modo sono sicuro che chiunque riesca ad eseguire il processo), ma comunque dipende da per chi (per quale soggetto) sto modellando il processo.

La **griglia metodologica** è uno strumento, che ci permette di andare a reingegnerizzare i processi. Essendo una griglia, essa va a incrociare:

- la descrizione delle **variabili** di analisi (che sono le nostre variabili organizzative);
- con la descrizione delle **fasi** di analisi → ovvero l'analisi che abbiamo fatto per riuscire a reingegnerizzare il processo.

La griglia metodologica, quindi, è la seguente:

Variabili	Fasi		
	Rilevazione della situazione esistente	Confronto e diagnosi	Riprogettazione
Flussi di attività	Macroprocessi. Processi. Attività. Fasi. Prestazioni.	Benchmarking con best practice. Analisi delle possibilità di integrazione.	Personalizzazione delle best practice. Prototipazione e simulazione.
Struttura aziendale	Struttura organizzativa. Meccanismi operativi. Stile di direzione.	Benchmarking. Analisi del ruolo delle diverse funzioni nell'integrazione dei processi.	Struttura organizzativa (cambiamento ruoli e/o nuova divisione delle attività). Meccanismi operativi e stile di direzione.
Risorse umane	Esame disponibilità al cambiamento. Inventario di professionalità e capacità esistenti.	Confronto con concorrenti o best in class. Diagnosi.	Definizione del nuovo sistema di valori. Predisposizione programmi di training e di assunzioni.
Misurazione delle prestazioni	Identificazione prestazioni strategiche.	Confronto con concorrenti o best in class. Diagnosi.	Definizione dei nuovi obiettivi e nuovi indici di prestazione.

Possiamo vedere, che le variabili che possiamo incrociare, ovvero sia le “cose” che possiamo cambiare per migliorare l'organizzazione, sono:

- flussi di attività → quindi, come sono organizzate le attività;
- struttura aziendale → quindi, com'è la mia struttura aziendale;

- risorse umane → quindi, quali sono le mie risorse umane a disposizione ed in questo caso cambiare le risorse umane, non significa licenziare dei dipendenti, bensì significa, ad esempio, fornirgli delle nuove competenze tramite corsi di aggiornamento;
- misurazione delle prestazioni → quindi, valutare come sta effettivamente andando la mia organizzazione.

Le fasi, invece, sono:

- rilevazione della situazione esistente;
- confronto (con altre organizzazioni, magari leader, del mio settore) e diagnosi;
- riprogettazione.

Per capire il funzionamento della griglia, andiamo a vedere la prima variabile:

- Flussi di attività → prendiamo, ad esempio, il processo per la produzione in un prodotto e quindi guardo effettivamente com'è, ovvero guardo:
  - quali sono le attività;
  - come sono le prestazioni;
  - quanto ci impiego;
  - se ci sono dei "colli di bottiglia", ovvero sia delle migliorie da adottare.

Quindi, vado a fare effettivamente l'analisi del processo. Successivamente, vado a vedere in quale modo posso migliorare il mio processo, per esempio:

- se vi è un collo di bottiglia, il primo passo sarà sicuramente capire come eliminarlo;
- se vi sono attività che faccio in sequenza, ma che potrebbero essere eseguite in parallelo, cercherò quindi di eseguirle in parallelo, in modo da risparmiare tempo e conseguentemente denaro.

Capiamo immediatamente, allora, che quando valutiamo l'esecuzione del processo, stiamo andando ad analizzare il suo tempo di esecuzione → di conseguenza, possiamo certamente dire, che miglioriamo il processo, se a parità di risultato ci impiego di meno. Infine, andiamo a reingegnerizzare e quindi a riprogettare il processo, facendone un prototipo e simulandolo.



vediamo, allora, che attraverso la griglia vado ad incrociare le variabili, che posso modificare della mia azienda, con le tre fasi indicate.

## Variabili organizzative

Analizziamo, a questo punto, le variabili organizzative. La trasformazione dei processi per avere successo deve integrare:

- innovazione tecnologica;
- innovazione organizzativa.

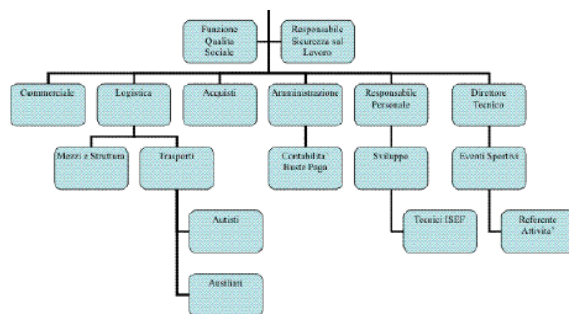
Secondo gli studi teorici del BPR è l'azione contemporanea e coordinata su tutte le variabili organizzative a garantire il successo. Le variabili organizzative, insieme con la tecnologia informativa, sono le leve attraverso cui sono trasformati i processi. Le variabili organizzative sono variabili di progettazione integrata dei processi ed in particolare sono:

- **Flusso delle attività** → insieme di attività coordinate per ottenere un determinato risultato. A parità di fattori, il flusso di attività, determina la durata del processo e quindi se le riorganizzo per ridurre la durata del processo, vado effettivamente ad ottimizzare il processo stesso. Quindi, il flusso di attività determina:
  - la durata del processo;
  - il livello di servizio → a seconda del grado di flessibilità (maggiore flessibilità comporterà un maggior livello di servizio);
  - qualità del prodotto (insieme alle tecnologie a disposizione e alle risorse umane).

La modellazione dei flussi (ovvero il modo in cui vado a rappresentare le attività che compongono il processo) può essere condotta a diversi livelli:

- schemi di sequenza → indicano solo la sequenza di attività che formano un processo (esempio: ricevimento ordini clienti, evasione ordini, spedizione, fatturazione);
- schemi più ricchi → indicano anche gli attori coinvolti, gli eventi che determinano le attività e le informazioni utilizzate. Richiedono, quindi, una raccolta di informazioni che può essere lunga e costosa.

- **Organizzazione del processo** → l'organizzazione ha a che fare con l'organizzazione aziendale, ovvero stiamo andando ad analizzare come l'organizzazione aziendale (ovvero: i ruoli all'interno dell'azienda, chi fa cosa, il modo in cui mi coordino) impatta sul mio processo. L'organizzazione, quindi, ha a che fare con chi fa cosa → da notare, che stiamo valutando la variabile organizzazione in funzione di miglioramento del processo e quindi, magari non cambio il flusso delle attività ma associo le attività che devono essere fatte alle figure ottime per eseguire una determinata attività. Per descrivere l'organizzazione si possono usare:
  - **organigrammi** → mostra la gerarchia delle responsabilità e delle autorità di una organizzazione a vari livelli di dettaglio. L'organigramma è molto utile, in quanto mi indica chi fa cosa;



- **tabelle di proprietà** → ci permettono di andare a descrivere le caratteristiche fondamentali delle strutture, andando a dire che cosa può fare ognuna delle strutture, ovvero, indicando:
  - su che cosa può agire ogni struttura;
  - quali sono i compiti assegnati;
  - quali sono i processi svolti.

Oltre a ciò, permette di descrivere l'organico (ad esempio, se abbiamo un solo corriere, ciò impatterà sul volume delle spedizioni) e il volume del lavoro che possono portare a termine.
- **Linear Responsibility Charting (LRC)** → offre una visione tabellare della responsabilità organizzativa, che integra quella dell'organigramma. LRC, quindi, è una tabella che, per ciascun processo, specifica il ruolo svolto da ogni struttura nel processo. Il ruolo può essere:
  - D = decide, autorizza e ratifica;
  - E = esegue le attività;

- A = partecipa a tempo parziale, fornisce assistenza operativa e supporto;
- I = è sistematicamente informato.

	Filiale vendita	Direzione commerciale	Direzione distribuzione	Magazzino Prodotti finiti	Spedizioni	Contabilità clienti
Ricezione ordini	E	D	I	I		A
Evasione degli ordini da magazzino	I		D	E	I	
Spedizione			I	A	E	I
Fatturazione			I			E

- **Competenza delle risorse umane che operano nel processo** →

l'adeguatezza delle competenze è condizione fondamentale per la trasformazione del processo a seguito di una innovazione tecnica. Le risorse umane, quindi, determinano la differenza tra il risultato effettivo di un processo e il massimo risultato teoricamente possibile in una data configurazione → la variabile risorse umane, infatti, è essenziale per il successo di un processo correttamente ingegnerizzato.

L'innovazione tecnologica causa la necessità di avere a disposizione figure professionali specializzate. Tali figure specializzate possono essere:

- formate internamente all'azienda riconvertendo personale → questa soluzione comporta dei problemi gestionali, quali:
    - il superamento dell'avversione al cambiamento;
    - l'accettazione a nuove tecnologie.
  - acquisite sul mercato → chiaramente il reperimento è complesso e costoso.
- **Sistema di misurazione e controllo delle prestazioni utilizzato per governare il processo e valutare gli attori aziendali che lo seguono.** Ogni processo è governato da un sistema di misurazione delle prestazioni che comprende:
    - il sistema di pianificazione e controllo → fissa gli obiettivi di efficienza e di efficacia del processo. Controlla, inoltre, periodicamente il raggiungimento degli obiettivi;
    - il sistema di promozione e di incentivazione → fissa gli obiettivi del processo (vendite, soddisfazione del cliente) e fissa le relative unità di misura globali e il metodo per misurare in contributo dell'individuo e calcolare il corrispondente incentivo;

- il sistema (scala) dei valori → posiziona gli obiettivi generali dell'azienda e decide i valori rispetto a cui incentivare. Nella scala dei valori tradizionale, la capacità di eseguire in modo altamente produttivo un segmento di attività, era considerato un valore da incentivare. In un processo orientato all'utente, è prioritaria la soddisfazione del cliente.

## Fasi della metodologia di analisi

Le fasi della reingegnerizzazione rappresentano i passi logici attraverso cui si svolge l'azione di analisi dei processi. Le fasi sono:

1. Rilevazione della situazione esistente.
  - ▶ Passo 1: Identificazione dei macroprocessi.
  - ▶ Passo 2: Dettaglio dei processi.
  - ▶ Passo 3: Incrocio processi / Unità organizzative.
  - ▶ Passo 4: Valutazione del processo.
2. Confronto con altre imprese e diagnosi dei problemi.
  - ▶ Passo 1: Confronto quantitativo e parametrizzazione.
  - ▶ Passo 2: Confronto qualitativo.
3. Ridefinizione dei processi.

Per ognuna di queste tre fasi, dobbiamo analizzare le variabili organizzative descritte precedentemente. Per le fasi si può adottare un approccio:

- **bottom-up** al miglioramento del processo esistente (approccio che consideriamo noi) → in cui si ha:
  - la rilevazione della situazione esistente;
  - confronto con altre imprese e diagnosi dei problemi;
  - ridisegno dei processi.
- **top-down** → in cui si ha:
  - disegno del processo con applicazione di criteri di ottimizzazione noti;
  - (limitata) analisi della situazione esistente a quanto necessario per poter attuare il progetto.

Analizziamo, infine, meglio i vari passi di cui sono composte le fasi:



## Fase 1: Rilevazione della situazione esistente

- ▶ Fase 1 - Passo 1: Identificazione dei macroprocessi.
  - ▶ Identificazione dei processi usando un opportuno modello (catena del valore, checklist, best practice).
- ▶ Rilevazione delle proprietà fondamentali:
  - ▶ Quali sono i clienti del processo (esterni / interni).
  - ▶ Tipo di processo (inside / buy-side / sell-side).
  - ▶ Input del processo (materie prime, competenze).
  - ▶ Output del processo (prodotto / servizio fornito al cliente).

## Fase 1: Rilevazione della situazione esistente

- ▶ Fase 1 - Passo 2: Dettaglio dei processi.
  - ▶ Scomposizione dei processi in fasi e attività, documentata da:
    - ▶ Diagrammi gerarchici (processo → fasi → attività).
    - ▶ Diagrammi di flusso (flussi fisici e/o flussi informativi).
    - ▶ Schede che descrivono le proprietà di processi, fasi, attività, operazioni.

## Fase 1: Rilevazione della situazione esistente

- ▶ Fase 1 - Passo 3: Incrocio processi / Unità organizzative.
  - ▶ Analisi della relazione fra strutture organizzative e processi da tre punti di vista:
    - ▶ Rilevazione delle strutture organizzative (organigrammi).
    - ▶ Definizione dei ruoli delle strutture nei processi (LRC).
    - ▶ Mappatura delle attività delle strutture nel flusso dei processi (opportuni diagrammi di flusso chiamati Responsibility Activity Diagram - RAD).

## Fase 1: Rilevazione della situazione esistente

- ▶ Fase 1 - Passo 4: Valutazione del processo.
  - ▶ Definizione dei parametri di funzionamento e giudizio sul valore dei prodotti.
  - ▶ Valutazione comprende:
    - ▶ Giudizio sul processo da parte degli esecutori.
      - ▶ Valutazione delle risorse dedicate al processo (costo del processo in termini di tempo usato dalle risorse).
      - ▶ Tempo di completamento del processo.
    - ▶ Giudizio sul processo da parte dei clienti interni o esterni.
      - ▶ Valutazione del prodotto in termini di:
        - ▶ Utilità/valore (prezzo che il cliente è disposto a pagare).
        - ▶ Livello del servizio (puntualità e velocità).
        - ▶ Qualità (contenuto del prodotto e assenza di difetti).

## Fase 2: Confronto e diagnosi

- ▶ Fase 2 - Passo 1: Confronto quantitativo e parametrizzazione.
  - ▶ Per il confronto tra prestazioni di aziende concorrenti.
    - ▶ Scelta di un insieme di parametri (produttività, livello di servizio, durata attività) da usare nel confronto.
    - ▶ Scelta di un significativo e accessibile campione di confronto.

## Fase 2: Confronto e diagnosi

- ▶ Fase 2 - Passo 2: Confronto qualitativo.
  - ▶ Esame delle variabili organizzative per individuare le aree in cui una diversità può indicare un legame causa-effetto.
    - ▶ Evidenziazione delle differenze di modello organizzativo o tecnologico.
  - ▶ Scopo: misurare le cause della diversità rispetto a valori di mercato.
  - ▶ Fase decisiva per la scelta di eventuali cambiamenti organizzativi.

## Fase 3: Ridefinizione

- ▶ Fase 3 comprende:
  - ▶ Definizione della *vision* che dà una rappresentazione sintetica degli elementi fondamentali della soluzione proposta.
    - ▶ Illustra la nuova struttura del flusso di attività e gli aspetti più significativi della configurazione delle variabili gestionali.
    - ▶ La *vision* è solitamente basata sugli schemi di processi best practice.
  - ▶ Analisi del cambiamento.
    - ▶ Specificazione dell'impatto organizzativo del progetto.
      - ▶ Incrocia le variabili organizzative con gli stakeholder, cioè con gli enti esterni e interni interessati al cambiamento.
    - ▶ Valutazione di rischi, benefici, tempi e costi del cambiamento.

# Business Process Model and Notation (BPMN)

Il BPMN è lo standard di modello grafico, che andremo ad utilizzare per modellare le attività dei nostri processi aziendali. Cercheremo, quindi, di costruire **modelli concettuali** dei nostri processi aziendali, avendo quindi a disposizione dei costrutti grafici, i quali ci permettono di rappresentare/modellare le attività, che devono essere portate a termine per raggiungere un determinato obiettivo. Il BPMN mette a disposizione:

- **elementi di base;**
- **concetti avanzati.**

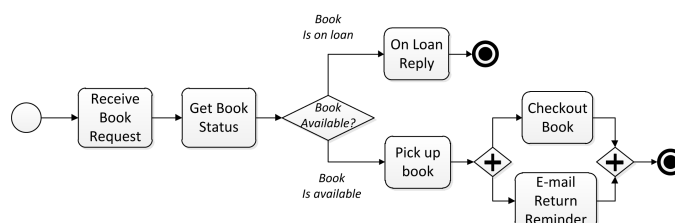


Il nostro obiettivo, quindi, è costruire una **rappresentazione grafica**, che ci permetta di descrivere/modellare e conseguentemente anche di documentare i processi presenti all'interno dell'azienda.

Il BPMN è molto utile, perchè una **notazione** per la modellazione grafica dei processi aziendali, definisce:

- i simboli per i vari elementi del processo;
- il loro corretto significato;
- le loro possibili combinazioni.

Pertanto, **una notazione è un linguaggio standardizzato per la descrizione dei processi aziendali** → quindi, pur non conoscendo la semantica dei costrutti e il loro utilizzo, **attraverso il BPMN abbiamo una rappresentazione grafica molto intuitiva per modellare i processi aziendali e di utilizzare tali modelli (di processi) per descrivere le attività**. Un primo esempio di modello di processo con il BPMN è il seguente:



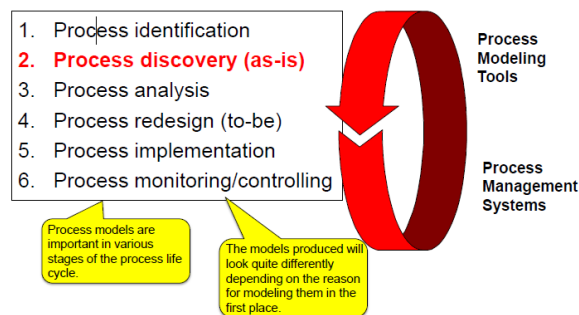
Anche in questo caso (come per il corso di Basi di Dati), quindi, abbiamo due concetti fondamentali:

- il concetto di **modello** → ovverosia una “cosa” che descriviamo ad alto livello di astrazione;
- il concetto di **istanza** → l'istanza di un processo è di fatto la singola esecuzione (del modello) del processo e di conseguenza, le istanze di un processo sono molte (riprendendo l'esempio sopra, il processo di prestito di un libro, avrà tante istanze; una per ogni persona che chiede in prestito un libro) → naturalmente, nell'istanza di processo, dove vi sono dei branch, in cui il cammino si divide, possiamo avere l'esecuzione di un'attività oppure di un'altra (a seconda del cammino che viene scelto).

---

Nel momento in cui andiamo a modellare un processo, ci stiamo ponendo nella condizione di capire come voglio raggiungere un determinato obiettivo e di conseguenza, dobbiamo capire:

1. **quali sono le attività che dobbiamo svolgere e come dobbiamo coordinarle per raggiungere l'obiettivo;**
2. una volta, che abbiamo capito quali attività dobbiamo svolgerle e come dobbiamo coordinarle, **dobbiamo progettare il modello di processo** (quindi dobbiamo modellare il processo);
3. successivamente **dovrò analizzare il modello di processo** ed eventualmente ottimizzarlo e di conseguenza, eventualmente dobbiamo riprogettare il modello di processo;
4. una volta che siamo certi di quello che abbiamo modellato, **andiamo effettivamente ad implementare il modello di processo;**
5. infine, svolgo tutta la **fase di monitoraggio e di controllo.**



Notiamo, che questo ciclo è molto simile al ciclo di vita del software, con l'unica differenza che in questo caso l'obiettivo è la modellazione di un processo e non la creazione di un Database.

Nel momento in cui vogliamo che il processo diventi eseguibile, abbiamo bisogno di mappare il modello di processo su un insieme di operazioni, ovvero su un linguaggio eseguibile (è la stessa cosa, che passare dal modello concettuale della Base di dati, al modello logico). Vi sono diversi linguaggi eseguibili, come per esempio:

- XPD L → è basato su XML;
- BPE L → linguaggio di esecuzione per i processi.

Sorge a questo punto spontanea la domanda: **“Che cos'è un sistema di gestione dei processi?”** Un sistema di gestione dei processi è l'analogo del DBMS, ovvero: il DBMS è un sistema che ci permette di gestire una Base dati e di conseguenza, **il PMS (Process Management Systems) è un sistema che ci permette di gestire i processi** (come per esempio: Signavio). **In particolare, il PMS è un sistema software, che ci permette di:**

- **modellare il processo;**
- **eseguire il processo;**
- **monitorare il processo;**
- **eseguire delle simulazioni del processo**, per capire se il modello del processo presenta dei problemi e di conseguenza è da sistemare.

---

Il BPMN si basa su XML ed è uno OMG standard. In particolare, **il BPMN rappresenta lo standard dedicato ai processi** ed è per questo motivo, che BPMN è molto importante, in quanto sempre di più l'idea di processo aziendale sta prendendo piede anche in azienda. L'obiettivo primario di BPMN è quello di fornire una notazione facilmente comprensibile da tutti gli utenti dell'azienda: dagli analisti, che creano la bozza iniziale dei processi, agli

processi e ai sviluppatori tecnici responsabili dell'implementazione e del monitoraggio.



L'idea, quindi, con la **versione 2.0 del BPMN**, è quella di creare un **ponte standardizzato** tra la progettazione dei processi aziendali e l'implementazione del processo.



Notiamo anche un ulteriore aspetto, ovvero: **il livello di astrazione che scegliamo per modellare un processo e per dargli una descrizione, dipende dallo scopo per cui stiamo modellando il processo.**

Nel momento in cui descriviamo un processo a **livello concettuale**, ci interessa solamente dare un'idea di quello che viene fatto e quindi, non arrivo a descrivere la singola attività. Il BPMN, per come è costruito, permette di scegliere qual è il livello di astrazione che vogliamo utilizzare e quindi, sono stati definiti **tre livelli di modellazione di processo**:

1. semplice descrizione delle attività e del coordinamento dei flussi dell'attività;
2. estendiamo la rappresentazione del 1° livello, aggiungendo delle informazioni addizionali, le quali sono più informative rispetto al 1° livello, ma comunque non sono sufficienti per dare un'effettiva esecuzione del processo;
3. livello dettagliato, che contiene abbastanza informazioni, per far sì che il processo sia eseguito.

A questo punto, vediamo gli **elementi di base** del BPMN, che essenzialmente sono quattro:



Event



Task



Flow

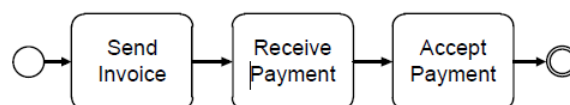


Gateway

- gli **eventi** → vengono rappresentati con un cerchio ed un "qualcosa" che succede in maniera **istantanea** e di conseguenza **non occupa tempo**. In particolare, vi sono:
  - eventi di **inizio**;
  - eventi di **fine**;

- eventi **intermedi** → spesso essi raccontano le eccezioni, che possono accadere durante l'esecuzione di un processo.
- i **task** (detti anche **attività**) → quando parliamo di attività, ricordiamoci che il processo è un insieme coordinato di attività. Anche in questo caso, il BPMN mette a disposizione diversi tipi di attività;
- i **flussi** → per gestire il fatto, che il processo è un insieme coordinato di attività, vengono utilizzati i flussi. In particolare, i flussi ci dicono in che modo sono ordinati nel tempo, gli oggetti che appartengono al nostro processo e di conseguenza, avremmo i flussi (i quali sono rappresentati con degli archi) tra gli altri tre elementi di base del BPMN → diciamo che i flussi ordinano, nel tempo, gli oggetti, perchè di fatto gli archi (ovvero i flussi) impongono dei **vincoli temporali** (ovvero dice cosa viene prima e cosa viene dopo);
- i **gateway** (che ci permettono di controllare il flusso) → per realizzare il coordinamento delle attività, oltre ai flussi, utilizzo i gateway, in quanto i gateway sono i punti, in cui il flusso del processo deve essere in un qualche modo controllato:
  - o per dire, che vi sono due cammini e ne deve essere preso uno a seconda del verificarsi o meno di una condizione;
  - oppure per dire, che vi sono più cammini e tutti devono essere percorsi.

Vediamo un esempio di processo (in questo caso, un processo sequenziale):



possiamo vedere, che abbiamo:

- un evento di inizio → **da notare, che gli eventi di inizio hanno sempre un solo arco uscente;**
- delle attività → **da notare, che esse possono avere degli archi entranti e degli archi uscenti;**
- i flussi, che dicono cosa viene prima e cosa viene dopo;
- un evento di fine → **da notare, che gli eventi di fine hanno sempre solamente degli archi entranti.**



Idealmente, per avere dei processi **ben strutturati**, vorremmo avere:

- un

**solo** punto di inizio e un **solo** punto di fine;

- ogni attività abbia un

**solo** arco entrante ed un **solo** arco uscente (in realtà è possibile avere due archi entranti per una stessa attività, ma certamente vogliamo evitare di avere più di un arco uscente per una attività, in quanto ciò rappresenterebbe ambiguità).

**Per riuscire a fare ciò, sfruttiamo il fatto che quando abbiamo un gateway di Split, dobbiamo avere il corrispondente gateway di Merge**

→ il vantaggio di avere dei processi ben strutturati, è che più facile gestire il flusso del processo.

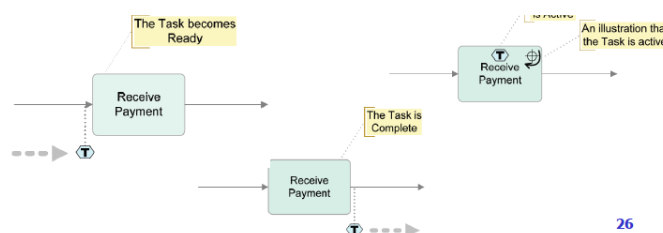
Naturalmente, ogni processo deve avere:

- almeno un evento di inizio, che ci indica la motivazione del perchè il processo inizia;
- almeno un evento di fine.

(Come abbiamo detto sopra, il nostro obiettivo è di avere un unico evento di inizio ed un unico evento di fine).

Un altro aspetto del BPMN è il **token** → esso **NON** è un costrutto del BPMN, bensì si tratta di qualcosa di puramente teorico, che noi utilizziamo per simulare l'esecuzione del processo. Diciamo, quindi, che il token:

- viene generato dall'evento di inizio;
- viene consumato dall'evento di fine;
- permette **a noi (nel senso, che non rappresentiamo il token nel modello del processo)**, di visualizzare/mostrare in maniera grafica come prosegue l'esecuzione del processo:



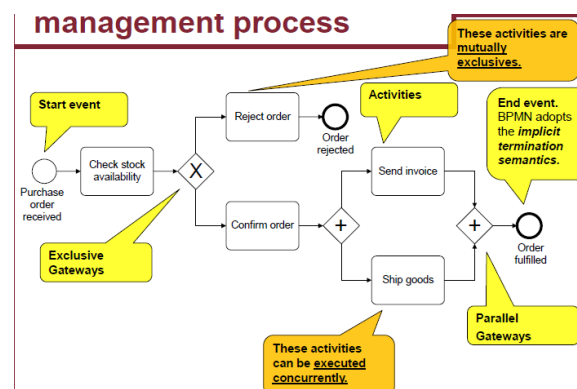
26

vediamo, ad esempio, che il token “entra” nell’attività ed essa, diventa pronta per essere eseguita. Successivamente, abbiamo l’esecuzione dell’attività e in questo caso, il token mi indica che si sta eseguendo l’attività. Quando abbiamo finito l’esecuzione dell’attività, il token è “uscito” dall’attività e quindi, si prosegue con il flusso.



Un aspetto da ricordarci, quando facciamo la simulazione di esecuzione di un processo, è il fatto che si presuppone che NON ci sia un tempo per il passaggio del token sul flusso → **il flusso, quindi, non consuma tempo e quindi il passaggio del token lungo le frecce (ovvero lungo i flussi) non consuma tempo**. Se vogliamo rappresentare che vi è un’attesa, dobbiamo utilizzare un evento intermedio.

Vediamo un primo esempio di modellazione con il BPMN:



Notiamo, che non diamo i dettagli dell’evento di inizio, in quanto vogliamo rimanere in un alto livello di astrazione. Notiamo, inoltre, che vi sono sia gateway esclusivi (che indichiamo con X) sia gateway paralleli (che indichiamo con +).

I **gateway**, quindi, sono un costrutto del BPMN che ci permettono di costruire dei cammini sul nostro processo. Quando, il token arriva al gateway, a seconda del tipo di **gateway (di split)**, il token può essere:

- instradato su un particolare cammino → nel caso del gateway esclusivo;
- instradato (e quindi duplicato) su tutti i cammini → nel caso del gateway parallelo.

Quando abbiamo **gateway di merge**, ovvero i gateway che raccolgono i cammini che sono stati creati, dobbiamo aspettare i token che arrivano (naturalmente, quindi, se abbiamo generato un solo token, dobbiamo aspettarlo e poi possiamo proseguire



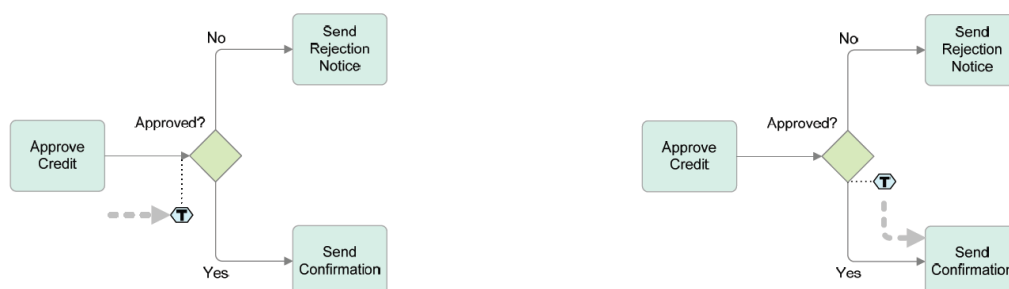
con l'esecuzione; se invece abbiamo generato diversi token, dobbiamo attenderli tutti e poi proseguire con l'esecuzione). Vediamo, poi, che abbiamo due tipologie di gateway:

1. **Gateway esclusivi (XOR)** → vengono rappresentati con un rombo, con all'interno una X (oppure sono rombi vuoti), e possono essere corredati con la domanda che ci dice cosa rappresenta il gateway. **Dal gateway esclusivo partono tanti cammini che sono in alternativa, ovvero sia viene preso solamente un cammino.**
2. **Gateway paralleli (AND)** → vengono rappresentati con un rombo, con all'interno un +, e rappresentano il fatto, che **dobbiamo eseguire tutte le attività, che stanno sui cammini che partono dal gateway parallelo.** I gateway paralleli, quindi, forniscono un meccanismo di sincronizzazione dei flussi paralleli e di creazione di flussi paralleli (AND-split), con attività che possono essere eseguite simultaneamente.

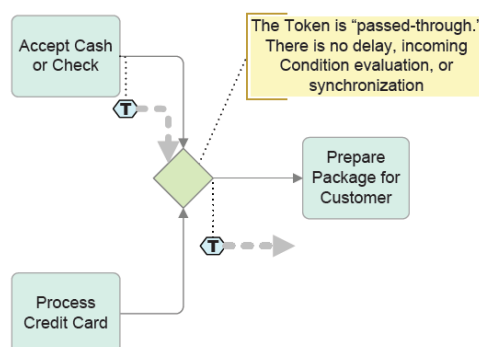


Per avere un **processo ben strutturato**, come avevamo per i gateway paralleli, in cui avevamo un gateway di split e un gateway di merge, anche per i gateway esclusivi dobbiamo comportarci allo stesso modo, ovvero sia che dobbiamo avere sia un gateway di split sia un gateway di merge.

Vediamo attraverso la rappresentazione con i token, di tutto ciò che abbiamo detto:



Vediamo che il gateway esclusivo viene rappresentato con un rombo vuoto e viene correlato con una domanda. Vediamo che una volta eseguita l'attività (nell'esempio, l'attività di Approve Credit) procediamo e il token arriva al gateway esclusivo di split. A seconda della risposta alla domanda, viene preso uno specifico cammino e quindi, il token viene instradato su un particolare cammino ed entra nella prossima attività (nell'esempio, il token entra nell'attività Send Confirmation). Per avere un processo ben strutturato, dobbiamo avere anche il gateway di merge, dove dobbiamo aspettare che arrivi un solo token.



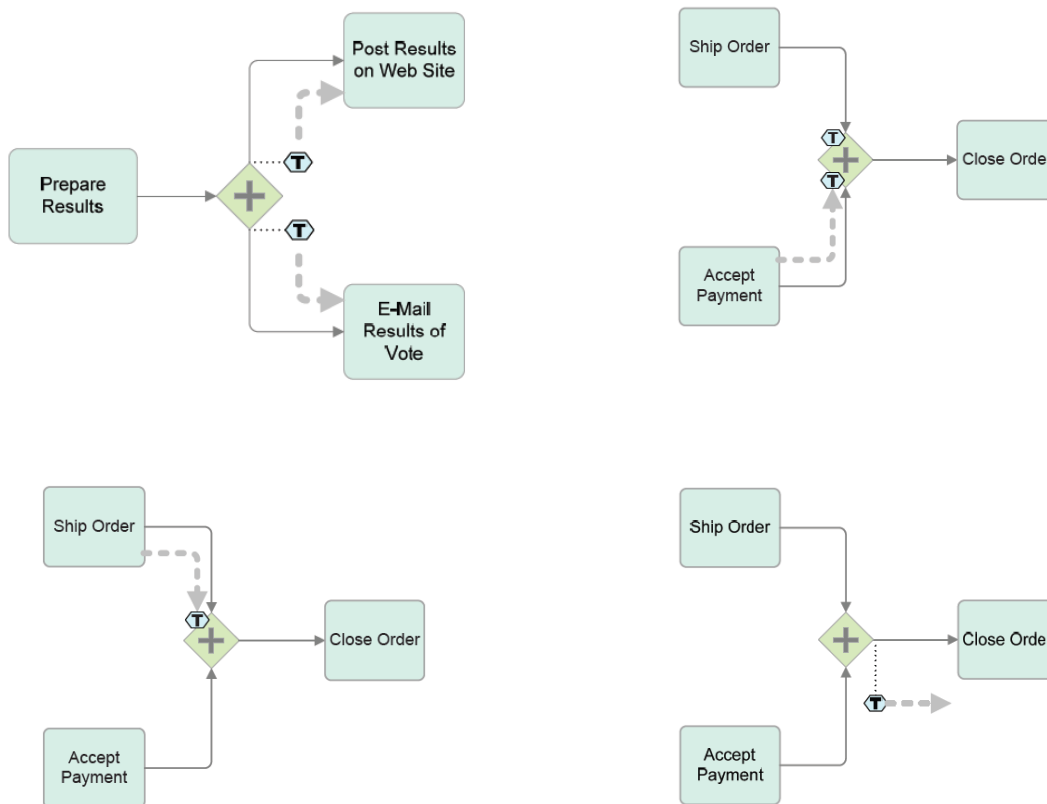
Possiamo anche specificare una condizione di default per il gateway, che serve per essere sicuri di non creare un punto in cui il gateway si ferma e non va più avanti e quindi, **per essere sicuri di arrivare alla terminazione del processo**. Naturalmente, se la domanda che corre da il gateway è ben fatta, la condizione di default non servirebbe, in quanto verrebbe sempre preso un cammino, ma non è sempre così (nel senso, che non sempre le domande sono ben fatte).



Notiamo anche il fatto, che un cammino che esce da un gateway esclusivo, potrebbe essere **senza attività** e che quindi fa avanzare l'esecuzione del processo.

Il gateway parallelo, invece, prevede che tutti i cammini che partono dal gateway sono paralleli. Da porre particolare attenzione al fatto, che in questo caso **“parallelo” NON vuol dire che le attività vengono eseguite in parallelo, bensì significa che tutte le attività che vi sono sui cammini (che partono dal gateway parallelo) devono essere eseguite prima di poter procedere** (l'ordine con cui vengono eseguite le attività a noi non interessa). In questo caso, non ha senso avere

un cammino senza attività e quindi, sui cammini che escono da un gateway parallelo, vi deve sempre esserci qualcosa. Inoltre, in questo caso, il numero di token che viene generato (a partire dal gateway parallelo) è pari al numero di cammini che vengono generati dal gateway parallelo e tutti i token devono arrivare al gateway di merge.



Per sincronizzare il flusso, quindi, il gateway parallelo attende l'arrivo di un token da ogni flusso di sequenza in arrivo. Quando arriva il primo token, non viene valutata alcuna condizione per il flusso di sequenza in arrivo, ma il token viene "trattenuto" dal gateway e non prosegue. Quando tutti i token sono arrivati, vengono uniti e il token viene "rilasciato" e si sposta lungo il flusso di sequenza in uscita.

A questo punto, vediamo un'altra tipologia di gateway, ovvero il **gateway inclusivo** → A volte potrebbe essere necessario prendere uno o più rami dopo un'attività decisionale

e per riuscire a fare ciò, viene utilizzato il gateway inclusivo. Esso viene rappresentato attraverso un rombo, con all'interno una O, e appunto mi indica, che a seconda della condizione che è stata specificata, uno o più cammini possono essere presi ed eseguiti. → NON abbiamo più il fatto, quindi, che i cammini siano mutualmente esclusivi.



Come avviene per le altre due tipologie di gateway, **il sistema sa** quanti token sono stati generati e quindi sa quanti token devono essere aspettati nel gateway di merge, prima di continuare con l'esecuzione del processo.

A questo punto, vediamo quella parte di DPMN che ci permette di gestire, il fatto che all'interno di un'organizzazione dobbiamo **coordinare** il lavoro e di conseguenza, questa parte di DPMN ha a che fare con:

- chi svolge determinate azioni;
- in che modo l'organizzazione dell'azienda, può impattare sulla modellazione dei processi.



**Con il BPMN, quindi, possiamo dire in che modo è organizzata l'azienda, o meglio dire quali parti dell'azienda (ovverosia **CHI**) si occupano di eseguire determinate attività di processo (il **QUANDO** è implicito, grazie alle linee di flusso che connettono i vari elementi nel processo).**

Concentriamoci, allora, su CHI fa cosa, per quanto riguarda la modellazione di processo → quando pensiamo ad un'organizzazione, ci immaginiamo immediatamente di avere:

- delle **risorse**, che sono a disposizione dell'azienda per raggiungere un determinato obiettivo (come per esempio: materie prime, risorse umane, software) → ovviamente, ogni singola persona è una risorsa e se andiamo a creare delle **classi di competenze**, possiamo dare una rappresentazione del fatto, che le aziende abbiano una determinata competenza. Questo è per dire, che CHI fa cosa non è necessariamente collegato ad una persona singola, bensì CHI ha una determinata competenza può eseguire una determinata attività del processo → quindi, oltre che a pensare alle risorse, possiamo pensare anche ad insiemi di risorse, che condividono determinate caratteristiche.

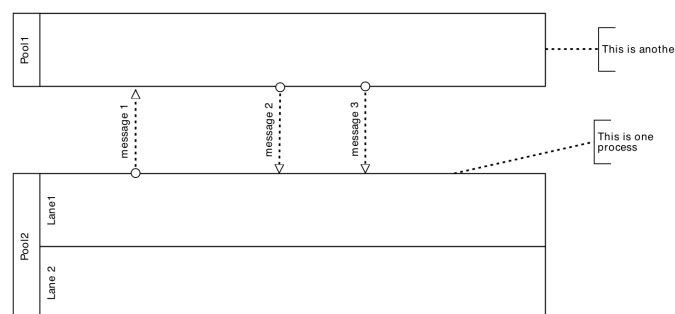


Quando parliamo di una classe di risorse, quindi, possiamo far riferimento ai **ruoli** (ovverosia le competenze e le skills che hanno tali risorse) ed eventualmente incasellarle nella struttura → ci ricollegiamo, quindi, alla risorsa organizzativa nella griglia metodologica.

Nel BPMN, i **costrutti** che ci permettono di modellare CHI fa cosa, sono essenzialmente due:

- **Pools** → permettono di rappresentare l'organizzazione. Solitamente, implicitamente il modello del processo è dentro una pool (nel caso in cui non rappresentiamo la pool, è perchè diamo per scontato che ci sia). Il processo, per come è pensato, è un insieme di attività interne all'azienda e questo vuol dire, che **il processo che mettiamo all'interno di una pool, rappresenta un insieme di attività fatte da un'unità indipendente, ovverosia che non condivide risorse con altre organizzazioni.**
- **Lanes** → Le sotto-parti in cui l'organizzazione/l'entità organizzativa è organizzata, le posso rappresentare con le lanes. Per esempio, se pensiamo ad un'azienda manifatturiera come entità organizzativa e inseriamo il relativo processo di modellazione all'interno di una pool, potremmo avere la lane relativa alla parte manageriale, una lane relativa alla parte produttiva e una lane relativa alla parte amministrativa. Le lanes, quindi, rappresentano più classi di risorse nella stessa pool e che condividono sistemi comuni.

Vediamo un esempio:



Questo potrebbe essere l'esempio del processo di immatricolazione di uno studente presso l'ateneo. In particolare, nella 1° pool abbiamo l'ateneo e nella 2° pool abbiamo lo studente, che vuole immatricolarsi e quindi nella 2° pool abbiamo il processo di immatricolazione.

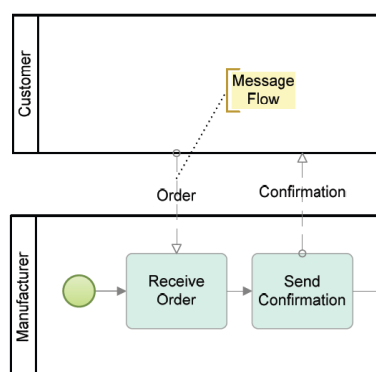
Abbiamo due pools, le quali mi permettono di rappresentare due organizzazioni diverse, dove per **organizzazione intendiamo un attore che partecipa al**

**processo** (quindi non intendiamo necessariamente due aziende, bensì possiamo intendere, ad esempio, medico-paziente che quindi sono due enti organizzativi diversi). In particolare, all'interno della 1° pool inserisco un processo e non specifico alcuna lane. Nella 2° pool, invece, abbiamo 2 lanes e quindi, avremo una **parte organizzativa** e un'altra **parte organizzativa**. Anche all'interno di questa 2° pool vado ad inserire un processo → **da notare, che il processo inserito nella 1° pool e il processo inserito nella 2° pool sono 2 processi distinti**. Notiamo, inoltre, che la comunicazione tra le pools avviene attraverso dei **messaggi** (i quali vengono rappresentati attraverso un arco tratteggiato, che ha un pallino all'inizio e un triangolo alla fine) → **questo tipo di arco sta SOLAMENTE tra pools diversi e MAI all'interno di una pool**. **Non sta all'interno di una singola pool, perchè il passaggio di informazioni all'interno di una pool viene considerato implicito**. **Al contrario, i sequence flow stanno SOLAMENTE all'interno di una pool, perchè il processo deve stare all'interno dell'organizzazione e quindi all'interno della pool**.



Il flusso di messaggio, inoltre, **a seconda del livello di astrazione** che si decide di utilizzare, può:

- partire dal bordo di una pool ed arrivare al bordo di un'altra pool;
- partire dal bordo di una pool ed arrivare ad una attività di un'altra pool;
- partire da un'attività di una pool ed arrivare al bordo di un'altra pool.

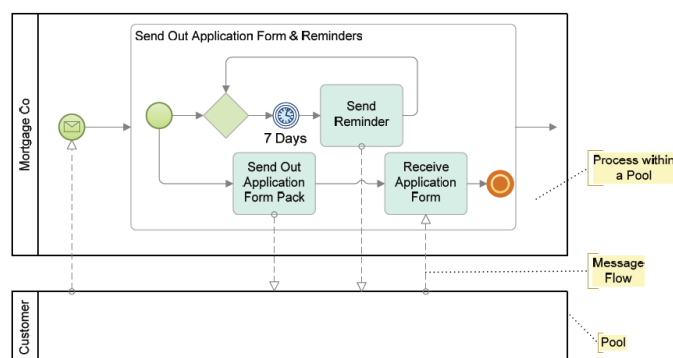


In questo esempio, abbiamo 2 organizzazioni (Customer-Manufacturer), ognuna delle quali porta avanti un proprio processo per raggiungere il proprio obiettivo e in un qualche modo si interfacciano tra loro.

Nell'esempio sopra, abbiamo considerato l'Ateneo come un ente organizzativo, il cui processo sta all'interno di una pool e lo studente, come un'altra organizzazione, che sta in un'altra pool, in quanto lo consideriamo esterno all'Ateneo (perchè fino a

quando lo studente non si è immatricolato, non fa parte dell'Ateneo. Una volta immatricolato, lo studente è interno all'Ateneo) → questo, chiaramente, dipende dal **punto di vista** che abbiamo utilizzato, ovvero è una **scelta di modellazione**. A seconda del punto di vista, quindi, la stessa risorsa può essere interna oppure esterna alla pool e di conseguenza, dovremmo organizzare le pools e le lanes in maniera differente.

Diamo un'altra definizione, ovvero quella di **Black Pool** → non è necessario che una pool contenga un processo. Conosciuta come Black Pool, esse non mostrano le attività o il flusso delle sequenze all'interno del suo confine. Nell'esempio sotto, la Pool "Customer" è una Black Pool.

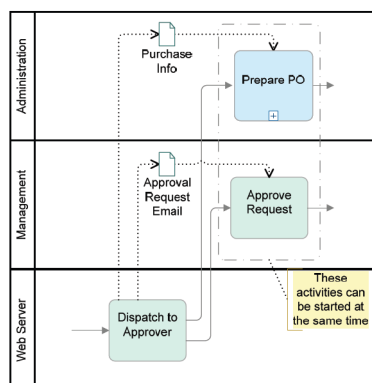


Sorge a questo punto spontanea la domanda: **“Perchè utilizziamo le Black Pool?”** Esse vengono utilizzate, perchè stiamo modellando dal punto di vista dell'organizzazione dell'altra pool (nell'esempio: Mortgage Co) e quindi, mi interessa solamente dire che il processo della pool che sto considerando, si interfaccia con una pool esterna e di conseguenza, il processo della pool esterna non è di nostra competenza/interesse → ci interessa solamente, quindi, il fatto che la pool esterna partecipa al nostro processo solamente come entità, che spedisce e riceve qualche messaggio.

A questo punto, analizziamo meglio il costrutto delle **Lanes** → le lanes, come abbiamo detto precedentemente, creano delle sotto-partizioni della mia organizzazione. Quindi, se vogliamo dettagliare quale sotto-parte esegue determinate attività, possiamo suddividere la pool in varie lanes. Le lanes spesso rappresentano i diversi ruoli dell'organizzazione (ad esempio: Manager, Administration, Associato, ecc.) ma possono anche rappresentare qualsiasi classificazione desiderata (ad esempio: tecnologia, dipartimenti organizzativi, prodotti dell'azienda, ecc.). Da notare che:

- i flussi di sequenza possono attraversare le lanes di una stessa pool;

- il flusso di messaggi non viene utilizzato all'interno o tra le lanes di un pool;
- le lanes possono essere annidate.



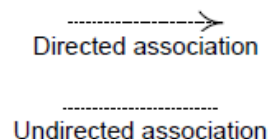
Oltre alle pools e alle lanes, nel BPMN possiamo avere degli **artefatti** → gli artefatti forniscono un meccanismo per catturare delle **informazioni aggiuntive** su un processo e di conseguenza, tali informazioni **non hanno un impatto diretto sul flusso di un processo**. Utilizzare o meno gli artefatti, dipende dalla **scelta di progetto** che si decide di adottare, ed in particolare sono disponibili tre tipi di artefatto:

- i **gruppi** → un gruppo viene utilizzato per circondare un insieme di oggetti del flusso, al fine di evidenziare e/o categorizzarli, ovvero, per esplicitare che un determinato insieme di attività riguardano un determinato aspetto del processo;
- le **annotazioni di testo** → un'annotazione di testo fornisce al modellatore, la possibilità di aggiungere ulteriori informazioni descrittive o note su un processo o dei suoi elementi.
- i **data objects** e le **associazioni** → ci permettono di dettagliare il passaggio di informazioni all'interno della pool. Se volessimo, quindi, esplicitare ad esempio che tra la 1° attività e l'ultima attività del processo vi è un passaggio di informazioni, possiamo utilizzare il data object. Il data object, quindi, rappresenta dell'informazione che viene gestita durante l'attività del processo e non necessariamente, tale informazione deve essere memorizzata da qualche parte. Quindi, una volta che il processo è stato eseguito, l'informazione viene persa, dato che non è stata memorizzata in una base dati. Se voglio rappresentare il fatto, che raccolgo e/o memorizzo le informazioni da una base dati, allora utilizzo il **Data Store** → i data store sono dei contenitori di data objects, che devono essere conservati oltre la durata di un'istanza di processo. Le attività di processo possono leggere/scrivere data objects da/verso i data stores.

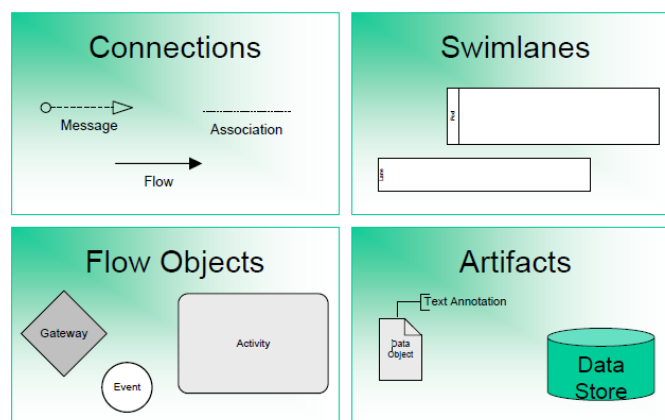


Per collegare i data objects e i data stores alle attività di processo, utilizzo le **associazioni** → le associazioni, quindi, vengono utilizzate per collegare gli artefatti (come i data objects e i data stores) con gli oggetti di flusso (ad esempio, attività e, a volte, eventi) e di conseguenza, **esse stanno all'interno di una pool**. Le associazioni si suddividono in:

- associazioni **dirette**;
- associazioni **indirette**.

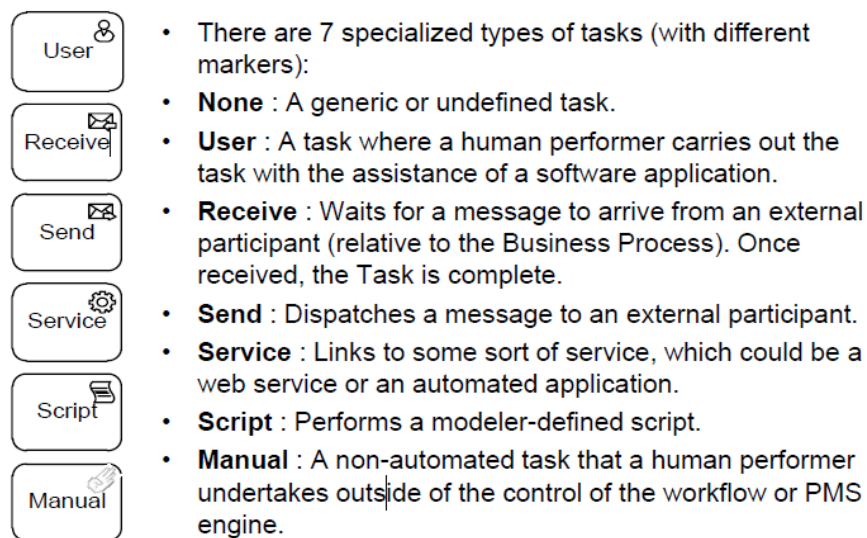


Prima di proseguire con gli argomenti, utilizziamo la seguente immagine per fare una panoramica, degli elementi del BPMN visti fino a questo momento:



Visto che abbiamo visto i concetti di base del BPMN, a questo punto possiamo andare ad analizzare i **concetti avanzati** del BPMN, dove per concetti avanzati intendiamo andare a dettagliare ulteriormente i concetti di base che abbiamo già visto → ad un alto livello di astrazione, abbiamo già tutti gli elementi per modellare qualsiasi tipo di processo. Se, invece, vogliamo utilizzare un livello di astrazione più basso e quindi dettagliare maggiormente il processo, abbiamo bisogno dei costrutti avanzati del BPMN. In particolare, **il BPMN 2.0 prevede 7 tipi diversi di tasks**, in modo tale da poter usare uno speciale tipo di task, per rappresentare uno speciale tipo di attività. Fino ad ora, abbiamo visto tasks “indefiniti”, ovvero senza alcun tipo di dettaglio e quindi **generico**, dato che a livello di modello del processo, non mi interessa/non mi serve dettagliare il task (stessa cosa vale anche per gli eventi). Se,

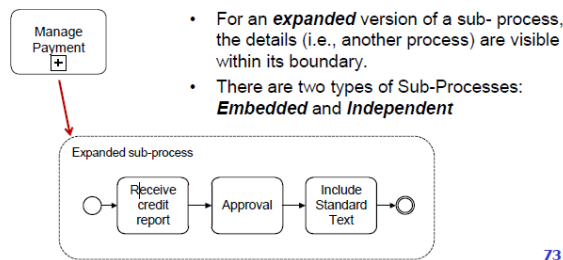
però, ci interessa **specializzare** un task (oppure un evento), il BPMN mette a disposizione 7 tipologie di tasks:



**Notiamo, che se abbiamo un task di ricezione, ovvero un task che aspetta di ricevere qualcosa, la simulazione dell'esecuzione del processo continua solamente se arriva “quel qualcosa” al task di ricezione. A livello di esecuzione del processo, quindi, esso (ovverosia il task) impatta sull'esecuzione del processo** → chiaramente, utilizziamo i none tasks oppure una delle altre 6 tipologie di tasks, a seconda: del livello di dettaglio, della scelta di progetto e a seconda di cosa mi serve il processo.

Un'attività è un lavoro che viene svolto all'interno di un processo aziendale. L'esecuzione può richiedere del tempo e coinvolge una o più risorse dell'organizzazione. Di solito richiede qualche tipo di input e produce una sorta di output e un'attività può essere:

- **atomica** (conosciuta anche come task);
- **composta** (non-atomica) → questa tipologia di attività viene chiamata **sotto-processo** (le attività composte, quindi, prendono il nome di sotto-processo). A livello di modellazione di processo, quindi, diciamo che si tratta di una **macro-attività**, la quale possiamo scomporla in un sotto-processo → scomporre un processo di grandi dimensioni in sotto-processi (ovvero scomporre un processo di grandi dimensioni in attività, le quali possono essere ulteriormente dettagliate), ci può tornare utile per avere una visione effettiva delle singole attività che dobbiamo svolgere, per completare il processo. Vediamo un esempio:



73

All'interno del processo, abbiamo l'attività "Gestione pagamenti", la quale è un sotto-processo e quindi la possiamo ulteriormente dettagliare.

Da sottolineare il fatto, che vi sono due tipi di sotto-processo:

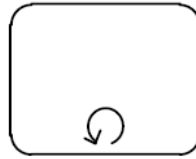
1. **embedded** → sotto-processi che sono all'interno del processo, che **non posso riutilizzare in altri processi**. Sono, quindi, una "sotto-porzione" di uno specifico processo. Nel caso di sotto-processi embedded, essi hanno un evento di inizio e un evento di fine, entrambi di tipo none;
2. **indipendenti** → sotto-processi che sono all'interno del processo, che **possono essere riutilizzati in altri processi**. In questa tipologia di sotto-processi, possiamo anche specificare eventi di inizio di tipo ricezione (oltre che ovviamente ad eventi di inizio di tipo none).



Ovviamente il token arriverà al sotto-processo e seguirà le attività in cui è stato dettagliato il sotto-processo, per poi uscire quando il sotto-processo è terminato.

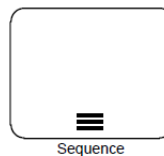
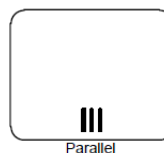
Altri tipi di attività, che possiamo utilizzare nella modellazione del processo sono:

- **attività di loop** → per una certa attività, posso specificare una condizione di loop, che determina il numero di volte in cui eseguire quell'attività. Esistono due varianti per il looping dell'attività:
  - While Loop (o While-Do) → la condizione di loop viene verificata **prima** che l'attività venga eseguita. Se la condizione del ciclo vera, l'attività viene eseguita. Nel caso contrario, l'attività viene completata e il processo continua (un token si muove lungo la sequenza in uscita). Il ciclo di controllo della condizione del ciclo e di esecuzione dell'attività continua finché la condizione del ciclo non è falsa;
  - Until Loop (o Do-While) → la condizione del ciclo viene verificata **dopo** che l'attività è stata eseguita.

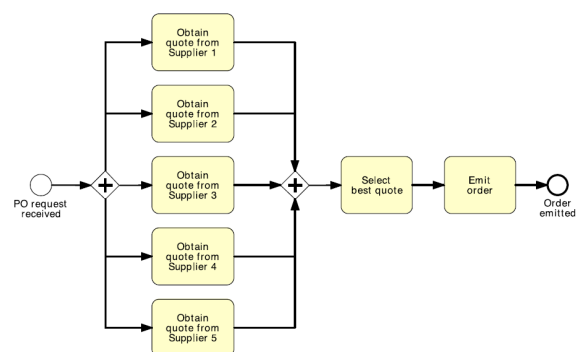
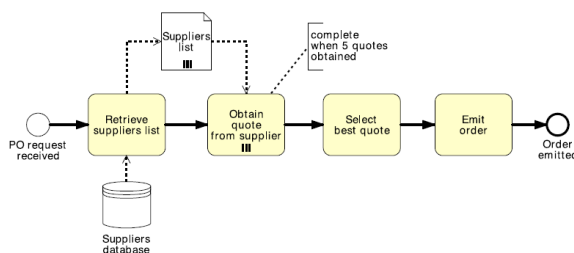


Utilizzando gli attributi dell'attività, è possibile impostare il numero massimo di cicli (loop maximum) per i cicli while e until. Dopo che l'attività ha raggiunto massimo di loop, l'attività si arresta (anche se la condizione di loop è ancora vera).

- **attività di multi-istanza** → la stessa attività viene eseguita più volte su dataset diversi, o meglio dire: vi sono più istanze della stessa attività all'interno dello stesso processo. A differenza delle attività di loop, in cui la stessa attività veniva eseguita più volte, con le attività di multi-istanza abbiamo un'attività che viene istanziata più volte (per esempio: l'attività "Richiesta preventivo" viene istanziata tante volte, quanti sono i preventivi che vogliamo avere) e quindi, è importante capire che **ogni esecuzione dell'attività è distinta dalle altre**. Le singole istanze di un'attività multi-istanza possono essere eseguite in:
  - parallelo;
  - sequenza.



Vediamo un esempio di attività multi-istanza e la corrispondente contropartita, utilizzando i concetti di base del BPMN:



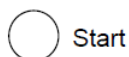


Le attività multi-istanza corrispondono esattamente ad istanziare diverse volte l'istanza (per esempio istanziare 5 volte l'attività di richiesta preventivo), con **il vantaggio che (in caso di cambiamenti) non dobbiamo modificare la struttura del processo** (immaginatoci, per esempio, che ci servono 7 preventivi e non più 5).

Analizziamo a questo punto le varie tipologie di **eventi**, che sono a disposizione nel BPMN → un evento è qualcosa che "**accade istantaneamente**

" nel corso di un di un processo aziendale e di conseguenza, **gli eventi non consumano tempo**. Un evento può influenzare il flusso del processo e solitamente ha una causa scatenante o un risultato. In particolare, possiamo avere tre tipologie di evento:

- evento di **inizio** → evento che, nella nostra idea di comportamento di processo, **genera il token, il quale proseguirà sul flusso di sequenza del processo**. Quindi, a fronte dell'accadimento di qualcosa, viene generato il primo token, il quale dà l'avvio al processo;
- evento **intermedio** → eventi che si mettono in mezzo al processo e danno un comportamento di:
  - raccolta/attesa di qualcosa (es: attesa di un messaggio);
  - produzione di qualcosa (es: invio di un messaggio).
- evento di **fine** → evento che consuma il token finale.



Start



Intermediate



End



Notiamo, inoltre, che quando modelliamo un processo, lo stesso accadimento (lo stesso fatto) potrebbe essere interpretato come un evento oppure come un'attività, a seconda che:

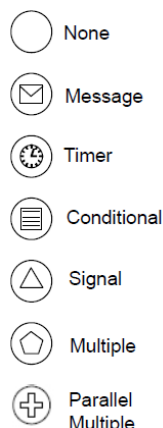
- l'interpretazione che diamo al fatto occupi tempo → in questo caso, il fatto viene interpretato come un'attività;
- l'interpretazione che diamo al fatto non occupi tempo → in questo caso, il fatto viene interpretato come un evento;

Analizziamo in maniera più approfondita le varie tipologie di evento e quindi, iniziamo con gli eventi di start → come abbiamo detto, gli eventi di start danno il via al processo. In particolare, abbiamo diverse tipologie di eventi di start, per indicare le diverse **circostanze** che possono appunto dare l'avvio al processo. Queste circostanze, come ad esempio l'avvio di messaggio o lo spegnimento di un timer, sono chiamati **inneschi/triggers**. Chiaramente, un evento di start può avere solo flussi di sequenza in uscita ed in particolare, gli eventi di start basati su trigger possono essere presenti **solamente** nei processi di livello superiore (quindi non vengono mai utilizzati nei sottoprocessi).

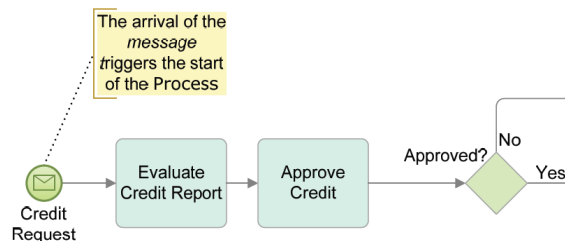


Sottolineiamo anche il fatto, che **l'evento di start di tipo None**, non è appunto basato su un trigger, ma **rappresenta un livello di astrazione più alto, ovvero non andiamo a dettagliare cosa sta succedendo**. Se, poi, vogliamo dettagliare il livello di astrazione, andremo ad utilizzare le altre tipologie di evento di start.

Oltre all'evento di start di tipo None, quindi, abbiamo le seguenti tipologie:



- **Messaggio** → il processo, per essere avviato, attende che venga ricevuto un messaggio (notare, quindi, che è rappresentato con la bustina bianca). Un messaggio è una comunicazione diretta tra due partecipanti e questi partecipanti devono trovarsi in pool separate (quindi non possono essere inviati da una lane ad un'altra all'interno di un'unica pool). Vediamo un esempio:



- **Segnale** → il processo, per essere avviato, attende che arrivi un segnale. Questo segnale è una comunicazione broadcast trasmessa dallo stesso processo, oppure da un altro processo. I segnali non hanno un destinatario o un bersaglio specifico, ovverosia tutti i processi e i partecipanti possono vedere il segnale e spetta a ciascuno di essi decidere se reagire o meno (l'evento che deve "raccogliere", quindi l'evento di throwing che raccoglie il segnale, lo indichiamo con la stessa etichetta dell'evento segnale di catching). A differenza dei messaggi, i segnali possono operare all'interno di un processo (magari tra un sottoprocesso e il suo processo genitore chiamante);
- **Timer** → il processo, per essere avviato, attende che si verifichi una determinata **condizione temporale**, la quale può essere specificata. Tale condizione temporale, potrebbe trattarsi di una data e un'ora specifiche (ad esempio, il 1° gennaio 2009 alle 8:00) o di un orario ricorrente (ad esempio, ogni lunedì alle 8:00).
- **Condizionale** → attende che si verifichi una condizione, ovverosia l'evento di start condizionato viene avviato quando una condizione predefinita diventa vera. Una condizione viene utilizzata per definire i dettagli della modifica dei dati che ci si aspetta e la condizione per l'evento deve diventare falsa e poi di nuovo vera, prima che l'evento possa essere nuovamente attivato → le condizioni che specifichiamo, possono avere anche un alto livello di dettaglio (per esempio: la merce in magazzino scende sotto il 15%);
- **Multipli** → l'evento di start multiplo, rappresenta un insieme di due o più o più inneschi di eventi di start. I triggers possono essere qualsiasi combinazione di messaggi, timer, condizioni e/o segnali;

- **Multipli paralleli** → per l'evento di avvio multiplo parallelo è necessaria una combinazione di triggers, prima che il processo possa essere istanziato.

Una volta analizzati gli eventi di inizio, analizziamo gli **eventi di fine** → a differenza dell'evento iniziale, l'evento finale **consuma** il nostro token e chiaramente, l'evento finale avrà solamente archi entranti (se il processo è ben strutturato, possibilmente vorremmo che l'evento finale abbia un solo arco entrante). Mentre gli eventi iniziali sono eventi di catching (nel senso che raccolgono qualcosa) che generano un token, gli eventi finali sono eventi throwing (nel senso che producono qualcosa) e di consumo del token (quindi, ad esempio: quando il token arriva all'evento finale, il token viene consumato e viene inviato il messaggio/il segnale...). Esistono diversi tipi di eventi finali, che indicano diverse categorie di **risultati** del processo, dove per risultato intendiamo qualcosa che si verifica alla fine di un determinato percorso del processo (ad esempio: l'invio di un messaggio o la trasmissione di un segnale), l'invio di un messaggio o la trasmissione di un segnale).

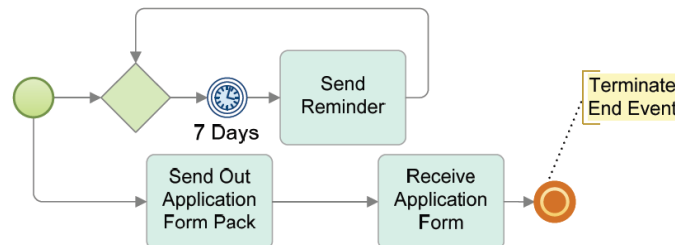


(Anche in questo caso, l'evento finale di tipo None, rappresenta il livello di astrazione più alto, ovvero rappresenta un generico evento finale, il quale può essere ulteriormente dettagliato).

Quando un token arriva ad un evento di fine, si verifica il risultato dell'evento (se presente) e il token viene consumato. Da notare, che è possibile avere uno o più percorsi (thread) che continuano anche dopo, che il token in un percorso ha un evento finale ed è stato consumato. Se il processo contiene ancora un token non consumato, il processo è ancora "**attivo**" e chiaramente, **dopo che tutti i percorsi "attivi" hanno raggiunto un evento di fine, il processo è completo** → notiamo, che se il processo è ben strutturato e quindi, ad ogni gateway di split corrisponde un gateway di merge, siamo sempre sicuri di creare delle parti del processo, che hanno un punto di ingresso e un punto di uscita e di conseguenza, questo in generale mi permette di avere un evento di inizio e un evento di fine → in questo caso, quindi, l'evento di fine sarà il cosiddetto **Terminate End Event**, ovvero l'evento di fine



che causerà la cessazione **immediata** dell'istanza del processo al suo livello attuale e per qualsiasi sotto-processo (anche se l'attività è ancora in corso) → chiaramente, in base alla scelta di progetto che vogliamo adottare e di conseguenza, in base al comportamento del processo, possiamo avere il Terminate End Event (nel caso in cui vogliamo che termini immediatamente l'istanza del processo) oppure possiamo scegliere di utilizzare un “normale” evento di fine (che va a consumare il token di un percorso).



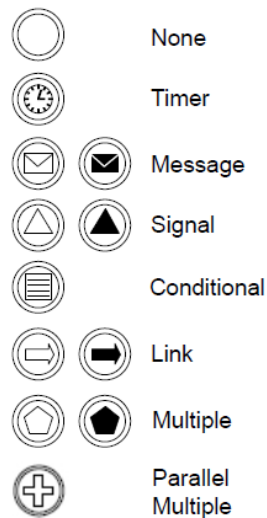
Una volta analizzati gli eventi di inizio e gli eventi di fine, passiamo all'ultima tipologia di eventi forniti dal BPMN, ovverosia gli **eventi intermedi** → troviamo gli eventi intermedi nel mezzo del processo e di conseguenza, solitamente essi hanno:

- un arco entrante;
- un arco uscente.

Quindi, un evento intermedio indica dove qualcosa accade/si verifica **dopo che un processo è iniziato e prima che sia terminato**. Inoltre, gli eventi intermedi possono essere di due tipi:

1. **eventi intermedi di catching**, ovverosia eventi intermedi che “raccolgono qualcosa”, come ad esempio: un messaggio oppure un segnale (quindi hanno il simbolo bianco). Al verificarsi di “quel qualcosa”, gli eventi intermedi lasciano passare il token e quindi, quest'ultimo (ovvero il token) procede lungo il processo. Quindi, un token che arriva ad un evento intermedio, aspetta finché non si verifica l'innesco. Successivamente, il token esce immediatamente e si sposta lungo il flusso di sequenza in uscita;
2. **eventi intermedi di throwing**, ovverosia eventi intermedi che “producono qualcosa”, come ad esempio: inviano un messaggio oppure inviano un segnale (quindi hanno il simbolo nero). In questo caso, quando arriva il token, gli eventi intermedi “producono qualcosa”.

Vediamo quali sono i possibili eventi intermedi di catching e di throwing:



(Anche in questo caso, l'evento intermedio None rappresenta un evento generico, che posso ulteriormente dettagliare con un'altra tipologia di evento intermedio).

Notiamo l'evento intermedio di tipo **Link**, che può essere utile quando si hanno processi molto grandi e noi vogliamo spezzare la rappresentazione del processo in due parti. **Gli eventi di Link, quindi, mi permettono di linkare parti diverse del processo** → quindi, gli eventi di Link vengono utilizzati in coppia: uno di throwing e uno di catching.



Possiamo notare immediatamente, che non tutte le tipologie di evento di catching hanno il corrispettivo evento di throwing (come per esempio il Timer).

Un evento intermedio di **catching**, inoltre, può:

- essere posizionato tra due attività. Per esempio:



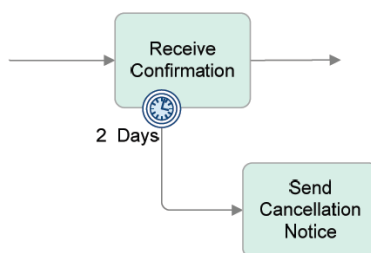
una volta eseguita l'attività di “Annunciare le questioni da discutere”, il token arriva all'evento intermedio di tipo Timer e si ferma per 6 giorni. Il processo, a questo punto, inizia a contare e una volta passati 6 giorni, allora l'evento

intermedio rilascia il token, il quale procede verso la prossima attività del processo → **questo mi permette di coordinare, dal punto di vista temporale, quanto tempo deve passare da un'attività ad un'altra.**



**Questo** (ovvero inserire un evento intermedio) **rappresenta l'unico modo, per passare dall'attività A all'attività B consumando del tempo, in quanto ricordiamo che il sequence flow non consuma tempo.**

- essere posizionato al **bordo** di un'attività (possono essere messi al bordo solamente gli eventi di catching). Vediamo un esempio:



abbiamo, quindi, che il token arriva all'attività "Ricezione conferma" e si inizia ad eseguire tale attività. Quando viene eseguita questa attività, abbiamo che:

- il token è fermo nell'attività;
- un altro token, che aspetta sull'evento intermedio.

Se l'attività non è terminata prima che sia scaduto il Timer (in questo caso 2 giorni), allora il token dell'evento intermedio si attiva e quindi si procede verso l'attività di "Inviare notifica di cancellazione". Se, invece, l'attività è terminata prima della scadenza del Timer, allora il token procede sul cammino di uscita "normale".

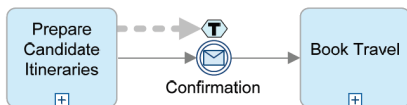


Questo evento intermedio posizionato sul bordo di un'attività, **mi serve per indicare che stiamo impiegando troppo tempo per eseguire l'attività** e che quindi, dobbiamo interrompere l'esecuzione di quella attività e proseguire con il processo.

Gli eventi intermedi di throwing, invece, sono quelli che nel momento in cui arriva il token, **immediatamente** "fanno qualcosa" (come ad esempio: inviare un messaggio oppure inviare un segnale) e successivamente il token procede.

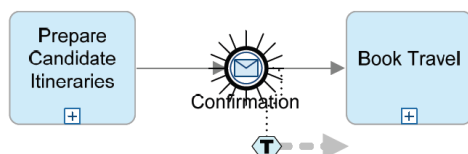
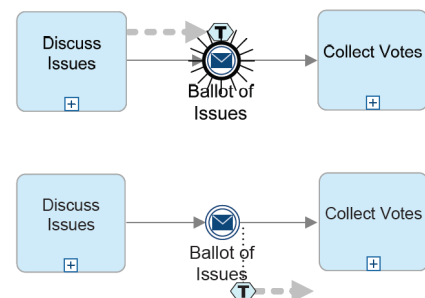


Da notare, che il messaggio che viene immediatamente inviato, è esattamente il messaggio di quando abbiamo parlato delle pools, ovvero: il messaggio è inviato da un processo ad un altro processo e quindi, NON è un messaggio interno al processo.



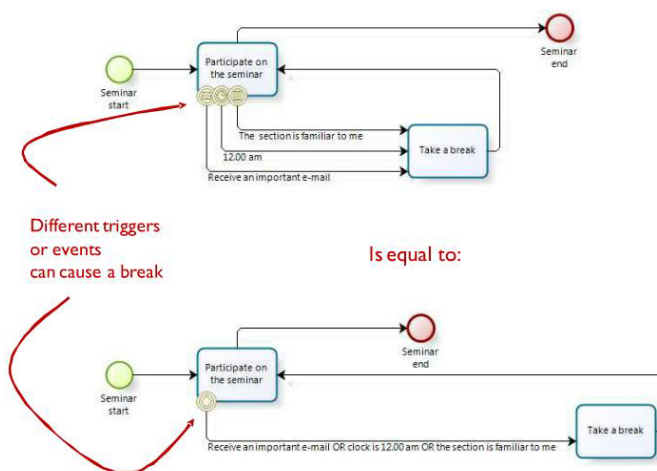
Quando un token arriva ad un evento intermedio di catching di tipo Messaggio, il token si blocca (si mette in pausa) sull'evento intermedio e riparte alla ricezione del messaggio, proseguendo lungo l'arco uscente.

Se invece, l'evento intermedio è di tipo throwing, il token arriva all'evento intermedio e il messaggio viene inviato immediatamente ad un processo specifico e il token prosegue lungo l'arco uscente. In questo caso, quindi, il token non aspetta/non si mette in pausa.



Se il token è in attesa presso l'evento intermedio di catching e il messaggio arriva, allora l'evento intermedio si attiva e il token prosegue.

Vediamo, infine, un esempio per capire gli eventi intermedi multipli:



Analizziamo a questo punto i **gateway** → abbiamo già parlato dei:

- gateway esclusivi;
- gateway paralleli;
- gateway inclusivi.

Andiamo a vedere, allora, altre due tipologie di gateway, ovvero:

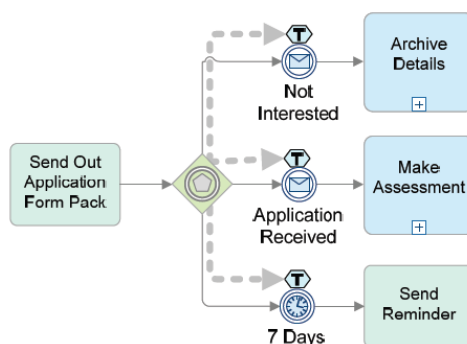
1. **gateway basato su eventi** → la sua rappresentazione è poco intuitiva, in quanto:
  - a. viene rappresentato come un diamante, con all'interno l'evento intermedio complesso;
  - b. ha una serie di archi uscenti, dove su ognuno dei quali è posizionato un evento intermedio di tipo catching.

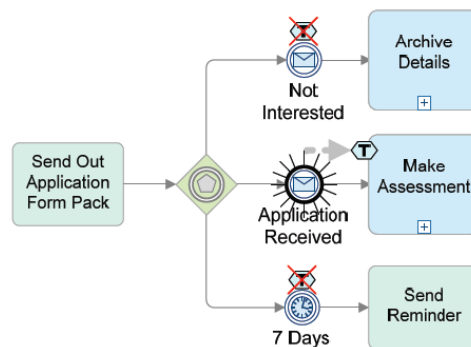
Il token, quindi, arriva al gateway basato su eventi, il quale (ovvero il gateway) manda un token su ogni arco uscente (nell'esempio sotto, vengono mandati 3 token) e tali token, sono in attesa sugli eventi intermedi di tipo catching. Da sottolineare il fatto che:



Il primo evento intermedio di tipo catching che si verifica sarà quello che farà procedere il token, mentre gli altri token generati (per gli archi uscenti) vengono consumati e quindi tali cammini non vengono eseguiti → questo ci fa capire, che **solamente quando l'evento intermedio di tipo catching si attiva, so effettivamente quale percorso devo scegliere. Prima che l'evento intermedio si attivi, effettivamente NON sappiamo quale cammino percorrere.**

Vediamo un esempio per capire meglio:

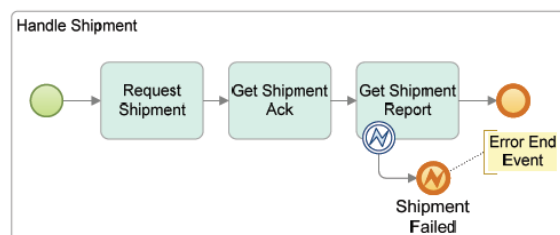




## 2. gateway complesso (che non abbiamo analizzato).

### Gestione degli errori e delle eccezioni

Possiamo avere degli eventi finali oppure degli eventi intermedi, che ci permettono di catturare un errore, dove l'errore può essere di qualsiasi tipo, come per esempio: errore nella fattura, errore nella gestione di un ordine. Vi è quindi, un problema che non mi permette di completare l'attività e di conseguenza la vogliamo interrompere. **Possiamo, allora, utilizzare un evento intermedio per interrompere un'attività, questo (ovvero l'evento intermedio) è attaccato al bordo dell'attività.**



Vediamo che abbiamo l'evento intermedio di errore. Succede il fallimento dell'attività e l'evento intermedio si attiva e quindi non si procede con l'esecuzione del processo verso l'evento di fine "classico", bensì si procede verso l'evento finale di gestione degli errori.

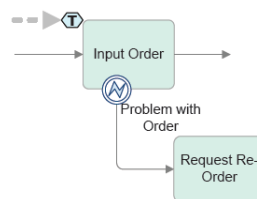
Oppure, **possiamo avere un evento finale, il quale ci permette di segnalare che in questo momento il processo viene terminato, perchè si è verificato un errore.** Quando si verifica un errore, tutto il "lavoro" fatto nel processo si blocca e quindi si segnala, che c'è qualcosa che non funziona e che conseguentemente deve essere gestito. Quindi, segnalare un errore è diverso rispetto a terminare il processo con un evento di fine.



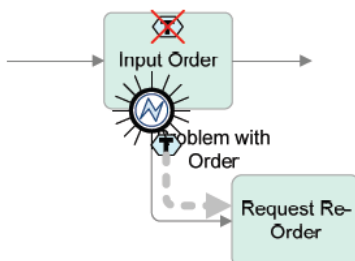
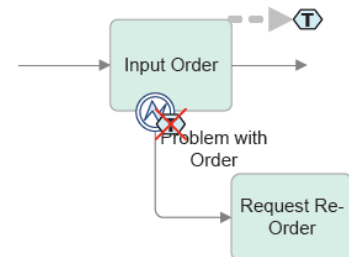
Lo scopo di visibilità dell'evento intermedio di gestione degli errori, è limitato al processo e di conseguenza non lanciamo un segnale ad un altro processo. La visibilità dell'errore, quindi, è limitato al processo e conseguentemente è limitata al processo padre (nel caso si abbiano sotto-processi) e gli altri processi non hanno visibilità dell'errore.

Come sempre, quando abbiamo un evento al bordo, abbiamo due token:

1. un token per l'attività in esecuzione;
2. un token per l'evento al bordo.



Se l'attività termina correttamente, allora, il token sull'evento al bordo viene consumato e quindi, si prosegue con la "normale" esecuzione del processo.



Se, invece, l'attività non termina correttamente, il token dell'attività in esecuzione viene consumato e viene attivato il token dell'evento al bordo e si prosegue con quel cammino.

Nel BPMN possiamo gestire anche delle **transazioni** → una transazione è un'attività che deve essere eseguita completamente, altrimenti non deve nemmeno essere eseguita (un esempio di transazione è il bonifico bancario: il bonifico ha successo se dal conto A arrivano al conto B. Se i soldi non arrivano al conto B, essi non devono partire dal conto A) → quindi, **se eseguiamo solamente un pezzo dell'attività e**

non arriviamo a compimento, dobbiamo eliminare il pezzo dell'attività che è stato eseguito.

Per gestire le transazioni utilizziamo:

- gli eventi di gestione degli errori;
- gli **eventi di cancellazione** → l'evento intermedio di cancellazione è progettato per gestire una situazione in cui una transazione viene annullata. Gli eventi intermedi di cancellazione possono solamente catturare una transazione di annullamento di una transazione. È compito poi, dell'evento finale di cancellazione lanciare l'annullamento.



- gli **eventi di compensazione** → esistono due tipi di eventi intermedi di compensazione:
  - throwing;
  - catching.

L'evento intermedio di compensazione di tipo catch può essere utilizzato solo da un'attività. Tuttavia, l'evento intermedio di tipo throw è utilizzato nel flusso normale. L'evento finale di compensazione indica che la fine di un percorso di processo, comporta l'attivazione di una compensazione.

Nella definizione dell'evento finale di compensazione, il nome di un'attività può essere identificato come l'attività che deve essere compensata.

Se l'attività nominata è stata completata e ha un evento intermedio di compensazione allegato, quell'attività sarà compensata.





# Data Warehousing

Arrivati a questo punto, iniziamo a parlare dei sistemi di supporto alle decisioni. Ci ricordiamo la **Piramide di Anthony** (in particolar modo, nella fascia medio-alta) e nel **portafoglio istituzionale**, abbiamo detto che i sistemi informativi, che possono dare supporto (alle varie organizzazioni) nelle decisioni, sono effettivamente i sistemi di supporto alle decisioni → questi sistemi sfruttano il fatto, che attualmente le organizzazioni hanno accumulato una grande quantità di dati e analizzati tali dati, al fine di essere di supporto alle decisioni. Di questo argomento, in realtà, ne abbiamo già parlato, quando abbiamo parlato di come l'avvento dell'informatica abbia impattato nell'azienda. In particolare, avevamo detto che la funzione svolta dalle basi di dati in ambito aziendale, è stata fino a qualche anno fa solo quella di memorizzare dati operazionali, ossia dati generati da operazioni svolte all'interno dei processi gestionali. Quindi,

**l'informatica veniva vista come una scienza di supporto, che permette di rendere più rapide ed economiche le operazioni di gestione delle informazioni, ma che non crea di per sé ricchezza.**

Il ruolo dei Sistemi Informatici è radicalmente cambiato dai primi anni '70 a oggi. I sistemi informatici si sono trasformati da semplici strumenti per migliorare l'efficienza dei processi a elementi centrali dell'organizzazione aziendale, in grado di rivoluzionare la struttura dei processi aziendali. Per questo motivo, si dice che l'informatica ha un duplice ruolo, ossia:

1. **Tecnologia di supporto alla gestione del sistema informativo;**
2. **Disciplina organizzativa che influenza i processi, servizi e la struttura aziendale** → questo ci fa capire, che l'informatica non è più solamente un supporto, bensì essa influenza pesantemente il modo di operare dell'organizzazione.

---

Riprendiamo un attimo il concetto di portafoglio applicativo → Il termine portafoglio applicativo indica l'insieme delle applicazioni e delle tecnologie presenti in un'organizzazione. Può essere diviso in tre segmenti principali:

- **Portafoglio direzionale** → è l'insieme delle applicazioni utilizzate dai manager aziendali per:
  - Analizzare lo stato dell'azienda;

- Prendere decisioni rapide;
- Prendere le decisioni migliori.

Si parla anche di piattaforma per la **Business Intelligence**, ossia la disciplina che consente a chi deve decidere in azienda di capire, attraverso soluzioni software, i fattori chiave del business e conseguentemente di **prendere le migliori decisioni in quel momento**. Si parla di piattaforma (per la Business Intelligence) poiché per consentire ai manager analisi potenti e flessibili, è necessario definire un'apposita infrastruttura hardware e software di supporto composta da:

- Hardware dedicato;
- Infrastrutture di rete;
- DBMS;
- Software di back-end;
- Software di front-end.

Il ruolo chiave di una piattaforma di Business Intelligence (e in senso più generale anche di un sistema di supporto alle decisioni) è la trasformazione dei **dati aziendali** in **informazioni** fruibili a diversi livelli di dettaglio.

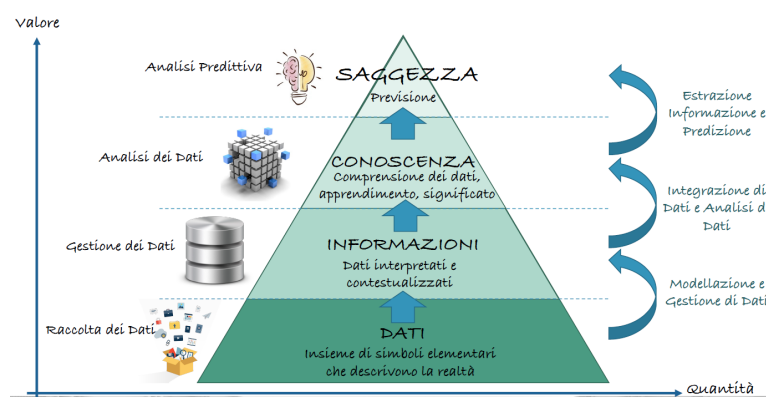
Capiamo, allora, che il concetto di dato e il concetto di informazione sono due concetti ben distinti: l'informazione è un bene a valore crescente, necessario per pianificare e controllare le attività aziendali con efficacia. Essa, quindi, costituisce la materia prima che viene trasformata dai sistemi informativi → **un dato, quindi, diventa informazione quando il mero dato (che può essere un numero, una stringa o effettivamente un insieme di dati) di cui sono in possesso, acquisisce un significato (ovverosia una semantica). Quindi, se siamo in grado di associare al dato il suo significato, allora siamo in grado di estrapolare l'informazione.**

Spesso, però, la disponibilità di troppi dati in azienda, rende arduo (se non impossibile) estrapolare le informazioni veramente importanti e di conseguenza i dati stessi hanno meno valore. Capiamo, allora, che il valore dei dati aumenta, man mano che riesco a sfoltire/sfrondare i dati stessi, ovverosia capire cosa effettivamente mi interessa e cosa non mi interessa per prendere una determinata decisione. Per ogni azienda, quindi, è fondamentale poter disporre in maniera rapida e completa delle informazioni necessarie al processo decisionale. Sorge a questo punto spontanea la domanda: **"Quando effettivamente otteniamo un valore**

**avendo a disposizione tantissime informazioni?"** Otteniamo un valore, quando effettivamente riusciamo a:

- **Selezionare** le informazioni che ci interessano, a seconda del contesto che ci interessa;
- **Aggregare** le informazioni (che abbiamo selezionato);
- **Organizzare** le informazioni;
- **Capire** le informazioni.

Quindi, una volta che abbiamo selezionato, aggregato, organizzato e capito le informazioni che ci interessano, riusciamo anche ad ottenere delle indicazioni strategiche, che saranno proprio quelle di supporto alle mie decisioni. Quindi maggiori sono i dati, minori è il loro valore, perchè le informazioni che riusciamo a ricavare da essi sono minori e di conseguenza anche le indicazioni strategiche saranno minori. Quindi, i dati e conseguentemente le informazioni **hanno un "valore minore" per prendere la decisione che ci interessa. Il modo con cui seleziono e gestisco le informazioni, quindi, è quello che mi permette renderle utili** (da notare, che non dobbiamo fraintendere il fatto, che avere tante informazioni vuol dire avere un basso valore. Avere tante informazioni è un grande potenziale, ma con il termine "basso valore" intendiamo, che le informazioni non possono essere utilizzate direttamente per prendere una decisione). Questo aumento esponenziale dei volumi di dati, fa sì che il ruolo dell'informatica (all'interno delle organizzazioni) diventi primario, in quanto per l'organizzazione diventa indispensabile avere dei sistemi informatici, in grado di trasformare i dati raccolti in informazioni utili. Per capire il ruolo dell'informatica all'interno delle organizzazioni è stata realizzata la **piramide dell'informazione**:

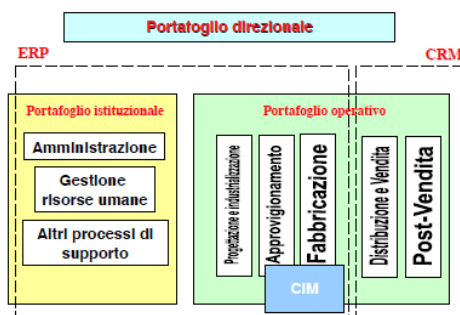


Al primo livello troviamo i dati grezzi, ovverosia i dati raccolti durante tutte le operazioni quotidiane. Nel momento in cui vogliamo interpretare tali dati e contestualizzarli, possiamo utilizzare delle classiche basi di dati relazionali (ovverosia i DBMS relazionali). Capiamo, allora, che già nel momento in cui gestisco i dati tramite un DBMS, otteniamo delle informazioni, in quanto non abbiamo più i dati disordinati, bensì li abbiamo organizzati in una base di dati e quindi abbiamo trasformato i dati raccolti in informazione. Il passo successivo è di ridurre ulteriormente l'informazione (ovverosia vogliamo capire il perchè succedono determinate cose) e quindi dalla nostra informazione, abbiamo estrapolato della conoscenza. Da notare, che la conoscenza è sempre un'informazione, ma mi permette di capire il perchè succedono determinate cose all'interno dell'organizzazione (ad esempio, perchè nel mese di novembre le vendite diminuiscono? Un motivo può essere che non siamo in periodo di saldi) → questo ci permette di comprendere i dati e le informazioni che abbiamo a disposizione e conseguentemente capire cosa sta succedendo all'interno dell'organizzazione → la conoscenza, quindi, ci permette di avere un vantaggio/un'informazione importante nel prendere una decisione. Quindi, i sistemi di supporto alle decisioni (e in particolare i Data Warehouse) mi permettono di fare analisi dei dati.

Da notare, inoltre, che una volta che si ha a disposizione della conoscenza (e quindi si è riusciti a fare un'analisi dei dati), essa ci permette di fare delle previsioni, in quanto se sappiamo come sono andate le cose nel passato, posso prevedere come andrà nel futuro e conseguentemente posso adottare dei comportamenti, che mi permettono di cambiare (in maniera positiva) cosa accadrà in futuro (ad esempio, so dagli scorsi anni, che a Novembre le vendite calano. Grazie alla comprensione dei dati riesco a capire il perchè le vendite calano e posso fare una previsione. Una volta fatta la previsione, posso adottare delle azioni che mi permettono di alzare le vendite nel mese di Novembre, come ad esempio iniziare gli sconti, in modo tale da migliorare la situazione).

- **Portafoglio operativo** → comprende le applicazioni informatiche per i processi primari dell'azienda;
- **Portafoglio istituzionale** → comprende le applicazioni informatiche per i processi di supporto, quali amministrazione, gestione delle risorse umane, contabilità.

Uno schema generale del portafoglio applicativo è dato dalla seguente immagine:



Quello che ci interessa (in questo momento) dello schema sopra, è che i sistemi di supporto alle decisioni si collocano nella parte alta del portafoglio direzionale (e nella parte medio-alta della piramide di Anthony).

Adesso che abbiamo fatto questa panoramica generale, concentriamoci sui sistemi di supporto alle decisioni. Quest'ultimi nascono negli anni '80 **per riuscire a descrivere il passato e quindi per riuscire a creare uno storico delle informazioni, che man mano si andava ad accumulare**. Quindi, i sistemi di supporto alle decisioni permettevano di ridurre i costi e di descrivere i problemi. Al giorno d'oggi, il ruolo dei sistemi di supporto alle decisioni, oltre che a:

- creare uno storico delle informazioni;
- ridurre i costi;
- descrivere i problemi.

Ma ci permettono anche, di cercare di **anticipare il futuro** e quindi:

- prendere delle decisioni;
- fare delle previsioni;
- capire cosa sta succedendo, in modo tale da:
  - essere di supporto alle decisioni da prendere nel futuro;
  - aumentare i profitti;
  - suggerire i cambiamenti da apportare.

Nel Passato	Nel Futuro
Descrivere il passato	Anticipare il futuro
Ridurre i costi	Aumentare i profitti
Descrivere i problemi	Suggerire i cambiamenti da apportare



Quindi, **i sistemi di supporto alle decisioni sono l'insieme di tutte le tecniche e degli strumenti informatici, che ci permettono di estrapolare informazioni (dove con il termine "estrapolare", intendiamo proprio cercare da una grande quantità di dati e informazioni, di estrapolare solamente l'informazione che ci serve per prendere una decisione).**

Notiamo, inoltre, che il progettista del sistema informativo impatta sul tipo di analisi che riusciamo a fare. Analizziamo meglio questo concetto: il progettista della base di dati, il quale decide cosa tenere e cosa non tenere, impatta sulle informazioni che andremo a raccogliere. Per il progettista del Data Warehouse vale lo stesso, in quanto nel Data Warehouse andiamo a raccogliere (a fattor comune) l'informazione che abbiamo a disposizione. In questo contesto, risulta importante anche la figura dell'utente del Data Warehouse, in quanto l'utente stesso "ci mette del suo" nel momento in cui va ad interrogare e a interagire con il sistema di supporto alle decisioni, perchè a fronte delle sue competenze manageriali o economiche (e quindi non necessariamente competenze informatiche) riesce a capire quali informazioni/concetti andare ad analizzare per capire effettivamente cosa sta succedendo.

Lo scenario tipico di utilizzo di un Data Warehouse è quello di una grande azienda, con numerose filiali, i cui dirigenti desiderano quantificare e valutare il contributo dato da ciascuna di esse al rendimento commerciale globale dell'impresa (da notare, che questo era lo scenario tipico, che valeva all'esordio del Data Warehouse, mentre oggi il Data Warehouse sta diventando uno strumento che non richiede di essere una grande azienda, bensì sta diventando un sistema informativo largamente utilizzato per organizzare l'informazione anche di medie imprese). Avere uno scenario (tipico) di questo tipo, vuol dire che le varie filiali hanno sistemi informativi diversi (come per esempio basi di dati diverse). Nonostante ciò, tutte queste differenze devono essere portate a fattor comune e vanno integrate.



Il Data Warehouse, quindi, è un sistema di supporto alle decisioni e in particolar modo, il Data Warehouse è un **raccoglitore/contenitore** di informazioni, le quali hanno la caratteristica di essere informazioni provenienti da sorgenti eterogenee. Il Data Warehouse, quindi:

- **raccoglie**;
- **integra**;
- **riorganizza** i dati provenienti da sorgenti eterogenee di varia natura (come per esempio: DB relazionali, DB non-relazionali, documenti XML).

Una volta che ho raccolto, integrato e riorganizzato i dati, quest'ultimi li ho a disposizione nel Data Warehouse, in modo tale da riuscire a fare analisi e valutazioni.

Notiamo un ulteriore aspetto: Come il progettista della base di dati decide (a fronte dei requisiti) quali dati raccogliere e come organizzarli nel Database, allo stesso modo c'è tutta una metodologia di progettazione del Data Warehouse, per capire quali dati (rispetto a tutti quelli che abbiamo a disposizione) sono "interessanti" e conseguentemente devono confluire nel Data Warehouse. Le sorgenti, quindi, vengono analizzate e successivamente integrate.

## OLTP e OLAP

L'elaborazione di tipo **analitico**, che viene fatta nel Data Warehouse, prende il nome di **OLAP (On-Line Analytical Processing)**. Capiamo immediatamente, che il termine "analitico" proviene da "analisi", in quanto (come abbiamo detto) le informazioni presenti nel Data Warehouse vengono utilizzate per fare analisi. Quando, invece, eravamo al livello della base di dati transazionale, utilizzavamo i dati legati al concetto di "transazioni" e quindi utilizzavamo un'elaborazione legata alle transazioni, che prende il nome di **OLTP (On-Line Transactional Processing)**. È importante separare i concetti di OLAP e OLTP, in quanto avere un sistema informatico che permette di fare sia interrogazioni OLAP sia interrogazioni OLTP significa:

- avere un sistema con delle prestazioni scadenti;
- andare a fare analisi su molteplici basi di dati, che possono contenere dati eterogenei e conseguentemente difficili da analizzare.



**La logica, quindi, di avere un Data Warehouse in cui travasare le informazioni delle mie varie sorgenti eterogenee, è proprio quella di tenere separata la parte analitica dalla parte transazionale, in modo tale da costruire un raccoglitore di informazioni, che integri i dati provenienti da fonti di varia natura, li organizzi in una forma appropriata e li renda disponibili per scopi di analisi e valutazione finalizzate alla pianificazione e al processo decisionale.**

Notiamo, inoltre, che se facessimo analisi sui dati contenuti nella nostra base di dati, non avremmo lo storico dei dati, in quanto la base di dati è una "fotografia" dei dati in questo preciso istante di tempo. Infatti, se ad esempio modifichiamo un'informazione con un'altra, l'informazione vecchia viene persa → **la logica del Data Warehouse, quindi, è anche di storicizzare le informazioni e quindi avere nel Data Warehouse lo storico delle informazioni e questo ha un ruolo fondamentale, nel momento in cui dobbiamo prendere una decisione.** In particolare, il Data Warehouse viene utilizzato in diversi ambiti:

- Commercio → per fare analisi delle vendite e dei reclami, controllo di spedizioni e inventari, cura del rapporto con i clienti;
- Manifattura → controllo dei costi di produzione, supporto fornitori e ordini;
- Trasporti → gestione del parco mezzi;
- Sanità → analisi dei ricoveri e dimissioni, contabilità per centri di costo.

Sorge a questo punto spontanea la domanda: **"Che cos'è il Data Warehousing?"** **Il Data Warehousing è una collezione di metodi, tecnologie e strumenti di ausilio al knowledge worker (dirigente, amministratore, gestore, analista), per condurre analisi dei dati finalizzate all'attuazione di processi decisionali e al miglioramento del patrimonio informativo** → da notare, che solitamente chi utilizza un Data Warehouse non ha grandi competenze informatiche, ovverosia che non necessariamente la sua professionalità è il campo informatico, ma bensì **la sua professionalità riguarda il dominio applicativo e le regole del dominio applicativo stesso**, ovverosia l'utente del



Data Warehouse conosce che tipo di informazione sta interrogando. Quindi, quando si progetta un Data Warehouse si deve tenere conto del fatto, che l'utente solitamente ha competenze esterne al mondo informatico e di conseguenza, ha bisogno del supporto del sistema informatico per rispondere a delle domande che si fa (questo non vale per gli utenti che utilizzano una base di dati, in quanto per scrivere delle query, per estrapolare le informazioni di cui necessitano, gli utenti hanno bisogno di competenze informatiche). In questo senso, **le caratteristiche del processo di warehousing sono:**

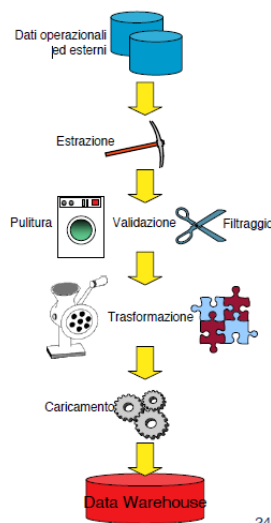
- **Accessibilità** a utenti con conoscenze limitate di informatica e strutture dati (quindi il Data Warehouse deve essere accessibile **anche** agli utenti che non sanno scrivere una query SQL);
- **Integrazione dei dati** sulla base di un modello standard d'impresa, ovverosia si parte da tutti i dati disponibili, decido quali di questi dati sono interessanti rispetto ad un determinato aspetto e li vado ad integrare nel Data Warehouse. Capiamo, allora, che per quanto riguarda l'integrazione dei dati gioca un ruolo fondamentale il progettista, il quale deve capire quali sono le esigenze e conseguentemente far confluire nel Data Warehouse le informazioni utili d'interesse (quindi non tutte le informazioni disponibili sono utili e quindi non tutte le informazioni devono finire nel Data Warehouse);
- **Flessibilità di interrogazione** per trarre il massimo vantaggio dal patrimonio informativo esistente, ovverosia permettere all'utente di maneggiare e analizzare l'informazione in modo interattiva;
- **Sintesi** per permettere analisi mirate ed efficaci e per riuscire a fare ciò vengono **aggregate** le informazioni (quindi, quando parliamo di sintesi, non intendiamo che andiamo ad eliminare le informazioni, bensì esse vengono aggregate, in modo tale da riuscire ad effettuare analisi efficaci e mirate);
- **Rappresentazione multidimensionale** per offrire all'utente una visione intuitiva ed efficacemente manipolabile delle informazioni;
- **Correttezza e completezza** dei dati integrati:
  - Completezza → perchè nel Data Warehouse inserisco tutti i dati che mi interessano;
  - Correttezza → perchè nel momento in cui travasiamo i dati nel Data Warehouse, dobbiamo "ripulire" i dati (per esempio, immaginiamoci di avere in un DB l'indirizzo di residenza di quando abitavo coi i miei

genitori, mentre in un altro DB ho l'indirizzo di residenza attuale di casa mia. Nel momento in cui andiamo ad inserire i dati nel Data Warehouse, dobbiamo risolvere queste incongruenze).

Quindi, al centro del processo di data warehousing vi è il Data Warehouse, il quale è un contenitore di dati che si fa garante dei requisiti appena sopra citati (correttezza e completezza, accessibilità...). In particolare, **un Data Warehouse è una collezione di dati di supporto per il processo decisionale, che presenta le seguenti caratteristiche:**

- **è orientato ai soggetti di interesse** → quindi, il Data Warehouse si focalizza su un certo concetto e su chi dovrà accedere a tale informazione (quindi, non è più orientato all'applicazione, come poteva esserlo invece una base di dati). Nei Data Warehouse, quindi, **l'accesso è sui soggetti**, mentre nei DB relazionali l'accento è sulle applicazioni;
- **è integrato e consistente** → questa caratteristica ha a che fare con la correttezza e completezza dei dati integrati. In particolare:
  - integrato → perchè i dati provengono da fonti eterogenee;
  - consistente → perchè i dati prima di essere travasati nel Data Warehouse, hanno bisogno di un processo di ripulitura ed inoltre, i dati devono essere inseriti nel Data Warehouse con lo stesso formato (per esempio, tutti i numeri di telefono devono avere lo stesso formato).

Il DW si appoggia a più fonti di dati eterogenee: dati estratti dall'ambiente di produzione, e quindi originariamente archiviati in basi di dati aziendali, o addirittura provenienti da sistemi informativi esterni all'azienda. Di tutti questi dati, il DW restituisce una visione unificata. Questi concetti possono essere osservati meglio con la seguente immagine:



Possiamo vedere, che partiamo da dati operazionali ed esterni (che sono le nostre sorgenti eterogenee) e dobbiamo arrivare a caricare, sul Data Warehouse, dei dati integrati e consistenti. Per riuscire a fare ciò, vengono utilizzati degli **strumenti di ETL (Extraction, Transformation and Loading)**. In particolare:

- **estrarre** i dati → vuol dire capire a partire dai dati operazionali ed esterni, rispetto a tutta l'informazione che abbiamo a disposizione, qual è l'informazione che ci interessa travasare nel Data Warehouse;
- una volta estratti, i dati devono essere **ripuliti, validati, filtrati e trasformati**.

Questi passaggi, ci consentono di ottenere l'informazione integrata e consistente (di cui parlavamo prima).

Infine i dati devono essere **caricati** nel Data Warehouse e nel momento in cui i dati vengono effettivamente caricati, avremo una visione unificata (orientata ad un aspetto che ci interessa analizzata) delle fonti di dati eterogenee da cui siamo partiti.

- **è rappresentativo dell'evoluzione temporale** → dato che abbiamo detto, che nel Data Warehouse vogliamo lo storico dei dati (e non la "fotografia" dei dati come nel caso di una base di dati). **Periodicamente (e ovviamente questa periodicità dipenderà da quanto le nostre basi di dati cambiano nel tempo)**, infatti, andiamo a travasare nel Data Warehouse i dati delle nostre varie sorgenti e quindi ogni volta vado ad accumulare dei dati (da notare, che non cancelliamo e non sovrascriviamo mai nulla). Chiaramente, questa storicità dei dati mi permette di fare analisi temporali sui dati, le quali

sono fondamentali per riuscire a prendere decisioni per il futuro. Per riassumere questi concetti, possiamo osservare la seguente immagine:



- **non volatile** → nel Data Warehouse abbiamo grandi volumi di dati ed in particolare, all'interno del Data Warehouse non facciamo operazioni di cancellazione e/o di modifica delle informazioni, bensì facciamo solamente operazioni di append (ovverosia di aggiunta).

Vediamo a questo punto, le **differenze** tra le interrogazioni OLTP e le interrogazioni OLAP. In particolare, abbiamo che:

- **interrogazioni OLTP** → esse sono le interrogazioni tipiche delle basi di dati ed eseguono transazioni, che leggono e scrivono un **ridotto** numero di record da diverse tabelle legate da semplici relazioni. Quindi, il nucleo sostanziale del carico di lavoro è "congelato" all'interno dei programmi applicativi;
- **interrogazioni OLAP** → esse sono le interrogazioni analitiche, che quindi andiamo a fare sui dati memorizzati in un Data Warehouse. In particolare, tali interrogazioni effettuano un'analisi dinamica e multidimensionale, che richiede la scansione di un'**enorme** quantità di record (perché abbiamo la storicità dei dati) per calcolare un insieme di dati numerici di sintesi che quantifichino le prestazioni dell'azienda. Quindi, **l'interattività è una caratteristica irrinunciabile delle sessioni di analisi e fa sì che il carico di lavoro effettivo vari continuamente nel tempo** (ovverosia nelle interrogazioni OLAP è fondamentale permettere a chi interroga, di seguire il flusso di pensiero/le idee che gli vengono in mente, vedendo man mano i risultati delle interrogazioni che sta facendo).

Riassumendo, quindi, abbiamo che:

	Database operazionali	Data warehouse
utenti	migliaia	centinaia
carico di lavoro	transazioni predefinite	interrogazioni di analisi <i>ad hoc</i>
accesso	a centinaia di record, in lettura e scrittura	a milioni di record, per lo più in lettura
scopo	dipende dall'applicazione	supporto alle decisioni
dati	elementari, sia numerici sia alfanumerici	di sintesi, prevalentemente numerici
integrazione dei dati	per applicazione	per soggetto
qualità	in termini di integrità	in termini di consistenza
copertura temporale	solo dati correnti	dati correnti e storici
aggiornamenti	continui	periodici
modello	normalizzato	denormalizzato, multidimensionale
ottimizzazione	per accessi OLTP su una frazione del database	per accessi OLAP su gran parte del database
sviluppo	a cascata	iterativo

## Architetture dei Data Warehouse

Ricordiamoci, che la logica è di partire da sorgenti di dati eterogenee e quello che **vogliamo è che tutte queste sorgenti di dati, dopo il processo di ETL, finiscano in un repository, ovvero sia il nostro Data Warehouse**. Da notare, che nel momento in cui andiamo a costruire l'architettura del Data Warehouse, dobbiamo tenere conto di molteplici requisiti:

- **Separazione** → l'elaborazione analitica e quella transazionale devono essere mantenute il più possibile separate (quindi dovremmo avere la possibilità di eseguire le classiche query transazionali sui dati transazionali e le query analitiche sui dati analitici);
- **Scalabilità** → l'architettura hardware e software deve poter essere facilmente ridimensionata a fronte della crescita nel tempo dei volumi di dati da gestire ed elaborare (dato che con il Data Warehouse teniamo una traccia dei dati) e del numero di utenti da soddisfare;
- **Estendibilità** → nel Data Warehouse deve essere possibile accogliere nuove applicazioni e tecnologie di analisi (quali per esempio, le tecniche di Data Mining), senza riprogettare integralmente il sistema;
- **Sicurezza** → esattamente come avviene per le basi di dati, il controllo sugli accessi è essenziale a causa della **natura strategica** dei dati memorizzati nel Data Warehouse (da notare, che i dati memorizzati nel DW hanno una natura strategica, in quanto essi sono stati opportunamente selezionati, integrati e contestualizzati, che mi permettono di prendere delle decisioni strategiche);
- **Amministrabilità** → la complessità dell'attività di amministrazione non deve risultare eccessiva, ovvero sia che l'attività di amministrazione (quindi chi

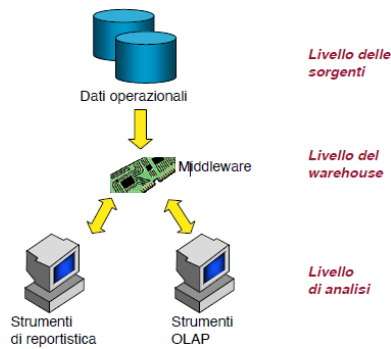
effettivamente gestisce il Data Warehouse) non deve risultare complessa.

Una volta analizzati i requisiti, passiamo ad analizzare singolarmente le possibili architetture di un Data Warehouse. In particolare, quindi, abbiamo:

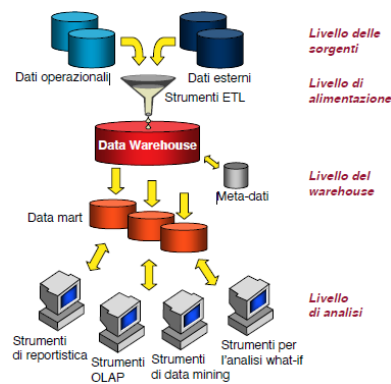
- **Architettura ad 1 livello** → in questa tipologia di architettura abbiamo:
  - il livello delle **sorgenti** → ovverosia l'insieme dei dati operazionali, i quali sono le nostre sorgenti eterogenee (di cui parlavamo nel capitolo precedente), che sono effettivamente quelle che voglio travasare in un Data Warehouse;
  - il livello del **warehouse** → nell'architettura ad 1 livello, abbiamo che in pratica il livello del Data Warehouse viene implementato e simulato tramite un **middleware**. Quest'ultimo, quindi, va a simulare il nostro Data Warehouse → attraverso questa soluzione, però, **capiamo immediatamente che viene a mancare il requisito della separazione**, in quanto **simuliamo** l'esistenza di un Data Warehouse e non lo andiamo effettivamente a creare. Di conseguenza, non andiamo a:
    - ripulire i dati;
    - analizzarli;
    - integrarli;
    - caricarli.

bensi abbiamo semplicemente le nostre basi di dati e abbiamo un livello di middleware, il quale accede alle basi di dati e mi permette di interrogare i dati come se fossero dentro un Data Warehouse. Tutto questo per dire, che **non viene garantito il requisito di separazione, in quanto con il livello middleware, sia le interrogazioni OLPT sia le interrogazioni OLAP interagiscono con le basi di dati. Inoltre, sottolineiamo il fatto, che nel momento in cui il DBMS non funziona, non possiamo fare né query transazionali né query analitiche;**

- il livello di **analisi** → comprende strumenti di reportistica (che possiamo utilizzare sulle nostre basi di dati relazionali) e strumenti OLAP. Quindi, possiamo dire che il livello di analisi, rappresenta quello che riusciamo a fare sui dati del Data Warehouse, ovverosia rappresenta l'insieme degli strumenti, che utilizziamo per analizzare le informazioni contenute nel Data Warehouse.



- **Architettura a 2 livelli** → essa può essere osservata nella seguente immagine:



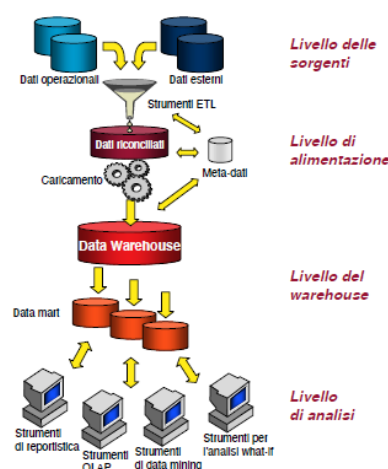
possiamo osservare, che anche in questa architettura abbiamo:

- il livello delle **sorgenti**;
- il livello di **alimentazione** → viene messo in atto dagli strumenti ETL;
- il livello di **warehouse** → in questo livello viene rappresentato anche il **Data mart**, il quale è un sotto-insieme o un'aggregazione dei dati presenti nel Data Warehouse. Per esempio, se abbiamo un Data Warehouse per i dati di un'azienda, possiamo avere un Data mart per i dati relativi alla parte di marketing, un altro Data mart per la parte della logistica e un altro Data mart per gli indicatori di funzionalità dell'azienda. L'insieme dei vari Data mart vanno a formare il Data Warehouse;
- il livello di **analisi** → con questa tipologia di architettura, in cui effettivamente abbiamo il Data Warehouse, abbiamo più strumenti di analisi. Possiamo osservare che abbiamo: strumenti di reportistica,

strumenti OLAP, strumenti di data mining e strumenti per l'analisi what-if.

L'architettura a 2 livelli gode di molteplici **vantaggi**, che sono:

- a livello del warehouse è continuamente disponibile informazione di buona qualità anche quando, per motivi tecnici oppure organizzativi, è temporaneamente precluso l'accesso alle sorgenti;
  - l'interrogazione analitica effettuata sul DW non interferisce con la gestione delle transazioni a livello operativo → quindi **con l'architettura a 2 livelli, viene garantito il requisito della separazione tra OLTP e OLAP e quindi possiamo interrogare separatamente le mie basi di dati e il mio Data Warehouse;**
  - l'organizzazione logica del DW è basata sul modello multidimensionale, mentre le sorgenti offrono in genere modelli relazionali o semistrutturati;
  - c'è una discordanza temporale e di granularità tra sistemi OLTP, che trattano dati correnti e al massimo livello di dettaglio, e sistemi OLAP che operano su dati storici e di sintesi;
  - a livello del warehouse è possibile impiegare tecniche specifiche per ottimizzare le prestazioni per applicazioni di analisi e reportistica.
- **Architettura a 3 livelli** → in questa tipologia di architettura, abbiamo che il livello in più (rispetto all'architettura a 2 livelli) è il **livello dei dati riconciliati**. Vediamo la seguente immagine:



Possiamo vedere che abbiamo:



- il livello delle sorgenti → il quale comprende i dati operazionali e i dati esterni, i quali (ricordiamoci) rappresentano le nostre sorgenti eterogenee;
- gli strumenti ETL → essi non vanno ad effettuare il caricamento delle informazioni direttamente nel Data Warehouse, bensì essi vanno a popolare una classica base di dati, la quale contiene i cosiddetti **dati riconciliati**, **ovverosia dati operazionali ottenuti a valle del processo di integrazione e ripulitura dei dati sorgente. Quindi si tratta di dati integrati, consistenti, corretti, volatili, correnti e dettagliati** → possiamo capire allora, che il risultato del processo di ETL va a caricare i dati in una base di dati. Quindi, l'integrazione dei dati non finisce direttamente nel Data Warehouse, bensì finisce in un repository intermedio esistente, nel quale andiamo ad inserire i nostri dati riconciliati (da notare, che tali dati riconciliati sono i dati con le caratteristiche nominate precedentemente, ovverosia: sono integrati, ripuliti, consistenti, corretti, volatili e dettagliati). I processi di ETL, quindi, producono una base di dati relazionale, la quale verrà caricata nel livello di alimentazione.

Il **vantaggio** principale del livello dei dati riconciliati è che esso **crea un modello di dati comune e di riferimento per l'intera azienda, introducendo al contempo una separazione netta tra le problematiche legate all'estrazione e integrazione dei dati dalle sorgenti e quelle inerenti l'alimentazione del DW. D'altro canto, i dati riconciliati introducono un'ulteriore ridondanza rispetto ai dati operazionali sorgente.** Introduce un'ulteriore ridondanza, in quanto prendo le mie sorgenti di dati eterogenee e successivamente vado a integrare, ripulire e caricare i dati nella base di dati al livello dei dati riconciliati. Successivamente, tali dati riconciliati vengono utilizzati e vengono caricati nel Data Warehouse. **Lo svantaggio di introdurre ridondanza riguarda l'aumento dei costi ed eventuali problemi di coerenza, in quanto se vogliamo che i dati riconciliati siano allineati con le mie sorgenti eterogenee, dovremmo periodicamente mantenere l'allineamento delle informazioni.**

## Processo di ETL

Una volta analizzate le possibili tipologie di architetture, possiamo analizzare in maniera più approfondita il processo di ETL, che nell'ambito del Data Warehouse è un processo molto importante, in quanto è il processo che ci permette di partire dalle nostre sorgenti di dati eterogenee ed integrare l'informazione, ripulirla, trasformarla ed infine caricarla nel Data Warehouse. **Quindi, il processo di ETL ci permette di passare dall'informazione**

**eterogenea di partenza** (per esempio, con eterogenea intendiamo, che potremmo dover mettere insieme dei documenti XML con delle tuple di una base di dati relazionale, oppure dover mettere insieme tuple di basi di dati relazionali riguardanti argomenti completamente differenti, oppure dover mettere insieme tuple di basi di dati relazionali che contengono le stesse informazioni, ma esse sono rappresentate con schemi diversi, come per esempio il caso in cui l'indirizzo è un singolo attributo, mentre in un altro DB l'indirizzo è suddiviso in: città, via, CAP, numero civico) **e garantisce la qualità dei dati, in quanto all'interno del Data Warehouse vogliamo dati di qualità, ovvero sia dati:**

- **completi;**
- **corretti.**

Il nostro obiettivo, quindi, è di fare in modo che il processo di ETL venga fatto in maniera corretta. Da notare che siamo noi a progettare il processo di ETL, ovvero sia siamo noi a progettare in che modo i dati devono essere integrati, ripuliti e trasformati per poter essere caricati nel Data Warehouse. Innanzitutto, dovremmo tenere in considerazione eventuali situazioni/problemi a **livello di schema** (come per esempio il caso dell'attributo indirizzo rappresentato in maniera diversa, oppure dati in una tabella e dati in un documento XML). Una volta che avremmo risolto questi primi problemi e che quindi, avremmo generato il nostro **schema integrato** (che per nostra comodità possiamo pensare come uno schema relazionale), il passo successivo è di **capire come mappare i dati di partenza sullo schema integrato e questo vuol dire trasformarli** (chiaramente se vi sono delle incongruenze, dobbiamo andare a sistemare i dati) → **questi passaggi sono fondamentali per garantire la qualità dei dati.** Sorge spontanea la domanda: **"Perché voglio la qualità dei dati?"** **Voglio la qualità dei dati nel Data Warehouse, perché su di essi dovrò prendere delle decisioni e quindi possiamo capire, che il processo di ETL viene fatto ogni volta che dobbiamo caricare dei dati nel Data Warehouse.**



L'unica differenza che possiamo avere da quando viene popolato per la prima volta il Data Warehouse rispetto al suo caricamento periodico, è il momento di studio dell'integrazione dell'informazione (ovverosia se l'integrazione dei dati nel Data Warehouse non cambia mai, allora possiamo studiare solamente la prima volta come integrare i dati, individuando le relative regole di mappatura, e poi continuerò ad utilizzare le regole di mappatura individuate).

Passiamo, a questo punto, ad analizzare in maniera più approfondita gli **strumenti di Extraction, Transformation and Loading:**

- **Estrazione** → i dati rilevati vengono estratti dalle sorgenti. In particolare, l'estrazione si suddivide in:
  - estrazione **statica** → viene effettuata quando il DW deve essere popolato per la **prima volta** e quindi concettualmente, consiste in una fotografia dei dati operazionali. Andiamo, quindi, a capire:
    - quali dati abbiamo a disposizione;
    - che cosa (ovverosia quali dati) vogliamo mettere nel DW;
    - una volta capito cosa abbiamo nel DW, possiamo estrarre l'informazione che ci interessa. Chiaramente, questa estrazione viene fatta rispetto/successivamente ad un'**analisi dei requisiti**. Ricordiamo, che vi sono due metodologie per progettare un DW:
      1. Progettazione basata sui requisiti → la vedremo successivamente, ma intuitivamente possiamo dire, che tale progettazione del DW funziona esattamente come quella della base di dati, con l'unica differenza, che in questo caso (ovvero quello della progettazione del DW) abbiamo in più che il requisito è relativo ai dati che ci interessa analizzare e dobbiamo partire da delle sorgenti di dati, che abbiamo a disposizione;
      2. La 2° tipologia di progettazione di un DW, consiste nel partire dalle sorgenti di dati e quindi prendere i dati che abbiamo a disposizione e in qualche modo, ripulire quest'ultimi sulla base dei requisiti.
  - estrazione **incrementale** → viene utilizzata per l'aggiornamento periodico del DW e cattura solamente i cambiamenti avvenuti nelle

sorgenti dall'ultima estrazione. Questa tipologia di estrazione **dipende dall'istanza**, nel senso che: attraverso l'estrazione incrementale non vado a cambiare l'informazione caricata nel DW, in quanto quest'ultima (ovvero l'informazione) l'ho già decisa all'inizio (per esempio, ho deciso che il mio DW contiene le informazioni relative alle vendite). All'inizio, quindi, carico nel DW tutte le informazioni, che ho nella base di dati, relative alle vendite e periodicamente vado ad aggiungere al DW, le nuove informazioni relative alle vendite.

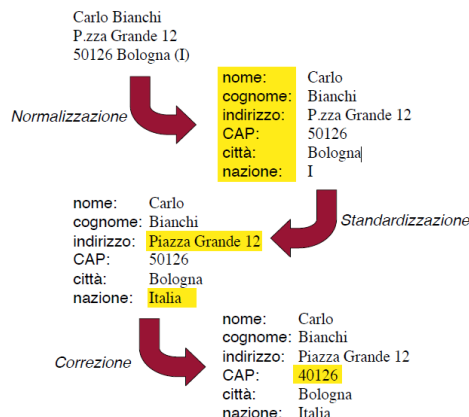
- **Pulitura** → si incarica di migliorare la qualità dei dati delle sorgenti, che andranno a finire nel mio DW. A questo punto, quindi, abbiamo estratto i dati di nostro interesse e dobbiamo andare a verificare che tipo di dati abbiamo a disposizione. Quindi, dobbiamo verificare se abbiamo:
  - dati duplicati;
  - inconsistenza tra valori logicamente associati;
  - dati mancanti;
  - uso non previsto di un campo;
  - valori errati;
  - valori inconsistenti per la stessa entità dovuti a differenti convenzioni.

Questi "errori/divergenze" appena sopra citati devono essere opportunamente gestiti e di conseguenza, dovranno essere costruite delle procedure che vanno a sistemare l'informazione di partenza, al fine di poterla caricare nel DW in maniera corretta (es: creiamo una procedura che mi permette di manipolare gli indirizzi, in modo tale da ripulirli in maniera corretta).

- **Trasformazione** → corrisponde al momento in cui andiamo a mappare i dati nelle sorgenti, nei dati che devono finire nel DW. Quindi, la trasformazione converte i dati dal formato operativo sorgente a quello del DW. La corrispondenza con il livello sorgente è complicata dalla presenza di fonti distinte eterogenee, che richiede una complessa fase di integrazione. Capiamo, allora, che dobbiamo definire delle regole di mappatura, le quali mi permettono di andare a definire come devono essere convertiti i dati per finire nel DW, a seconda di come sono i dati nelle sorgenti → chiaramente, **la trasformazione è estremamente facilitata, se i dati sono riconciliati, in quanto se abbiamo il livello dei dati riconciliati, vuol dire che abbiamo già integrato e riconciliato tutti i dati (e di conseguenza la struttura dei dati è**

uguale per tutti) e in questo modo, dobbiamo definire un'unica regola di mappatura.

Vediamo un esempio di pulitura e trasformazione:



- **Caricamento** → il caricamento dei dati nel DW si suddivide in due tipologie:
  - **Refresh** → i dati nel DW vengono riscritti integralmente, sostituendo quelli precedenti (tecnica normalmente utilizzata solo per popolare inizialmente il DW) → capiamo, allora, che non possiamo utilizzare il Refresh nei caricamenti dei dati successivi a quello iniziale, in quanto non saremmo in grado di mantenere uno storico dei dati (dato che con il Refresh i dati vengono riscritti integralmente);
  - **Update** → i soli cambiamenti occorsi nei dati sorgenti vengono aggiunti nel DW (tecnica normalmente utilizzata per l'aggiornamento periodico del DW).

Cerchiamo ora di capire cos'è il **modello multidimensionale** → il modello multidimensionale ci permette di rispondere alle seguenti domande:

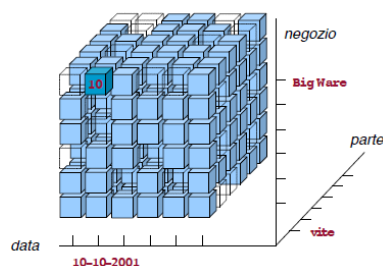
- "Che incassi sono stati registrati l'anno passato per ciascuna regione e ciascuna categoria di prodotto?". Notiamo, che gli incassi rappresentano la quantità numerica che vogliamo analizzare. Ci ricordiamo, che i dati che andiamo a mettere nel DW sono principalmente numerici e di conseguenza, la misura (ovverossia la "cosa" che vogliamo quantificare) è l'incasso. Il fatto, poi, che siamo registrati l'anno passato, mette in evidenza che consideriamo:
  - l'aspetto temporale;

- i dati in maniera aggregata (ovvero, prendo tutti gli incassi del 2023 e li metto insieme).

Inoltre, "l'anno scorso", "ciascuna regione" e "ciascuna categoria" rappresentano le 3 **dimensioni**, dove per dimensioni intendiamo: le dimensioni sono i punti di vista, che vogliamo utilizzare, per svolgere una determinata analisi (ad esempio, l'analisi relativa alle vendite). Quindi, ad esempio la vendita la vogliamo analizzare sotto tre punti di vista (ovvero sotto tre dimensioni):

1. il **tempo** → questa dimensione vi è sempre;
2. lo **spazio** (quindi le regioni);
3. la **categoria** di prodotti.

I dati saranno modellati con il modello multidimensionale, il quale è il modello di riferimento per la rappresentazione e l'interrogazione dell'informazione nei DW. Infatti, i fatti di interesse (nel nostro caso, le vendite) sono rappresentati attraverso dei **cubi**. Capiamo, allora, che il concetto che mi interessa analizzare è il cubo e se noi siamo interessati a 10 concetti diversi, allora avremo 10 cubi (che poi sarebbero i nostri Data mart) nel nostro DW. Ad esempio, essendo che per le vendite abbiamo considerato 3 dimensioni e di conseguenza otteniamo, come modello multidimensionale, un cubo. Il cubo delle vendite è il seguente:



(Naturalmente, il DW può utilizzare anche più di 3 dimensioni. Abbiamo fatto l'esempio di 3 dimensioni, in modo tale da poter visualizzare il cubo).

In particolare, abbiamo che:

- ogni cella del cubo contiene delle misure numeriche (nel nostro esempio delle vendite, la misura numerica è l'incasso), che quantificano il fatto d'interesse (ovvero la vendita) da diversi punti di vista. Per esempio, nel nostro caso, una cella va a specificare quanti prodotti sono stati venduti in una certa data e in un certo negozio;

- ogni asse rappresenta una dimensione di interesse per l'analisi e quindi un punto di vista, che possiamo utilizzare per fare l'analisi;

Fino a questo punto, notiamo che manca la parte di aggregazione dei dati e di conseguenza, per poter avere l'aggregazione, abbiamo bisogno che per ogni dimensione possiamo rappresentare una gerarchia. In particolare:

- ogni dimensione può essere la radice di una gerarchia di attributi, i quali vengono utilizzati per aggregare i dati memorizzati nei cubi base. Per esempio, la data, la posso aggregare in base al mese e una volta che ho i mesi, posso aggregare in base all'anno e poi in base al trimestre. Nel modello multidimensionale, aggregare in base alla data ad esempio, significa che vado a prendere tutte le coordinate riferite al mese di Novembre ad esempio e per tutti i numeri che trovo, vado a fare la somma. In questo modo, quindi, ho calcolato il valore aggregato del mese di Novembre. Chiaramente, oltre a prendere le coordinate relative al mese di Novembre, posso anche considerare le coordinate relative al negozio.



Questo per dire, che **il modello multidimensionale ci permette di incrociare i punti di vista che vogliamo utilizzare, al fine di fare analisi.**

Sorge a questo punto spontanea la domanda: **"Quali sono i concetti principali, che mi interessano quando considero un DW?"** I concetti principali, che mi interessano quando considero un DW, sono quelli che rappresentano il modello multidimensionale, ovvero:

- i **fatti di interesse**, che vogliamo analizzare (che ricordiamo essere rappresentati attraverso i cubi);
- le **measure** → sono le quantità numeriche che vogliamo poter analizzare;
- le **dimensioni** → rappresentano le dimensioni di analisi, ovverosia i punti di vista da cui possiamo analizzare un fatto d'interesse;
- siccome è importante poter fare analisi sui dati aggregati, per ogni dimensione possiamo specificare una **gerarchia**.

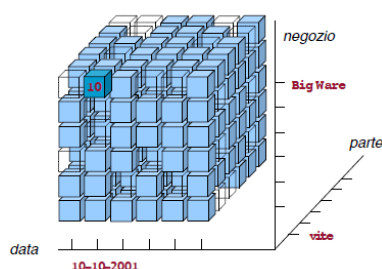
La logica è: A fronte della domanda iniziale "Che incassi sono stati registrati l'anno passato per ciascuna regione e ciascuna categoria di prodotto?" La risposta alla domanda, viene costruita applicando un processo di progettazione (che vedremo successivamente), in cui andiamo a definire:

- qual è il **fatto di interesse**;
- quali sono le **misure** che vogliamo analizzare;
- quali sono le **dimensioni** di analisi;
- quali sono le **gerarchie**.

Rispetto a questi "aspetti", i quali descrivono lo schema del DW, dobbiamo definire il matching tra lo schema della base di dati e lo schema del DW, in modo tale da capire quali dati mappare su queste informazioni e quindi quali dati devono finire del DW.

Per capire tutti questi concetti, vediamo meglio l'esempio delle vendite: Il fatto d'interesse è rappresentato dalla vendita e di conseguenza, la vendita la rappresento come un cubo. Le dimensioni di analisi, ovvero i punti di vista rispetto a cui voglio analizzare il fatto di interesse (ovvero le vendite) sono:

- la data;
- il negozio;
- la parte venduta (ovverosia il prodotto).



Possiamo notare, che se incrociamo una certa data per un certo tipo di prodotto e in un certo negozio, otteniamo il numero di prodotti venduti. Il numero di prodotti rappresenta la quantità e di conseguenza, rappresenta la misura (quindi il numero di prodotti venduti è la mia misura).

## Le gerarchie

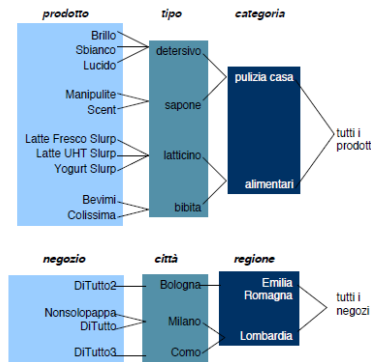
Abbiamo già detto, che quando progettiamo il nostro DW, dobbiamo definire:

- qual è il fatto d'interesse;
- quali sono le **misure** che vogliamo analizzare;
- quali sono le **dimensioni** di analisi;



- un'altra cosa che dobbiamo modellare è la **gerarchia** → **ci interessa modellare la gerarchia, perchè vogliamo aggregare l'informazione e quindi vogliamo fare analisi su dati aggregati.**

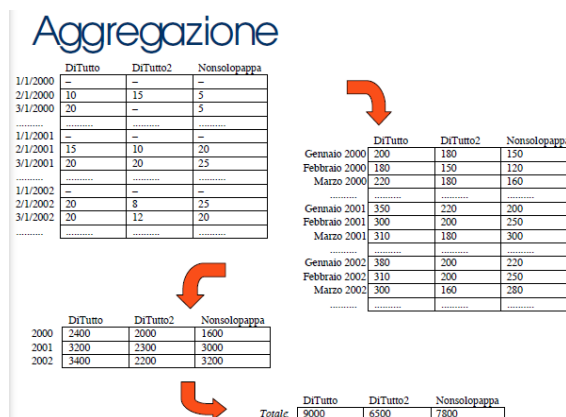
La gerarchia, quindi, ci permette di definire in che modo aggregare l'informazione. Vediamo la seguente immagine:



Vediamo, che abbiamo una gerarchia sul prodotto: Prodotto (posso aggregare i prodotti in base al tipo) → Tipo di prodotto (i tipi di prodotto li posso aggregare per categoria) → Categoria di prodotto → Tutti i prodotti e una seconda gerarchia sul negozio: Negozio (ogni negozio è in una certa città) → Città del negozio (le città le posso aggregare per regione) → Regione del negozio → Tutti i negozi.



Ovviamente, quando definiamo una gerarchia, vuol dire che il DW mi permetterà di aggregare rispetto alla gerarchia appena definita.



## Tecniche di analisi

Una volta che i dati sono stati ripuliti, integrati, trasformati e caricati nel nostro DW, occorre capire come trarne il massimo vantaggio informativo. In particolare, esistono in sostanza tre approcci differenti, supportati da altrettante categorie di strumenti, all'interrogazione dei dati nel nostro DW da parte degli utenti finali, che sono:

- la **reportistica** → non richiede conoscenze informatiche e infatti, essa può essere realizzata anche attraverso una base di dati. La reportistica (quali ad esempio: grafi a torta oppure un grafico dell'andamento degli incassi) è orientata agli utenti, che hanno necessità di accedere, a intervalli di tempo predefiniti, a informazioni strutturate in modo pressoché invariabili;
- **OLAP** → è la principale modalità di interrogazione dei dati contenuti in un DW. Rappresenta la principale modalità, perchè OLAP parte dal presupposto, che i dati sono modellati con il modello multidimensionale ed in particolare, vengono messi a disposizione degli operatori OLAP, i quali lavorano/operano sul cubo e ci permettono di esplorare in maniera interattiva i dati memorizzati nel nostro DW. OLAP, quindi, consente a utenti, le cui necessità di analisi non siano facilmente identificabili a priori, di analizzare ed esplorare interattivamente i dati sulla base del modello multidimensionale. OLAP lavora sul cubo e permette agli utenti di fare interrogazioni in maniera interattiva, usando competenze non strettamente informatiche, e semplificando la costruzione di query complicate. Quindi, mentre gli utenti degli strumenti di reportistica svolgono un ruolo essenzialmente passivo, gli utenti OLAP sono in grado di costruire attivamente una sessione di analisi complessa, in cui ciascun passo effettuato è conseguenza dei risultati ottenuti al passo precedente. Una sessione OLAP consiste in un percorso di navigazione, che riflette il procedimento di analisi di uno o più fatti di interesse sotto diversi aspetti e a diversi livelli di dettaglio. Questo percorso si concretizza in una sequenza di interrogazioni spesso formulate non direttamente, ma per differenza rispetto all'interrogazione precedente. Alcuni operatori OLAP sono:
  - Operatore **Roll-up** → aumenta l'aggregazione dei dati, eliminando un livello di dettaglio dalla gerarchia che abbiamo progettato → questo significa, che nel momento in cui andiamo a progettare la gerarchia per il nostro DW, **dobbiamo mettere a disposizione tutte le eventuali granularità (livelli di granularità) di aggregazioni, che ci possono essere utili;**

- Operatore **Drill-down** → è l'operatore opposto all'operatore Roll-up ed in particolare, diminuisce l'aggregazione dei dati introducendo un ulteriore livello di dettaglio dalla gerarchia che abbiamo progettato;
- Operatore **Slice-and-dice** → riduce la dimensionalità del cubo (intuitivamente, questo operatore fa una fetta del cubo di partenza o dei sotto-cubi. Quindi, per esempio, avendo inizialmente gli incassi annuali del 97 e 98 per categoria di prodotto e per ciascuna regione, attraverso l'operatore Slice-and-dice posso considerare solamente gli incassi del 98);
- Operatore di **Pivoting** → permette una rappresentazione diversa della stessa informazione;
- Operatore **Drill-across** → abbiamo detto, che il nostro DW è formato da molti Data mart, dove ogni Data mart contiene un fatto d'interesse. Vorremmo, quindi, mettere insieme e fare analisi su informazioni provenienti da due o più Data mart e per riuscire a fare ciò, vi è l'operatore Drill-across, il quale permette di stabilire un collegamento tra due o più cubi correlati, al fine di compararne i dati.
- **Data mining** → una volta che abbiamo i dati nel nostro DW, possiamo cercare di scoprire informazioni nascoste (o meglio dire **informazioni implicite**). Con le tecniche di Data mining, quindi, possiamo andare a riconoscere dei modelli/pattern significativi. Ovviamente, il Data mining è una tecnica di intelligenza artificiale, che ci permette di scoprire (partendo dalla grande mole di informazioni che abbiamo a disposizione) una serie di informazioni implicite interessanti. Queste tecniche di Data mining possono essere utili in diversi ambiti, come ad esempio:
  - ricerche di mercato;
  - studio dell'efficacia del marketing;
  - analisi delle abitudini di acquisto;
  - analisi dei rischi;
  - rilevamento di attività fraudolente.

Una delle tecniche del Data mining sono le **regole associative** → esse permettono di evidenziare regole di implicazione logica, che sono presenti nella nostra informazione e quindi nella nostra base di dati. Un esempio è: Chi compra scarpe compra anche calze. Un esempio di applicazioni delle

regole associative è lo **studio delle abitudini di acquisto, per la pubblicità mirata e l'organizzazione della merce sugli scaffali del supermercato.**

Questo ci permette di definire delle regole di implicazione, in cui la regola X implica Y si dice che possa avere un supporto e una confidenza, dove:

- il **supporto**, ci dice quante sono le transazioni (sul totale) che includono sia scarpe sia le calze. Il supporto, quindi, ci dice quante volte la regola associativa è vera rispetto al totale;
- la **confidenza**, ci dice che le persone che comprano scarpe, comprano anche le calze. La confidenza, quindi, ci dice quante volte è vero che una volta che si è verificata la prima parte/cosa, si verifica anche la seconda.

Il Data mining, quindi, va a costruire le regole associative (relative all'analisi che si sta svolgendo) e fornisce nel risultato, le regole associative che hanno un supporto e una superiore ad una certa soglia e anche una confidenza superiore ad una certa soglia → quindi, una regola associativa sarà significativa, nel momento in cui avrà un supporto e una confidenza abbastanza alte (e chiaramente per avere un supporto e una confidenza alta, la regola associativa deve essere frequente).

Per fare Data mining, possiamo utilizzare anche il concetto di **clustering** → data una popolazione di oggetti rappresentabili come punti in uno spazio multidimensionale in cui ciascuna dimensione corrisponde a una caratteristica di interesse, effettuare un clustering significa raggruppare gli oggetti in un ridotto numero di insiemi (cluster) che caratterizzino al meglio la popolazione. Il clustering, quindi, ci permette di:

- effettuare la segmentazione della clientela in categorie;
- valutazione di casi clinici sulla base della sintomatologia.

Possiamo utilizzare anche gli **alberi decisionali** → essi ci permettono di classificare le cause che portano al verificarsi di un evento e di conseguenza, gli alberi decisionali vengono spesso utilizzati per:

- la valutazione delle categorie di rischio dei clienti, per le società che concedono mutui e prestiti.

Infine, possiamo utilizzare le **serie temporali** per fare Data mining → in riferimento al tempo, andiamo ad individuare pattern ricorrenti o atipici in sequenze di dati complesse.

## ROLAP e MOLAP

Una volta, che abbiamo capito stiamo parlando di dati del modello multidimensionale e quindi, abbiamo a disposizione una serie di dati, che vengono presi dalle sorgenti e vengono travasati in un sistema di DW, in cui andiamo a considerare soprattutto dati quantitativi rispetto a determinate dimensioni di analisi, possiamo avere due tipologie di approcci alla modellazione dei dati:

1. Approccio **ROLAP (Relational OLAP)** → giustificato dall'enorme lavoro svolto in letteratura sul modello relazionale, dalla diffusa esperienza aziendale sull'utilizzo e l'amministrazione di basi di dati relazionali e dall'elevato livello di prestazioni e flessibilità raggiunto dai DBMS relazionali. Vi è quindi, la necessità di elaborare tipologie specifiche di schemi, che permettano di traslare il modello multidimensionale sul modello relazionale (vedremo più avanti che parleremo dello schema a stella). L'approccio ROLAP, quindi, permette di sfruttare al massimo le competenze e i sistemi che già abbiamo a disposizione, al fine di ottenere il sistema di supporto alle decisioni;
2. Approccio **MOLAP (Multidimensional OLAP)** → basato su un modello logico ad hoc, sul quale i dati e le operazioni multidimensionali possono essere direttamente rappresentati. I dati vengono fisicamente memorizzati in vettori e l'accesso è di tipo posizionale. Il grosso **vantaggio** dell'approccio MOLAP, rispetto a quello ROLAP, è che le operazioni multidimensionali sono realizzabili in modo semplice e naturale, senza necessità di ricorrere a join e di conseguenza, le prestazioni risultano ottime. Lo **svantaggio**, però, è che non esistendo ancora uno standard per il modello logico multidimensionale, le diverse implementazioni MOLAP hanno veramente poco in comune: in genere, solo l'utilizzo di tecnologie di ottimizzazione specifiche per trattare il problema della sparsità.

---

## Ciclo di vita del Data Warehouse

Molte organizzazioni mancano della necessaria esperienza e capacità per affrontare con successo le sfide implicite nei progetti di data warehousing. Uno dei fattori che maggiormente minaccia la riuscita dei progetti è la mancata adozione di un approccio metodologico, che minimizza i rischi di insuccesso essendo basato su un'analisi costruttiva degli errori commessi. In particolare, esistono due approcci possibili per la realizzazione di un DW, ovvero:

- Approccio **top-down** → analizza i bisogni globali dell'intera azienda e successivamente pianifica lo sviluppo del DW per poi progettare e realizzarlo nella sua interezza. Questo significa, che già dall'inizio pensiamo di fare l'analisi di tutte le aree, che possono poi essere interessate ad analizzare qualcosa, ovvero si partiamo facendo l'analisi dei requisiti di tutti i Data mart (per esempio: il data mart delle vendite, il data mart del marketing e quello della logistica), in modo tale da ottenere una visione completa dell'intera azienda e cercando di progettare il DW nella sua interezza. Questo approccio ha un unico **vantaggio**, ossia:
  - promette ottimi risultati, poiché si basa su una visione globale dell'obiettivo e garantisce in linea di principio di produrre un DW consistente e ben integrato.

Gli **svantaggi**, invece, dell'approccio top-down sono:

- il preventivo di costi onerosi a fronte di lunghi tempi di realizzazione, scoraggia la direzione dall'intraprendere il progetto;
  - affrontare contemporaneamente l'analisi e la riconciliazione di tutte le sorgenti di interesse è estremamente complesso;
  - riuscire a prevedere a priori nel dettaglio le esigenze delle diverse aree aziendali impegnate è pressoché impossibile, e il processo di analisi rischia di subire una paralisi;
  - il fatto di non prevedere la consegna a breve termine di un prototipo, non permette agli utenti di verificare l'utilità del progetto e ne fa scemare l'interesse e la fiducia.
- Approccio **bottom-up** → con questo approccio, invece, il DW viene costruito in modo incrementale, assemblando iterativamente più Data mart. Questo significa, che partiamo da un Data mart (che deve essere quello che gioca il ruolo più strategico per l'azienda e quindi deve ricoprire un ruolo centrale e di riferimento per l'intero DW. Capiamo, allora, che il Data mart deve appoggiarsi su dati già disponibili e consistenti) e facciamo l'analisi dei requisiti relativi al Data mart scelto. Via via, quindi, andiamo a costruire altri Data mart, che andranno a costruire e completare il DW globale. L'unico **svantaggio** di questo approccio è:
    - determina una visione parziale del dominio di interesse (in quanto inizialmente ci concentriamo su un unico Data mart).

I **vantaggi**, invece, sono:

- determina risultati concreti in tempi brevi;
- non richiede elevati investimenti finanziari;
- permette di studiare solo le problematiche relative al Data mart in oggetto;
- fornisce alla dirigenza aziendale un riscontro immediato sull'utilità del sistema;
- mantiene costantemente elevata l'attenzione sul progetto.

Il ciclo di sviluppo di un Data mart è analogo al ciclo di sviluppo di una base di dati, che come sappiamo a sua volta, è analogo al ciclo di sviluppo di un qualsiasi altro software. In particolar modo, quindi, il ciclo di sviluppo di un Data mart è il seguente:



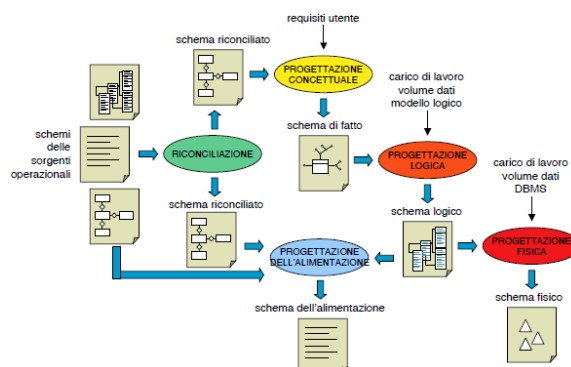
Possiamo vedere che dall'Analisi dei requisiti alla Progettazione logica (analisi dei requisiti, progettazione concettuale, raffinamento del carico di lavoro e progettazione logica), il ciclo di sviluppo del Data mart ricalca perfettamente il ciclo di sviluppo di una base di dati. In più, quindi, abbiamo:

- Analisi e riconciliazione delle sorgenti → abbiamo questo livello in più, in quanto i dati che andiamo a caricare nel DW non sono semplicemente dati per cui devo andare a definire una struttura, bensì devo definire la struttura del DW tenendo conto dei requisiti disposti dall'utente e anche delle sorgenti di dati che abbiamo a disposizione (nel senso che, se vogliamo analizzare un dato che non è presente nelle nostre sorgenti, ovviamente non possiamo andare a realizzare il DW).

(Anticipiamo, dicendo che per la progettazione di un DW possiamo:

- fare l'analisi dei requisiti, tenendo conto dell'analisi e della riconciliazione delle sorgenti;
- oppure possiamo costruire lo schema dei dati riconciliati, partendo dallo schema delle sorgenti e in maniera semi-automatica riuscire a costruire lo schema del DW.)
- Progettazione dell'alimentazione → una volta che ho tutti gli schemi iniziali e lo schema finali, dobbiamo effettivamente capire come mappare i dati iniziali con i dati finali, che devono essere caricati nel DW. Quindi, la progettazione dell'alimentazione consiste in tutti quei meccanismi che vengono utilizzati, per matchare i dati iniziali con i dati finali, che devono essere caricati nel DW;
- Progettazione fisica (di cui non ci occupiamo).

Con il seguente grafico, vediamo in maniera intuitiva tutti i passi che dovremo svolgere per la progettazione del Data mart:



) Vediamo che partiamo dagli schemi delle sorgenti operazionali, i quali sono gli schemi delle nostre sorgenti eterogenee di partenza (per sottolineare il fatto che le sorgenti sono eterogenee, possiamo osservare che abbiamo uno schema ER, un documento XML o un diagramma UML). Rispetto alle sorgenti eterogenee, abbiamo la fase di RICONCILIAZIONE → in questa fase, capiamo effettivamente le sorgenti che abbiamo a disposizione e vogliamo produrre uno schema riconciliato, ovvero uno schema integrato delle informazioni che abbiamo a disposizione, il quale quindi ci permette di avere una visione unificata globale delle sorgenti eterogenee. Ovviamente lo schema riconciliato, sarà il punto di partenza per capire cosa dovrà effettivamente finire nel nostro DW. Con lo schema riconciliato, quindi con lo schema delle informazioni relative alle sorgenti, possiamo procedere con la PROGETTAZIONE CONCETTUALE, tenendo conto dei requisiti dell'utente (tali requisiti utente mi



diranno quale sarà il fatto d'interesse, che tipo di misure dovremmo analizzare, che tipo di analisi dovremmo fare, le dimensioni e le gerarchie). A questo punto, possiamo costruire lo schema di fatto, il quale utilizza il Dimensional Fact Model, e ci permette di costruire lo schema concettuale del nostro DW. Lo schema concettuale viene dato in input alla progettazione logica (tenendo conto del carico di lavoro, del volume dei dati e del modello logico che vogliamo utilizzare, che nel nostro caso sarà il modello relazionale), la quale a sua volta (ovvero la progettazione logica) ci permetterà di produrre lo schema logico relazionale. Quest'ultimo, infine, ci permetterà di andare a costruire fisicamente il nostro DW. Guardando, poi, la parte bassa dello schema, vediamo che una volta prodotto lo schema riconciliato, passiamo alla progettazione dell'alimentazione, attraverso cui andremo a definire un insieme di regole di matching, che mi permetteranno di mappare i dati di partenza con i dati finali, che finiranno nel nostro DW.

## **Analisi e riconciliazione delle sorgenti operazionali**

Partiamo con la **progettazione del livello conciliato** → ci permette di partire dai nostri schemi sorgente e attraverso l'analisi e la riconciliazione dei dati, ci permette di integrare quest'ultimi (ovvero i dati). Ricordiamoci, che una volta che otteniamo lo schema riconciliato e gli strumenti per fare il mapping dai dati iniziali a quelli finali, tutta la parte di travaso, pulitura e di trasformazione dei dati avrà a che fare con gli strumenti ETL.

Abbiamo poi la fase di **ricognizione e normalizzazione** → questa fase è bene farla anche nel caso in cui vi sia un'unica sorgente e anche nel caso in cui non andiamo a materializzare il livello di dati riconciliato, in quanto questa fase ci permette avere cognizione di quali dati abbiamo a disposizione. Quindi, nell'analisi e nella riconciliazione:

- partiamo dalle sorgenti a disposizione;
- analizziamo le sorgenti;
- integriamo gli schemi;
- definiamo la corrispondenza con le sorgenti e a questo punto, sappiamo come mappare le sorgenti in maniera integrata sullo schema riconciliato.

La fase di integrazione non si deve limitare a evidenziare le differenze di rappresentazione dei concetti comuni a più schemi locali, ma deve anche identificare l'insieme di concetti distinti e memorizzati in schemi differenti che sono correlati attraverso proprietà semantiche.

## Analisi dei requisiti

L'analisi dei requisiti ha l'obiettivo di capire quali sono le esigenze di utilizzo del Data mart (e quindi del DW) espresse dai suoi utenti finali. Da notare, che durante la fase di analisi dei requisiti dobbiamo tenere in considerazione quello che abbiamo imparato durante l'analisi e la ricognizione delle sorgenti e quindi, dobbiamo essere consapevoli dei dati che abbiamo a disposizione. Dopodiché, l'analisi dei requisiti mira ad intervistare gli utenti finali e capire cosa effettivamente vogliono. Inoltre, l'analisi dei requisiti ha importante strategica, in quanto influenza le decisioni da prendere riguardo:

- lo schema concettuale dei dati;
- il progetto dell'alimentazione;
- l'architettura del sistema;
- le linee guida per la manutenzione e l'evoluzione del sistema.

La fonte principale da cui attingere i requisiti sono i futuri utenti del Data mart (i quali ricordiamo hanno poche competenze informatiche e di conseguenza si ha una differenza di linguaggio usato dai progettisti e dagli utenti finali). Mentre per gli aspetti più tecnici, saranno gli amministratori del sistema informativo a fungere da riferimento per il progettista.

I **fatti** sono i concetti su cui gli utenti finali del Data mart baseranno il processo decisionale; ogni fatto descrive una categoria di eventi che si verificano in azienda. Da notare, che fissare le dimensioni di un fatto è importante poiché significa determinarne la **granularità**, ovvero il più fine livello di dettaglio a cui i dati saranno rappresentati. La scelta della granularità di un fatto nasce da un delicato compromesso tra due esigenze contrapposte: quella di raggiungere un'elevata flessibilità d'utilizzo e quella di conseguire buone prestazioni. Per ogni fatto, poi, occorre definire l'**intervallo di storicizzazione**, ovvero l'arco temporale che gli eventi memorizzati dovranno coprire → le interviste, quindi, sono mirate a raccogliere e a capire quali sono i fatti d'interesse, in quanto ricordiamo che vogliamo sapere:

- quali fatti modellare → chiaramente se il fatto non si verifica, non abbiamo il dato da analizzare;
- che misure devono avere ;
- che dimensioni devono avere;

- quali gerarchie costruire sulle dimensioni → nel momento in cui costruiamo la gerarchia, andiamo anche a definire la granularità con cui fare analisi sul fatto d'interesse.

## Progettazione concettuale

Con la progettazione concettuale, andiamo a vedere qual è il **modello concettuale** per modellare un DW → un modello concettuale è un modello grafico, che mi permette di rappresentare ad alto livello d'astrazione:

- i fatti d'interesse;
- le dimensioni;
- le misure.

**Il nostro obiettivo, quindi, è di riuscire a costruire uno schema di fatto, al fine di modellare il nostro DW.** Per riuscire a fare ciò, intuitivamente verrebbe da utilizzare il modello Entity/Relationship, il quale è molto diffuso nelle imprese come formalismo per la documentazione dei sistemi informativi relazionali, ma non può essere usato per modellare il DW. Di conseguenza, il modello concettuale grafico che andremo ad utilizzare prende il nome di **Dimensional Fact Model** → esso ci consente di generare un insieme di schemi di fatto, in cui possiamo rappresentare graficamente i nostri fatti d'interesse, le relative misure, dimensioni e gerarchie. Il Dimensional Fact Model, analogamente al modello E/R, ha i vantaggi di uno schema grafico ad alto livello d'astrazione e quindi, ci permette di:

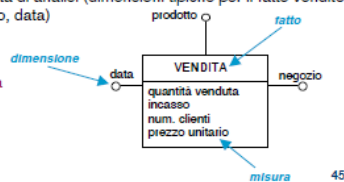
- supportare efficacemente il progetto concettuale;
- creare un ambiente su cui formulare in modo intuitivo le interrogazioni dell'utente;
- permettere il dialogo tra progettisti e utenti finali;
- restituire una documentazione a posteriori espressiva e non ambigua.

Vediamo a questo punto, i **costrutti di base** del Dimensional Fact Model:



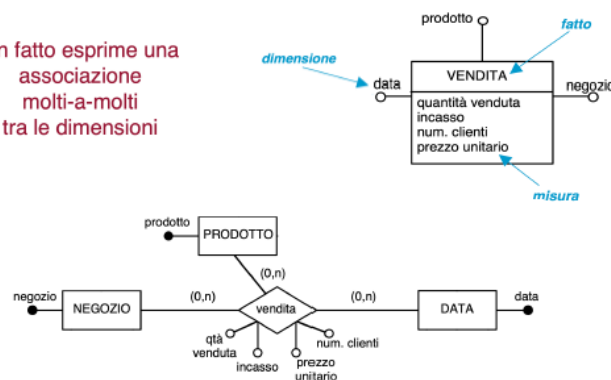
- Un **fatto** è un concetto di interesse per il processo decisionale; tipicamente modella un insieme di eventi che accadono nell'impresa (ad esempio: vendite, spedizioni, acquisti, ...). È essenziale che un fatto abbia aspetti dinamici, ovvero evolva nel tempo
- Una **misura** è una proprietà numerica di un fatto e ne descrive un aspetto quantitativo di interesse per l'analisi (ad esempio, ogni vendita è misurata dal suo incasso)
- Una **dimensione** è una proprietà con dominio finito di un fatto e ne descrive una coordinata di analisi (dimensioni tipiche per il fatto vendite sono prodotto, negozio, data)

Un fatto esprime una associazione multi-a-molti tra le dimensioni



45

Un fatto esprime una associazione multi-a-molti tra le dimensioni



- Con il termine generale **attributi dimensionali** si intendono le dimensioni e gli eventuali altri attributi, sempre a valori discreti, che le descrivono (per esempio, un prodotto è descritto dal suo tipo, dalla categoria cui appartiene, dalla sua marca, dal reparto in cui è venduto)
- Una **gerarchia** è un albero direzionato i cui nodi sono attributi dimensionali e i cui archi modellano associazioni multi-a-uno tra coppie di attributi dimensionali. Essa racchiude una dimensione, posta alla radice dell'albero, e tutti gli attributi dimensionali che la descrivono



Notiamo allora che per il fatto che ci interessa analizzare, modelliamo una serie di misure, le quali sono le misure quantitative che ci permetteranno di fare l'analisi. Le misure, quindi, saranno i valori su cui ci concentreremo per fare l'analisi del fatto d'interesse (sono valori quantitativi, in quanto su di essi abbiamo bisogno di applicare degli operatori aggregati). Il punto di vista da cui partiamo per fare l'analisi, lo rappresentiamo tramite una dimensione. Per le dimensioni, possiamo specificare una gerarchia. La gerarchia, quindi, ci

permette di definire in che modo andremo ad aggregare i dati, che abbiamo a disposizione per fare analisi. Abbiamo bisogno di aggregare i dati, perchè se vogliamo cercare di analizzare l'informazione, magari abbiamo bisogno di avere informazioni sintetizzate, che mi permettono di capire come stanno andando le cose. Per ogni dimensione, quindi, definiamo una gerarchia, la quale è formata da **attributi dimensionali**. Andiamo, allora, a costruire un albero, in cui abbiamo associazioni N:1 (associazioni molti a uno) e questo ci permetterà, ad esempio, rispetto alla gerarchia della data (che possiamo vedere nella figura sopra) di aggregare le date che abbiamo in un mese, i mesi che abbiamo in un trimestre e i trimestri che abbiamo in un anno.

Come avviene per il modello E/R, anche in questo caso abbiamo delle convenzioni sui nomi, che sono:

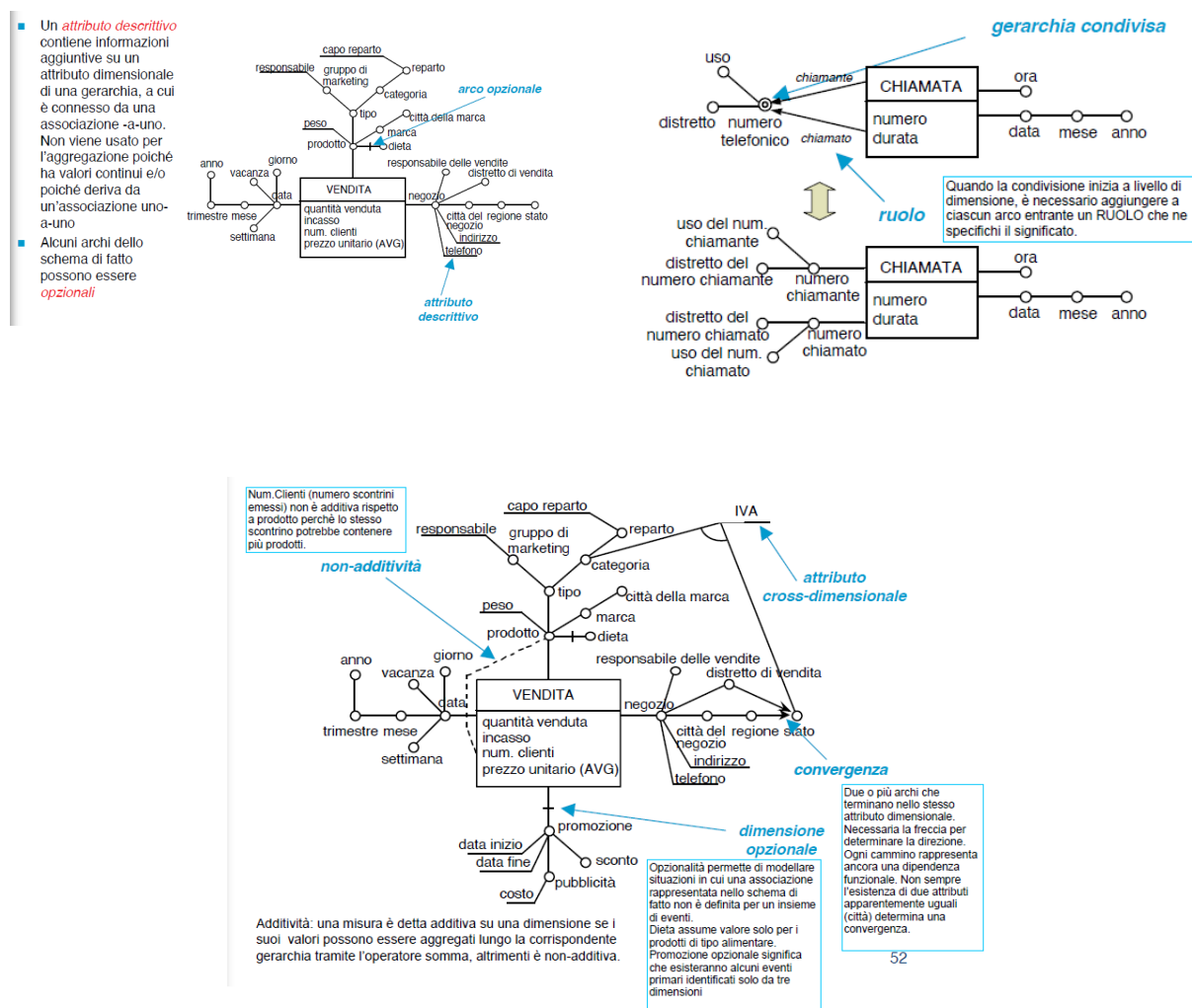
- Tutti gli attributi dimensionali in ciascuno schema di fatto devono avere nomi diversi;
- Eventuali nomi uguali devono essere differenziati qualificandoli con il nome di un attributo dimensionale che li precede nella gerarchia;
- I nomi degli attributi non dovrebbero riferirsi esplicitamente al fatto a cui appartengono;
- Attributi con lo stesso significato in schemi diversi devono avere lo stesso nome.

Abbiamo poi già detto, che l'occorrenza di un fatto d'interesse (ovverosia il verificarsi di un fatto, che nel nostro caso è la vendita) è un **evento**. In particolare, un **evento primario** è una particolare occorrenza di un fatto, individuata da una ennupla costituita da un valore per ciascuna dimensione. A ciascun evento primario è associato un valore per ciascuna misura → possiamo quindi dire, che **l'evento primario riguarda l'evento, visto dal punto di vista della radice delle nostre dimensioni** (Con riferimento alle vendite, un possibile evento primario registra per esempio che, il 10/10/2001, nel negozio NonSoloPappa sono state vendute 10 confezioni di detersivo Brillo per un incasso complessivo pari a 25 euro).

Dato un insieme di attributi dimensionali (pattern), ciascuna ennupla di loro valori individua un **evento secondario**, che aggrega tutti gli event primari corrispondenti. A ciascun evento secondario è associato un valore per ciascuna misura, che riassume in sé tutti i valori della stessa misura negli eventi primari corrispondenti → l'evento secondario, quindi, si ottiene nel momento in cui andiamo ad aggregare e di conseguenza, l'evento secondario va ad

aggregare tutti gli eventi primari corrispondenti. Pertanto, **le gerarchie definiscono il modo in cui gli eventi primari possono essere aggregati e selezionati significativamente per il processo decisionale; mentre la dimensione in cui una gerarchia ha radice ne definisce la granularità più fine di aggregazione, agli altri attributi dimensionali corrispondono granularità via via crescenti.**

A questo punto, vediamo alcuni **costrutti avanzati** del Dimensional Fact Model:



## Addittività

L'addittività è molto importante, perchè nel momento in cui andiamo ad aggregare le nostre informazioni, **dobbiamo capire in che modo comporre i valori delle misure che caratterizzano gli eventi primari in valori da abbinare a ciascun evento secondario.** Quindi, a seconda del tipo di misura con cui abbiamo a che fare, dovremmo specificare operatori aggregati differenti e in questo senso, è possibile distinguere tre categorie di misure:

- **Misure di flusso** → si riferiscono a un periodo, al cui termine vengono valutate in modo cumulativo (il numero di prodotti venduti in un giorno, l'incasso mensile, il numero di nati in un anno). Per quanto riguarda le misure di flusso, sulle gerarchie temporali e sulle gerarchie non temporali, posso utilizzare i seguenti operatori aggregati: SUM, AVG, MIN, MAX;
- **Misure di livello** → vengono valutate in particolari istanti di tempo (il numero di prodotti in inventario, il numero di abitanti di una città). Per quanto riguarda le misure di livello, sulle gerarchie temporali posso utilizzare i seguenti operatori aggregati: AVG, MIN, MAX, mentre sulle gerarchie non temporali, posso utilizzare gli operatori aggregati: SUM, AVG, MIN, MAX;
- **Misure unitarie** → vengono valutate in particolari istanti di tempo, ma sono espresse in termini relativi (il prezzo unitario di un prodotto, la percentuale di sconto, il cambio di una valuta). Per quanto riguarda le misure unitarie, sulle gerarchie temporali e sulle gerarchie non temporali, posso utilizzare i seguenti operatori aggregati: AVG, MIN, MAX.



**Una misura è detta additiva su una dimensione se i suoi valori possono essere aggregati lungo la corrispondente gerarchia tramite l'operatore di somma, altrimenti è detta non-additiva. Una misura non-additiva è non-aggregabile se nessun operatore di aggregazione può essere usato su di essa.**

Parliamo brevemente della progettazione concettuale. Diciamo, innanzitutto, che per la progettazione concettuale esistono due approcci:

- **Basata sui requisiti** (che è quella che utilizzeremo) → il progettista deve essere in grado di enucleare, dalle interviste condotte presso l'utente, un'indicazione precisa circa i fatti da rappresentare, le misure che li descrivono e le gerarchie attraverso cui aggregarli utilmente. Il problema del collegamento tra lo schema concettuale così determinato e le sorgenti operazionali viene affrontato in un secondo tempo (ovvero attraverso questo approccio parto dai requisiti, tenendo però sempre in considerazione le sorgenti, al fine di costruire lo schema di fatto);
- **Basata sulle sorgenti** → è possibile definire lo schema concettuale in funzione della struttura delle sorgenti, evitando il complesso compito di stabilire il legame con esse a posteriori. Inoltre, è possibile derivare uno

schema concettuale prototipale dagli schemi operazionali in modo automatico (ovvero attraverso questo approccio parto dalle sorgenti, tenendo però sempre in considerazione i requisiti, al fine di costruire lo schema di fatto).

## Progettazione logica

Ricordiamoci innanzitutto la differenza tra modello logico e modello concettuale. Il modello concettuale ci permette di rappresentare, con un certo livello di astrazione, l'informazione senza poi preoccuparci di come andremo effettivamente a modellare l'informazione e quindi con il modello concettuale, abbiamo una visione astratta completamente distaccata dal modello logico. Il modello logico, invece, è un modello concettuale, che in un qualche modo ricalca l'organizzazione che i dati avranno. Capiamo, allora, che mentre la modellazione concettuale è indipendente dal modello logico prescelto per l'implementazione, evidentemente lo stesso non si può dire per i temi legati alla modellazione logica. Dobbiamo, allora, capire qual è il modello logico che dovremmo utilizzare e i modelli logici che prendiamo in considerazione sono due:

- Modello logico **MOLAP (Multidimensional On-Line Analytical Processing)** → memorizza i dati utilizzando strutture intrinsecamente multidimensionali (es. vettori multidimensionali). Quindi, l'organizzazione dell'informazione è multidimensionale e quindi intuitivamente possiamo dire, che tale modello MOLAP ricalca la rappresentazione multidimensionale che abbiamo visto fino ad ora (ovvero quella del cubo). MOLAP rappresenta una soluzione naturale e può fornire ottime prestazioni poiché le operazioni non devono essere "simulate" mediante complesse istruzioni SQL. Pone, però, il problema della sparsità: in media solo il 20% delle celle dei cubi contiene effettivamente informazioni, mentre le restanti celle corrispondono a fatti non accaduti. È frenato dalla mancanza di strutture dati standard: i diversi produttori di software utilizzano strutture proprietarie che li rendono difficilmente sostituibili e accessibili mediante strumenti di terze parti. Per questi motivi, quindi, di fatto l'approccio utilizzato è il modello logico ROLAP;
- Modello logico **ROLAP (Relational On-Line Analytical Processing)** → si basa sul modello relazionale per la rappresentazione dei dati multidimensionali. Con questo modello, l'organizzazione dei dati è meno

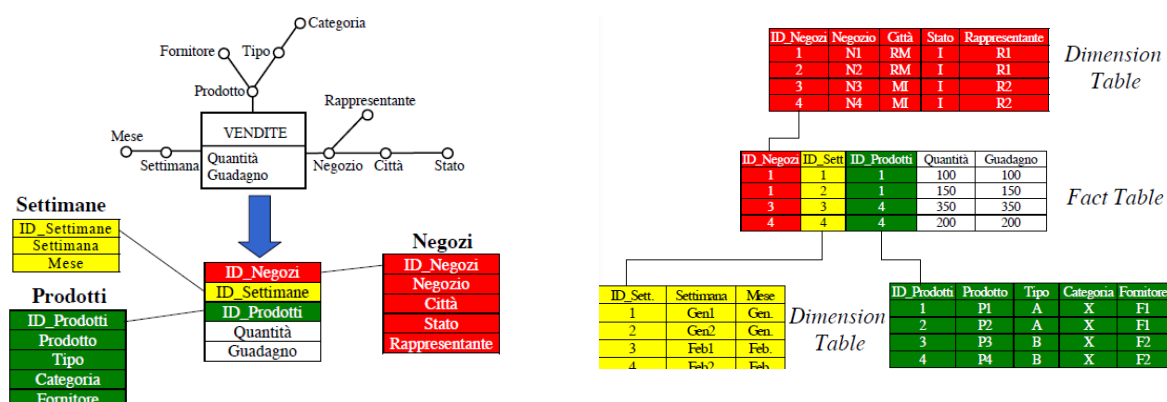


intuitiva, ma dall'altra parte sfrutta il fatto che il modello relazionale è ampiamente diffuso.

## ROLAP: lo schema a stella

La modellazione multidimensionale su sistemi relazionali, ovvero partiamo dalle nostre sorgenti di dati (qualunque esse siano: relazionali, semi-strutturate, XML) e svolgiamo tutta la procedura ETL e la progettazione concettuale. Nel momento in cui arriviamo alla progettazione logica del nostro DW, utilizziamo come modello logico dei dati nuovamente il modello relazionale dei dati. Esso è basata sul cosiddetto **schema a stella**, il quale è uno schema relazionale dei dati (quindi è una base di dati relazionale), in cui abbiamo:

- un insieme di relazioni DT1..., DTn, chiamate **dimension table**, ciascuna corrispondente ad una dimensione dello schema di fatto. Ogni DTi è caratterizzata da una chiave primaria (tipicamente surrogata) di e da un insieme di attributi che descrivono le dimensioni di analisi ai diversi livelli di aggregazione (quindi, ogni gerarchia che abbiamo modellato nello schema di fatto corrisponde ad una dimension table);
- una relazione FT, chiamata **fact table**, che importa le chiavi di tutte le dimension table. La chiave primaria di FT è data dall'insieme delle chiavi esterne dalle dimension table, d1..., dn; FT contiene inoltre un attributo per ogni misura.



Possiamo immediatamente osservare, che attraverso le **dimension table** andiamo a **valorizzare le misure** ed inoltre osserviamo, che le dimension table sono **completamente denormalizzate** e questo ha come conseguenze dirette che:

- è sufficiente una join per recuperare tutti i dati relativi ad una dimensione;

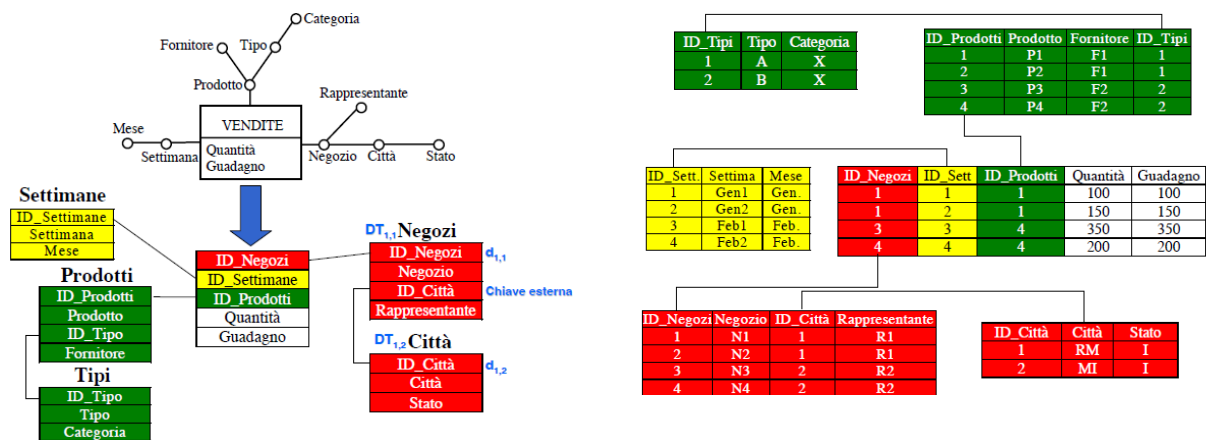
- lo svantaggio, però, è che la denormalizzazione introduce una forte ridondanza dei dati (Per esempio vediamo che per ogni prodotto, che appartiene alla stessa categoria, dobbiamo comunque ripetere la categoria).

Notiamo, inoltre, che la **fact table** contiene le tuple relative ai diversi livelli di aggregazione e di conseguenza:

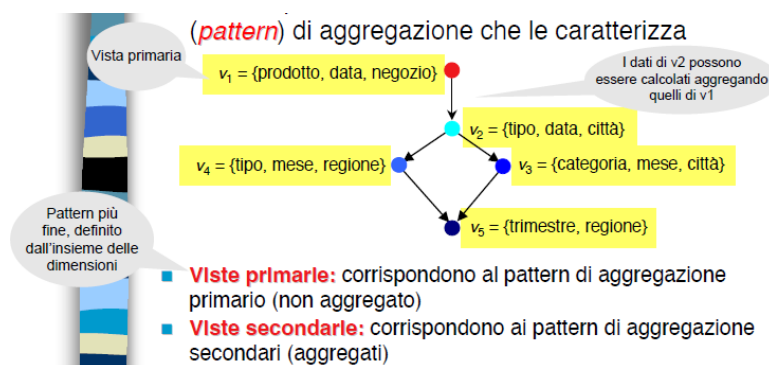
- l'elevata dimensione incide sui tempi di accesso ai dati;
- il vantaggio, però, è che non si hanno problemi di sparsità, in quanto vengono memorizzate soltanto le tuple corrispondenti a punti dello spazio multi-dimensionale per cui esistono eventi.

---

Una variante dello schema a stella, che riduce la denormalizzazione delle dimension table DT e quindi che cerca di migliorare dal punto di vista della ridondanza dei dati, è lo **schema a fiocco di neve** → In particolare, lo schema a fiocco di neve (snowflake schema) riduce la denormalizzazione delle dimension table DTi degli schemi a stella eliminando alcune delle dipendenze transitive che le caratterizzano. Lo schema a fiocco di neve, quindi, va a creare delle **dimensional table secondarie**, le quali vanno a rappresentare le parti ridondanti. Denominiamo, quindi, **primarie** le dimension table le cui chiavi sono importate nella fact table, **secondarie** le rimanenti → chiaramente, attraverso lo schema a fiocco di neve andiamo a ridurre la ridondanza e di conseguenza lo spazio richiesto per la memorizzazione dei dati si riduce grazie alla normalizzazione, però abbiamo come **svantaggio** che non possiamo più utilizzare un solo join per recuperare tutti i dati relativi ad una dimensione, bensì abbiamo bisogno di due join (perché le informazioni che ci interessano sono suddivise su 3 tabelle e non più su 2) ed inoltre, è necessario inserire nuove chiavi surrogate che permettano di determinare le corrispondenze tra dimension table primarie e secondarie.



L'analisi dei dati al massimo livello di dettaglio è spesso troppo complessa e non interessante per gli utenti che richiedono dati di sintesi. L'aggregazione rappresenta il principale strumento per ottenere informazioni di sintesi. L'elevato costo computazionale connesso con l'aggregazione induce a precalcolare i dati di sintesi maggiormente utilizzati. Con il termine **vista** si denotano le fact table contenenti dati aggregati e le viste possono essere identificate in base al livello (pattern) di aggregazione che le caratterizza e in questo senso, possiamo avere **viste primarie** e **viste secondarie**:



Chiaramente, a seconda dell'interrogazione che dobbiamo fare, andremo a considerare o meno la necessità di creare una vista, in quanto ricordiamo che avere una vista significa avere già a disposizione i dati aggregati per la mia interrogazione. In questo senso, quindi, possiamo creare un **reticolo** (di viste), che ci permette di vedere qual è il modo più vantaggioso di risolvere una determinata interrogazione.

