

# LINGUAGGI PER LA MODELLAZIONE DEI PROCESSI AZIENDALI

Fabio Casati, Barbara Pernici

## 1. Analisi dei processi aziendali

Per progettare un sistema informativo è necessario identificare tutti i suoi elementi e descriverli in modo preciso, non ambiguo e che possa essere compreso dagli utenti che partecipano alla progettazione del sistema, tra i quali troveremo i futuri utilizzatori del sistema. Tra gli elementi da descrivere come componenti di un sistema informativo troviamo i *processi*, e cioè insiemi di attività elementari che vengono svolte per raggiungere un certo obiettivo nel sistema.

I processi che potranno essere descritti durante la progettazione di un sistema informativo possono essere di diverse tipologie:

- *processi fisici*, che descrivono attività di elaborazione di oggetti fisici del sistema; ad esempio, possiamo avere descrizioni di flussi di materiali all'interno di un processo di produzione
- *processi informativi*, funzioni che creano, gestiscono, elaborano e forniscono informazioni. Ad esempio, un processo informativo in un'organizzazione sarà quello di gestire le informazioni relative all'emissione di un ordine per l'acquisto di materiali;
- *processi aziendali* (o business process), rappresentano funzioni legate all'attività complessiva dell'organizzazione o dell'impresa, quale la produzione di un'automobile, la pubblicazione di un libro, la gestione di una compagnia aerea, la consegna della posta, l'insegnamento agli studenti. Per esempio, per un'organizzazione che si occupa di insegnamento agli studenti, sarà necessario definire i processi aziendali per la diffusione delle informazioni sulla scuola, per l'iscrizione degli studenti, per la composizione delle classi e per la definizione dell'orario delle lezioni, e così via.

Nella progettazione di un sistema informativo la rappresentazione dei processi aziendali assumerà un'importanza particolare. Infatti questo sarà il punto di partenza per una descrizione dettagliata dell'attività dell'azienda o dell'organizzazione e quindi per la definizione dei requisiti del sistema. I processi aziendali sono infatti legati alla missione aziendale e quindi agli obiettivi globali dell'organizzazione. Per questo motivo sarà spesso opportuno legare i processi aziendali a una loro valutazione e quindi a una

riorganizzazione dei processi per una migliore efficienza e efficacia delle attività aziendali.

La descrizione accurata dei processi aziendali è anche la base per la realizzazione dei sistemi informativi basati su elaboratore. In tali sistemi si descriveranno in dettaglio tutti i processi informativi legati ai processi aziendali definiti nella fase di analisi nello sviluppo del sistema informativo. Per ogni processo si potranno definire diversi livelli di automazione, che andranno da uno svolgimento prevalentemente operativo o decisionale da parte di persone a uno svolgimento totalmente automatizzato.

Le tecnologie per la realizzazione delle applicazioni di supporto ai processi informativi potranno essere molteplici. Il sistema informativo potrà essere realizzato con tecniche tradizionali, basate sullo sviluppo di applicazioni informatiche, con l'utilizzo di software per la gestione di basi di dati e di middleware per la realizzazione di applicazioni distribuite. Potranno essere utilizzate tecnologie innovative rivolte alla gestione di flussi di attività, quali i Workflow Management System (WFMS), che consentono di realizzare applicazioni specificatamente rivolte alla gestione di processi informativi.

## 2. Definizione di processo aziendale

Un processo aziendale è caratterizzato da un insieme di *attività*, collegate tra loro, per fornire un certo *output* a partire da *input* definiti. L'output può essere un prodotto o un servizio e viene utilizzato da determinati *clienti* o utilizzatori. Il processo aziendale nella sua esecuzione può richiedere l'interazione con diverse fonti di informazioni, anche nel corso della sua esecuzione.

Uno schema generale di processo aziendale è illustrato in figura 1.

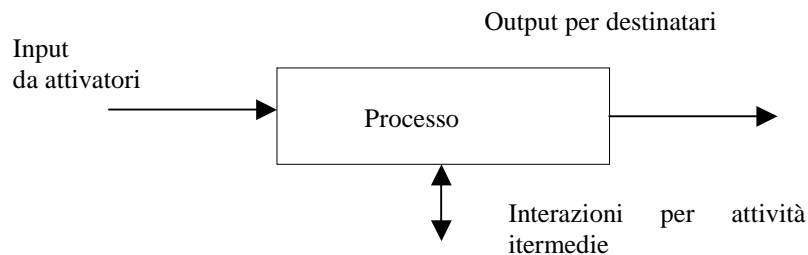


Figura 1 - Descrizione generica di un processo

I processi vengono in generale descritti a diversi livelli di dettaglio. Si può parlare di macroprocessi quando si tratta di processi complessi che al loro interno possono essere scomposti in altri sottoprocessi e attività.

Nella tabella seguente troviamo alcuni esempi di macroprocesso all'interno di un'organizzazione complessa.

### **Esempio: Macroprocessi nel Ministero delle Finanze**

- Assistere i cittadini, le imprese e gli intermediari fiscali nel rispetto degli adempimenti
- Attribuire il codice fiscale
- Riscuotere le imposte
- Ricevere le dichiarazioni e atti
- Controllare la correttezza delle imposte versate
- Eseguire i rimborsi
- Accertare l'evasione
- Fornire certificazioni e visure
- Gestire il personale e pianificare le attività (autoamministrazione)

*Tabella 1 - Esempi di macroprocessi*

#### *Identificazione dei processi*

L'analisi dei processi richiede di rappresentare i processi all'interno delle organizzazioni. Per potere procedere alla loro rappresentazione, è però necessario prima identificare quali sono i processi presenti all'interno dell'organizzazione. Tale identificazione può essere spesso difficile e laboriosa, perché i processi sono spesso costituiti da attività svolte nell'organizzazione, ma i processi stessi non identificati esplicitamente e possono avere degli elementi di sovrapposizione, quali lo svolgimento di due processi da parte dello stesso ufficio, o da parte della stessa persona, oppure attività comuni. Le stesse attività elementari che compongono il processo possono essere di difficile identificazione o difficilmente separabili da altre attività collegate. All'interno dell'organizzazione sarà necessario esaminare le modalità di esecuzione delle attività e tutta la documentazione di tipo normativo relativa allo svolgimento di attività, spesso costituita da testi in formato libero.

Al fine dell'identificazione dei processi può essere utile esaminare i seguenti elementi nella documentazione raccolta:

- *Eventi*: si considerano gli eventi che danno inizio ad un processo, quali ad esempio l'arrivo di una richiesta di attribuzione di codice fiscale, gli eventi che portano alla terminazione di un processo, sia essa per il completamento del processo, sia per una sua interruzione dovuta a una situazione anomala, e gli eventi significativi nello svolgimento del processo.
- *Verbi*: un elemento utile per scomporre un processo nelle sue attività elementari può essere dato dall'esame dei verbi contenuti nella documentazione raccolta: verbi

quali predisporre, approvare, richiedere il parere e così via, sono tutti indicativi di attività da svolgere all'interno del processo.

- *punti di decisione*: come si vedrà più in dettaglio quando nel seguito del capitolo si parlerà di modelli per la rappresentazione dei processi, una parte significativa degli stessi è data da quei momenti nel loro svolgimento in cui devono essere prese delle decisioni; tali decisioni possono portare a diverse strade nell'esecuzione delle attività successive del processo, possono iniziare attività di negoziazione tra parti, possono portare a diversi tipi di terminazione del processo.

Nel seguito del capitolo si esamineranno le caratteristiche dei vari tipi di modelli, per cui le indicazioni sopra elencate risulteranno più chiare, e si vedrà anche come diversi modelli enfatizzano più alcuni aspetti rispetto a altri.

### **3. Modelli per la rappresentazione dei processi aziendali**

Una prima differenziazione tra i diversi modelli per la rappresentazione di processi può essere fatta prendendo in considerazione le seguenti caratteristiche:

- *aspetto osservato*: l'enfasi può essere sulla rappresentazione delle relazioni interpersonali tra le persone coinvolte nell'esecuzione del processo oppure sulle tecnologie utilizzate per assistere gli utenti nel suo svolgimento;
- *livello di formalizzazione*: la descrizione dei processi può essere formalizzata, utilizzando tecniche basate su una formalizzazione matematica di tutti gli elementi del processo, oppure essere basata soprattutto su alcuni elementi grafici con una definizione volutamente non sempre precisa e integrati con descrizioni o elementi testuali che consentono la comprensione della descrizione anche da parte di non specialisti.

Attualmente non c'è un modo unanimemente riconosciuto ad applicato per classificare i processi. Una prima classificazione, proposta da Georgakopoulos et al [Geo95], divide i processi a seconda della complessità della loro struttura. I processi *scarsamente strutturati* sono caratterizzati da attività da eseguire in sequenza. I processi *altamente strutturati* sono invece caratterizzati da strutture complesse, con cicli, task eseguiti in parallelo e sincronizzazioni fra attività.

Un'ulteriore classificazione proposta in [Geo95] divide i processi in *human-oriented* e *system-oriented*. Nei processi *human-oriented*, agenti umani cooperano per eseguire le attività del processo e per garantire la correttezza dei risultati. I processi *system oriented* sono invece altamente automatizzati, includono attività pesanti dal punto di vista computazionale e di solito richiedono che sia il sistema a valutare la correttezza dell'esecuzione e delle informazioni prodotte.

Gli analisti tipicamente dividono i processi in *ad hoc*, *amministrativi* e *produttivi* [McC92]. I processi *ad hoc* sono processi senza una struttura fissa e ben definita, caratterizzati da alta variabilità. Per questi processi è difficile definire uno schema, dato che la sequenza delle attività non è prestabilita, ma viene definita dinamicamente,

durante l'esecuzione del processo. La riorganizzazione di un'azienda o un'azione di una partita di calcio sono esempi di processi ad hoc.

Un sistema che supporta un processo ad hoc deve consentire la cooperazione tra persone che contribuiscono alla sua esecuzione, al fine di definire (dinamicamente) la struttura del processo man mano che questo procede.

I processi amministrativi sono processi prevedibili e ripetibili, per i quali è possibile definire uno schema che viene seguito da molte o tutte le istanze (esecuzioni) del processo. Le varie attività che fanno parte di un processo amministrativo sono in genere eseguite da agenti umani (sono processi human-oriented), e comportano tipicamente l'inserimento di dati o la scrittura di documenti. La prenotazione di un viaggio, la valutazione di una richiesta di pensione di invalidità e l'assunzione di un nuovo impiegato sono esempi di processi amministrativi.

Data la loro ripetitività, i processi amministrativi si prestano bene ad essere automatizzati; uno strumento di supporto all'esecuzione dei processi amministrativi deve consentire la specifica del processo e deve poi supportarne l'esecuzione, programmando le attività da ed eseguire e fornendo i dati e i documenti necessari agli agenti (umani) che svolgono le attività.

I processi produttivi sono system-oriented, altamente strutturati e comportano un elevato numero di transazioni che accedono a vari sistemi informativi. La costruzione di un aereo e l'aggiornamento di un *data warehouse* sono esempi di processi produttivi. Anche i processi produttivi, come quelli amministrativi, sono prevedibili e ripetibili, e come tali possono essere automatizzati. In questo caso però, il software di supporto deve essere molto sofisticato, in quanto deve essere in grado di cooperare con sistemi informativi esterni che tipicamente sono eterogenei, autonomi e distribuiti.

Le "dimensioni" che fanno parte delle notazioni per la descrizione dei processi possono essere molte, ma in generale in tutte le rappresentazioni si troveranno i seguenti elementi, che potranno essere caratterizzati in modo diverso e quindi porteranno a definire le caratteristiche specifiche di un dato modello:

- *attività* (o task): i processi sono composti da attività da eseguire, che possono essere elementari, e quindi avere una definizione precisa e immediata di specifici compiti, o essere a loro volta costituite da un insieme di attività da svolgere; si parla in quest'ultimo caso di descrizione a più livelli o gerarchica dei processi, e i processi si definiscono come costituiti da *sottoprocessi*.
- *dati/oggetti manipolati*: le attività di un processo saranno basate su operazioni svolte su oggetti; tali oggetti potranno essere di natura diversa, quali le materie prime in un processo produttivo e i documenti ufficiali protocollati in una pratica amministrativa.
- *agenti/ruoli*: si parlerà di esecutori dei processi e a tale scopo diversi approcci potranno essere utilizzati – in tale contesto si troveranno termini quali esecutore, agente, ruoli e così via; in generale un esecutore potrà essere una persona, una

macchina o un particolare programma software, oppure l'esecuzione potrà richiedere diversi esecutori di tipologie diverse nello stesso momento; in particolare, nelle strutture aziendali e amministrative, si porrà il problema di identificare chi all'interno di una struttura organizzativa dovrà eseguire una determinata parte di un certo processo. E' importante che l'assegnamento delle attività agli esecutori possa essere effettuato in modo flessibile e in modo da minimizzare i cambiamenti necessari a fronte di modifiche nella struttura del processo o dell'organizzazione.

- *Punti di decisione*: come già indicato nei paragrafi precedenti i punti di decisione sono importanti all'interno dei processi, perché solitamente comportano strade alternative nell'esecuzione dei passi successivi del processo e possono portare a diverse fasi di negoziazione prima di raggiungere il risultato.
- *Eccezioni*: una problematica che si sta affrontando in tempi recenti sia a livello di modellizzazione che a livello di strumenti di supporto all'esecuzione dei processi è quella delle eccezioni; è difatti difficile prevedere anticipatamente in modo esaustivo l'intera gamma di situazioni che potranno presentarsi durante l'esecuzione del processo, in particolare se queste situazioni non fanno parte di una pratica consolidata o normata nell'esecuzione, ma richiedono un trattamento ad hoc di specifiche attività. Si è visto nella pratica dell'uso di tali modelli e strumenti che non è opportuno congelare un processo definendo tutte le situazioni possibili, ma è utile definire, oltre alle situazioni di esecuzione normale o usualmente prevedibile di esecuzione, anche condizioni particolari che possono presentarsi durante l'esecuzione, magari anche in parti diverse del processo, e definire il comportamento di tipo generale da tenere in tali situazioni.

Sulla base degli elementi sopra indicati, una possibile classificazione dei modelli di rappresentazione dei processi è la seguente:

- *modelli basati sui dati*: i modelli basati sui dati enfatizzano il flusso di dati e/o documenti nell'esecuzione del processo; un esempio classico di tale modello, i diagrammi di flussi di dati o DFD viene presentato nel seguito.
- *basati su attività*: molti modelli hanno come elemento caratterizzante la descrizione delle attività da svolgere e la loro sequenza, descritta in modo più o meno formale e articolato; nel seguito verrà presentato il modello WIDE come esempio di tale modello.
- *basati su comunicazione*: un modello basato sull'interazione tra agenti e sulle fasi di negoziazione necessarie nell'esecuzione del processo è il modello Action workflow presentato nel seguito, che presenta numerosi aspetti originali ed è particolarmente adatto a rappresentare processi in cui gli elementi di decisione sono da considerarsi importanti.

## 4. Tre tipi di modelli per la rappresentazione dei processi

### 4.1. Data Flow Diagram

I data flow diagram sono un insieme di notazioni per la descrizione dei processi utilizzate soprattutto nell'ambito della progettazione di sistemi informativi, e in particolare nella progettazione congiunta di dati e funzioni, e nella progettazione del software, per descrivere le funzionalità del sistema.

Il concetto su cui si basano i data flow diagram (DFD o diagrammi di flussi di dati) è quello di processo che elabora dati in ingresso e produce dati in uscita. Il sistema viene visto come un insieme di processi collegati tra loro da flussi di dati, che indicano come i dati prodotti da un processo vengono utilizzati da altri processi. La tecnica dei DFD è di larga diffusione soprattutto grazie alla sua notazione grafica, che rende agevole la lettura delle specifiche del sistema anche agli utenti e ai fornitori dei requisiti per il sistema stesso. La notazione grafica utilizzata per i DFD viene inoltre complementata da informazioni contenute in un dizionario dei dati, in cui i dati e i processi illustrati nei diagrammi possono essere descritti in maggior dettaglio.

Gli elementi che caratterizzano un diagramma DFD sono i seguenti:

- *processi*, che rappresentano le unità di elaborazione dei dati;
- *flussi di dati*, che indicano scambi di dati tra i processi;
- *archivi (o file)*, che rappresentano informazioni memorizzate in modo permanente nel sistema;
- *sorgenti e pozzi (o destinazioni) di dati*, che rappresentano i fornitori dei dati e i destinatari dei dati esterni al sistema

Una notazione grafica adottata nei diagrammi per rappresentare questi elementi è illustrata in figura 2.

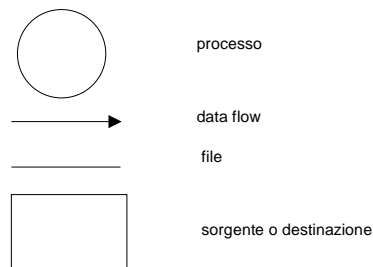


Figura 2 - Notazione grafica per diagrammi di flussi di dati

A ciascun elemento grafico è associato un nome, che deve essere univoco e che deve rappresentare in modo chiaro per il lettore del diagramma l'uso dell'elemento nel sistema.

Per i processi, il nome deve essere un verbo, o un verbo con oggetto, per indicare il tipo di trasformazione di dati svolta dal processo. Per i flussi di dati, vengono utilizzati sostantivi, sostantivi composti, oppure con attributi. Anche in questo caso il nome del flusso è determinante per la comprensione del diagramma, e deve essere univoco. È necessario inoltre evitare l'uso di nomi di tipo generico, quali "elabora" per i processi o "dato" per i flussi di dati, in quanto il loro uso non renderebbe più chiara la lettura dei diagrammi.

Ai processi inoltre è associato un numero identificativo univoco, che ha significato anche nell'ambito della scomposizione gerarchica dei processi che verrà illustrata nel seguito.

Come esempio di DFD, illustriamo la rappresentazione del sistema informativo per il caso di una società di organizzazione di corsi. Una sintetica descrizione informale del processo è contenuta nel riquadro di figura 3.

1. La società è una ditta di consulenza e formazione di medie dimensioni che tiene corsi pubblici e presso clienti nelle maggiori città europee e a volte altrove.
2. I partecipanti si iscrivono ai corsi per posta e per telefono. Ogni iscrizione è seguita da una lettera di conferma e da una fattura inviata al partecipante.
3. I pagamenti arrivano per posta. Ogni pagamento deve essere associato a una delle fatture relative ai pagamenti attesi.
4. Vi è un meccanismo che consente di cancellare l'iscrizione di un partecipante a un corso.
5. Una volta che un partecipante ha seguito uno dei corsi della ditta, o espresso interesse in uno di essi, il suo nome è inserito in una lista di persone utilizzata in seguito per pubblicizzare i corsi. Questa base di dati contiene informazioni su circa 10000 persone in numerose organizzazioni diverse.
6. Oltre ai normali rapporti sulle vendite, si deve poter rispondere a domande quali:
  7. Quali persone di una certa organizzazione hanno frequentato il seminario sui DFD? Come lo hanno valutato?  
Che istruttore insegnerà il corso di Analisi e Progettazione a Londra il prossimo mese?

*Figura 3 - Esempio di processo aziendale*



Nella figura seguente si presenta una possibile rappresentazione tramite DFD del processo di gestione corsi sopra descritto. Come si avrà occasione di notare, non tutti gli elementi della descrizione testuale compariranno nella rappresentazione del processo. Alcuni saranno omissi, altri saranno descritti in modo più dettagliato all'interno di un dizionario dei dati, come verrà illustrato nel paragrafo successivo.

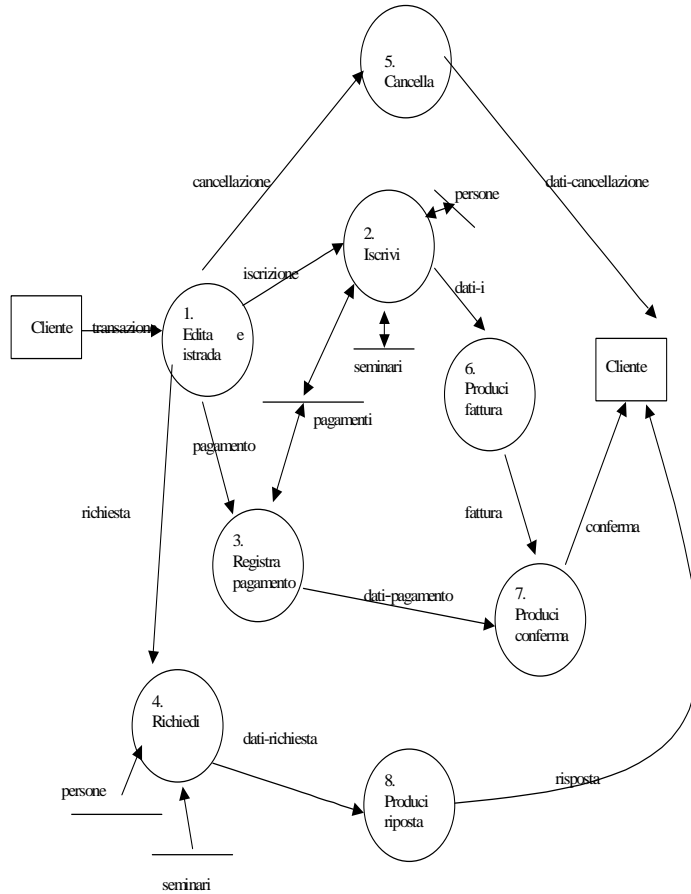


Figura 4 - DFD per esempio organizzazione corsi

Nella figura si vede come gli elementi della notazione grafica vengono utilizzati per rappresentare il processo. Sorgente e destinazione dei dati è in questo caso il cliente, che viene graficamente rappresentato due volte, per una migliore chiarezza di lettura del diagramma, separando i dati in ingresso da quelli in uscita. Da notare che si tratta di un unico elemento, rappresentato in questo modo solo per comodità notazionale, e pertanto identificato univocamente tramite il nome. Analoghe considerazioni possono essere fatte

per gli archivi persone e seminari. In generale, sarà opportuno evitare di avere diagrammi con linee che si incrociano (anzi, queste sono da alcuni autori esplicitamente vietate), e ricorrere a eventuali duplicazioni di elementi. Questo però è possibile solo per i dati (archivi, sorgenti o destinazioni). Si noterà nel diagramma come si è intenzionalmente evitato di avere flussi di dati tra processi con lo stesso nome: infatti utilizzare lo stesso nome verrebbe dire indicare lo stesso flusso. Invece flussi quali iscrizione e dati-i, pur contenendo entrambi dati relativi a iscrizioni, sono semanticamente diversi, perché in un caso si tratta di richiesta di iscrizione, nel secondo caso di iscrizione già valutata e quindi inserita nell'archivio seminari come nuova iscrizione. Verranno anche effettuati controlli sulla validità della richiesta, quali la verifica che nella data richiesta venga effettivamente erogato il seminario, e il controllo che il seminario non sia ancora completo.

Come si noterà, tutti i flussi di dati hanno un nome, eccetto quelli diretti o provenienti da archivi, per i quali si intende un accesso ai dati dell'archivio. Le frecce in ingresso negli archivi indicano inserimento o aggiornamento, quelle in uscita indicano lettura e quelle bidirezionali operazioni di lettura e scrittura.

#### 4.1.1. Dizionario dei dati

Come accennato nel paragrafo precedente, i soli nomi dei processi e dei dati nel diagramma sono molto significativi, ma non sono sufficienti da soli a identificare con precisione il suo contenuto.

Si associa pertanto a ciascun insieme di diagrammi un dizionario dei dati, che fornisce i dettagli relativi ai dati contenuti nei diagrammi, Per quanto riguarda i processi, si rimanda al paragrafo successivo.

Ciascun dato potrà essere dettagliato indicando i suoi componenti elementari utilizzando la seguente notazione:

- =           equivalenza
- +           composizione di dati
- [ ] scelta di una delle opzioni tra parentesi, che possono essere separate da una barra verticale | o da opportuna giustificazione del testo
- ()          indica elemento opzionale
- { }         indica un elemento di tipo iterativo, ripetibile più volte

Alcuni esempi di descrizioni di archivi o flussi di dati nel dizionario dei dati sono i seguenti:

```

dati-pagamento =       nomecliente +
                          indirizzo-cliente +
                          numero-fattura +
                          ammontare-pagamento

numero-fattura =        codice-provincia +
                          numero-conto-cliente +
                          numero-venditore +
                          numero-sequenziale-fattura

```

persone = {persona}  
ordine = buono-ordine + (pagamento)  
ammontare-pagato = [ammontare-lire | ammontare-euro]

Nel dizionario dei dati si descrivono i dati finché il loro significato risulta chiaro ai lettori dei DFD. Non è necessario pertanto giungere a una descrizione dei dati precisa come ad esempio è richiesto dai linguaggi di programmazione.

#### 4.1.2. Scomposizione gerarchica dei processi

Anche per i processi non è sufficiente indicare il nome del processo per descriverlo in modo completo.

Per quanto riguarda i processi, la loro descrizione dettagliata può passare da un “raffinamento” di un processo, in cui il processo viene scomposto gerarchicamente utilizzando per la sua descrizione ai livelli successivi in altri diagrammi di flusso dei dati. Questa scomposizione gerarchica dei processi può essere effettuata a più livelli, fino a arrivare a processi molto semplici per cui tale scomposizione non avrebbe più significato. Anche questi processi, detti *processi elementari*, devono però essere descritti più in dettaglio, per indicare quali tipo di elaborazioni di dati verranno effettuate all’interno del processo. I processi elementari sono descritti nel dizionario dati utilizzando una descrizione testuale, che potrà anche essere di tipo semi-formale. Uno dei formalismi più adottati per descrivere i processi nel dizionario dei dati è il cosiddetto *linguaggio strutturato*, che si basa sui classici costrutti della programmazione strutturata:

- sequenza
- ciclo (“ripeti ... finché”, oppure “finché è vera una certa condizione esegui determinate azioni”)
- alternativa (“se ... altrimenti”)

Un esempio di specifica di processo elementare tramite linguaggio strutturato è il seguente:

##### *Processo Produci fattura*

Se l'*ammontare* della *fattura* supera L. 1000000  
  se il *conto* del *cliente* ha *fatture* non pagate con più di 60 giorni di ritardo  
    *conferma* condizionata al pagamento del debito  
    altrimenti (conto OK)  
      emetti *conferma* e *fattura*  
altrimenti (*fattura* di 1000000 o meno)  
  se il *conto* ha *fatture* non pagate con più di 60 giorni di ritardo  
    emetti *conferma*, *fattura* e  
    scrivi un messaggio a rapporto delle azioni di credito  
altrimenti (conto OK)  
  emetti *conferma* e *fattura*

Nel processo indicato nell'esempio si vedono sia esempi di alternative, anche annidate tra loro, che di sequenze di azioni. Alcuni dei termini indicati nel testo possono avere una descrizione nel dizionario dei dati, e vengono evidenziati nel testo (ad esempio fattura, ammontare, e così via). In questo modo si può effettuare una verifica di consistenza tra i termini utilizzati nei diagrammi e le relative descrizioni dei processi; si può inoltre verificare se i processi utilizzano solo dati descritti nel dizionario dei dati.

I processi vengono considerati elementari solo se la loro descrizione testuale è contenuta. In genere si richiede che la descrizione testuale di un processo non superi la lunghezza di una pagina di testo. Qualora i processi siano più complessi, si procede a una scomposizione gerarchica dei processi. Per ciascun processo complesso viene cioè creato un diagramma DFD, che viene identificato tramite il numero associato nel DFD di più alto livello al processo stesso. Ad esempio il processo 1 sarà scomposto nel DFD 1. I processi al suo interno saranno numerati progressivamente e utilizzando una numerazione di tipo gerarchica. Ad esempio il DFD conterrà i processi 1.1, 1.2, 1.3, il DFD 1.1. conterrà i processi 1.1.1, 1.1.2, 1.1.3, e così via. A volte si utilizza la notazione abbreviata .1, .2, .3 per indicare i sottoprocessi X.1, X.2, X.3 del processo X nel relativo DFD.

Nella scomposizione devono essere seguite alcune regole che garantiscono la consistenza dei diagrammi. È necessario che tutti i flussi in ingresso al processo che viene scomposto siano presenti anche nel DFD dettagliato corrispondente; la stessa regola vale anche per i flussi in uscita, siano essi diretti a altri processi o esternamente al sistema. È possibile inoltre basarsi sulle descrizioni del dizionario dei dati per scomporre flussi di dati in flussi di maggior dettaglio. Ad esempio, con riferimento alla figura 5, il flusso A in ingresso nel processo 2.2 è scomposto nei flussi A1 e A2 nel diagramma DFD 2.2. È inoltre possibile aggiungere archivi, qualora questi non vengano utilizzati esternamente al processo considerato, e siano pertanto da considerarsi archivi interni al processo stesso.

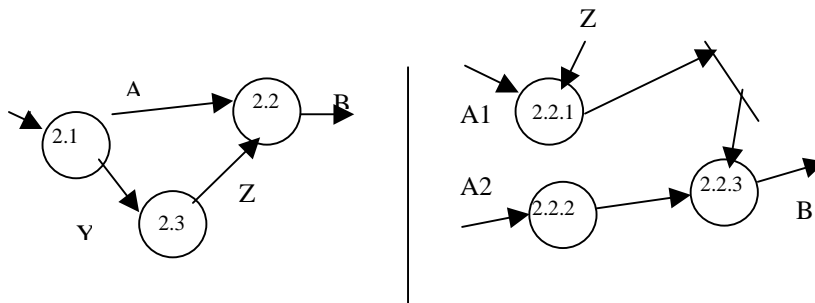


Figura 5 - Scomposizione gerarchica dei processi

## 4.2. Processi in WIDE

Il modello per la rappresentazione dei processi proposto nel progetto EU-ESPRIT WIDE (Workflows

on an Intelligent and Distributed database Environment) si colloca nel filone dei modelli che consentono di descrivere processi come insiemi di attività tra loro collegate da vincoli di precedenza e punti di sincronizzazione. Caratteristiche particolare del modello WIDE sono la possibilità di descrivere processi in modo flessibile, in particolare per quanto riguarda il trattamento delle eccezioni, e la definizione di un modello transazionale associato ai processi.

Il modello dei processi di WIDE è associato a altri due modelli che completano la descrizione dei processi con la descrizione delle informazioni a essi associate e degli agenti che svolgono attività nei processi. Pertanto il modello WIDE descrive i processi e le loro caratteristiche tramite tre modelli tra loro collegati:

- il *modello dei processi*, che definisce le attività che fanno parte del processo e l'ordine in cui tali attività devono essere eseguite;
- il *modello delle informazioni*, che consente di descrivere i dati e i documenti necessari all'esecuzione di un processo, in particolare in vista di un supporto informatizzato tramite workflow;
- il *modello dell'organizzazione*, che consente di descrivere la struttura dell'organizzazione e gli agenti che ne fanno parte, indipendentemente dalla descrizione dei singoli processi.

Nel seguito descriveremo brevemente i tre modelli. La descrizione completa del modello e dell'architettura di WIDE è presentata in [Gre99].

### 4.2.1. Il modello dei processi

La descrizione del flusso delle attività in WIDE è basata sulla descrizione formale del flusso di controllo delle attività nel processo.

Un workflow in WIDE è specificato essenzialmente da un insieme di attività (task) e da connettori che specificano l'ordine in cui i task devono essere eseguiti. Oltre a task e connettori, la specifica di un workflow può includere altri due tipi di elementi:

- *unità di modularizzazione, distribuzione e transazionali*: consentono di descrivere i processi a diversi livelli di dettaglio, isolandone alcune parti che debbono essere ritenute unitarie dal punto di vista della distribuzione del lavoro o dal punto di vista transazionale
- *eccezioni*: consentono di descrivere in modo compatto alcune situazioni di tipo anomalo che si possono verificare durante l'esecuzione del processo e richiedono un particolare trattamento, quale l'esecuzione di specifiche attività o l'aggiornamento di alcuni dati del processo o l'alterazione del normale flusso di esecuzione.

Un esempio di workflow che descrive un processo di prenotazione viaggi è presentato in figura 6.

Prenotazione viaggi

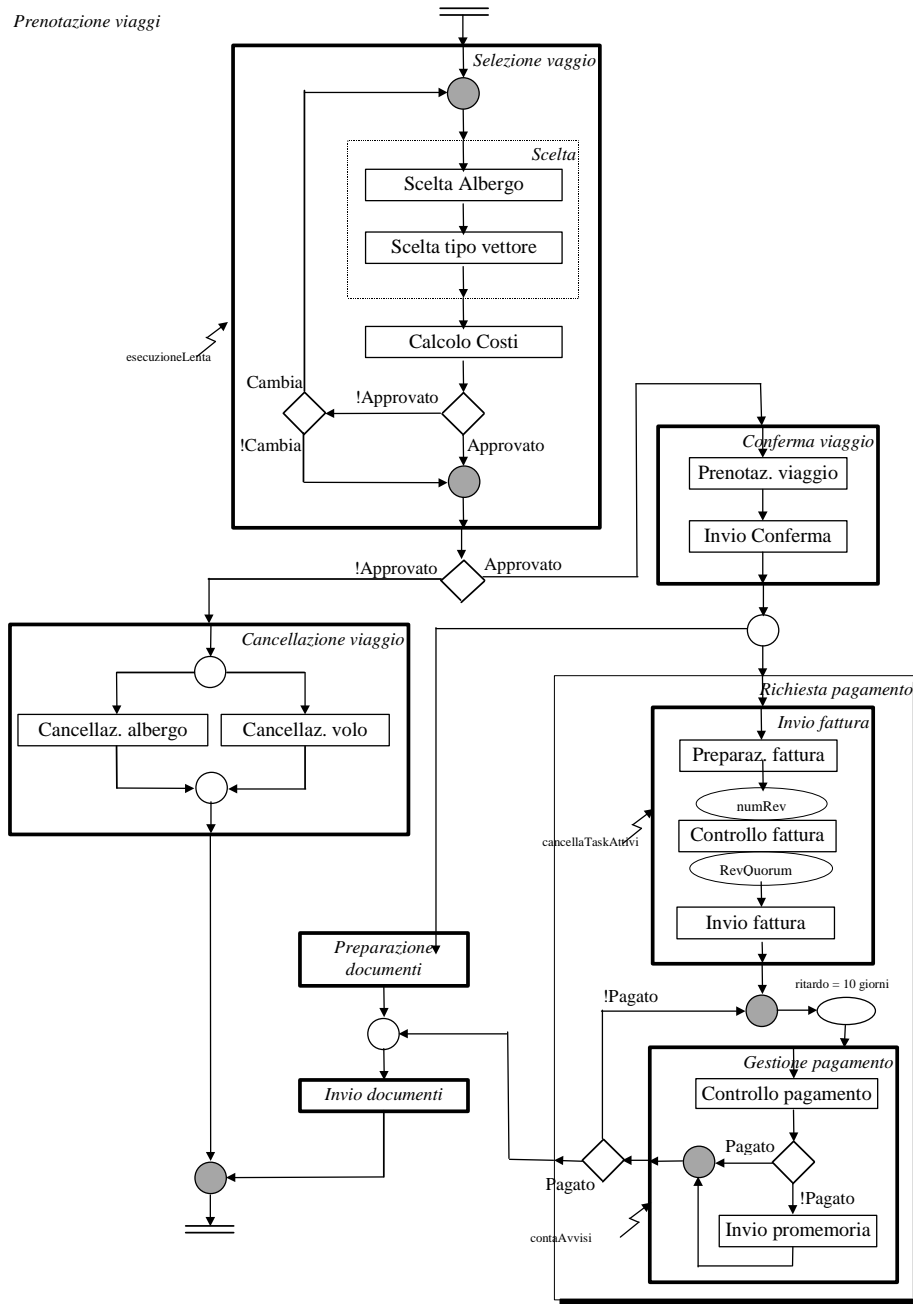


Figura 6 – Specifica WIDE di un processo per la prenotazione di un viaggio

Nella figura, i task sono rappresentati da rettangoli, mentre gli altri simboli sono connettori, e definiscono la struttura del processo. Una *istanza* di un workflow (chiamata *case* nella terminologia WIDE) è un'esecuzione del processo. Varie istanze dello stesso processo possono essere in esecuzione nello stesso momento (ad esempio, molte istanze del processo di prenotazione viaggi possono essere contemporaneamente attive). Nel seguito della sezione presenteremo nel dettaglio i diversi costrutti del modello WIDE e descriveremo poi la semantica del workflow di figura 6.

### Task

I task sono le attività elementari che compongono un processo. Un task è caratterizzato da:

- un *nome*, unico all'interno del processo;
- una *descrizione*, in linguaggio naturale, che descrive lo scopo del task;
- un insieme di *ruoli*, che indica le capacità necessarie per eseguire il task. L'esecutore del task deve essere scelto fra quelli che ricoprono almeno uno dei ruoli specificati. I ruoli sono l'anello di collegamento tra il modello di processo e quello dell'organizzazione, come meglio descritto in seguito;
- un insieme di dati associati al task, necessari per la sua esecuzione. I dati possono essere sia strutturati e di un tipo noto al sistema (ad esempio interi, reali, stringhe) che non strutturati (documenti prodotti con un programma di videoscrittura);
- un insieme di *azioni* predefinite che è possibile effettuare sul task; ad esempio, è possibile sospendere e poi riprendere l'esecuzione di un task, terminare un task o delegarlo per l'esecuzione ad un altro agente.

Uno stesso task può essere attivato (istanziato) più volte nello stesso case. Differenti istanze di task sono caratterizzate da un diverso (e progressivo) *numero di attivazione*<sup>1</sup>.

### Connettori

I connettori modellano l'interazione fra i task e definiscono la *struttura* di un workflow. Un task può avere una sola connessione in uscita ed una in ingresso. Due task A e B possono essere connessi direttamente (tramite una freccia, nel linguaggio grafico), con la semantica che non appena A termina, B viene mandato in esecuzione. In tutti gli altri casi, le connessioni fra task sono mediate da *connettori*. I connettori possono iniziare l'esecuzione parallela di task (connettori *fork*), o sincronizzare al termine di esecuzioni parallele (connettori *join*).

#### *Connettori fork*

I connettori di tipo fork sono preceduti da un task, chiamato predecessore, e seguiti da due o più task, chiamati successori. I connettori di tipo fork sono classificati come segue:

- *totale*: al termine del predecessore il sistema attiva tutti i successori.

---

<sup>1</sup> Fa eccezione il caso di istanze attivate nel contesto di un multitask, descritto nel seguito.

- *condizionale*: ad ogni successore è associata una condizione: al termine del predecessore, il sistema attiverà tutti i successori la cui condizione è vera.

### Connettori join

I connettori di tipo join sono preceduti da due o più task, chiamati predecessori, e seguiti da un task, chiamato successore. I connettori di tipo join sono classificati come segue:

- *totale*: il successore viene attivato solo al termine di tutti i predecessori;
- *parziale*: al connettore join è associato un valore  $k$ : il successore viene attivato non appena  $k$  predecessori con lo stesso numero di attivazione sono terminati. La terminazione di ulteriori predecessori non ha nessun effetto.  $k$  può essere una costante o una variabile del processo. Per default, il join parziale ha  $k=1$ .
- *ciclico*: un'istanza del successore viene attivata tutte le volte che un predecessore termina.

La rappresentazione grafica di task e connettori è mostrata in figura 7.

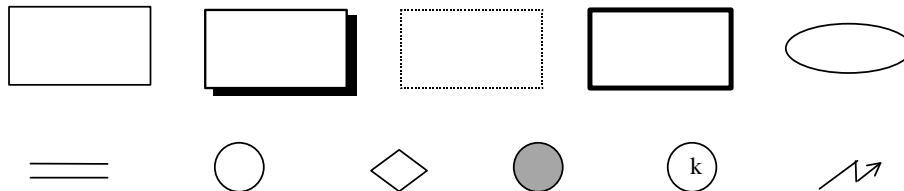


Figura 6 - Simboli del modello WIDE. Da destra a sinistra, dall'alto al basso: task, sottoprocesso, supertask, business transaction, wait task; simbolo di inizio-fine, fork/join totale, fork condizionale, ciclo, join parziale, trigger

### Simboli di inizio e fine processo

Ogni workflow ha un simbolo di inizio e uno o più simboli di fine processo. Il simbolo di inizio ha uno o più successori (nel caso di più di un successore, il simbolo di inizio deve essere seguito da un connettore fork), e analogamente il simbolo di fine ha uno o più task predecessori (nel caso di più successori, il simbolo di fine deve essere preceduto da un connettore join).

Quando un workflow viene istanziato, il sistema attiva il successore (o i successori) del simbolo di inizio. L'istanza terminerà al termine di un predecessore del simbolo di fine workflow.

Quando l'istanza termina, alcuni task possono essere ancora attivi: questi task vengono cancellati, e i rispettivi agenti vengono informati della cancellazione.



**Wait task**

Il wait task è un particolare tipo di task che non compie azioni e che non deve essere assegnato ed eseguito da un agente. Il suo compito è di attendere che una certa condizione si verifichi. La condizione associata alla definizione del wait task può essere definita come un predicato sui dati del workflow, sul tempo, o sull'occorrenza di eventi esterni. Non appena la condizione è verificata, il task termina, attivando così il task successivo.

**Multitask**

Un multitask definisce un insieme di task che eseguono la stessa funzione in parallelo. L'attivazione del multitask corrisponde all'attivazione contemporanea di più istanze dello stesso task, assegnate ad agenti differenti. Tutte le istanze hanno lo stesso numero di attivazione, che coincide con quello del multitask (differenti attivazioni del multitask hanno comunque numeri di attivazione differenti e progressivi). Il multitask ha un duplice scopo: consente di specificare in modo compatto un insieme di task che compiono la stessa funzione e consente di definire a tempo di esecuzione il numero delle istanze che devono essere attivate, che può dipendere dal valore di una variabile del workflow (indicato con  $j$  in figura 8).

Ad esempio, nel workflow di prenotazione viaggi, revisione di un documento, il numero dei revisori che controllano una fattura può essere definito in funzione dell'importo della fattura. Un multitask termina quando termina un numero  $k$  di istanze di task.  $k$  è *quorum* del multitask, e può essere una costante o può identificare una variabile di processo il cui valore viene verificato ogni volta che una istanza del multitask termina. Ad esempio, è possibile specificare che il multitask di controllo fattura termini non appena la maggioranza dei revisori ha dato un parere concorde (favorevole o contrario), senza dover necessariamente attendere il giudizio di tutti i revisori.

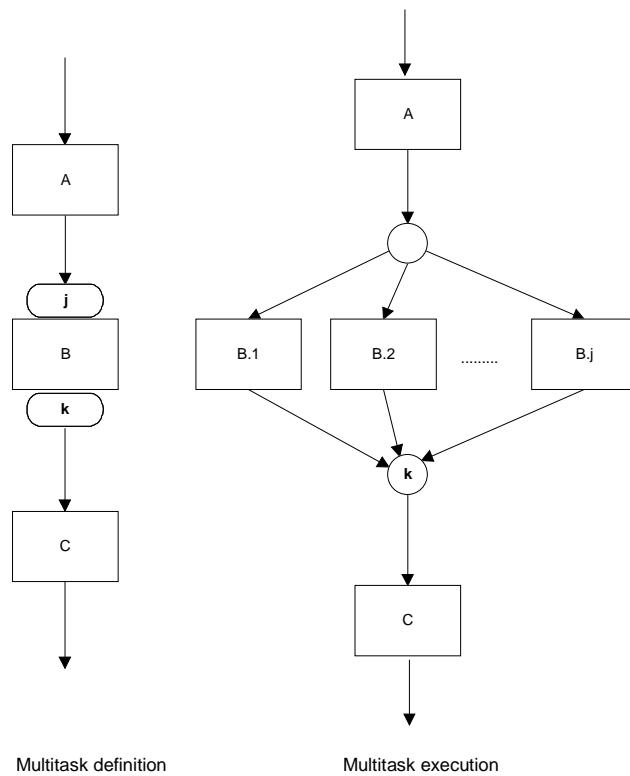


Figura 7 - Specifica del multitask (sinistra) e sua esecuzione (destra)

### Sottoprocessi, supertask, e business transactions

Sottoprocessi, supertask e business transaction sono costrutti che raggruppano insiemi di task, consentendo di modularizzare la specifica di un workflow e di definire proprietà transazionali. Nessuno di questi costrutti può essere direttamente eseguito (istanziato): essi devono essere definiti all'interno di un workflow, e sono istanziati quando vengono raggiunti dal flusso di controllo, come per i task ordinari.

I *sottoprocessi* sono del tutto analoghi a workflow, salvo per il fatto che non possono essere direttamente istanziati. Essi hanno un insieme di dati in ingresso e in uscita, passati (per copia/risultato) al sottoprocesso e ritornati al processo padre al termine del sottoprocesso. Il sottoprocesso è una "scatola nera" per il processo padre, ed è la base per la definizione di specifiche riusabili: infatti, un sottoprocesso può essere riutilizzato (invocato) nel contesto di diversi workflow.

Come i sottoprocessi, anche i *supertask* sono composti da un insieme di task, collegati tramite connettori. Il supertask non ha però parametri di ingresso o di uscita, e vede le

stesse variabili del processo nel quale è definito. Inoltre, il supertask può essere definito ed istanziato solo nel contesto di un workflow, e non può quindi essere riusato per altri workflow.

La *business transaction* è l'elemento di base del modello transazionale di WIDE. Il concetto di business transaction ha lo scopo di consentire la specifica, a livello di workflow, di proprietà transazionali caratteristiche delle operazioni su basi di dati. Un workflow ha però caratteristiche molto diverse da una classica transazione eseguita su di una base di dati. Infatti, l'esecuzione di un workflow comporta azioni che hanno effetto nel mondo reale, quali l'invio di documenti per posta o la vendita di un prodotto. Tali operazioni non possono essere annullate, come lo possono invece essere le operazioni di manipolazione dati. Inoltre, è praticamente impossibile eseguire un intero case in modo isolato, dato che un case può durare parecchi giorni, o anche mesi, ed altre applicazioni hanno tipicamente la necessità di accedere agli stessi dati in quel lasso di tempo. Il concetto di business transaction e il modello transazionale di WIDE hanno quindi lo scopo di fornire determinate proprietà transazionali ai workflow, tenendo conto delle caratteristiche sopra citate.

Una *business transaction* raggruppa task che formano un'unità transazionale, ovvero che devono essere eseguiti in modo atomico e isolato rispetto agli altri task dello stesso o di altri case. In WIDE, ogni task deve fare parte di una business transaction o deve essere esso stesso una business transaction. Pertanto, un workflow può essere visto come un insieme di business transaction. All'interno di una business transaction i vincoli di isolamento possono essere definiti in modo flessibile: è possibile specificare se i risultati di un task devono essere resi visibili o meno agli altri task della business transaction.

Nel caso che l'esecuzione di una istanza debba essere annullata, il sistema garantisce l'atomicità dell'esecuzione come segue: le business transaction attive vengono abortite; questo è possibile, dato che le business transaction hanno le classiche proprietà "acide" delle transazioni. Le business transaction completate vengono invece compensate, dal punto di vista semantico, eseguendo un task di compensazione associato ad ogni business transaction. Ad esempio, se una business transaction ha il compito di inviare un documento ad un cliente, il task di compensazione potrebbe comportare una telefonata o un'altra lettera al cliente che lo informa che il contenuto della precedente lettera non deve essere considerato valido. La compensazione avviene in ordine inverso rispetto all'esecuzione delle business transaction, e procede fino a che il case è stato compensato.

### **Il workflow Prenotazione Viaggi**

Nel seguito descriveremo la semantica del workflow Prenotazione Viaggi di figura 6. Una nuova istanza del processo viene attivata da un agente alla richiesta della prenotazione di un viaggio da parte di un cliente. L'impiegato propone al cliente un viaggio (business transaction *Selezione viaggio*), verifica la disponibilità del mezzo di trasporto e della sistemazione alberghiera e calcola il costo del viaggio. Il cliente può decidere se accettare o meno il viaggio proposto dall'agente (l'agente assegna alla variabile *Approvato* il valore *true* o il valore *false* a seconda della decisione del cliente),

e in caso di rifiuto può decidere di provare con un'altra proposta (alla variabile *Cambia* viene assegnato il valore *true*) o cancellare la sua richiesta.

In caso di approvazione, il processo continua con la preparazione e l'invio della fattura al cliente (business transaction *Conferma viaggio*) e con la preparazione dei documenti di viaggio (business transaction *Preparazione documenti*). In parallelo alla preparazione dei documenti viene attivato il sottoprocesso *Richiesta pagamento*, composto dalle business transaction *Invio fattura* e *Gestione pagamento*. Si noti, all'interno di *Invio fattura*, il multitask *Controllo fattura*, descritto in precedenza, la cui molteplicità è definita dalla variabile del case *NumRev*, ed il cui quorum è definito dalla variabile di case *RevQuorum*. Queste variabili sono opportunamente impostate dal task *Preparazione fattura*. Da ultimo, quando il viaggio è stata pagato e i documenti sono stati preparati (si noti il join totale che attende il termine del task *Prepara documenti* e del sottoprocesso *Richiesta pagamento*), il processo prevede, come ultimo task, l'invio dei documenti al cliente.

Nel caso che invece il cliente non accetti i viaggi proposti dall'agente, la richiesta viene cancellata (business transaction *Cancellazione viaggio*), ed il case termina subito dopo.

#### 4.2.2. Modello delle informazioni

Il modello delle informazioni descrive le informazioni che possono essere utilizzate, modificate o prodotte dai workflow. Le informazioni sono in generale indipendenti dai processi, nel senso che le stesse informazioni possono essere utilizzate da vari processi.

Il modello WIDE consente la specifica di quattro tipi di informazioni: variabili, form, documenti e cartelle:

- **variabili:** in WIDE è possibile definire variabili di diversi tipi: intero, reale, booleano, stringa, durata, data, enumerazione. Una volta definita, una variabile può essere associata ad uno o più workflow. In ogni caso, ogni istanza di workflow vede e modifica una copia locale della variabile. Le variabili di workflow possono essere utilizzate per definire predicati da associare ai wait task, ai fork condizionali, alla molteplicità di un multitask o al suo quorum;
- **form:** le variabili possono essere raggruppate in *form*, sia per definire proprietà comuni ad un insieme di variabili (come, ad esempio, la specifica di quale insieme di variabili può essere letto o modificato da un certo task), sia per definire come le variabili vengono, graficamente, presentate agli agenti. Inoltre, è possibile associare alla form un predicato sul valore delle sue variabili che deve essere verificato al termine di determinati task o case;
- **documenti:** sono insiemi di informazioni in un formato ignoto al sistema di gestione di workflow, e a cui il sistema non può accedere. I documenti sono creati e manipolati per mezzo di applicazioni esterne. Esempi di documenti sono immagini, file di testo o in formati proprietari (come Microsoft Word), sequenze televisive. Il sistema si fa solamente carico di passare i documenti agli agenti opportuni e di controllare l'accesso ad essi;
- **cartelle:** sono combinazioni di variabili, form, documenti o anche di altre cartelle. Una cartella può essere inizialmente vuota o può avere un contenuto predefinito. Le

cartelle sono utili quando c'è la necessità di raggruppare logicamente un insieme di elementi che devono essere trasferiti tra i task al procedere del case. È inoltre possibile specificare quali elementi di una cartella possono essere letti o modificati da un certo task.

#### 4.2.3. Modello dell'organizzazione

Lo scopo del modello dell'organizzazione è di descrivere la parte dell'organizzazione coinvolta nella gestione e nell'esecuzione dei processi, e di come questa è collegata ai task e ai workflow, al fine di consentire la specifica degli agenti (o dei gruppi di agenti) che hanno il diritto di eseguire un certo task.

Una caratteristica importante di WIDE è la netta separazione tra la descrizione dell'organizzazione e quella dei processi. Infatti, molti workflow sono in genere definiti nell'ambito della stessa organizzazione, e viceversa lo stesso workflow può essere eseguito nel contesto di diverse organizzazioni.

Lo schema dell'organizzazione è tipicamente definito in varie fasi: durante la specifica di un processo, viene definito il workflow insieme alle competenze necessarie (i *ruoli*) per l'esecuzione dei task del workflow. Questo tipo di informazione è indipendente dall'organizzazione.

La popolazione dello schema organizzativo (ad esempio, la definizione di quali agenti ricoprono un dato ruolo) è un passo separato, eseguito in modo indipendente dal precedente.

Questo approccio consente di modificare la popolazione dell'organizzazione senza modificare la specifica dei workflow; questa caratteristica è molto importante in particolar modo per le grandi aziende, che spesso definiscono il processo in modo centralizzato e lo distribuiscono ai vari dipartimenti per l'esecuzione.

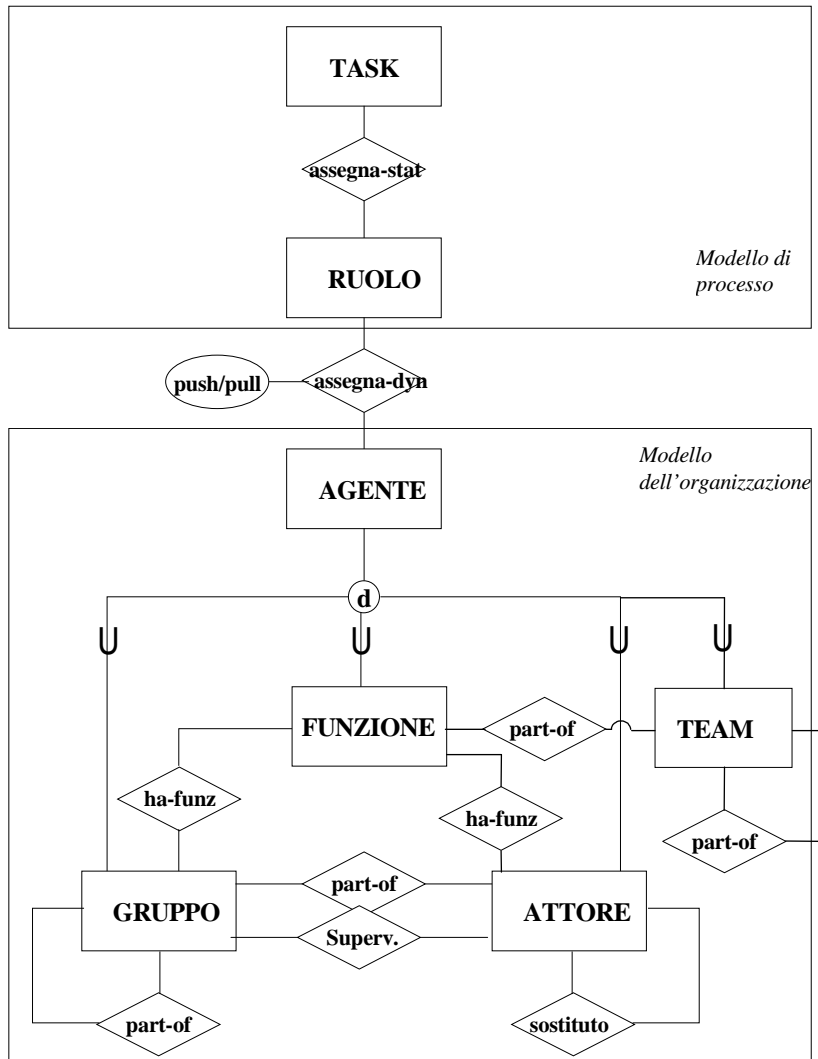


Figura 8 - Modello dell'Organizzazione di WIDE e sua relazione con il modello dei processi

Le differenti entità che compongono il modello organizzativo, assieme alle relazioni fra loro e con il modello dei processi, sono mostrate in figura 9. Il modello comprende le entità Task, Agente, Ruolo, Attore, Funzione, Gruppo, Team:

- un **Task** è una attività atomica dal punto di vista del sistema, eseguita da un singolo agente, che può essere umano o automatizzato;

- il **Ruolo** rappresenta una descrizione delle capacità necessarie per l'esecuzione di un task;
- un **Agente** è una qualunque entità in grado di eseguire un task. Un agente può a sua volta essere un Attore, un Gruppo, una Funzione o un Team:
- un **Attore** è un umano o un componente meccanico o elettronico in grado di eseguire attività, e quindi di interfacciarsi con il sistema di gestione dei workflow;
  - un **Gruppo** rappresenta un insieme di attori che hanno caratteristiche comuni, ad esempio sono assegnati allo stesso progetto o sono nella stessa sede;
  - una **Funzione** è una specifica di un insieme di gruppi, attori e team che hanno le stesse capacità;
  - un **Team** è una lista di funzioni necessarie a svolgere un certo compito; ad esempio, un team può essere definito come composto da un senior manager, un junior manager, e una segretaria;

Gli attori ed i gruppi sono concetti “concreti”, nel senso che un task può essere assegnato per l'esecuzione ad un attore o ad un gruppo. Le funzioni e i team sono invece concetti astratti, che sono stati introdotti per strutturare e raggruppare in modo flessibile attori e gruppi.

La figura 9 mostra anche le relazioni tra le entità. Le relazioni sono fondamentalmente di due tipi: quelle che collegano il modello organizzativo e di processo tramite la specifica di quale agente può ricoprire un certo ruolo, e quelle che sono interne e specifiche del modello organizzativo e che definiscono le relazioni fra le entità del modello. Il primo gruppo comprende le relazioni assegnamento statico (*assegna-stat*) e assegnamento dinamico (*assegna-dyn*).

- la relazione di assegnamento statico (*assegna-stat*) collega un task ad uno o più ruoli, al fine di descrivere le capacità richieste per eseguire il task; la cardinalità della relazione è molti a molti. Questa relazione è tipicamente popolata durante la specifica di un workflow;
- la relazione di assegnamento dinamico (*assegna-dyn*) collega i ruoli con gli agenti che sono in grado di ricoprire quel ruolo; la cardinalità della relazione è molti a molti, dato che più agenti possono ricoprire lo stesso ruolo, e un agente può ricoprire più ruoli. Anche questa relazione è popolata durante la specifica di un nuovo workflow. La semantica dell'attributo push/pull verrà dettagliata nel seguito

Le relazioni nel secondo gruppo dipendono invece dall'organizzazione più che da specifici workflow, e sono tipicamente definite in fase di specifica dell'organizzazione. Questo gruppo contiene le seguenti relazioni:

- *Ha funzione di* (*ha-funz*) definisce le funzioni di un attore o di un gruppo. La cardinalità di queste relazioni è molti a molti;

- *Part-of* collega un attore o sottogruppo al gruppo (padre) di cui fa parte, un team al team padre, o una sottofunzione alla funzione della quale fa parte. La cardinalità di queste relazioni è molti a molti;
- *Sostituto* collega un attore ad un altro attore che lo può sostituire nel caso che il primo attore decida di delegare un task. La cardinalità di questa relazione è uno a molti (un attore può avere un solo sostituto);
- *Supervisiona* collega un gruppo ad un attore che gestisce il gruppo ed i suoi membri. La cardinalità di questa relazione è uno a molti (un gruppo ha un solo supervisore).

### **Assegnamento di un task**

L'assegnazione di un task ad un agente può avvenire in due modi: il sistema può scegliere l'agente, nel rispetto dei vincoli definiti dal progettista del workflow, e assegnare direttamente il task ad un agente (modalità *push*), oppure può inserire il task nella lista delle attività pendenti di tutti gli agenti autorizzati ad eseguire il task. Quando un agente autorizzato seleziona il task per eseguirlo, il task viene rimosso dalla lista delle attività degli altri agenti (modalità *pull*). La maggior parte dei sistemi commerciali adotta la modalità *pull*, sia perché è più flessibile (non impone l'esecuzione ad un agente specifico ma la propone a molti), sia perché la modalità *push* è a volte considerata "fastidiosa" dall'impiegato, che si vede assegnato il lavoro da svolgere dal sistema.

Quando un agente ha terminato l'esecuzione del task, notifica al sistema che il task è completato ed il sistema determina il prossimo task da attivare nel case. Un task assegnato può anche essere delegato ad un altro agente (il sostituto), se l'operazione è fra quelle consentite per quel task.

### **Agenti particolari in un sistema di gestione di workflow**

In un sistema di workflow vi sono alcuni agenti che svolgono funzioni specifiche rispetto alla gestione e all'esecuzione dei workflow:

- l'*esecutore del task* è l'agente incaricato di eseguire una specifica istanza di task in un certo case;
- il *responsabile del case* è l'agente che supervisiona il case, e al quale vengono notificati i problemi e le situazioni eccezionali relative a quel case;
- l'*esecutore del case* è l'agente che ha attivato il case. Come per i task, anche per i workflow è possibile specificare il ruolo che deve ricoprire l'agente autorizzato ad istanziare un workflow, ovvero a creare un nuovo case;
- il *progettista* di un workflow è l'agente che ha specificato il workflow;
- l'*amministratore* del sistema è l'agente (o l'insieme di agenti) che gestisce il sistema di workflow, e che ha i diritti di installare nuove definizioni di workflow o di modificare la struttura dell'organizzazione.



#### 4.2.4. Gestione delle eccezioni

Uno dei problemi principali degli attuali sistemi di workflow è la loro limitata flessibilità: infatti, tali sistemi sono in grado di supportare in modo molto efficiente solo i processi che vengono eseguiti seguendo esattamente le specifiche. Spesso, però, si possono verificare situazioni “eccezionali” che non consentono di eseguire il processo nel modo previsto. Un incidente d’auto in un processo che gestisce il noleggio di automobili, la cancellazione di un ordine in un processo per la gestione degli ordini o guasti di componenti hardware o software sono esempi di situazioni eccezionali, che rendono necessaria una deviazione dall’esecuzione “normale” del processo.

In queste situazioni, gli utenti sono costretti a “imbrogliare” il sistema e ad eseguire azioni al di fuori del suo controllo al fine di gestire la situazione eccezionale, facendo invece credere al sistema che il processo è stato eseguito secondo le specifiche. In queste situazioni, i sistemi di gestione di workflow sono più di ostacolo che di supporto all’esecuzione dei processi.

Una delle caratteristiche più significative ed interessanti di WIDE è la possibilità di definire e gestire situazioni eccezionali, ovvero situazioni che fanno parte della semantica di un processo, pur non facendo parte della sua esecuzione normale.

Le eccezioni hanno caratteristiche diverse rispetto al flusso normale del processo; infatti, le eccezioni sono spesso asincrone, ovvero possono capitare in qualunque momento durante l’esecuzione del processo, e non solo in corrispondenza dell’esecuzione di uno specifico task. Ad esempio, la cancellazione di un viaggio in un processo di prenotazione viaggi o l’incidente di un’auto in un processo di noleggio auto possono capitare in vari momenti durante l’esecuzione di questi processi. Inoltre, molto spesso l’azione necessaria per gestire un’eccezione non comporta necessariamente l’attivazione di un task: al contrario, l’azione opportuna può essere l’invio di un messaggio ad un agente per segnalare la situazione eccezionale, o la cancellazione del processo.

Dato che un’eccezione non è direttamente causata dall’esecuzione di un task, e non viene gestita attivando un task, non è possibile rappresentarla tramite diramazioni nel grafo che descrive il processo.

Per questo, nel progetto WIDE sono stati studiati altri formalismi per la specifica dei comportamenti eccezionali, ed il formalismo che è stato scelto è quello di regole Evento-Condizione-Azione (ECA), nelle quali l’evento indica l’occorrenza di una situazione potenzialmente eccezionale, la condizione verifica che l’evento corrisponde effettivamente ad una situazione eccezionale che deve essere gestita, mentre l’azione, eseguita solo se la condizione vera, reagisce all’evento, al fine di gestire la situazione eccezionale. Il linguaggio di WIDE per le eccezioni si chiama *Chimera-Exc*, e deriva dal linguaggio per basi di dati attive *Chimera*, che è stato modificato per renderlo applicabile al contesto dei workflow.

Nel seguito descriveremo quali sono gli eventi, le condizioni e le azioni che si possono definire in *Chimera-Exc*.

*Chimera-Exc* consente la specifica di quattro tipi di eventi:

- eventi di *manipolazione dati*, che corrispondono alla modifica delle variabili del case. Una tipica applicazione di una regola con evento di manipolazione dati è il controllo di un vincolo definito sui dati;
- eventi *esterni*, definiti e segnalati da applicazioni esterne. Esempi di eventi esterni sono l'arrivo di una telefonata, o la richiesta di cancellazione di una prenotazione da parte dell'utente;
- eventi di *workflow*, che corrispondono all'attivazione ed alla terminazione di un task. Questi sono gli unici eventi sincroni rispetto all'esecuzione "normale" del processo;
- eventi *temporali*, a loro volta suddivisi in eventi di tipo deadline (scadenza), intervallo, o periodici. Ad esempio, è possibile definire regole attivate il primo giorno di ogni mese, o regole attivate quando è trascorso un certo tempo dall'attivazione di un task o di un case.

La parte *condizione* di una regola Chimera-Exc è un predicato sullo stato della base di dati del sistema di gestione di workflow, che comprende sia le variabili dei processi sia dati di sistema relativi all'esecuzione di task e case. La condizione, oltre a determinare se la parte azione deve essere eseguita o meno, identifica i case e gli agenti coinvolti.

La parte *azione* reagisce all'eccezione eseguendo operazioni *informative*, come l'invio di messaggi all'esecutore di un task o al responsabile del case, o *correttive*, quali l'istanziamento di un workflow o la terminazione di un case attivo.

WIDE consente di specificare regole globali o regole valide all'interno del contesto di uno specifico workflow. Le regole globali sono tipicamente utilizzate per definire politiche aziendali, valide per tutti i workflow. Ad esempio, una regola globale può definire il comportamento da seguire a fronte di una improvvisa indisponibilità di un agente che stava eseguendo un task critico.

Qui sotto riportiamo un esempio di regola Chimera-Exc definita per il processo di prenotazione viaggi di figura 6.

```

define trigger troppiAvvisi for prenotazioneViaggi
  events          modify(prenotazioneViaggi.numeroAvvisi)
  condition       prenotazioneViaggi(P),
                  occurred
                  (modify(prenotazioneViaggi.numeroAvvisi), P),
                  P.numeroAvvisi>3
  actions         notify (P.responsible, "Sono già stati spediti tre avvisi
al cliente      P.nomeCliente)
end

```

Figura 9 - Esempio di trigger

La regola viene attivata da una modifica della variabile *numeroAvvisi* in un case del workflow *Prenotazione Viaggi* (evento *modify(prenotazioneViaggi.numeroAvvisi)*). La condizione dapprima identifica, nella variabile P, i case per i quali la variabile *numeroAvvisi* è stata modificata dall'ultima esecuzione della regola (predicato *prenotazioneViaggi(P), occurred (modify (prenotazioneViaggi.numeroAvvisi), P)*), e quindi restringe P a rappresentare i case per i quali la variabile ha un valore maggiore di 3 (*P.numeroAvvisi>3*). Se la condizione è verificata (ovvero, se l'insieme dei case rappresentato da P non è vuoto), viene eseguita l'azione che notifica al responsabile del case che sono già stati spediti tre avvisi al cliente, in modo che vengano presi provvedimenti.

### 4.3. Action Workflow

Nell'approccio proposto negli Action Workflow il flusso di lavoro (ed il processo correlato) non è fatto soltanto di azioni ed attività finalizzate alla trasformazione di informazioni, ma è fatto soprattutto di interazioni tra persone, che comportano reciproche richieste, promesse e assunzioni di impegni. L'enfasi viene quindi posta sull'interazione tra fornitori e clienti di servizi nell'esecuzione del processo.

Gli elementi principali del modello Action Workflow sono i seguenti:

- **Attività** - Ogni compito lavorativo svolto dalle persone
- **Azione** - Ciò che le persone fanno quando chiedono o prendono reciprocamente impegni le une con le altre con l'obiettivo di ottenere una condizione di soddisfazione
- **Impegno** - Modalità di interazione tra cliente e fornitore
- **Processo** - Insieme di attività scatenati da azioni, sotto la responsabilità di un operatore, e che avvengono in un determinato spazio o tempo

Un processo è dunque una rete di transazioni tra persone (clienti ed esecutori) che produce azioni coordinate ai fini della soddisfazione del cliente.

#### 4.3.1. L'action workflow come modello per la descrizione dei processi

I processi nell'action workflow sono descritti gerarchicamente tramite cicli di interazioni tra clienti e esecutori. Ciascun ciclo quindi descrive compiutamente il processo di interazione tra il cliente che richiede il servizio e l'esecutore dello stesso. Un ciclo è composto da quattro fasi, a ciascuna delle quali possono essere associati uno o più cicli per descrivere i dettagli dello svolgimento del processo. Ciascuna di queste fasi fa parte di un ciclo di negoziazione, che parte da una richiesta di un cliente e arriva all'accettazione del lavoro svolto dall'esecutore o fornitore del servizio. All'interno di ciascuna di queste fasi, ulteriori cicli di negoziazione possono essere iniziati e svolti, in alternativa o in sequenza tra loro, ottenendo così una scomposizione gerarchica delle attività in successivi livelli di dettaglio.

In tal caso si parla di annidamento gerarchico dei cicli e i cicli annidati vengono denominati workflow secondari.

Ciascun ciclo di descrizione di un processo nell'action workflow prevede la suddivisione del processo in quattro fasi distinte:

- **Preparazione/richiesta** - Cliente o esecutore propongono che l'esecutore esegua il lavoro
- **Negoziazione/accettazione** - Cliente e esecutore raggiungono un accordo
- **Esecuzione o produzione** - L'esecutore esegue il lavoro e lo completa
- **Soddisfazione/erogazione** - Il cliente valuta il lavoro ed il livello di soddisfazione

In ciascuna fase cliente e esecutore interagiscono secondo una serie di modalità di interazione prestabilite nel modello, che determinano l'evoluzione del processo. Si viene così a definire una rete di impegni tra cliente e esecutore, che porta all'esecuzione del processo. Le modalità di interazione previste dal modello sono indicate nella tabella seguente:

<b>Chi</b>	<b>Che cosa</b>
· Cliente	Richiesta
· Esecutore	Accettazione della richiesta
· Esecutore	Dichiarazione di completamento
· Cliente	Dichiarazione di soddisfazione
· Cliente	Dichiarazione di insoddisfazione
· Cliente	Cancellazione
· Esecutore	Rifiuto della richiesta
· Esecutore	Revoca
· Esecutore	Contro-offerta
· Cliente	Accettazione della contro-offerta
· Cliente	Contro-offerta alla contro-offerta
· Cliente	Insistenza

Un esempio generico di action workflow è presentato nella figura 11, in cui la fase di richiesta presenta un workflow secondario per gestire una richiesta di chiarimento, la fase di accettazione è dettagliata nelle fasi di negoziazione e autorizzazione, la fase di produzione comporta una delega nell'esecuzione di tutto o parte del processo e la fase di soddisfazione o erogazione comporta la necessità di una conferma da parte dell'operatore.

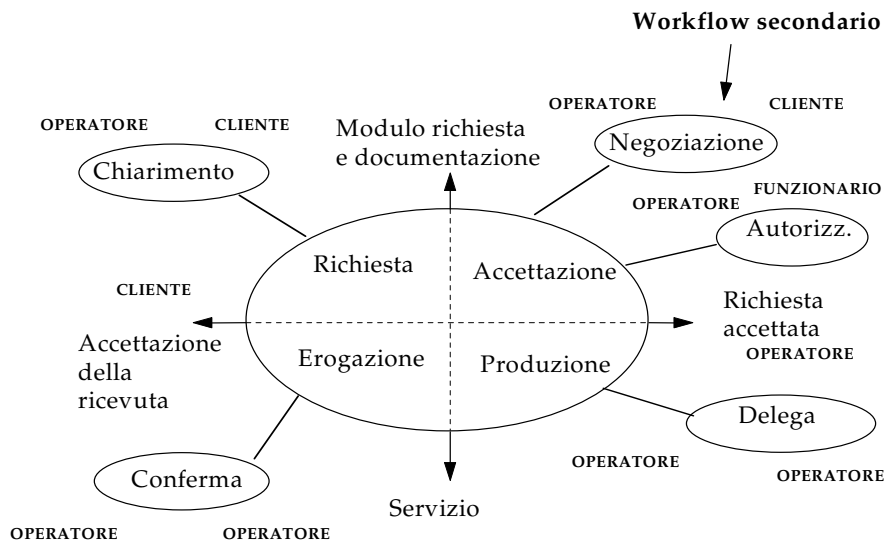


Figura 10 - Action workflow

### Esempio: concessione di crediti

Nel seguito verrà esaminato un caso di processo per la gestione della concessione dei crediti in una banca. Il cliente della banca presenta la sua richiesta di credito allo sportello, che però non la gestisce direttamente, ma passa la pratica al direttore dell'agenzia di credito che è responsabile di tutte le pratiche di questo tipo. Lo schema di base del processo di concessione di crediti è illustrato nella figura 12.

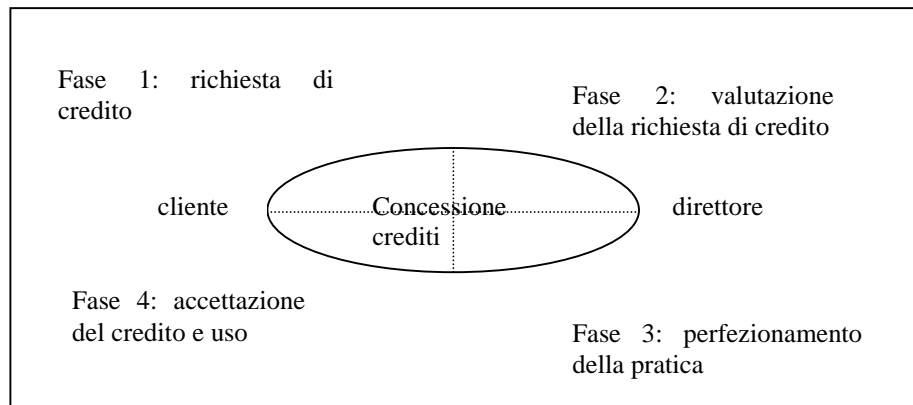


Figura 11 - Action workflow: concessione di crediti

Nella prima fase il cliente formula la sua richiesta al direttore utilizzando i moduli di richiesta predisposti a tale scopo. Il cliente solitamente riceve questi moduli allo sportello, dove può anche richiedere spiegazioni, e li presenta compilati al direttore dell'agenzia.

Nella seconda fase (vedere figura 12) il direttore valuta la richiesta del cliente. In questa fase vengono attivati alcuni processi secondari che coinvolgono altri impiegati nella banca per stabilire se il credito può essere concesso. Sulla base dei dati raccolti, il direttore calcola la situazione del cliente e il rischio relativo e decide se concedere il credito. A seconda dell'ammontare del credito possono essere necessarie ulteriori autorizzazioni all'interno della banca. La concessione del credito richiede altre elaborazioni per il perfezionamento della pratica. Questo viene effettuato nella fase 3, in particolare per crediti che richiedono garanzie. Il ciclo viene chiuso con la firma da parte del cliente di una lettera di accettazione del credito, che contiene tutte le condizioni relative alla concessione del credito.

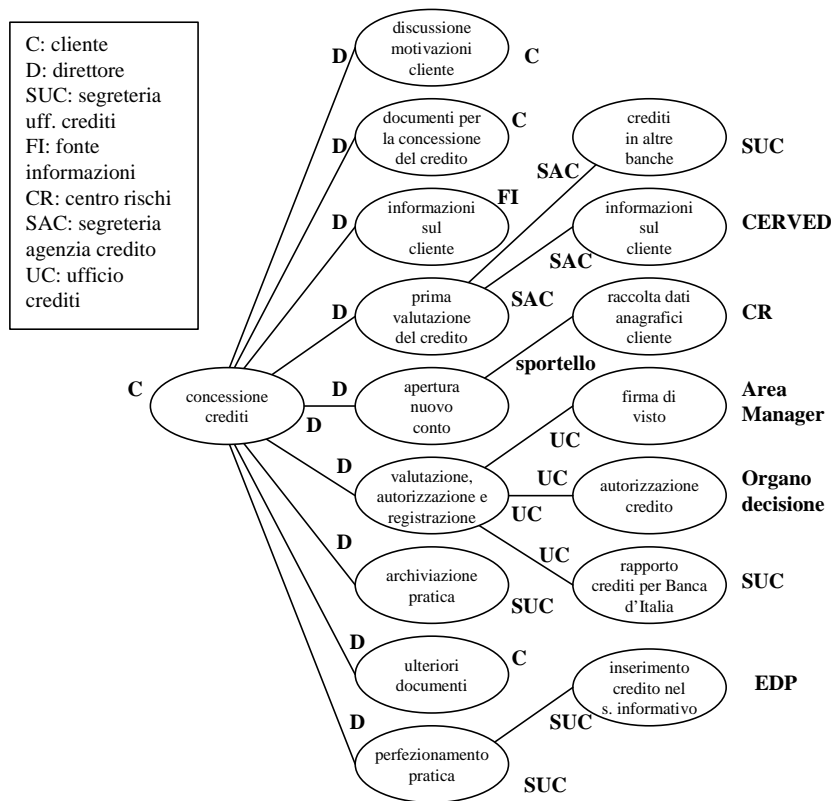


Figura 12 - Valutazione della richiesta del cliente

## 5. Diagramma delle attività UML

UML (Unified Modeling Language) [Boo99] è un linguaggio grafico che consente di specificare e documentare sistemi software, tramite un approccio orientato agli oggetti. UML nasce dall'integrazione di tre diverse notazioni proposte negli anni '80 e nei primi anni '90 da Booch, Jacobson e Rumbaugh, ed è stato sviluppato dai tre autori sopra citati in collaborazione e con il contributo di decine di ricercatori e analisti. UML è attualmente mantenuto e sviluppato dal consorzio OMG (Object Management Group), che ne ha fatto un suo standard.

UML comprende vari tipi di notazioni e di diagrammi, per specificare i diversi aspetti di un sistema software. Per la specifica di processi, UML propone l'utilizzo dei *diagrammi delle attività* (*activity diagrams*). I diagrammi delle attività sono essenzialmente dei grafi nei quali i nodi rappresentano le attività del processo e gli archi specificano l'ordine in cui i task devono essere eseguiti.

La Figura 14 mostra la specifica del processo *Prenotazione viaggi* tramite diagramma delle attività UML.

### 5.1. Attività e azioni

Le attività possono essere di due tipi: atomiche o non atomiche. Le attività atomiche, chiamate *azioni* in UML, descrivono operazioni elementari, concettualmente non decomponibili in azioni più semplici. Esse sono graficamente rappresentate da una losanga, ed hanno un'etichetta associata che indica l'azione da svolgere.

Le *attività* vere e proprie rappresentano invece azioni composte, costituite da altre attività o azioni. La loro struttura viene dettagliata separatamente, tramite un ulteriore diagramma delle attività, più di dettaglio. Le attività sono utili quando il processo è complesso, e non è possibile o non è conveniente rappresentarlo nel dettaglio in un unico diagramma (il concetto di attività in UML è analogo al concetto di supertask in WIDE). Le attività vengono graficamente specificate con lo stesso simbolo usato per la definizione delle azioni. Nel seguito useremo il termine attività per indicare sia attività che azioni.

Per specificare processi con strutture complesse, UML propone l'uso di costrutti che hanno funzioni simili a quelle dei connettori in WIDE.

### 5.2. Diramazioni condizionali

In un diagramma delle attività è possibile specificare diramazioni del processo che devono essere seguite solo se sono verificate determinate condizioni. L'esecuzione condizionale è specificata graficamente da un rombo che ha una transizione in ingresso e due o più transizioni in uscita (come la fork condizionale in WIDE); alle transizioni in uscita sono associate delle condizioni, che devono essere in mutua esclusione (questa limitazione non è invece presente in WIDE). Le condizioni associate alle transizioni

vengono valutate all'attivazione della transizione in ingresso, e viene quindi attivata la diramazione in uscita alla quale è associata una condizione vera, che deve esistere ed essere unica.

### 5.3. Fork e Join

L'esecuzione parallela di attività e la sincronizzazione fra attività, sono specificate tramite le *barre di sincronizzazione (synchronization bar)*. Le barre di sincronizzazione possono essere di due tipi: *fork* e *join*, e la loro semantica è del tutto analoga a quella della fork e del join totale in WIDE. La barra di tipo *fork* ha un arco in ingresso e due o più archi in uscita: non appena si attiva la transizione in ingresso, la fork attiva tutte le transizioni in uscita, in parallelo. La barra di tipo *join* ha due o più transizioni in ingresso e una in uscita: quando tutte le transizioni in ingresso sono state attivate, la barra di tipo *join* attiva la transizione in uscita.

### 5.4. Corsie

In un diagramma delle attività è possibile specificare, per ogni attività, qual è l'entità (ad esempio, il dipartimento o l'organizzazione) responsabile della sua esecuzione. Le responsabilità sono specificate partizionando le attività di un diagramma in gruppi, e associando un responsabile ad ogni partizione.

Graficamente, la partizione del grafo viene rappresentata tramite *corsie (swimlanes)*, generalmente delimitate da segmenti paralleli, anche se talvolta la complessità del grafo rende necessario l'utilizzo di linee curve. Le attività del processo *Prenotazione viaggi* di figura 14 sono partizionate in due gruppi: quelle eseguite dall'agenzia di viaggi e quelle invece di responsabilità dell'ufficio amministrazione.

### 5.5. Flusso dei dati

Un altro aspetto importante nella specifica del processo è la descrizione dei dati manipolati dalle attività. UML consente di specificare, all'interno di un diagramma delle attività, a quali oggetti le attività accedono in lettura e/o in scrittura. Inoltre è possibile specificare lo stato in cui si dovrebbe trovare un oggetto al termine di una attività. Graficamente, gli oggetti sono rappresentati da rettangoli, collegati alle attività da frecce tratteggiate. Gli accessi in lettura sono rappresentati da frecce orientate dall'oggetto all'attività, mentre gli accessi in scrittura sono rappresentati da frecce orientate dall'attività all'oggetto. Per semplificare la presentazione, in figura 14 abbiamo solamente rappresentato il flusso dei dati (oggetto *P* di tipo *Prenotazione*) relativi alla prenotazione dall'attività *Conferma viaggio* all'attività *Richiesta pagamento*.

Come si può notare, i diagrammi delle attività sono concettualmente simili al modello di processo di WIDE (e degli altri sistemi di workflow). La differenza fondamentale tra i due è che UML è un linguaggio che ha lo scopo di supportare l'analista nella modellazione di un processo e nell'interazione con il cliente, mentre WIDE è un linguaggio completo che consente di specificare compiutamente un workflow, che viene poi interpretato da un apposito componente (il motore del workflow).



I diagrammi delle attività non forniscono costrutti necessari per la specifica di una serie di aspetti di un processo che sono invece necessari per la sua implementazione, quali le caratteristiche transazionali, la specifica delle situazioni eccezionali e vincoli di priorità e temporali. I diagrammi delle attività sono infatti spesso utilizzati per la specifica ad alto livello di un processo, e più in generale per la specifica degli aspetti dinamici di un sistema software, anche grazie alla loro integrazione con gli altri diagrammi e formalismi UML. Nel complesso, UML fornisce un insieme di notazioni semi-formali che consentono l'analisi e la specifica dei diversi aspetti di un sistema software, e che sono sufficientemente semplici ed intuitive da semplificare l'interazione con il cliente durante la fase di analisi di un sistema. Tali specifiche dovranno poi essere completate con aspetti di dettaglio e tradotte in un linguaggio formale (ad esempio il linguaggio WIDE nel caso dei processi), adatto ad essere interpretato o compilato da un sistema informatico.

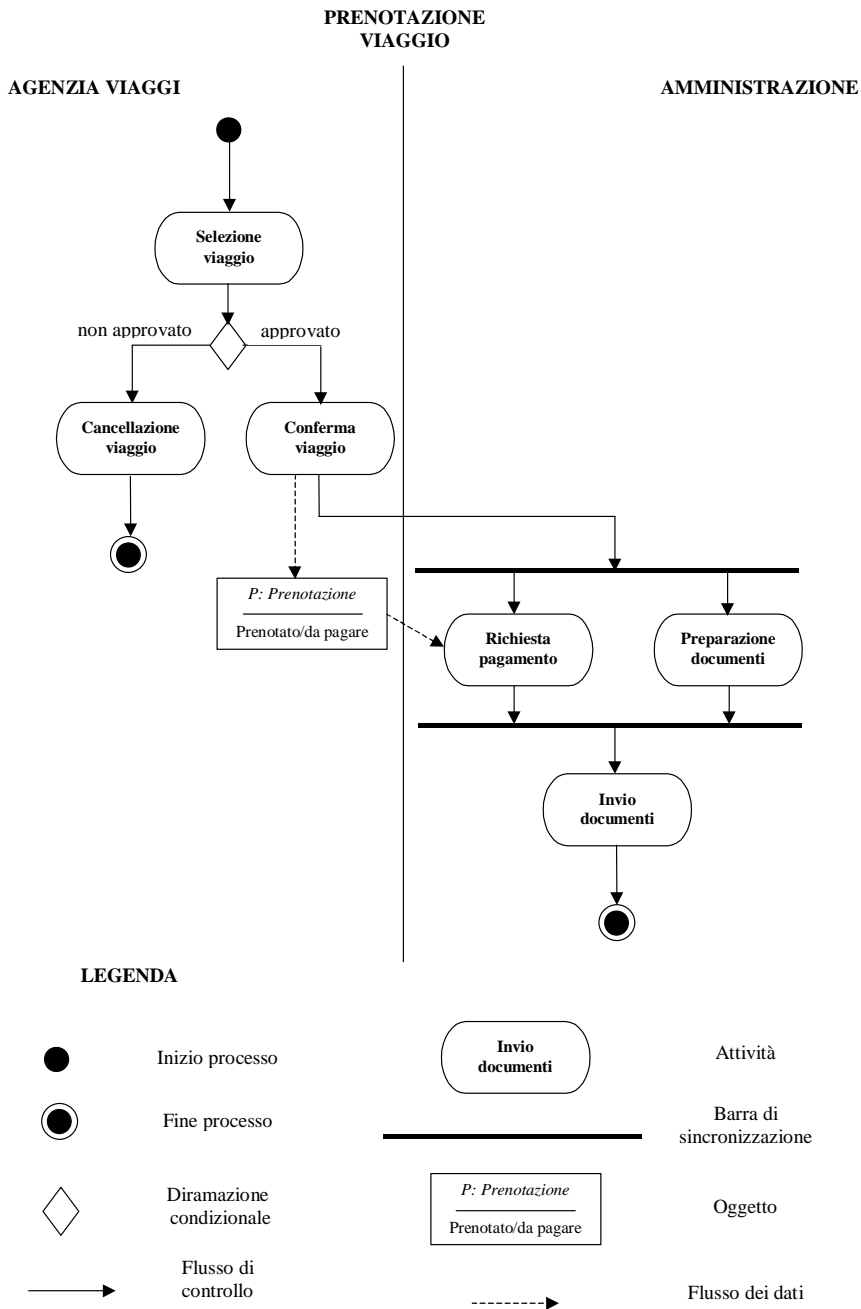


Figura 13 - Notazione degli activity diagrams e descrizione del processo Prenotazione Viaggi

## 6. Bibliografia

- [Bat92] C. Batini, S. Ceri, S. Navathe. Conceptual Database Design - An Entity-Relationship Approach. Benjamin Cummings, 1992
- [Boo99] G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language User Guide. Addison-Wesley, Object Technology Series, 1999.
- [Bra97] G. Bracchi, G. Motta. Processi aziendali e sistemi informativi. Franco Angeli, 1997
- [Geo95] D. Georgakopoulos, M. Hornick, A. Sheth: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases* 3(2): 119-153 (1995)
- [Gre99] P. Grefen, B. Pernici, G. Sanchez (a cura di). Database Support for Workflow Management : The Wide Project. Kluwer Academic Publishers, 1999.
- [Med92] Raul Medina-Mora, Terry Winograd, Rodrigo Flores, Fernando Flores, The Action workflow approach to workflow management technology, Proc. CSCW 1992, novembre 1992
- [Sch96] T. Schael, Workflow management systems for process organisations, LNCS 1096, Springer, 1996
- [You90] E. Yourdon, Analisi Strutturata dei Sistemi, Jackson-Prentice Hall Intl., 1990

### Capitolo 2 del Libro:

**P. Atzeni, C. Batini, F. Casati, B. Pernici, L. Saladini,  
"Modelli e Progettazione" Volume II della Collana "SISTEMI INFORMATIVI"  
(a cura di C. Batini, B. Pernici, G. Santucci), Franco Angeli, 2001.**