

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Modelli di sistema

- La modellazione dei sistemi aiuta gli analisti a comprendere le funzionalità del sistema
- I modelli servono per comunicare con i clienti
- I modelli sono astratti: alcune informazioni vengono deliberatamente trascurate
 - **Rappresentazione:** Mantiene le caratteristiche del sistema
 - **Astrazione:** Semplifica deliberatamente e trascura alcune caratteristiche

Crediti: E. Leonardi, Corso di Ingegneria del Software, Università di Ferrara, AA 2000/2001

Tipi di modelli

- E' possibile rappresentare un sistema tramite diversi tipi di modelli
 - *Data-processing model:* Diagramma data-flow che illustra come i dati vengono processati nei diversi stadi del sistema
 - *Composition model:* Diagramma entità-relazioni mostra come le entità del sistema sono composte di altre entità, e sono in relazione tra di loro
 - *Classification model:* Diagramma di classe/ereditarietà mostra le caratteristiche comuni tra diverse entità
 - *Stimulus-response model:* Modelli a transizioni di stati mostrano come il sistema reagisce ad eventi interni ed esterni
 - *Process model:* Mostrano le principali attività che devono essere compiute per eseguire i processi

Modello concettuale

- Il modello concettuale è centrato sul **dizionario dei dati**, un magazzino che contiene la descrizione di tutti gli oggetti-dato usati dal software
- Su questo dizionario sono basati 3 diagrammi:
 - il **diagramma entità-relazioni** (Entity-Relation Diagram o ERD)
 - riassume le relazioni tra i singoli oggetti-dato
 - il **diagramma di flusso dei dati** (Data Flow Diagram o DFD)
 - indica come i dati sono trasformati o si spostano nel sistema
 - rappresenta le funzioni che trasformano i dati
 - il **diagramma stati-transizioni** (State-Transition Diagram o STD)
 - illustra il comportamento del sistema in seguito a eventi esterni
 - rappresenta i modi operativi del sistema (stati) e le transizioni tra essi

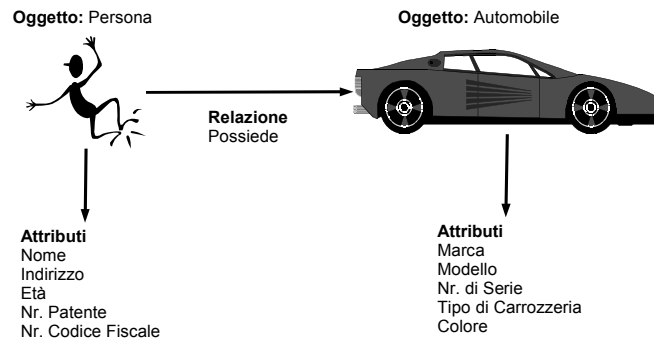
Modellazione dei dati

- La modellazione dei dati risponde alle seguenti domande
 - quali sono gli oggetti-dato che il sistema deve trattare?
 - come sono composti gli oggetti-dato e che attributi hanno?
 - quali sono le relazioni tra gli oggetti?
- L'ERD rappresenta gli oggetti-dato in modo grafico come un reticolo di relazioni

Gli oggetti-dato

- Un DO è la rappresentazione di un'informazione composta che il software dovrà trattare
 - N.B. una singola quantità non è considerata un DO
- Un DO può rappresentare praticamente qualsiasi entità
 - una cosa, un evento, un'unità organizzativa, un luogo, una struttura
- La descrizione di un DO include l'identificativo dell'oggetto e tutti i suoi attributi
- Tra i DO esistono *relazioni* che indicano un qualche legame tra essi

Esempio




Gli Attributi

- Gli attributi di un DO rientrano in 3 categorie
 - **attributi identificativi**: danno un nome ad un esemplare del DO
 - **attributi descrittivi**: descrivono l'esemplare
 - **attributi referenziali**: rimandano ad esemplari di altri DO
- Uno o più attributi identificativi svolgono il ruolo di identificatore del DO e fanno da chiave nella ricerca
 - P.es. per il DO automobile il Nr. di Serie è un buon identificatore
- Gli attributi utili a definire un DO variano a seconda del contesto in cui agisce il sistema analizzato
 - P.es. gli attributi di un'automobile che interessano un concessionario sono, almeno in parte, differenti da quelli che interessano un costruttore

Rappresentazione tabellare

- Poiché la descrizione di un DO include solo gli attributi e non fa riferimenti alle operazioni che agiscono su di esso, è possibile usare una rappresentazione tabellare del DO



Identificatore

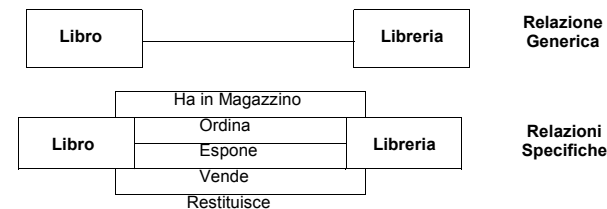
Marca	Modello	Nr.Serie	Carrozzeria	Colore	Proprietario
FIAT	Millecento	AB234	Utilitaria	Bianco	UF
Ferrari	Testarossa	XYZ22	Sport	Rosso	LCM
Ford	Taurus	QQ77	Station-Wagon	Verde	CC

Istanza

Attributi Denominativi
Attributi Descrittivi
Attributo Referenziale

Relazioni tra DO

- Tra due oggetti possono esistere delle relazioni
- Il tipo di relazione dipende dal ruolo dei due DO nel sistema software che si sta realizzando



- Ogni relazione implica un rapporto bidirezionale
 - “la libreria ordina il libro” e “il libro è ordinato dalla libreria”

Cardinalità delle relazioni

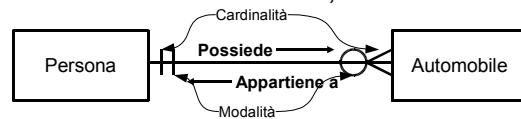
- Oltre a specificare il tipo, una relazione deve anche specificare il numero di DO messi in relazione
 - Es. una madre ha N figli ($N \geq 0$) ma un figlio ha una sola madre
- La cardinalità di una relazione specifica quanti DO di ciascun tipo sono collegati da quella relazione
- La cardinalità può essere
 - Uno-a-uno** (1:1) un DO di tipo A è collegato ad un solo DO di tipo B e viceversa (es. marito/moglie)
 - Uno-a-molti** (1:N) un DO di tipo A può essere collegato a N DO di tipo B ma ogni DO di tipo B può essere collegato a un solo DO di tipo A (es. madre/figli)
 - Molti-a-molti** (M:N) un DO di tipo A può essere collegato a molti DO di tipo B e viceversa (es. zii/nipoti)

Cardinalità/Modalità delle relazioni

- La cardinalità definisce il numero massimo di oggetti collegati dalla relazione ma non se la relazione deve obbligatoriamente esistere
- La modalità di una relazione indica se una relazione è facoltativa o meno, ossia se il minimo numero di DO di un dato tipo che partecipa alla relazione può essere o meno zero
 - La modalità si indica con 1 (relazione obbligatoria) o con 0 (relazione facoltativa)
- Esempio: un'auto ha sempre una sola persona che la possiede mentre una persona può avere 0 o più auto. La relazione ha cardinalità 1:N e modalità 1:0

Simbologia delle relazioni

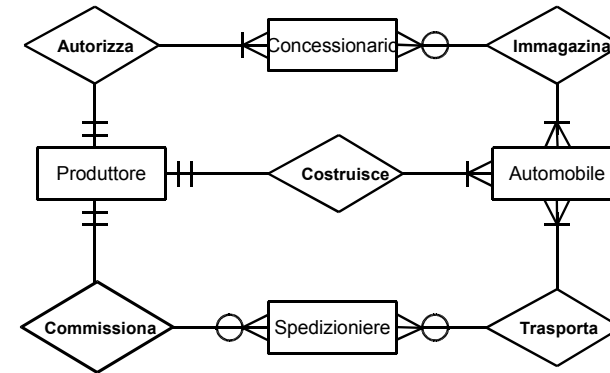
- Cardinalità: 1 si indica con + , molti si indica con <
- Modalità: 1 si indica con + , 0 con ○



- Una automobile appartiene ad una ed una sola persona mentre una persona può possedere zero o più automobili
- Una persona può o meno possedere automobili, ma una automobile ha sicuramente un proprietario
- In una variante della notazione il nome della relazione (possiede) è incluso in un rombo

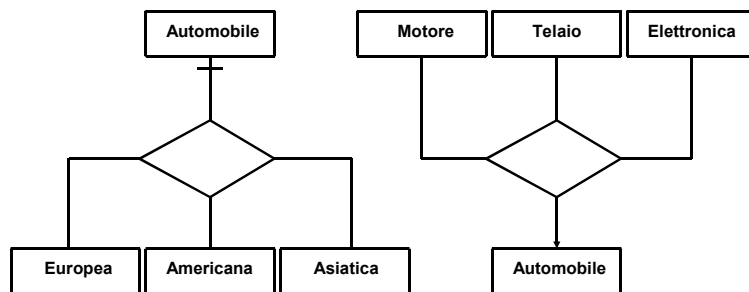


Esempio: ERD



Relazioni gerarchiche

- Se un DO può essere di vari tipi o se più DO concorrono a formare un DO complesso si possono inserire nell'ERD delle relazioni gerarchiche

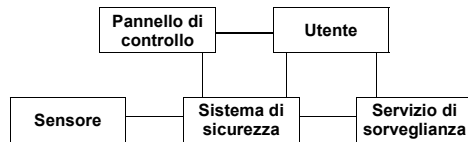


Creare un ERD

- Il primo passo consiste nell'esame dei requisiti per identificare gli oggetti entranti e uscenti nel sistema e gli eventuali produttori/consumatori di dati esterni
- Per ogni oggetto vanno identificate le relazioni, inclusive di cardinalità e modalità, che lo legano ad altri oggetti
- Si definiscono gli attributi degli oggetti identificati
- Si definisce il diagramma entità-relazioni
- Si individuano le possibili lacune del diagramma identificando nuovi oggetti e si ripete il processo fino al suo completamento

Esempio: creazione di un ERD / 1

- Nel sistema di controllo per la casa si identificano:
 - utente (padrone di casa), pannello di controllo, sensori, sistema di sicurezza, servizio di sorveglianza
- ...con le seguenti relazioni:

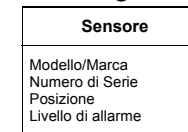


Esempio: creazione di un ERD / 2

- Le relazioni vanno poi specificate includendo cardinalità e modalità



- Infine vanno identificati gli attributi di ciascun oggetto

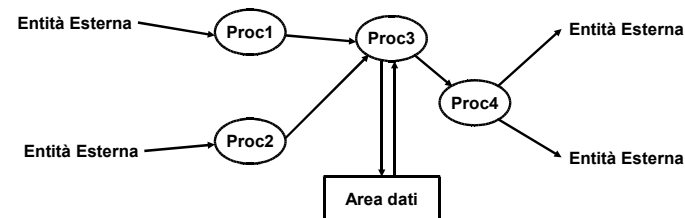


Modelli data-flow

- Mostrano come i dati vengono trasformati e fluiscono attraverso il sistema
- Questi modelli vengono sviluppati come parte intrinseca di molti metodi di analisi
- Sono basati su una notazione semplice e intuitiva che tutti possono comprendere
- Indicano complessivamente come i dati in ingresso vengono trasformati nei risultati della computazione

Diagrammi data-flow (DFD)

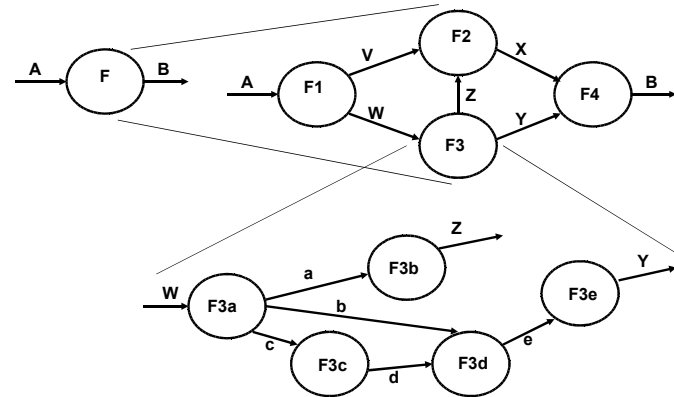
- Nel diagramma di flusso vengono rappresentate le unità funzionali del sistema, ciascuna collegata ai generatori/ricevitori di dati da essa trasformati
- Nel diagramma vanno anche indicati i generatori/ricevitori di dati esterni al sistema



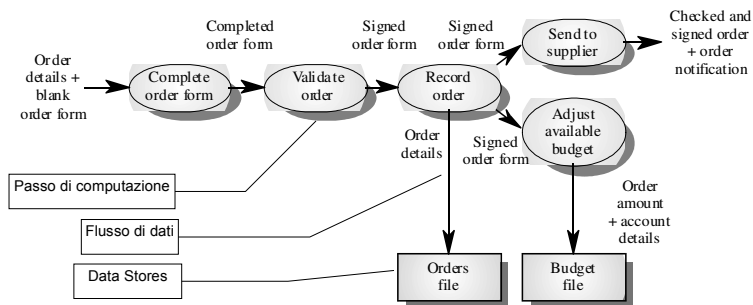
DFD / 1

- Semplici e intuitivi da comprendere
- Il DFD può essere rappresentato a più livelli di dettaglio, a partire dal modello contestuale in cui l'intero sistema è rappresentato come una singola unità funzionale
 - Nel diagramma *non* compaiono indicazioni esplicite sulla sequenza di elaborazione o della logica con cui i dati fluiscono nel sistema: questi dettagli vengono rimandati alla fase di progettazione del software
- Possono essere utilizzati per illustrare come i dati vengono scambiati tra i diversi sottosistemi che compongono il sistema
- Non sono adeguati per descrivere le interfacce del sistema

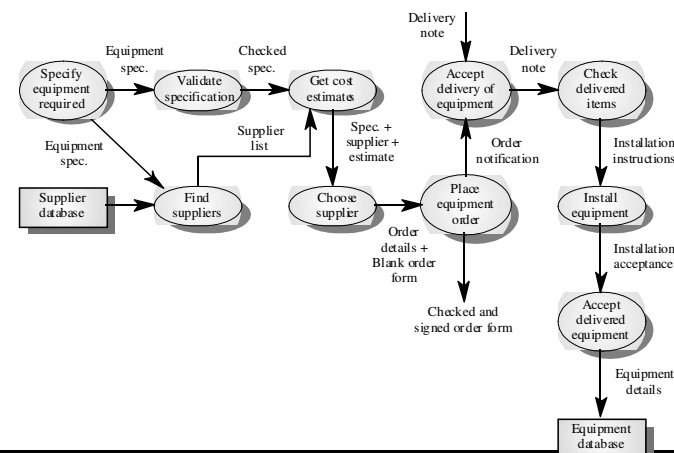
DFD / 2



Esempio: Gestione degli ordini



Esempio: DFD per l'acquisto di una apparecchiatura



DFD / 3

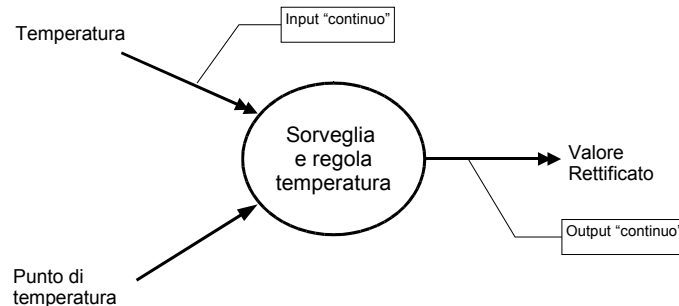
- Per descrivere i dati che fluiscono lungo le varie frecce del DFD si ricorre al Dizionario dei Dati (vedremo dopo)
- Il DFD deve essere accompagnato dalla specifica di processo: una descrizione delle singole entità funzionali in termini di input, output ed algoritmo applicato.
- La specifica di processo deve anche riportare le restrizioni e i vincoli che la funzione deve rispettare e i vincoli progettuali che la possono condizionare

Estensioni per sistemi real-time / 1

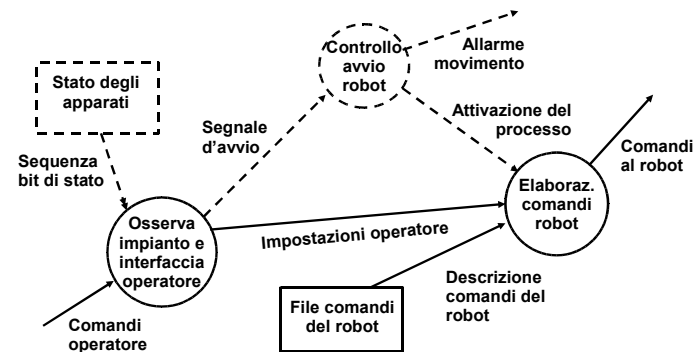
(Estensioni di Ward e Mellor)

- Le applicazioni real-time introducono nel flusso di dati la necessità di un controllo temporale
- Ward e Mellor hanno esteso il DFD per
 - gestire flussi continui di dati
 - gestire il controllo del sistema e l'elaborazione associata
 - consentire esemplari multipli della stessa funzione (multi-tasking)
- La freccia a doppia punta \longleftrightarrow rappresenta un flusso continuo di dati
- Gli eventi di controllo e il loro flusso è rappresentato con frecce e bolle tratteggiate \dashrightarrow
- Il multi-tasking si rappresenta con più bolle sovrapposte

Esempio



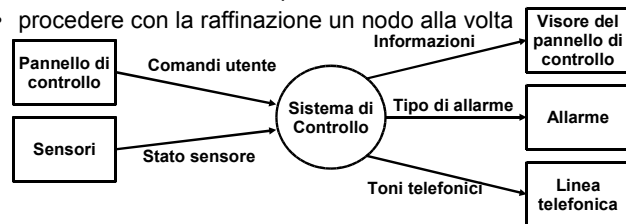
Estensioni per sistemi real-time / 2



Creazione di un DFD

- Il modello parte dalla rappresentazione del sistema come un nodo singolo (livello 0) a cui fanno capo tutti gli I/O esterni per poi scendere nei dettagli facendo attenzione a

- mantenere la continuità del flusso informativo tra i vari livelli
- utilizzare nomi descrittivi per nodi e flussi
- procedere con la raffinazione un nodo alla volta



Creazione di un DFD / 2

- Il passaggio dal livello 0 al livello 1 si può effettuare tramite “analisi grammaticale” dei requisiti:
 - dal testo della specifica si evidenziano nomi e verbi
 - i nomi corrispondono a entità esterne (riquadri) oggetti-dato e oggetti-controllo (freccie) o depositi di dati (doppie freccie)
 - i verbi corrispondono a funzionalità o processi (nodi)
- Una volta identificati i processi a un dato livello, una descrizione più dettagliata di ciascun processo può fare da base per il successivo livello di DFD
- Il processo si ferma quando ogni nodo rappresenta una funzionalità elementare

Esempio

Il software di SafeHome permette all'utente di configurare il sistema di sicurezza al momento in cui viene installato, sorveglia i sensori collegati e interagisce con l'utente mediante una tastiera e alcuni tasti funzione contenuti nel pannello di controllo.

Durante l'installazione si utilizza il pannello di controllo per “programmare” e configurare il sistema. A ogni sensore si assegnano un numero e un tipo; si registra una password per attivare e disattivare il sistema; si registrano i numeri di telefono da comporre in caso di segnale da un sensore.

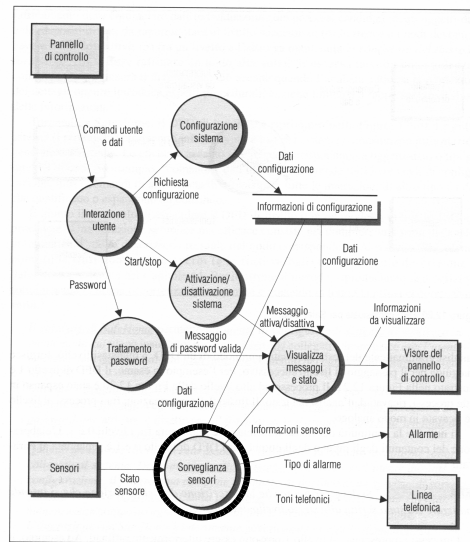
Quando un sensore rileva l'evento associato, il software fa scattare un allarme sonoro collegato al sistema. Dopo un intervallo di tempo specificato dall'utente durante la configurazione, il programma compone il numero di telefono di un servizio di sorveglianza, fornisce informazioni sulla posizione e sulla natura dell'evento rilevato. Il numero viene composto ogni venti secondi fino a ottenere la comunicazione.

L'interazione con SafeHome è interamente governata da un sottosistema apposito, che legge i dati inseriti mediante la tastiera e i tasti funzione, presenta messaggi di richiesta su un visore. L'interazione mediante tastiera prende la forma seguente...

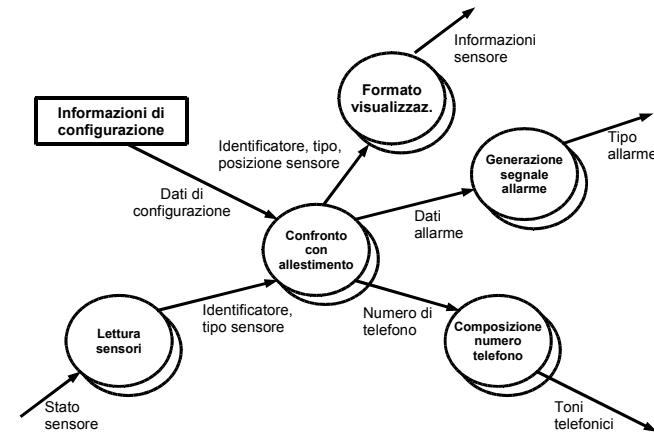
Risultato analisi grammaticale

- I verbi corrispondono ai processi, cioè ai nodi del DFD
- I nomi possono essere
 - Entità esterne (rettangoli)
 - Oggetti-dato e Oggetti-controllo (freccie)
 - Flussi continui di dati (freccie doppie)
- L'analisi grammaticale fornisce una “guida” intuitiva per passare a raffinare la descrizione del DFD di livello 0

DFD di livello 1



DFD di livello 2 del processo "sorveglianza sensori"



Modellare il comportamento

- Il comportamento di un sistema può essere rappresentato attraverso l'insieme degli stati (modalità operative) in cui il sistema può trovarsi e delle transizioni che possono avvenire tra questi stati in seguito a determinati eventi
 - Esempio: una lampadina si trova nello stato "spenta" finché l'evento "uso dell'interruttore" la fa transire allo stato "accesa"
- Il diagramma stati-transizioni (STD) rappresenta il modello di comportamento con una serie di stati (riquadri) collegati tra loro da frecce di transizione etichettate con l'evento che provoca la transizione

Diagramma stati-transizioni

- Uno stato è un comportamento osservabile
 - Rappresentato da rettangoli
- Una transizione è composta da due righe
 - La riga superiore indica il nome dell'evento (o degli eventi) che causano la transizione
 - La riga inferiore indica quale azione viene eseguita in seguito all'evento
 - La transizione conduce ad un nuovo stato

Diagramma stati-transizioni software di una fotocopiatrice

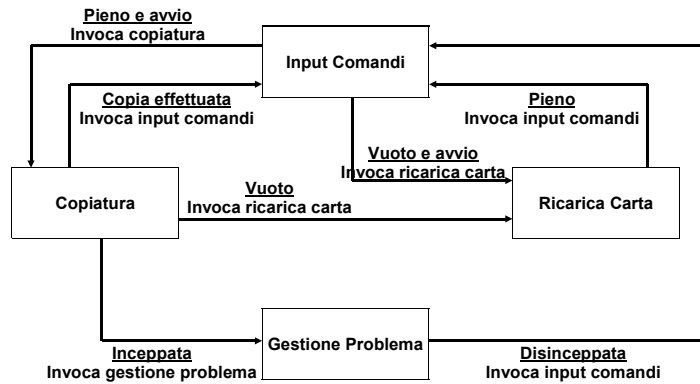


Diagramma stati-transizioni software di un forno a microonde

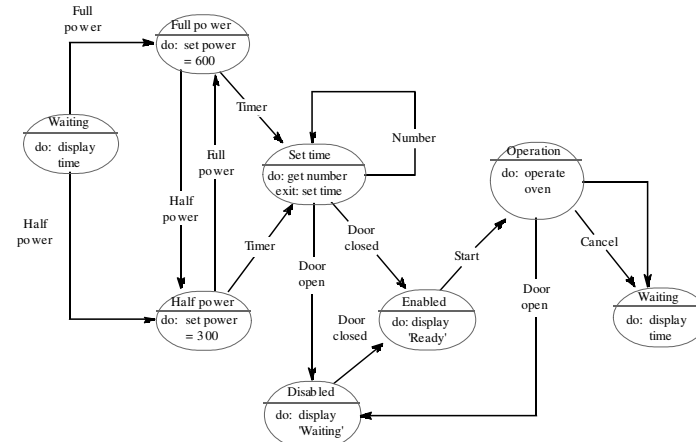


Diagramma stati-transizioni software di un forno a microonde, descrizione stati

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for 5 seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

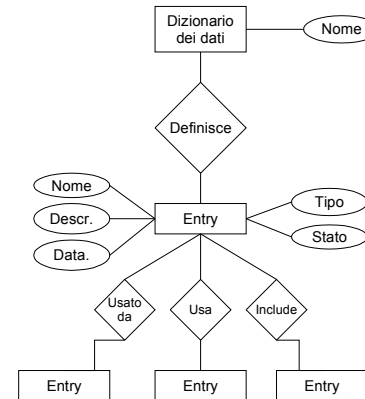
Dizionario dei dati

- Il dizionario dei dati è una lista di nomi con associate delle descrizioni di entità utilizzate nel sistema
- Rappresenta un repository condiviso di informazioni sul sistema
- Serve come
 - Meccanismo per gestione dei nomi. Dato che il modello di sistema può essere sviluppato da persone diverse, c'è il rischio di conflitti dei nomi (usare due volte lo stesso nome con significati diversi)
 - Collegamento tra l'analisi e l'implementazione

Dizionario dei dati: definizione

- *Il dizionario dei dati è un elenco strutturato degli elementi-dato pertinenti al sistema, con definizioni precise e rigorose che diano all'utente e all'analista una vista comune degli input, degli output, dei dati memorizzati e anche dei calcoli intermedi*

Modello ERD del tipico dizionario dei dati



Entries del dizionario dei dati

- E' necessario includere nel dizionario tutti i nomi utilizzati nel modello del sistema, e anche nelle successive fasi di design e implementazione
- Deve esistere software di supporto per creare, mantenere e interrogare il dizionario
- Il dizionario dei dati può essere integrato in uno strumento CASE (*Computer Aided Software Engineering*, ingegneria del software assistita da calcolatore) in grado di automatizzarne in parte la costruzione e il mantenimento

Elementi comuni ai dizionari dei dati

- **Nome**
 - Il nome del dato, dell'elemento del controllo, dell'archivio o dell'entità esterna
- **Alias**
 - Un nome alternativo
- **Dove e come**
 - Elenco dei processi che utilizzano l'elemento e il modo di impiego (ad esempio come input, output, memoria o entità astratta)
- **Contenuto**
 - Una rappresentazione del contenuto
- **Altre informazioni**
 - Informazioni sui tipi di dati, sui valori prefissati, eventuali vincoli o limitazioni e così via

Esempio di entries

Nome numero di telefono
Alias nessuno
Dove e come confronta con allestimento (output)
"componi numero di telefono" (input)
Descrizione numero di telefono = [interno | esterno] Alternativa
interno = prefisso + numero di accesso Sequenza
esterno = 1 + codice area + numero locale
codice area = [800 | 888 | 561]
Commento prefisso = *un numero di tre cifre che non inizi con 0 o 1*
numero di accesso = *una stringa di quattro numero qualsiasi*

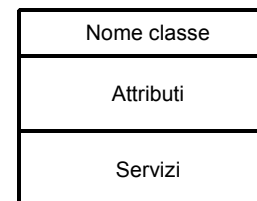
Modelli ad oggetti

- I modelli ad oggetti descrivono il sistema in termini di classi di oggetti
- Una classe è una astrazione di un insieme di oggetti con attributi e servizi (operazioni) comuni, forniti da ciascun oggetto
- Possono essere definiti diversi tipi di modelli ad oggetti
 - Modelli di ereditarietà
 - Mostrano chi deriva da cosa
 - Modelli di aggregazione
 - Mostrano la struttura interna di composizione degli oggetti
 - Modelli di servizio
 - Mostrano quale oggetto usa i servizi di altri oggetti

Modelli ad oggetti

- Rappresentano una notazione molto "naturale" per descrivere entità concrete del mondo reale
- Entità più di alto livello possono essere difficili da descrivere con questo approccio
 - Un "word processor", un "sistema di elaborazione dati medici", un "sistema libreria"...
- Identificare le classi degli oggetti è talvolta difficoltoso, e richiede una buona comprensione del dominio applicativo
- Classi che rappresentano entità del dominio possono essere solitamente riutilizzate per sistemi diversi riferiti allo stesso dominio

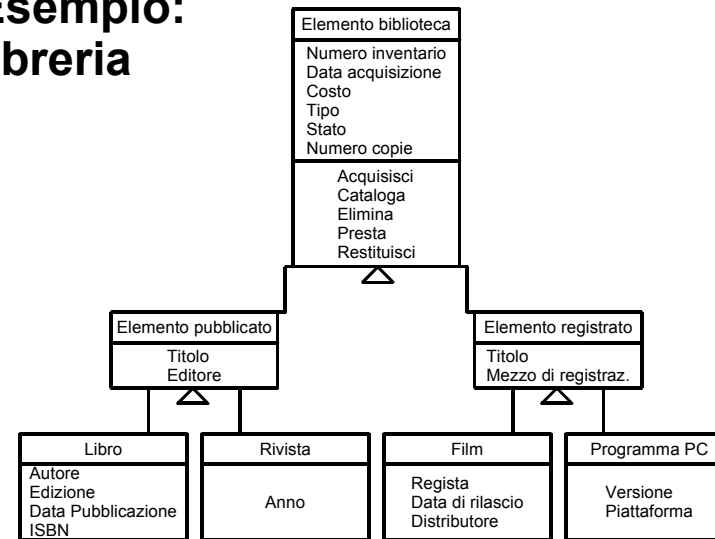
Notazione di classe



Modelli di ereditarietà

- Organizzano le classi di oggetti del dominio in una gerarchia
- Le classi in cima (alla radice) della gerarchia contengono le caratteristiche comuni a tutte le classi da esse derivate
- Le classi ai livelli intermedi della gerarchia ereditano i loro attributi e servizi da una o più super-classi.
 - Attributi e operazioni possono eventualmente essere specializzati (ridefiniti) nelle classi derivate
- Definire una gerarchia di classi è difficile se si desiderano evitare duplicazioni di attributi/operazioni in rami diversi della gerarchia

Esempio: libreria



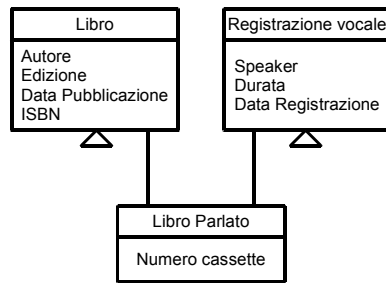
Esempio: utenti libreria



Ereditarietà multipla

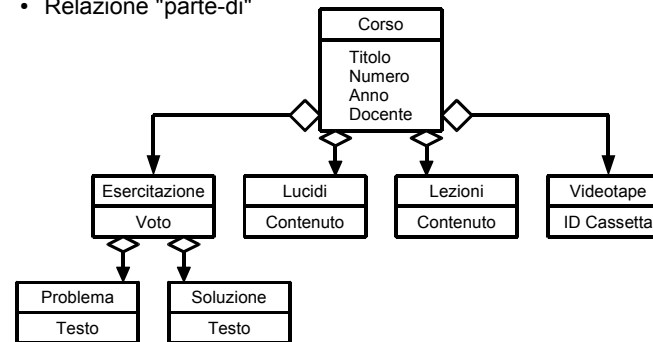
- Una classe può ereditare i propri attributi e operazioni da molteplici super-classi, piuttosto che da una singola
- L'ereditarietà multipla può produrre conflitti semantici nel caso in cui attributi e/o servizi con lo stesso nome appaiano in diverse super-classi con significato diverso
- L'ereditarietà multipla può rendere la riorganizzazione della gerarchia più difficile che nel caso in cui si faccia uso solo di ereditarietà singola

Ereditarietà multipla



Modelli di aggregazione

- I modelli di aggregazione mostrano come le classi sono composte da collezioni di oggetti di altre classi
- Relazione "parte-di"



Modelli di utilizzazione dei servizi

- Questi modelli mostrano in che modo i servizi forniti da un oggetto sono usati da altri
- Questi modelli devono essere usati con parsimonia, poiché alcuni oggetti forniscono servizi comuni che sono usati da moltissimi altri oggetti del sistema
 - Per mantenere i diagrammi leggibili, è opportuno evitare di indicare l'uso di questi servizi comuni

Uso dei servizi

