



UNIVERSITÀ  
DEGLI STUDI  
DI TERAMO



# User Experience (UX)

Prof.ssa Romina Eramo

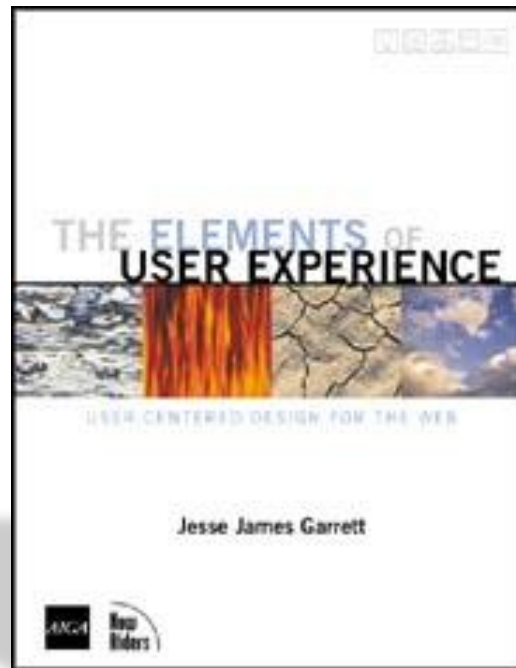
Università degli Studi di Teramo

Dipartimento di Scienze della Comunicazione

[rerao@unite.it](mailto:rerao@unite.it)

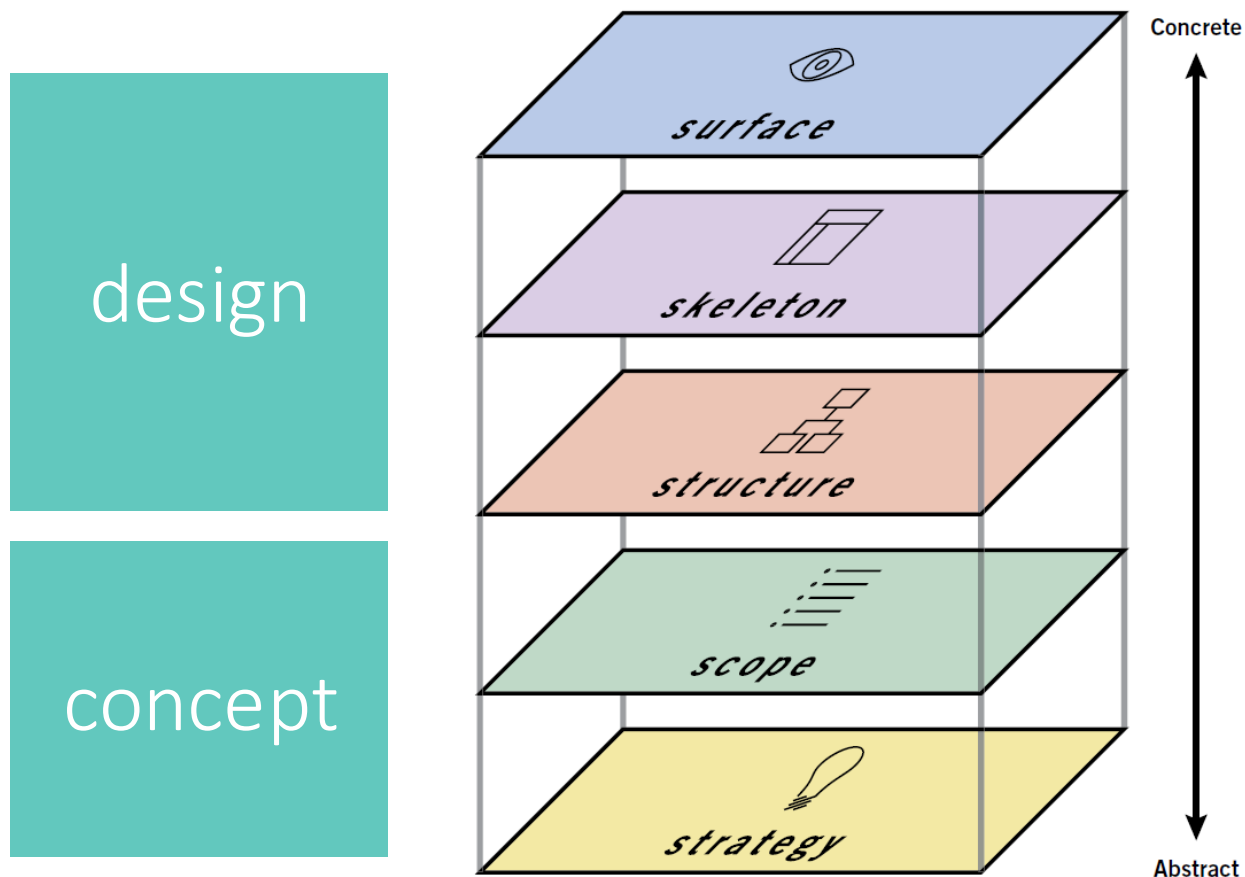
# Riferimenti

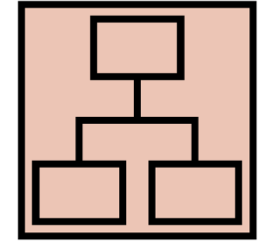
---



<http://jjg.net/ia/visvocab>

# User Experience 5 piani (livelli)





# Structure

---

- » Progettazione strutturale dello spazio informativo
  - da decisioni **astratte** → a decisioni più **concrete**,
    - » che influenzano direttamente ciò che l'utente **sperimenta**
- » Come l'utente si muove e dà un senso alle attività e alle informazioni
  - **Interaction Design** → lato *funzionalità*
    - » Navigazione tra le viste tramite un **MODELLO DI NAVIGAZIONE**
  - **Information Architecture** → lato *contenuti*
    - » Visualizza definizione e **NOMENCLATURA DEI CONTENUTI**

# Interaction Design: l'idea di base

---

» Si occupa di:

- descrivere i **comportamenti possibili** dell'utente,
- definire come il **sistema risponde** a questi comportamenti.

» Metafora della **danza**:

- l'utente agisce → il sistema reagisce → l'utente reagisce alla risposta, e così via.

» Obiettivo:

- progettare software che funzioni **bene per le persone**, non solo per la macchina.

# Conceptual Models (o Domain Models)

---

- » Un **modello concettuale** è il modo in cui l'utente “si immagina” come funziona un componente:
  - è un **oggetto** che possiedo?
  - un **luogo** che visito?
  - qualcosa che **consumo**?
- » Esempio:
  - **Carrello** e-commerce → modello di *contenitore*:  
“metto dentro”, “tolgo”, “vado alla cassa”.
- » **Modello concettuale del dominio** che incorpora sia il comportamento che i dati

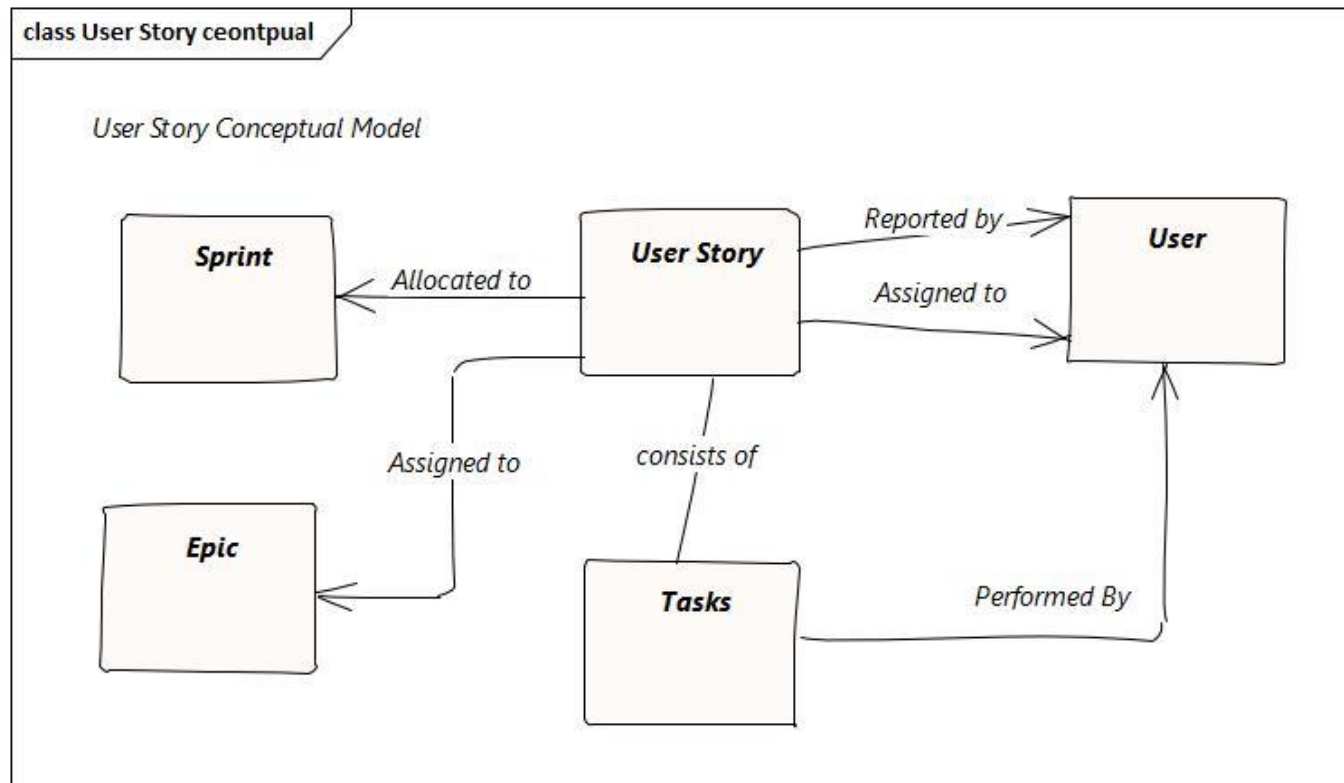
# Conceptual Models (o Domain Models)

---

- » Sono diagrammi che descrivono idee complesse e sono suddivisi in sostantivi.
  - I sostantivi possono essere familiari o sconosciuti.
  - I sostantivi sono collegati da verbi
- » Elementi:
  - Concetti nominali
    - » Questi sono rappresentati come classi/elementi nel diagramma e sono i sostantivi del dominio
  - Concetti verbali
    - » Questi sono rappresentati dalle linee di relazione tra i concetti nominali
    - » Le linee delle relazioni non sono esclusivamente verbi, ma possono anche rappresentare
      - Categorie, Classificazioni o Ruoli

# Esempi di Domain Models

Una User Story viene segnalata da un utente. Ma anche assegnato a un Utente. Una user story consiste in compiti svolti dagli utenti. Le User Stories sono assegnate agli Epici. Una storia utente viene assegnata a uno Sprint.



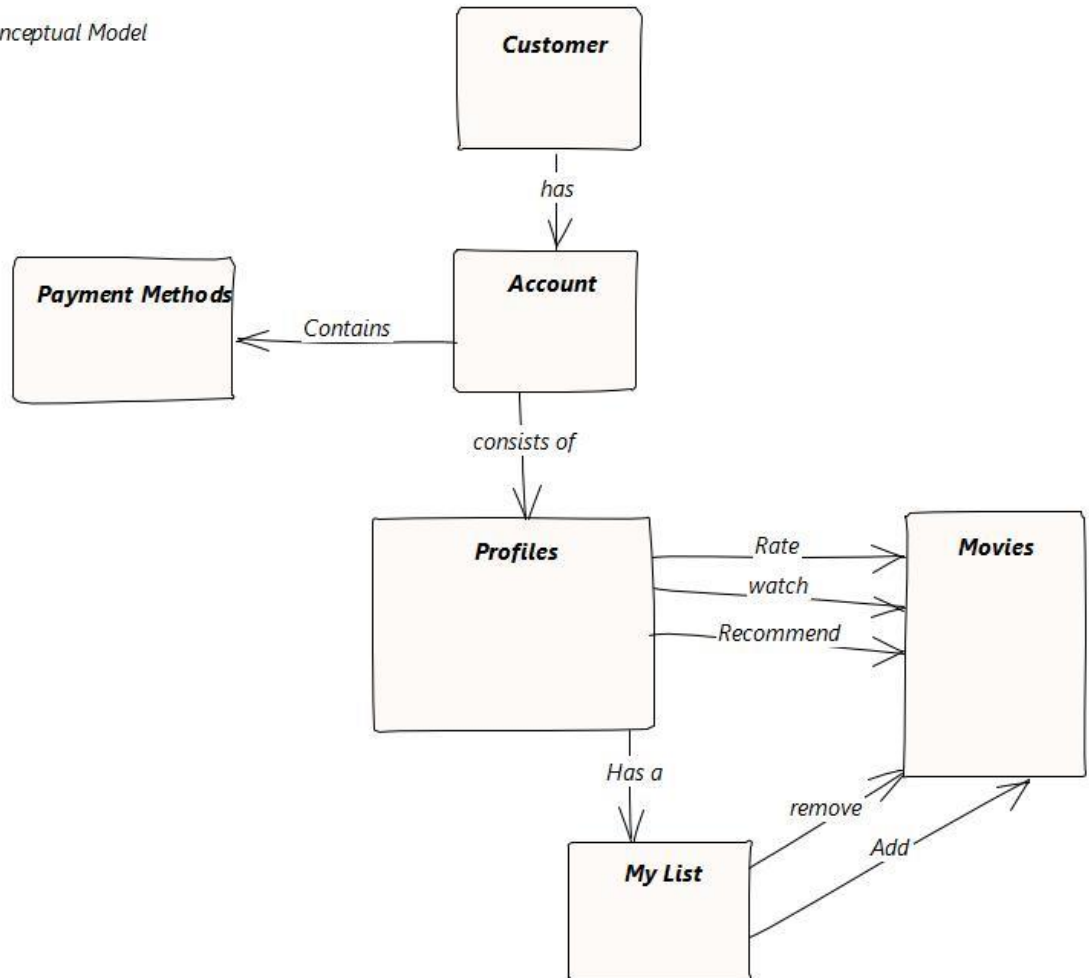


# Esempi di Domain Models

A netflix customer has an Account. This account contains a Payment method. An Account consists of Profiles. With a profile you can watch/rate and be recommended movies. A Profile has a list. And a My List can either add or remove movies.

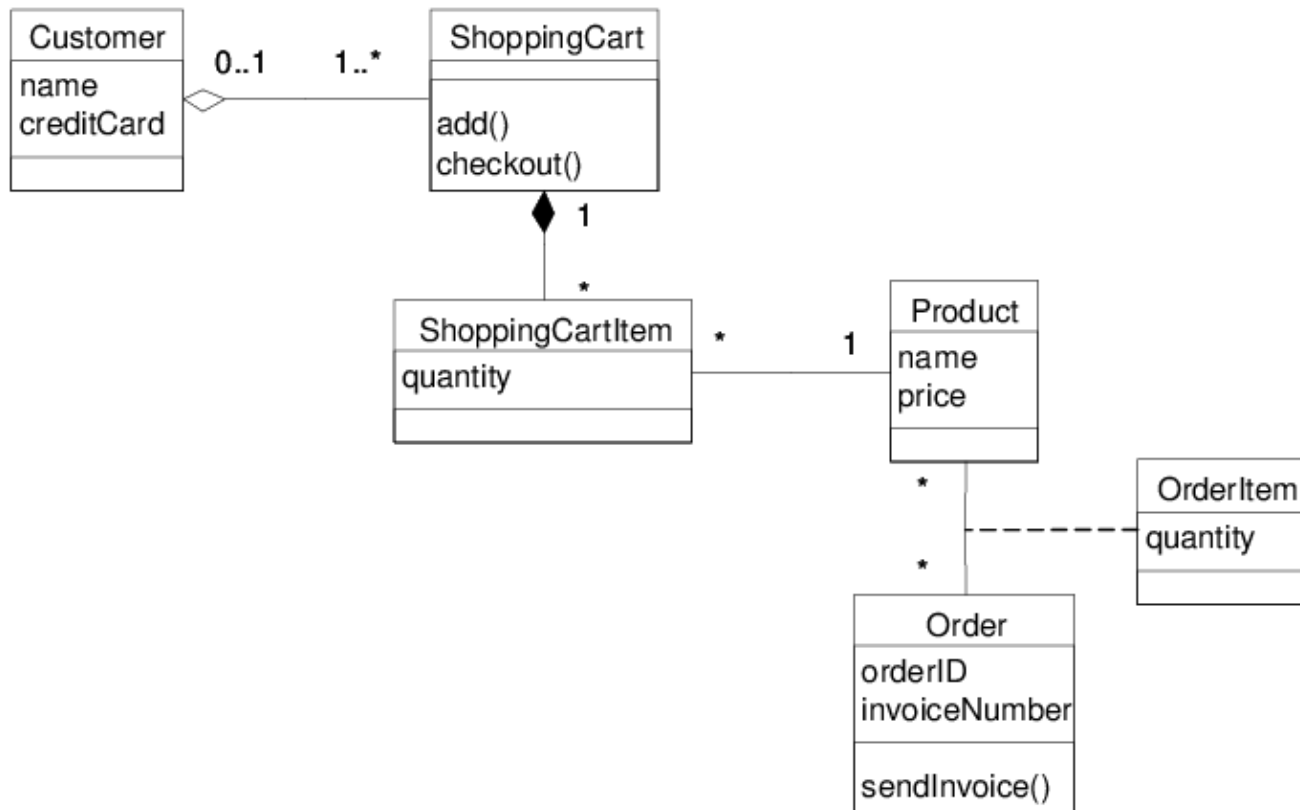
class Netflix conceptual Model

NetFlix Conceptual Model



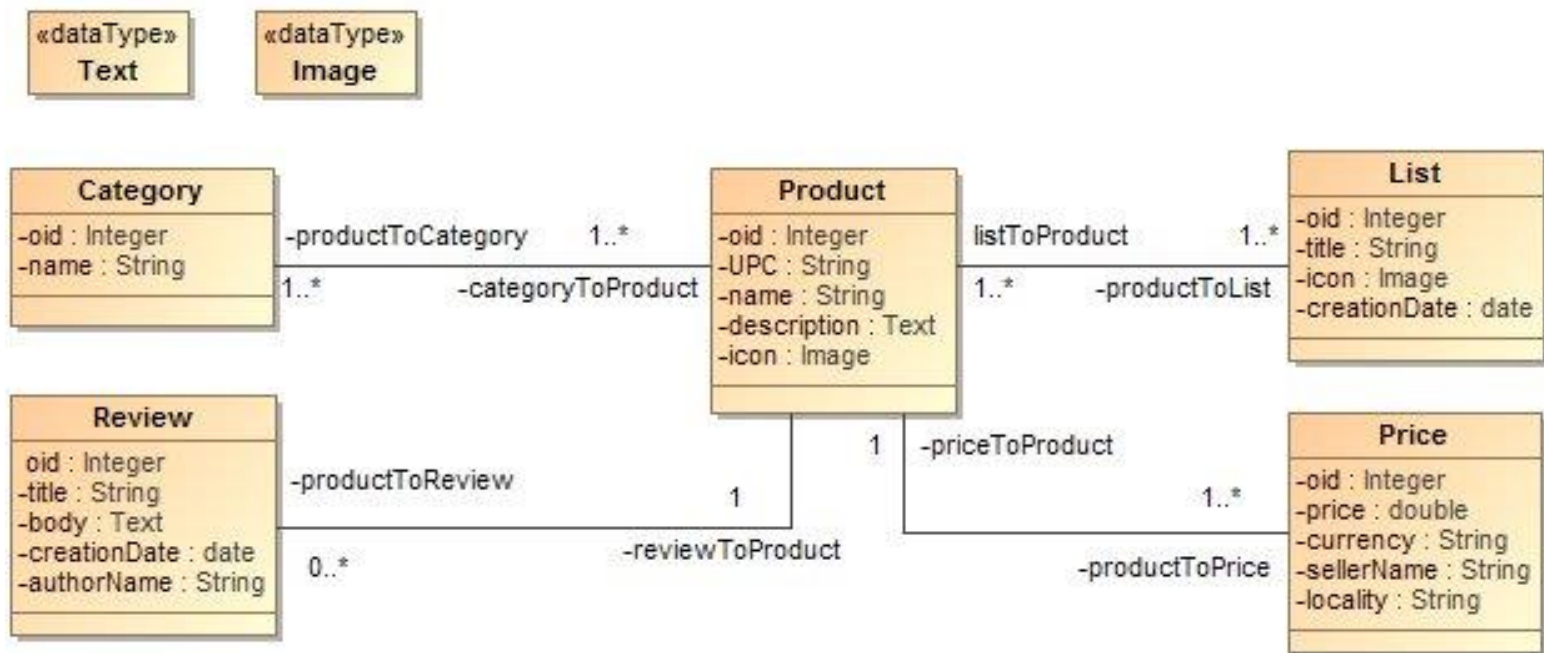
# Esempi di Domain Models

[UML Class Diagram]



# Esempi di Domain Models

[UML Class Diagram]



# Error Handling (gestione degli errori)

---

Tre livelli principali:

## » Prevenzione

- Progettare il sistema in modo che l'errore sia **impossibile** o molto difficile.
- Es.: bloccare azioni pericolose se mancano dati obbligatori.

## » Correzione / Aiuto

- Messaggi di errore chiari, suggerimenti, correzioni automatiche **non invadenti**.

## » Recupero

- Consentire di tornare indietro: **undo**, annulla, versioni precedenti.
- Usare con moderazione i pop-up “Sei sicuro?”, solo quando conta davvero.

# Information architecture: di cosa si occupa

---

» L'Information Architecture (IA) riguarda:

- come **organizziamo, raggruppiamo e presentiamo** le informazioni,
- per permettere agli utenti di **capire, trovare e usare** i contenuti.

» È cruciale:

- nei siti orientati ai contenuti (siti informativi, corporate, news),
- ma utile anche in prodotti più funzionali (app, sistemi complessi).

# Strutturare i contenuti

---

## Top-down approach

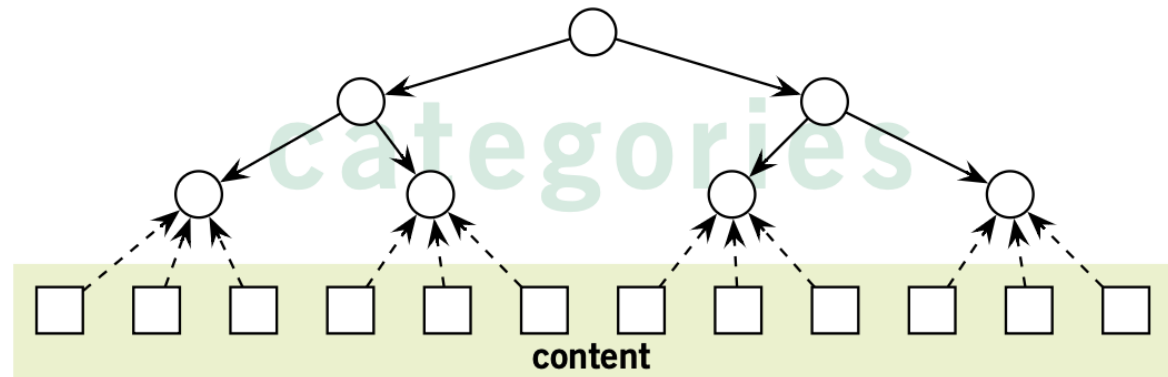
- » Parte da:
  - obiettivi di prodotto + bisogni utenti (Strategia),
  - definisce grandi **categorie**, poi sottocategorie.
- » Crea una “scatola vuota” dentro cui inserire contenuti e funzioni.

## Bottom-up approach

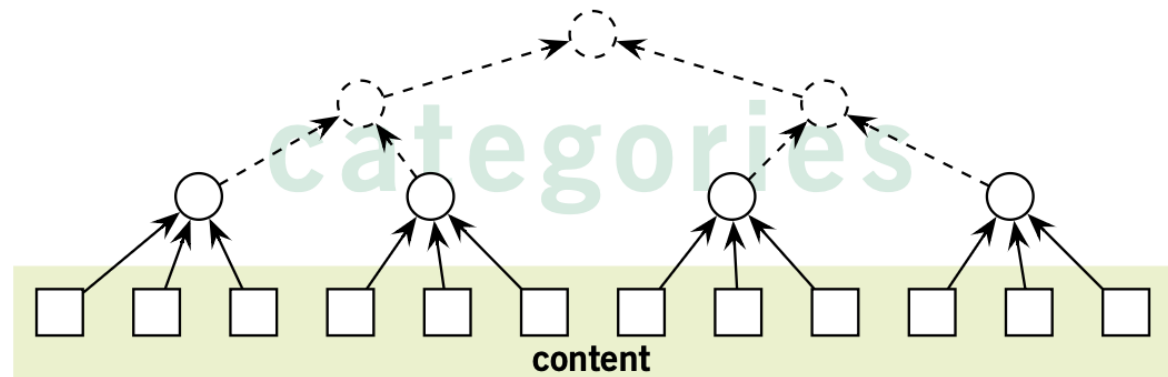
- » Parte dai **contenuti reali** (o previsti), li raggruppa:
  - piccoli gruppi → categorie → categorie più alte.
- » Rischi:
  - Top-down: ignorare dettagli importanti.
  - Bottom-up: struttura troppo “su misura” sull’esistente, poco flessibile.
- » Soluzione: bilanciare **top-down + bottom-up**.

# Strutturare i contenuti

A top-down architectural approach is driven by considerations from the strategy plane.



A bottom-up architectural approach is driven by considerations from the scope plane.

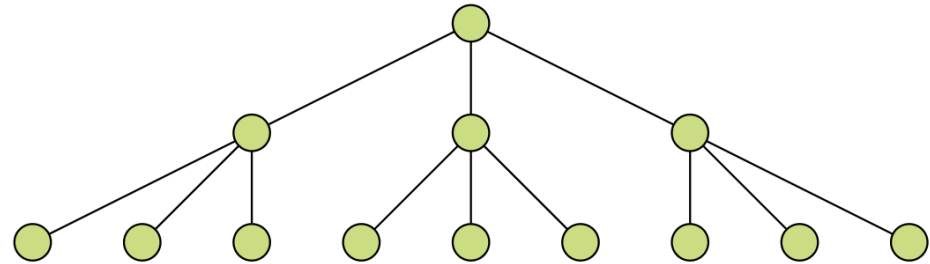


# Tipi di strutture

---

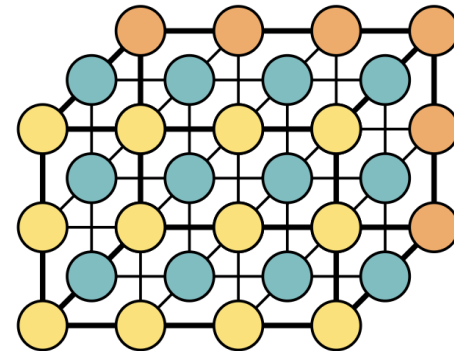
## » Gerarchica (ad albero)

- Nodi con relazioni padre/figlio.
- È la più comune (menu, sottosezioni...)



## » A matrice (matrix)

- Navigazione lungo più dimensioni (es. per **colore** e per **taglia**).
- Utile per supportare diversi modi di cercare le stesse cose.

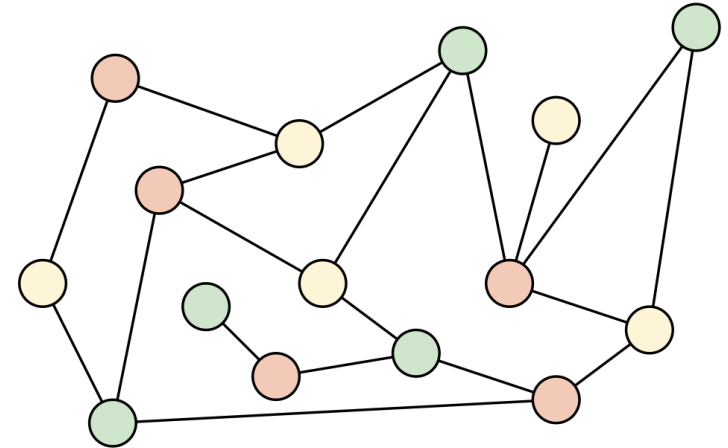




# Tipi di strutture

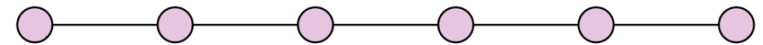
## » Organica

- Collegamenti “a rete”, senza schema rigido.
- Favorisce esplorazione libera, ma può disorientare.



## » Sequenziale

- Passi in ordine fisso (come un libro o un tutorial).
- Indicata quando l'ordine è parte fondamentale dell'esperienza.



# Principi di organizzazione

---

- » Un **principio di organizzazione** è il criterio con cui raggruppiamo i nodi:
  - per **tipo di utente** (Consumer, Business, Investor),
  - per **geografia** (Europa, Asia, America),
  - per **tempo** (ultime news, archivio),
  - per **tema** (Sport, Politica, Economia).
- » Ogni insieme di contenuti può avere **diverse possibili strutture**:
  - la sfida è trovare quella che **meglio risponde** a:
    - » obiettivi di prodotto,
    - » bisogni degli utenti.
- » Concetto di **facets**:
  - attributi come marca, prezzo, categoria, colore...
  - attenzione: troppi facet = caos per l'utente.

# Linguaggio, Vocabolario controllato e Metadata

---

- » Anche la **nomenclatura** è parte della struttura:
  - etichette di menu, nomi categorie, titoli sezioni...
- » Strumento chiave: **vocabolario controllato**
  - lista di termini “ufficiali” da usare,
  - evita gergo interno incomprensibile per gli utenti,
  - garantisce **coerenza** tra diversi autori/editor.
- » **Thesaurus**:
  - mappa sinonimi, sigle, varianti → verso un termine preferito.
  - utile per:
    - » navigazione,
    - » suggerimenti di ricerca,
    - » collegamenti tra contenuti.

# Metadata e ricerca

---

- » **Metadata** = “informazioni sulle informazioni”
  - autore, data, tipo contenuto, tema, prodotto, settore, ecc.
- » **Vantaggi:**
  - maggiore **flessibilità** nel riorganizzare i contenuti,
  - facile creare nuove sezioni o viste (es. una nuova area “Emergenza” usando contenuti già esistenti),
  - ricerca più intelligente rispetto al semplice full-text.
- » **Collegando:**
  - **metadata + vocabolario controllato / thesaurus**
  - il motore di ricerca può:
    - » interpretare sinonimi,
    - » mostrare risultati più pertinenti,
    - » suggerire contenuti correlati.

# Documentare la Struttura

---

- » La documentazione strutturale può assumere forme diverse:
  - outline testuali
  - fogli di calcolo o database
  - diagrammi
- » Il **diagramma** è lo strumento principale per Information Architecture e Interaction Design.
- » Perché?
  - rappresenta visivamente relazioni, gerarchie, flussi
  - rende comprensibile ciò che a parole sarebbe complesso
  - facilita la comunicazione interna al team

# Chi gestisce la struttura del sito?

---

Dipende dall'organizzazione:

» **Team editoriali / marketing**

→ se il sito è orientato ai contenuti

» **Team tecnici**

→ se l'azienda ha cultura IT

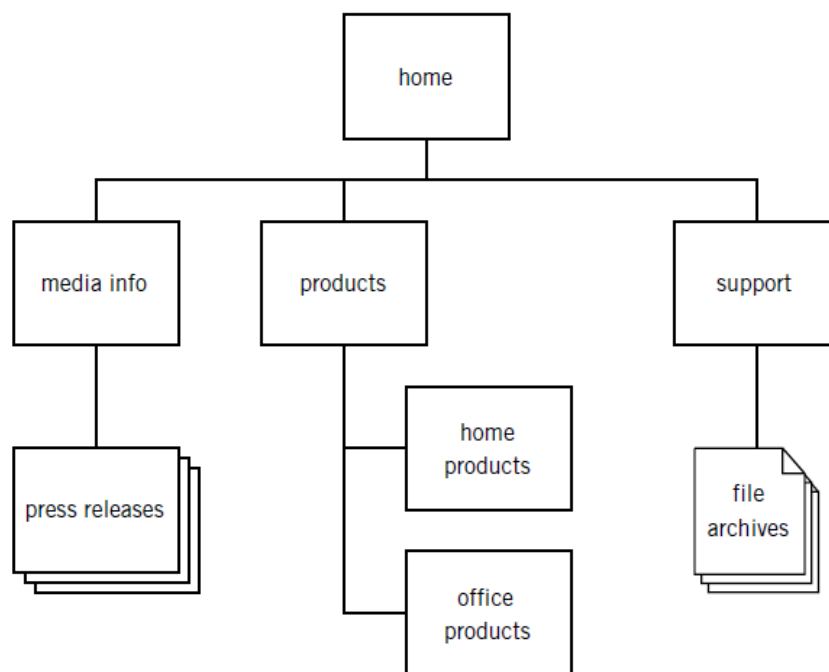
» **Project manager tecnico**

→ se manca una figura UX dedicata

# Navigation models (o sitemap)

» Rappresentano

- **relazione** del contenuto con altri contenuti e
- come l'utente **viaggia** attraverso lo spazio informativo



Per maggiori dettagli:  
<http://www.jjg.net/ia/visvocab/>

# Simbologia

---

- » Utilizzeremo una simbologia di base
  - per la diagrammazione dell'architettura delle informazioni
  - e concetti di progettazione dell'interazione

- » Proposto da **Jesse James Garrett**
  - tra l'altro, ha coniato il termine Aiace





# Simbologia

---

- » Per descrivere ad alto livello la **struttura** e il **flusso** della User Experience di un sito web o di un'app
  - Caratteristiche
  - Semplice
  - Indipendente dallo strumento
  - Piccolo
  - Autonomo
- » L'attenzione è rivolta alla macrostruttura
- » Un vocabolario visivo per descrivere l'architettura delle informazioni e la progettazione dell'interazione :  
<http://jjg.net/ia/visvocab>

# Audience

---

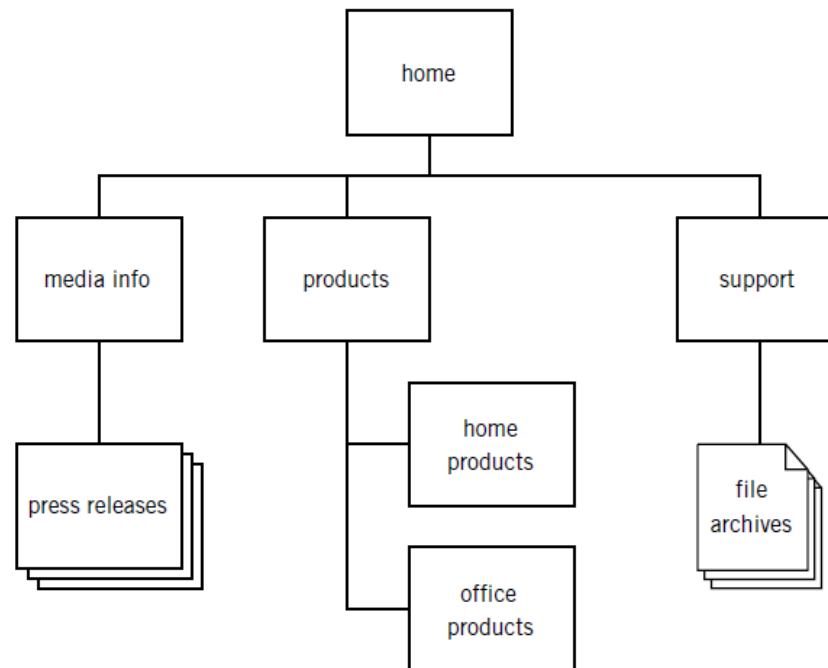
- » Project manager e sponsor
  - per avere un'idea generale del progetto
- » Produttori di contenuti
  - per derivare i requisiti di contenuto
- » Progettisti di interfacce
  - per derivare i requisiti di progettazione dell'interfaccia
- » Tecnologi
  - per derivare requisiti funzionali
- » Architetti dell'informazione
  - per derivare i requisiti di navigazione e di interfaccia

# Modello navigazionale (mappa del sito)

---

## » Concentrarsi su

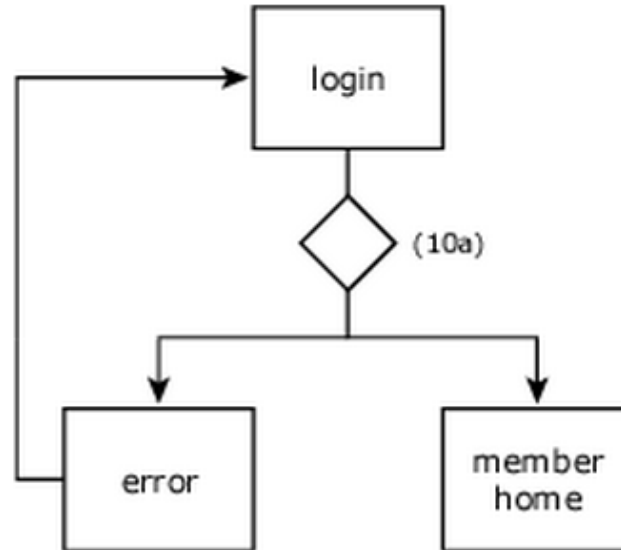
- Relazione del contenuto con altri contenuti e
- Come l'utente viaggia attraverso lo spazio informativo



# Modello navigazionale (mappa del sito)

---

- » Il sistema presenta all'utente dei **percorsi**
- » L'utente si muove lungo questi percorsi **attraverso azioni**
- » Queste azioni quindi fanno sì che il sistema **generi risultati**



# Pagine (Viste)

---

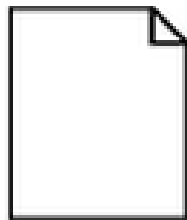
- » Le pagine sono l'unità fondamentale della presentazione
- » Sono anche chiamate viste
- » Non (necessariamente) un'unità di implementazione
- » Una pagina nel tuo diagramma può corrispondere a più file
- » Le pagine possono essere organizzate in una pila di pagine
- » Stack di pagine: pagine funzionalmente identiche con identico trattamento di navigazione



# File

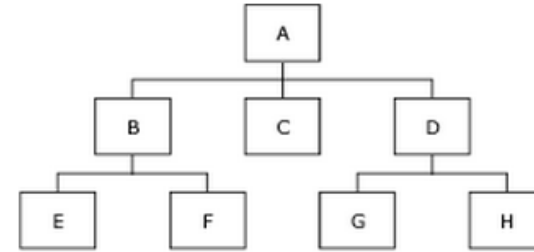
---

- » I file sono pacchetti di dati senza proprietà di navigazione
- » Vengono consegnati all'utente per l'utilizzo al di fuori dell'app
- » I file possono essere organizzati in una pila di file



# Connettori e frecce (1)

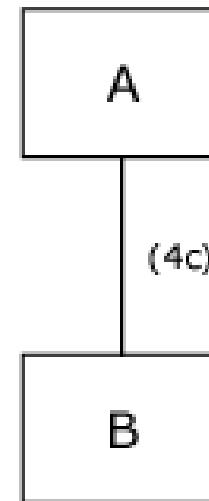
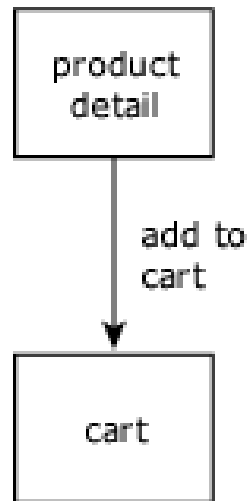
- » Le relazioni tra gli elementi sono rappresentate con linee semplici
- » Frecce per comunicare la direzionalità
  - indicano come l'utente si muoverà nel sistema
- » Una barra trasversale all'estremità opposta della freccia viene utilizzata per impedire il movimento a monte



# Connettori e frecce (2)

---

- » Anche i connettori e le frecce possono essere etichettati
- » Se le etichette diventano lunghe, indirizzare il lettore verso una nota a piè di pagina o una voce di appendice

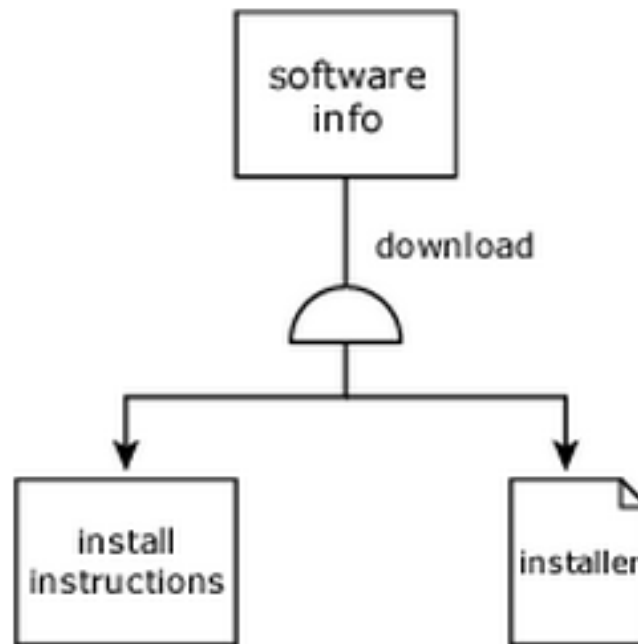




# Insieme concorrente

---

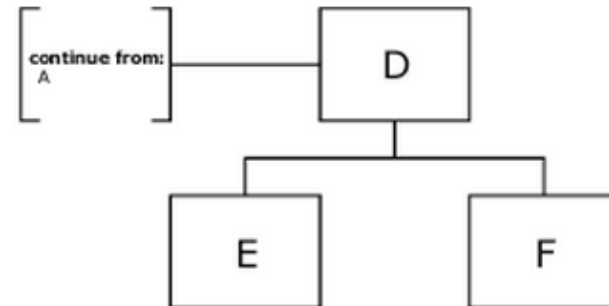
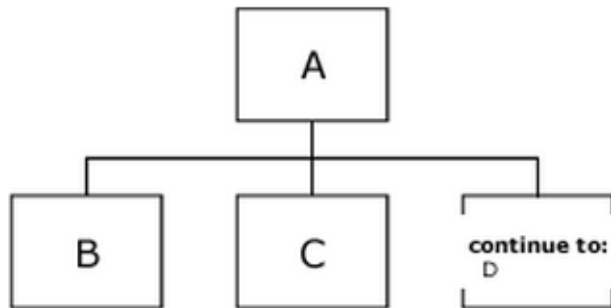
- » Viene utilizzato quando un'azione dell'utente genera più risultati simultanei



# Punti di continuazione

---

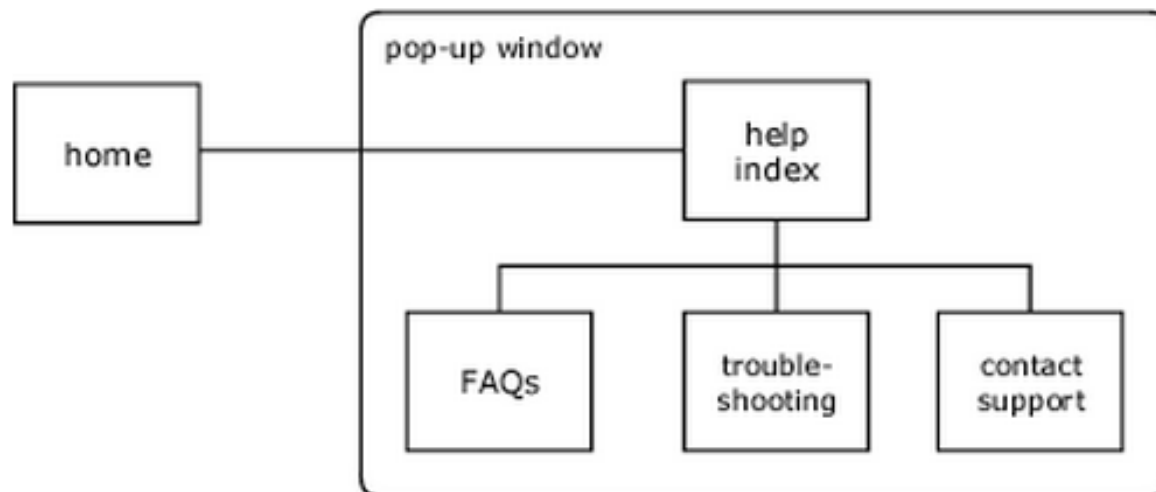
- » I punti di continuazione ci permettono di dividere i nostri diagrammi



# Area

---

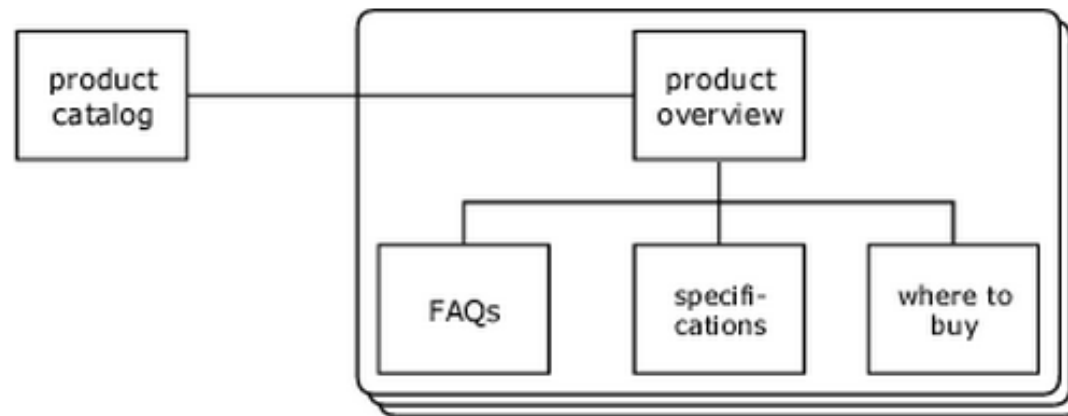
- » Sono utilizzati per identificare un gruppo di pagine che condividono uno o più attributi comuni
- » Esempio: apparire in una finestra pop-up con un trattamento di design unico



# Area

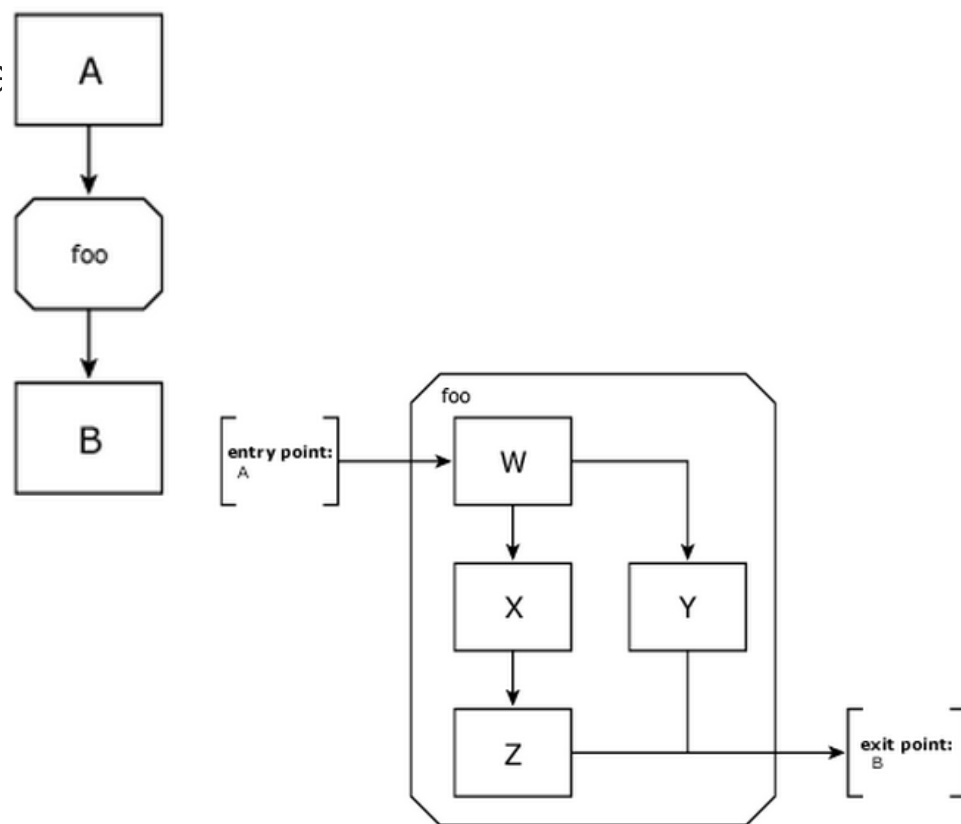
---

- » Sono utilizzati per rappresentare architetture che implicano la ripetizione della stessa struttura di base applicata a un numero di elementi informativi funzionalmente identici
- » Esempio: un catalogo prodotti in cui ogni prodotto ha un certo numero di pagine associate



# Area di flusso

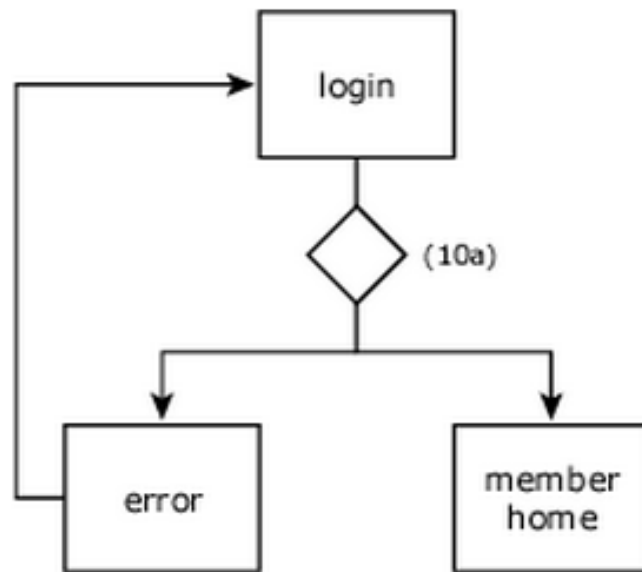
- » Racchiude una sequenza di passaggi che appariranno ripetutamente nel diagramma
- » Esempio: procedura di login
- » Richiedono due tipi speciali di punti di continuazione: punti di ingresso e punti di uscita



# Punto di decisione

---

- » Viene utilizzato quando un'azione dell'utente può generare uno dei numerosi risultati e il sistema deve decidere quale risultato presentare

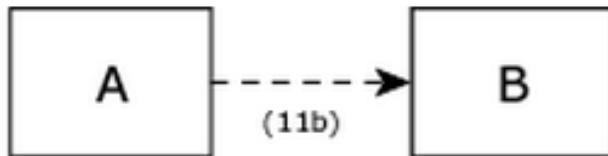


(10a) credenziali corrette?

# Connettore condizionale

---

- » Viene utilizzato quando un percorso può o non può essere presentato all'utente a seconda che siano soddisfatte una o più condizioni

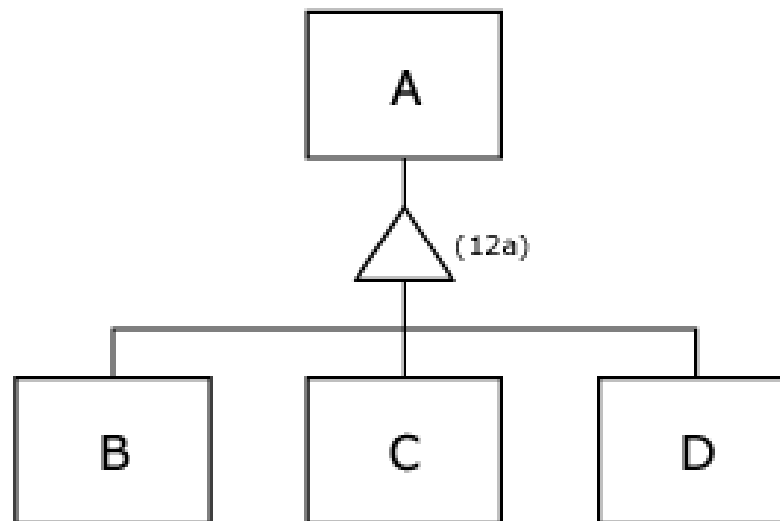


(11b) solo i dipendenti dell'azienda possono avere accesso a B

# Ramo Condizionale

---

- » Viene utilizzato quando il sistema ( **non in base all'azione dell'utente** ) deve selezionare un percorso tra una serie di opzioni reciprocamente esclusive

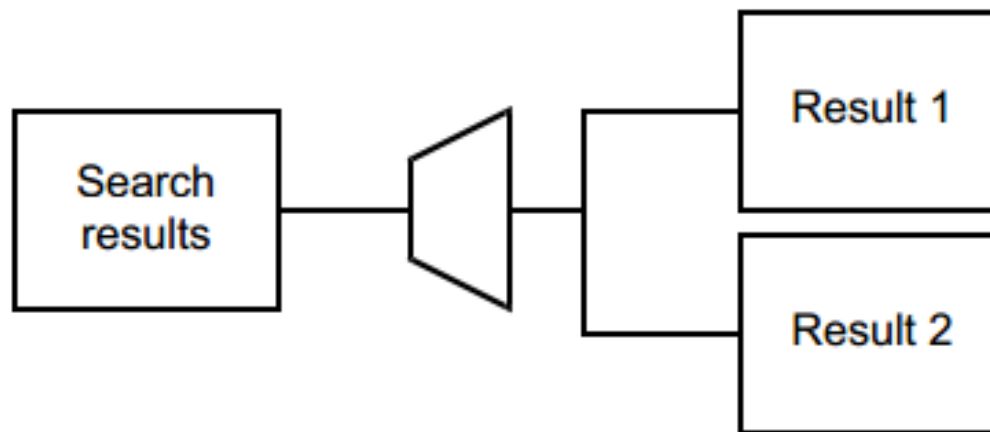




# Selettore Condizionale

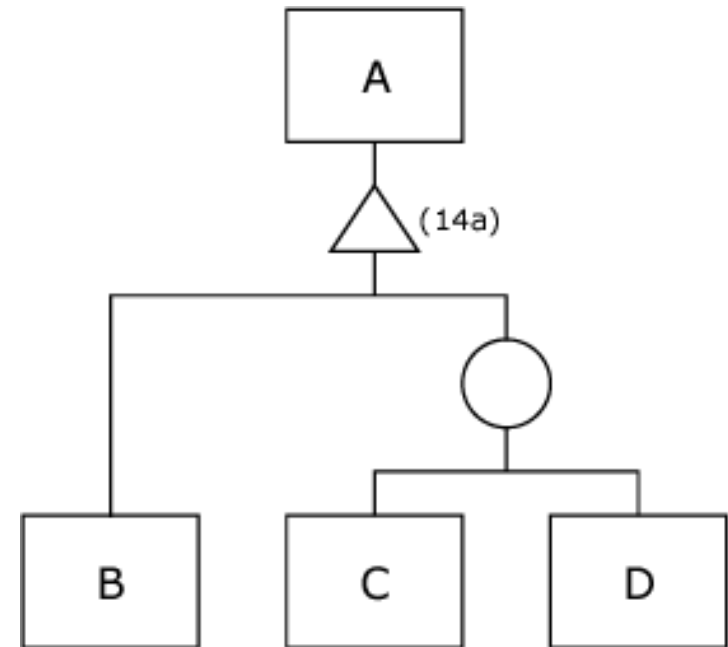
---

- » Uguale al ramo condizionale, ma i percorsi non sono mutuamente esclusivi
- » L'applicazione più comune del selettore condizionale è nei risultati generati da un motore di ricerca



# Grappolo

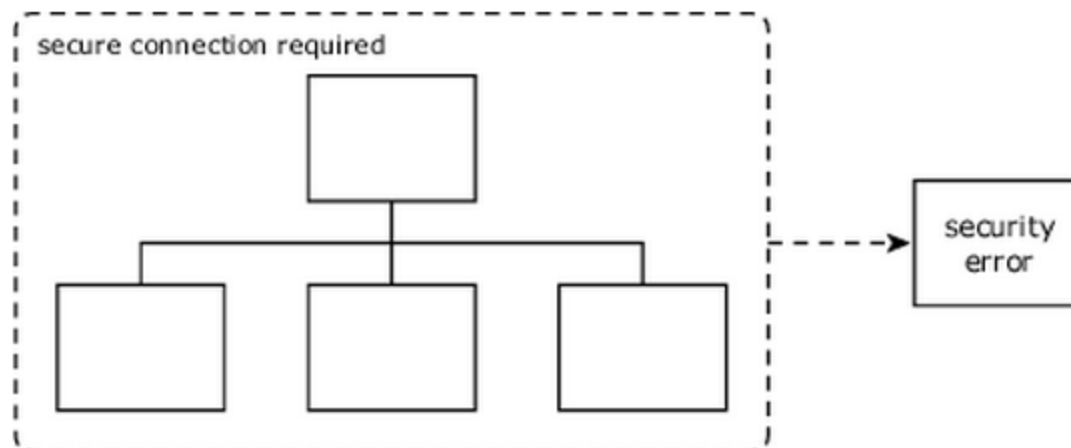
- » Simile al selettore condizionale, il sistema consente più di un percorso per determinate condizioni
- » Se l'attributo valutato ha valore **x**, l'utente vede un percorso verso la pagina B; ma se l'attributo ha valore **y**, l'utente vede percorsi sia verso la pagina C che verso la pagina D



# Area Condizionale

---

- » Viene utilizzato quando una o più condizioni si applicano a un gruppo di pagine
- » Viene generato un risultato in uscita se la condizione non è soddisfatta
- » Esempio: permessi di accesso



UNIVERSITÀ  
DEGLI STUDI  
DI TERAMO

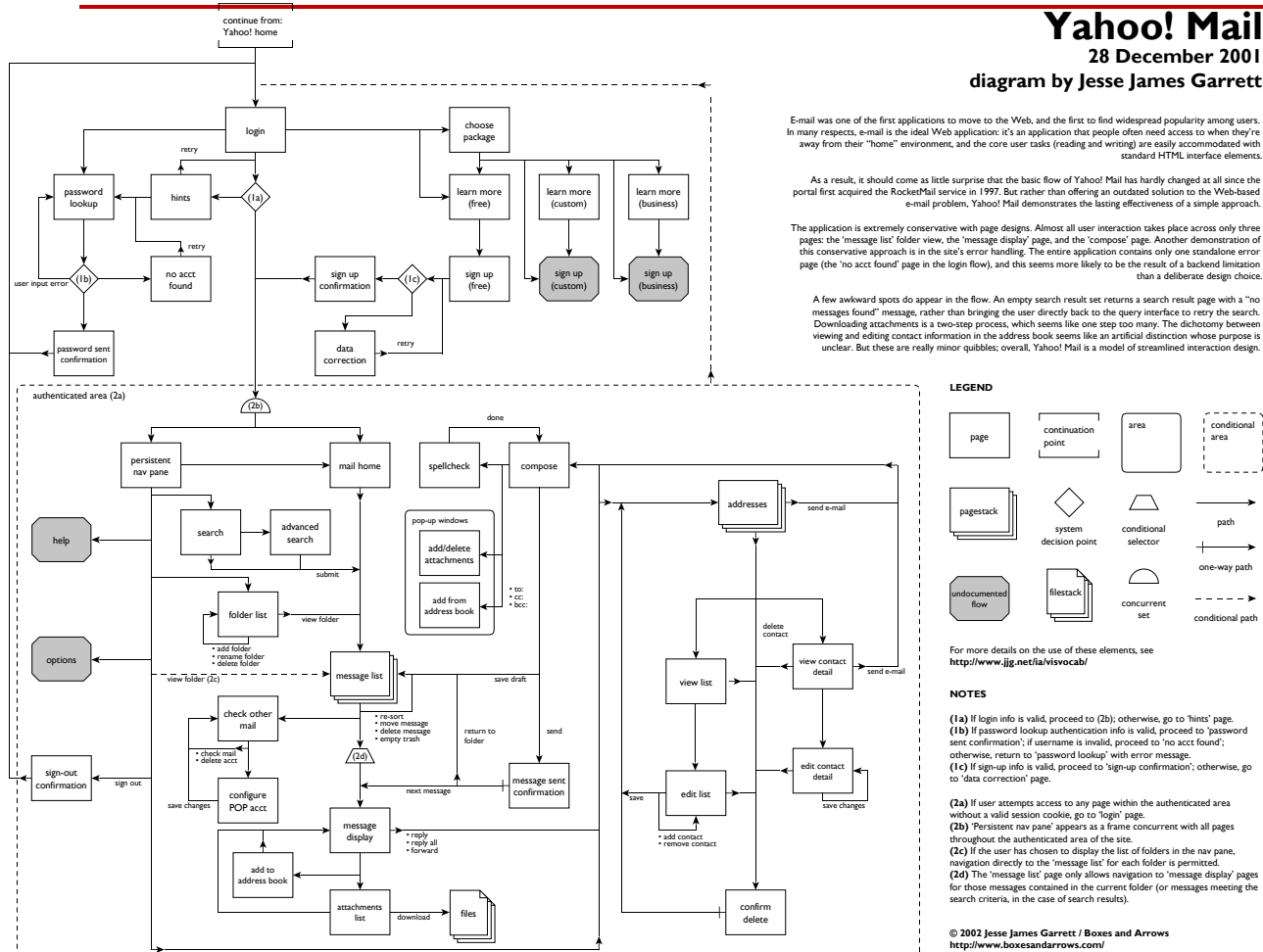
diagram by Jesse James Garrett

E-mail was one of the first applications to move to the Web, and the first to find widespread popularity among users. In many respects, e-mail is the ideal Web application: it's an application that people often need access to when they're away from their "home" environment, and the core user tasks (reading and writing) are easily accommodated with standard HTML interface elements.

As a result, it should come as little surprise that the basic flow of Yahoo! Mail has hardly changed at all since the portal first acquired the RocketMail service in 1997. But rather than offering an outdated solution to the Web-based e-mail problem, Yahoo! Mail demonstrates the lasting effectiveness of a simple approach.

The application is extremely conservative with page designs. Almost all user interaction takes place across only three pages: the 'message list' folder view, the 'message display' page, and the 'compose' page. Another demonstration of this conservative approach is in the site's error handling. The entire application contains only one standalone error page (the 'no acct found' page in the login flow), and this seems more likely to be the result of a backend limitation than a deliberate design choice.

A few awkward spots do appear in the flow. An empty search result set returns a search result page with a “no messages found” message, rather than bringing the user directly back to the query interface to retry the search. Downloading attachments is a two-step process, which seems like one step too many. The dichotomy between viewing and editing contact information in the address book seems like an artificial distinction whose purpose is unclear. But these are really minor quibbles; overall, Yahoo! Mail is a model of streamlined interaction design.



For more details on the use of these elements, see <http://www.jjg.net/ia/visvocab/>

## NOTES

- (1a) If login info is valid, proceed to (2b); otherwise, go to 'hints' page.
  - (1b) If password/login authentication info is valid, proceed to 'password sent confirmation'; if username is invalid, proceed to 'no acct found'; otherwise, return to 'password/login' with error message.
  - (1c) If sign-up info is valid, proceed to 'sign-up confirmation'; otherwise, go to 'data correction' page.
- 
- (2a) If user attempts access to any page within the authenticated area without a valid session cookie, go to 'login' page.
  - (2b) 'Persistent nav pane' appears as a frame consistent with all pages throughout the authenticated area of the site.
  - (2c) If the user has chosen to display the list of folders in the nav pane, navigation to the 'message list' for each folder is permitted.
  - (2d) 'The message list' page only allows navigation to 'message display' pages for those messages contained in the current folder (or messages meeting the search criteria, in the case of search results).

© 2002 Jesse James Garrett / Boxes and Arrows  
<http://www.boxesandarrows.com/>

# Strumenti

---

Strumento per costruire modello navigazionali e di dominio:

<https://www.draw.io>