

# **1: Introduzione alla modellazione concettuale**

2

Modellazione  
concettuale di  
applicazioni Web

1

## Modellazione concettuale di applicazioni Web

- Metodologia di sviluppo per applicazioni Web data-intensive

### ● WebML (Web Modeling Language) –

<http://webml.org>

- Modellazione del contenuto informativo

- Modellazione dell'iper testo

- WebRatio: un tool di sviluppo per WebML –

<http://www.webratio.com>

3

## Applicazioni Web data-intensive

- Sistemi per il Web, il cui scopo principale è gestire e pubblicare grandi quantità di dati

- Dotate di interfacce ipertestuali

- Supportano la navigazione tra i contenuti

- In alcuni casi sono personalizzate

### Esempi:

- Commercio: cataloghi elettronici, e-mail, aste elettroniche, ...
- Contenuto: quotidiani online, biblioteche elettroniche, siti istituzionali ...
- Servizi: e-banking, prenotazioni, monitoraggio degli ordini, ...
- Comunità: portali tematici, forum, ...
- Contesto: portali di ricerca, directories, ...

Tutte gestiscono e pubblicano grandi quantità di dati

4

## Problematrice di sviluppo

- **Manca di metodi adeguati**
  - I metodi centrali sui dati non soddisfano i requisiti di sviluppo del front-end
  - I metodi OO (per es. UML) non sempre catturano l'essenza dei sistemi basati sul Web
- **Manca di modelli concettuali**
  - Modelli di ipertesto non adeguati

## Alcuni fattori di complessità

- **Output su dispositivi diversi**
  - PC, telefoni WAP, palmari, ...
- **Personalizzazione One-to-One**
  - myYahoo, myCDNOW, ...
- **Carenza di personale IT esperto**
  - I Web designers si concentrano sul look-and-feel
  - Elevati sforzi di programmazione per codificare a mano software ridondante - difficoltà nella generazione di prototipi
- **Costi di mantenimento superiori ai costi per la prima versione del sistema**
- **Documentazione povera = difficile evoluzione dell'applicazione**

# Requisiti per la modellazione di applicazioni Web

- **Espressività**
  - Possibilità di specificare casi reali
  - Catalogazione dei design pattern frequentemente adottati
- **Facilità d'uso**
  - Notazioni visuali intuitive
  - Semantica non ambigua - Verifica automatica degli schemi
- **Implementabilità**
  - Mapping efficiente verso strutture fisiche dei dati
  - Generazione automatica di codice a partire da specifiche ad alto livello

8

# Vantaggi della modellazione concettuale

- **Svariati successi in diversi campi:**
  - ER per la progettazione dei dati
  - UML per l'analisi e la progettazione
- **Processo di sviluppo maggiormente strutturato**
- **Applicazioni finali maggiormente usabili e coerenti**
- **Maggiore qualità della documentazione**
- **Generazione automatica del codice**
  - Produzione di prototipi immediata
  - Minori sforzi di sviluppo (costi e tempo)

7

# Alcuni modelli concettuali

- Prime proposte per la modellazione di applicazioni ipermediali (*Communication of ACM, August 1995*)

- HDM (1993)
- OO-HDM (1995)
- RMM (1995)

- Proposte successive, per la modellazione di applicazioni Web

- ARANEUS (1998)
- ADM (Araneus Data Model)
- Strudel (1998)
- UGM (Unified Graph Model) + StruQL (Strudel Query Language)
- Web Modeling Language – WebML (1998)

- Modello ER per il contenuto + Primitive visuali per l'ipertesto
- Supportato da uno strumento CASE commerciale (<http://www.webraio.com>)

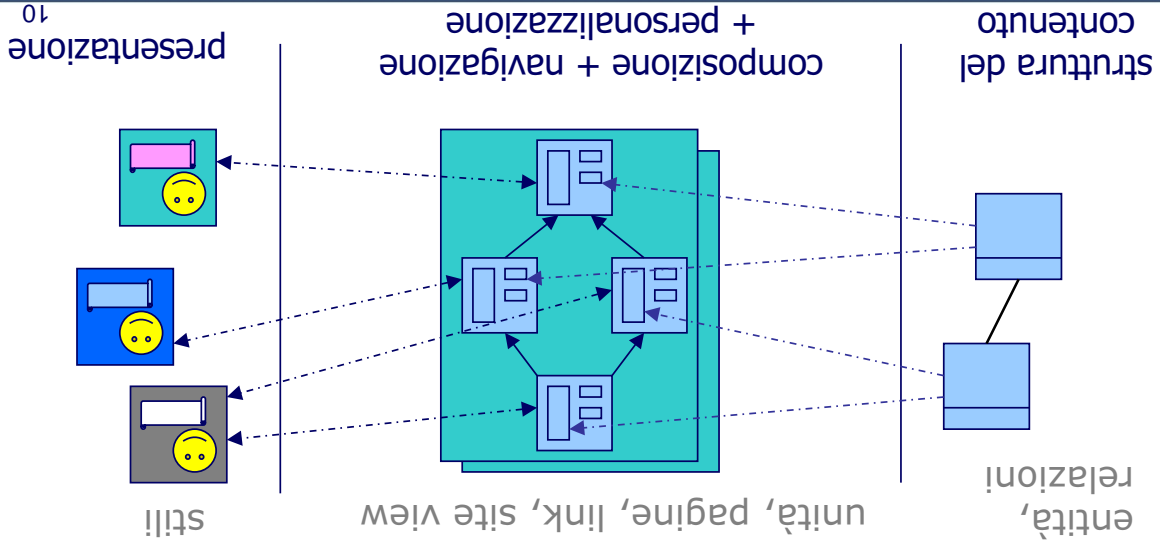
- Estensione di UML per il Web (*Jim Conallen, "Building Web Applications with UML", Addison Wesley, 2000*)

6

# WebML – concetti di base

Applicazione Web =

**Dati** + **Iper testo** + **Presentazione**



10

## 2. Modello dei Dati in WebML

11

### Obiettivi

MODELLO DEI DATI

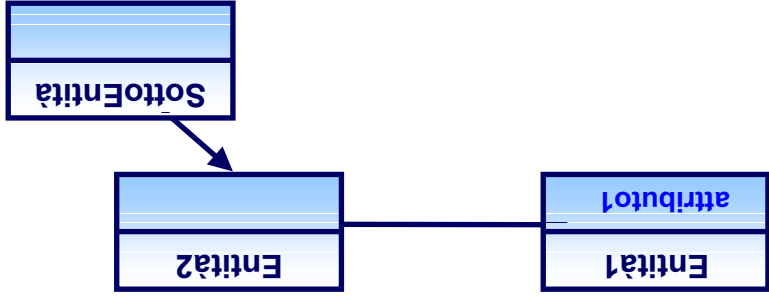
- **Tipiche domande:**
  - Quali sono gli oggetti informativi da pubblicare tramite l'applicazione?
  - Quali sono le proprietà che li caratterizzano?
  - In che modo i vari oggetti sono correlati?

- **Primitive del modello Entità-Relazioni:**

- **Entità:** una classe di oggetti nel dominio dell'applicazione
- **Attributo:** una proprietà di una entità
- **Relazione:** una connessione tra entità
- **Gerarchia IS-A:** costruito utilizzato per classificare o raggruppare

12

## Notazione grafica

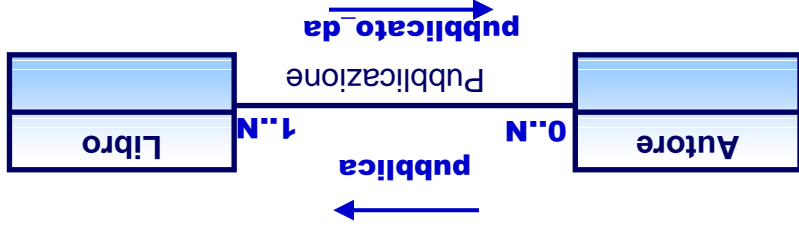


- Ogni entità identificata da un Object Identifier (OID), non rappresentato esplicitamente nello schema

13

## Ruoli e cardinalità delle relazioni

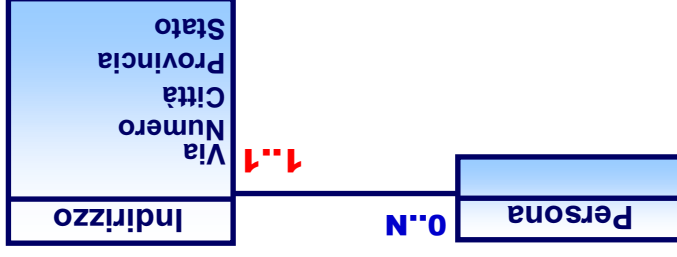
- Il ruolo di una relazione è una delle due "direzioni" in cui una relazione può essere interpretata
- Per ogni ruolo, possono essere specificati vincoli di cardinalità



14

## Attributi composti

- Attributi con una struttura interna (per es., un indirizzo può includere diversi campi)
- Rappresentati per mezzo di una entità e una relazione
- Es.: una persona ha uno o più indirizzi, ognuno formato da numero, via, città, provincia, stato



NOTA: L'entità Indirizzo è debole

## Attributi multi-valore

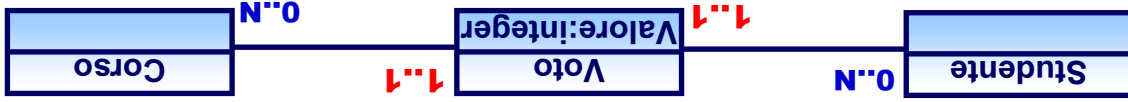
- Attributi di un oggetto che possono assumere un insieme di valori
- Reppresentati per mezzo di una entità più una relazione
- Es.: una persona ha più numeri telefonici



NOTA: L'entità Telefono è debole

## Relazioni con attributi

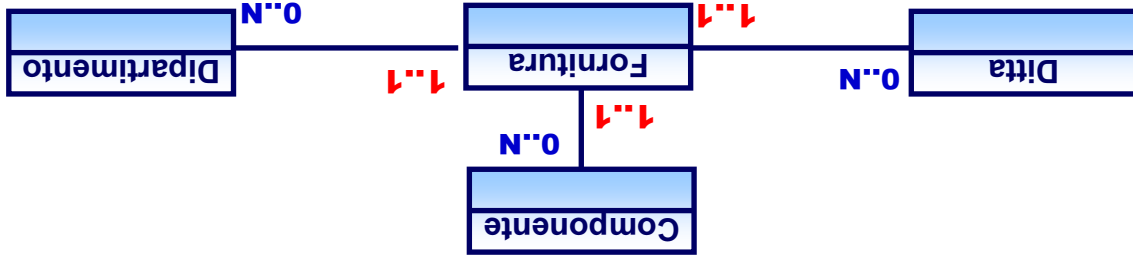
- Rappresentate da una entità centrale e due relazioni!
- Es.: il voto preso da uno studente nell'esame di un certo corso



NOTA: l'entità Voto è debole

## Relazioni N-arie

- Relazioni che coinvolgono N entità (N>2)
- Es.: la fornitura di componenti ad un dipartimento da parte di una ditta
- Rappresentabili da una combinazione di entità e N relazioni binarie



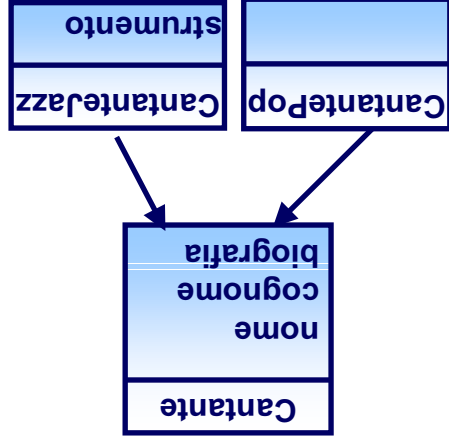
NOTA: L'entità Fornitura è debole

## Gerarchie IS-A

- Le sotto-entità ereditano gli attributi e le relazioni di una super entità
- Le sotto-entità possono anche avere proprietà localmente definite
- Le sotto-entità modellano collezioni!

### Alcune ipotesi restrittive in WebML:

- Non è permessa l'ereditarietà multipla
- E' possibile definire solo gerarchie esclusive
- Ogni entità può apparire al massimo in una gerarchia



## Esempio Catalogo Elettronico della ditta ACME

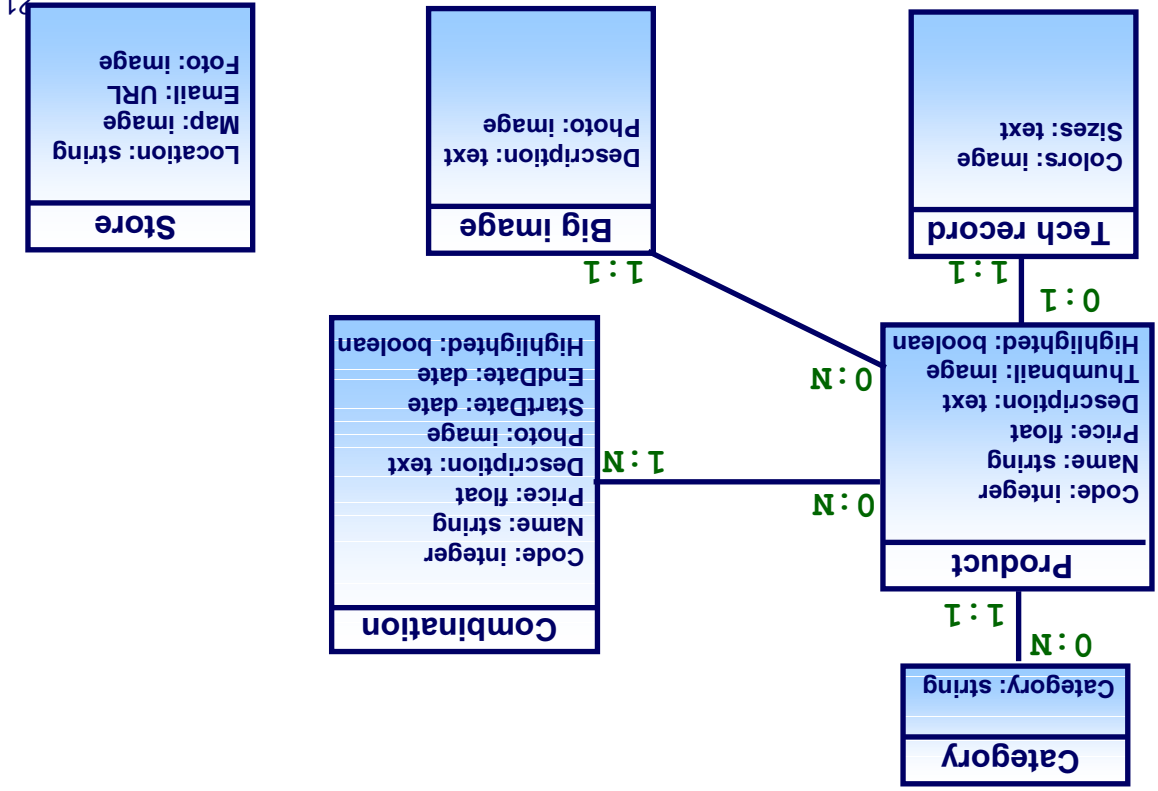
- ACME: una medio-piccola impresa di mobili, che vende i suoi prodotti attraverso punti vendita sparsi in tutto il mondo
- Necessità di sviluppare un sito Web per pubblicare il catalogo dei suoi prodotti.

● Il sito deve pubblicare informazioni su:

- prodotti
- offerte speciali: combinazioni di prodotti venduti ad un prezzo scontato
- punti vendita
- Ogni prodotto ha una scheda tecnica e diverse immagini
- I prodotti sono raggruppati in categorie

# ACME – Schema dei dati

CASO DI STUDIO



21

## Objetti derivati

- La derivazione permette di:

- Arricchire il contenuto di un'entità, aggiungendo sia attributi calcolati, sia attributi importati da oggetti correlati.

- Definire la popolazione di entità o di relazioni, sulla base di alcune proprietà degli oggetti coinvolti.

22

MODELLO DEI DATI

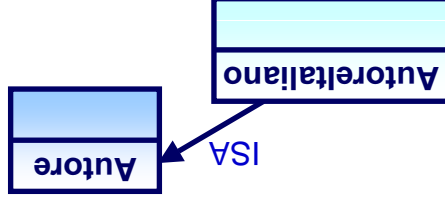
## WebML OQL

- La derivazione si specifica scrivendo espressioni chiamate "query di derivazione"
- WebML OQL (Object Query Language): un linguaggio per esprimere query di derivazione
- Concetti derivabili:
  - Entità, relazioni, attributi

- Le query di derivazione sono **automaticamente** trasformate in viste SQL installate nella base di dati

## Entità derivate

- La popolazione di Sotto-Entità nelle gerarchie IS-A può essere specificata per mezzo di query OQL



- Es.: "Un'autore italiano è un autore nato in Italia"
- WebML OQL:

```
value = "Autore as A where A.LuogoNascita contains `Italia`"
```

Value="Politecnico di Milano"

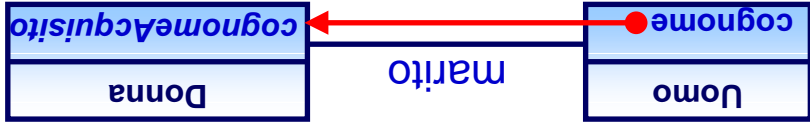
Affiliazione
DipendentePoi

## Attributi Costanti

- Quattro tipi di attributi derivati:
  - Attributi costanti
  - Attributi Importati
  - Attributi Calcolati
  - Attributi Aggregati

## Attributi derivati

## Attributi importati



```
value="Self.marito.cognome"
```

- La variabile `Self` denota l'entità corrente in cui si vuole definire l'attributo

## Attributi calcolati e aggregati



Prezzo totale dell'ordine come somma dei prezzi delle singole linee d'ordine:

1. prezzoTotale in Dettaglio come attributo calcolato: `Self.prezzo * Self.qta`
2. prezzoTotale in Ordine come attributo aggregato: `Sum (Self.Ordine_Dettaglio.prezzoTotale)`

Il numero di linee d'ordine per un certo ordine  
 Count (`Self.Ordine_Dettaglio`)

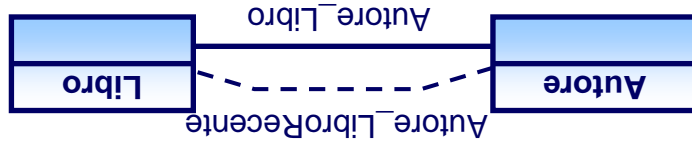
## Relazioni Derivate

Le relazioni possono essere derivate in due modi:

- Definendo restrizioni e/o composizioni di relazioni pre-esistenti
- Definendo nuove relazioni tra coppie di oggetti in base a condizioni

## Relazioni derivate: restrizione

**Restrizione di relazioni pre-esistenti:**



```
Value="Self.Autore_Libro as T where T.date >
\01/01/03'"
```

# Modello di ipertesto in WebML

## Relazioni derivate: concatenazione

**Concatenazione di relazioni:**



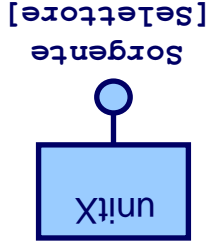
```
value="Self.Capitolo_Libro.Libro.Libro_Autore"
```

- Modellazione ad alto livello del front-end di una applicazione Web dinamica e delle interazioni con la logica e i dati del back-end
- Utilizzo di una notazione visuale semplice ma formale
- Generazione automatica di template di pagine dinamiche e di interrogazioni per l'accesso e la manipolazione dei dati

## Modello di Iper testo: obiettivi

MODELLO DI IPERTESTO

## 1. Motivazioni e Concetti di Base

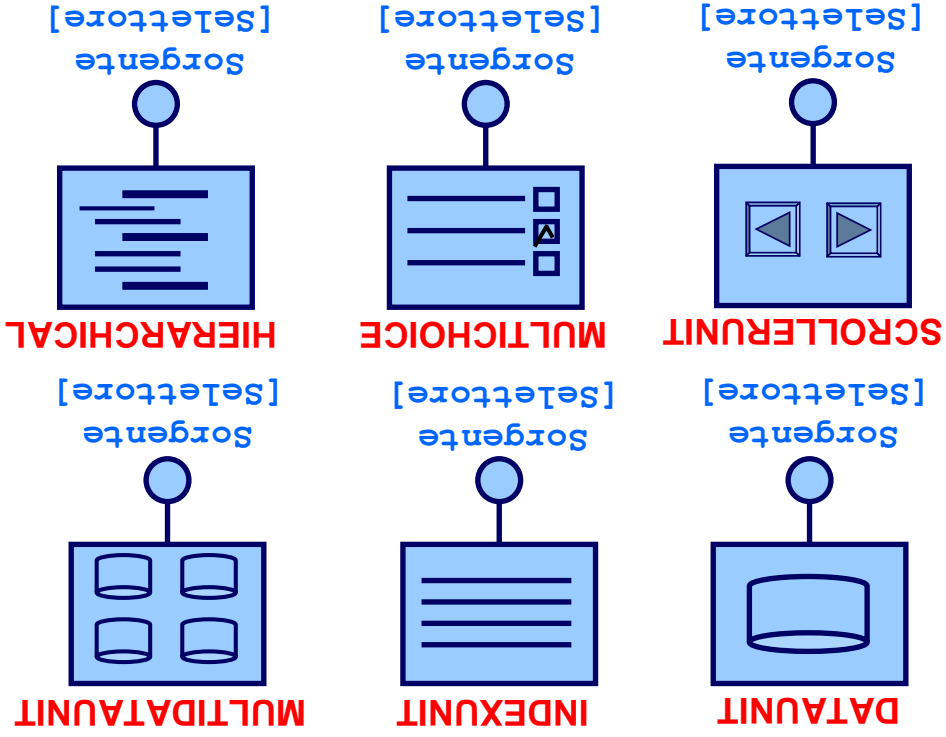


- Una **unit di contenuto** in WebML è l'elemento atomico per la pubblicazione dell'informazione
- Corrisponde ad una "vista" definita su di un contenitore di oggetti:
  - Tutte le istanze di un'entità sorgente
  - Le istanze di una entità che soddisfano una condizione di selezione chiamata settore

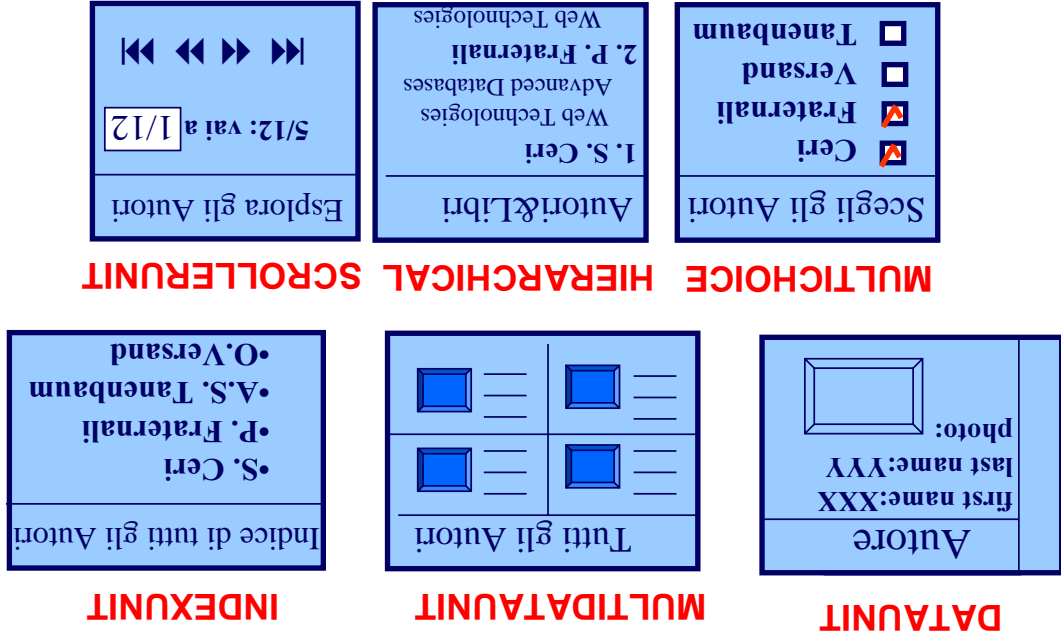
## Unità di contenuto

- ## Modellazione dell'Iper testo: Domande Tipiche
- Domande tipiche
- In che modo l'utente deve fruire del contenuto pubblicato tramite il sito?
  - Quali sono le pagine nell'ipertesto, tramite cui l'utente può accedere ai contenuti?
  - Quale informazione deve essere pubblicata in ogni pagina?
  - In che modo i nodi dell'ipertesto sono collegati tra loro?
- Primitive del modello di Iper testo
- Site view
  - Pagine
  - Unità di contenuto (o unit)
  - Link

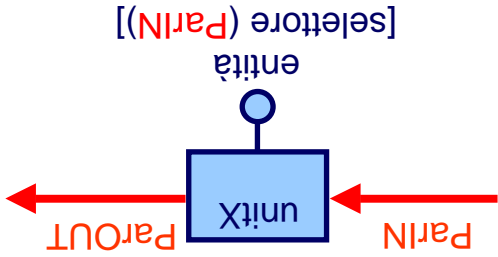
# Unità di Contenuto di Base



# Significato delle unità di contenuto



## Input e output delle unit



- Ogni unit può avere parametri di input e output
- I parametri in input sono necessari per calcolare la unit
- Parametri richiesti dal selettore della unit
- I parametri in output possono essere utilizzati per la computazione di una o più unit che dipendono dalla unit corrente

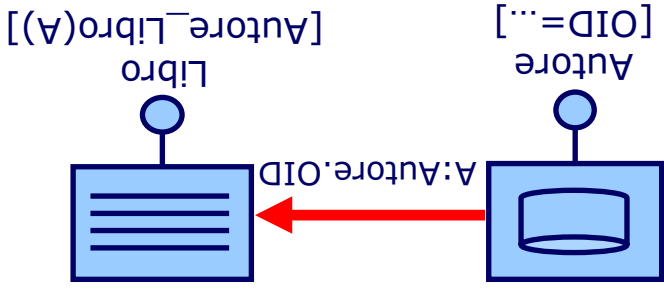
39

## 2. Link e parametri

40

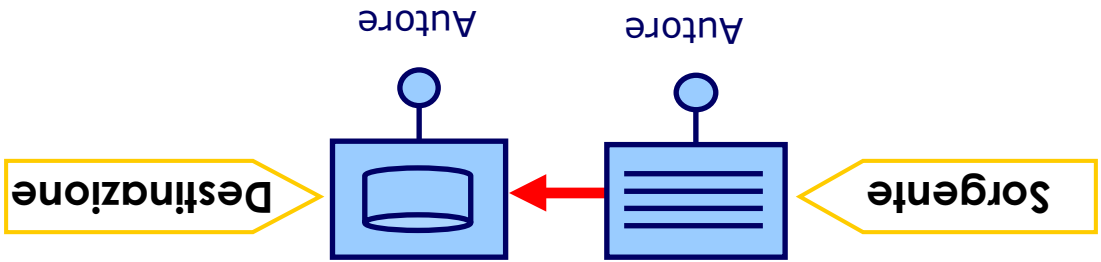
## Parametri sui link

- Il contesto è trasportato dai link attraverso l'uso di parametri definiti sui link
- Un parametro sui link ha un **Nome**.
- Il contenuto del parametro è un attributo della unit sorgente del link



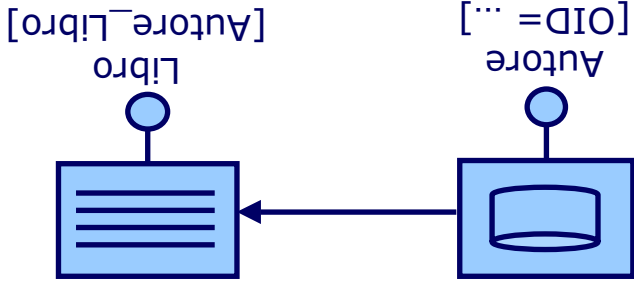
## Link contestuali

- Connessioni orientate tra due unit (sorgente e destinazione), la cui presentazione corrisponde ad ancore o bottoni "submit" che permettono all'utente di navigare tra nodi diversi dell'iper testo
- Trasportano informazione di contesto
- Attivano una computazione (effetto collaterale)



## Parametri sui link: default

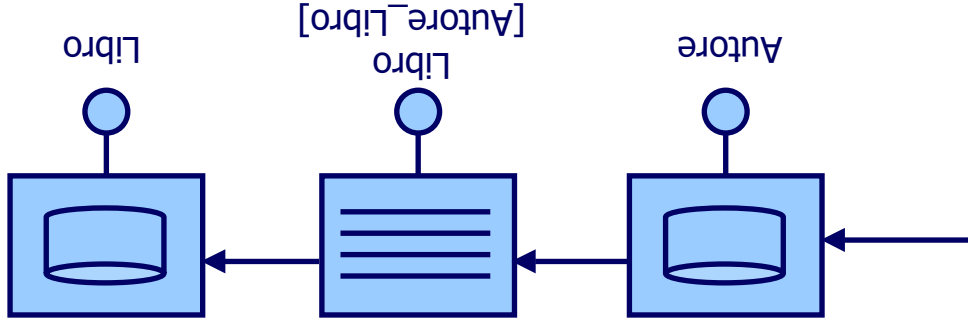
- Quando è possibile, i parametri sui link sono inferiti dal diagramma, senza bisogno di essere specificati esplicitamente
- I diagrammi acquistano maggiore chiarezza



43

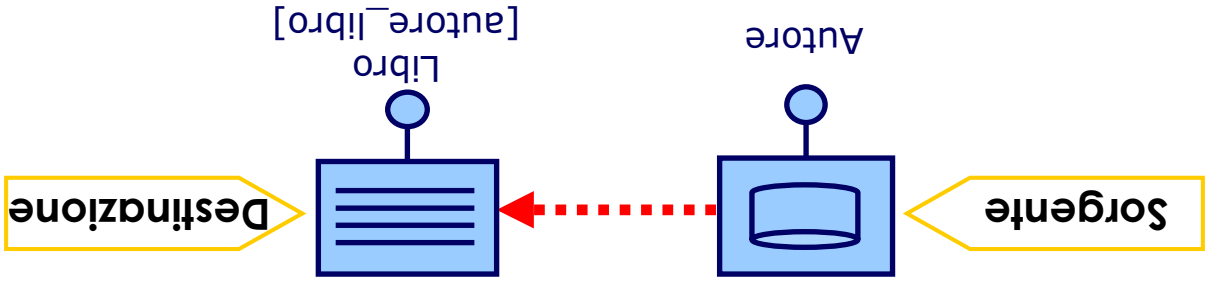
## Selettori: Default

- Quando possibile, i selettori e i loro parametri sono inferiti dal diagramma, senza dover essere specificati esplicitamente



44

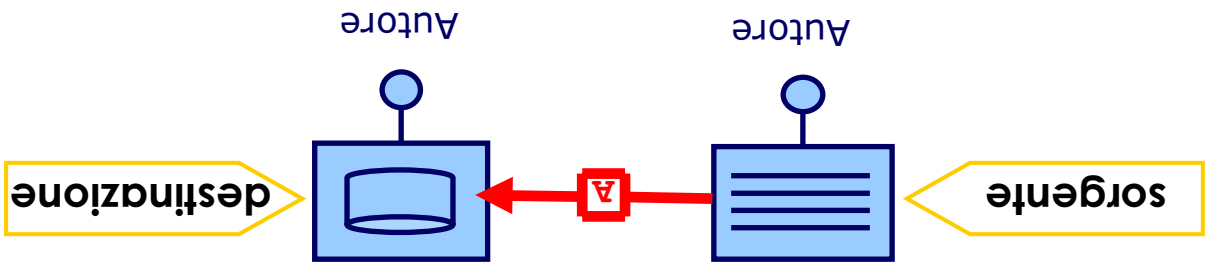
- Un link di trasporto passa il contesto alla unit di destinazione immediatamente dopo la visualizzazione della unit sorgente, senza la necessità dell'intevento dell'utente
- L'utente non può cambiare il contesto trasportato dal link; il link non è visualizzato tramite un ancora



## Link di trasporto

MODELLO DI IPERTESTO

- Passano il **contesto** alla target unit immediatamente dopo la visualizzazione della source unit, senza la necessità di un intevento dell'utente
- Successivamente, l'utente può cambiare il contesto passato, scegliendo un oggetto differente tramite l'ancora (o le ancora) che rappresentano il link



## Link automatici

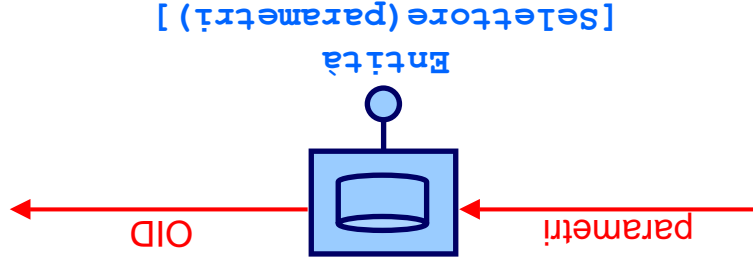
MODELLO DI IPERTESTO

### 3. Unit di Contenuto

47

48

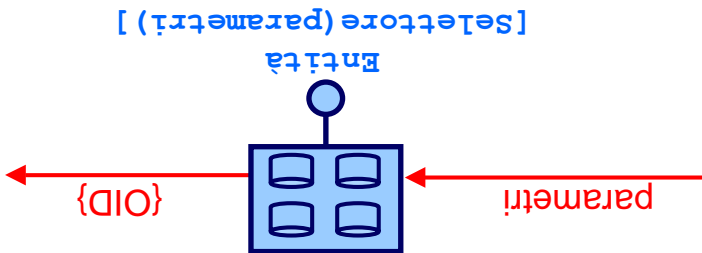
- **Contenitore: una entità, più (opzionalmente) un selettore**  
Parametri in input:
  - **OID dell'oggetto** che deve essere pubblicato, **OPPURE**
  - **Parametri richiesti** dalla computazione del selettore
  - **Parametri in output:**
  - **L'OID dell'oggetto** pubblicato, più ogni suo attributo
- **Pubblica informazione circa UNA SINGOLA ISTANZA**



**DataUnit**

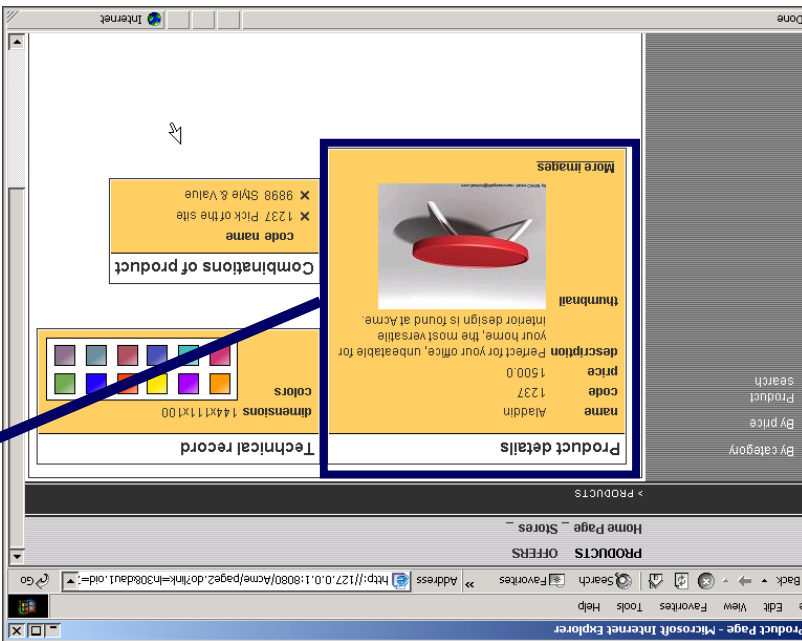
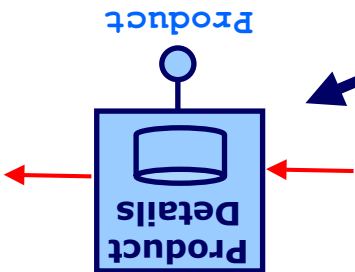
MODELLO DI IPERTESTO

- Presenta istanze multiple di una entità (insieme di oggetti)
- **Contentore:** una entità, piu (optionalmente) un selettore
- **Parametri di input:** quelli richiesti per la computazione del selettore
- **Parametri di output:** L'insieme di **OID** degli oggetti (piu gli attributi degli oggetti)



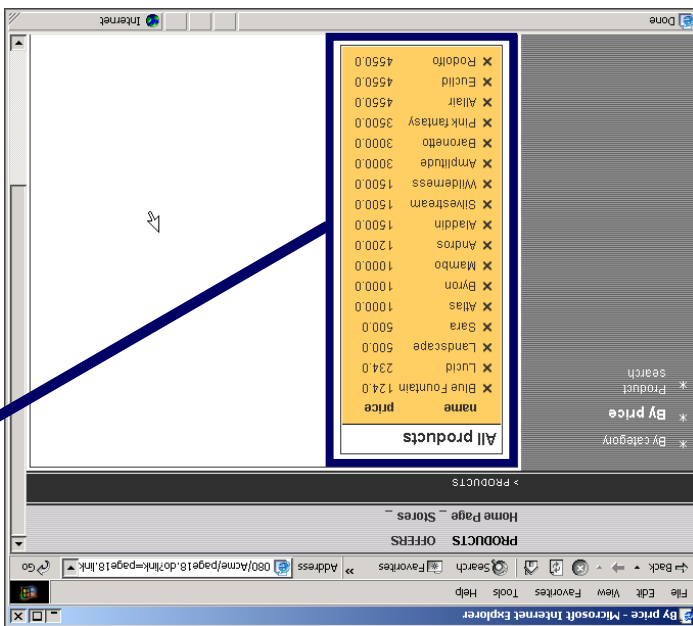
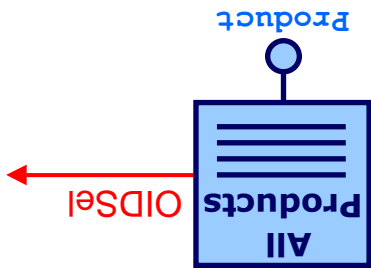
# MultidataUnit

Informazione su di un prodotto specifico



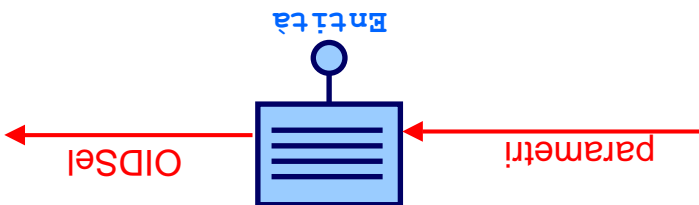
# ACME Esempio di DataUnit

L'utente può accedere ai singoli prodotti cliccando su uno degli elementi della lista



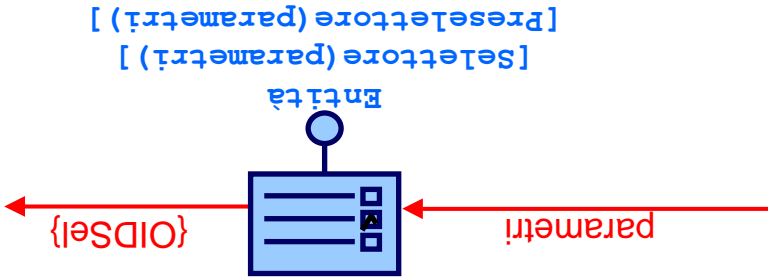
# ACME: IndexUnit nella pagina Products

- Contenitore: una entità, più (opzionalmente) un selettore (Parametri di input: quelli richiesti per la computazione del selettore Parametri in output: **OID dell'oggetto selezionato** dall'utente (più tutti i suoi attributi))
- Pubblica una lista di elementi (insieme di oggetti) [Selettore (parametri.) ]



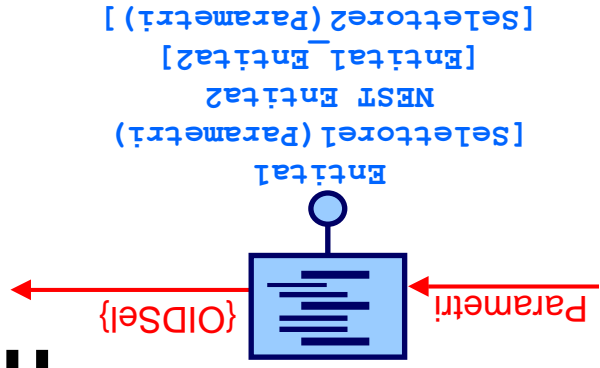
# IndexUnit

- **Pubblica indici di elementi (insieme di oggetti)** tra cui l'utente **seleziona uno o più elementi** (tramite checkbox)
- **Contenitore: una entità, più (opzionalmente) selettore e pre-selettore**
- **Parametri in input: quelli richiesti per la computazione dei selettori**
- **Parametri in output: **OID degli oggetti marcati** dall'utente (più tutti i suoi attributi)**



## MultichoiceUnit

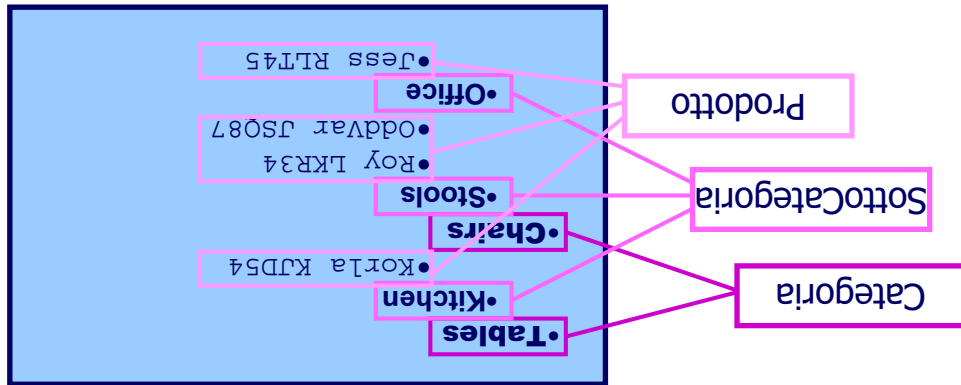
- **Pubblica una lista di oggetti** (appartenenti a più entità) organizzati gerarchicamente in base a relazioni definite tra le entità
- **Contenitore: un insieme di entità e le relazioni che le associano, più (opzionalmente) selettori ad ogni livello**
- **Parametri in input: quelli richiesti dalla computazione dei selettori**
- **Parametri in output: **OID** dell'oggetto selezionato dall'utente (più tutti i suoi attributi)**



## HierarchicalUnit

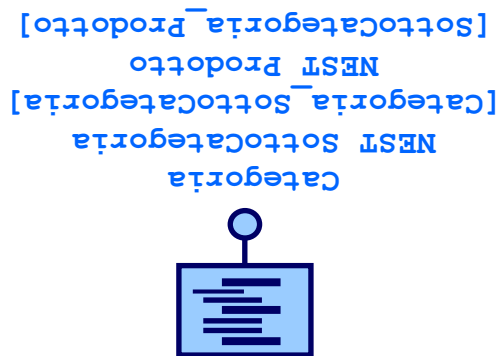
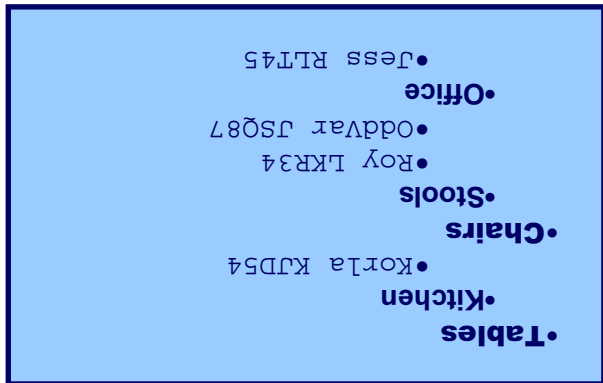
# HierarchicalUnit: Esempio

MODELLO DI IPERTESTO



# HierarchicalUnit: Esempio

MODELLO DI IPERTESTO



- Contenitore: una entità, più (opzionalmente) un selettore
  - Parametri in input: quelli richiesti per la computazione dei selectori
  - Parametri in output: l'**insieme di OID** del blocco di oggetti corrente
- **Block factor** = numero di oggetti visualizzati in blocco in un passo di navigazione
- Permette di definire il browsing in un insieme di oggetti:
- Visualizza link al primo, al precedente, al prossimo, all'ultimo oggetto nell'insieme

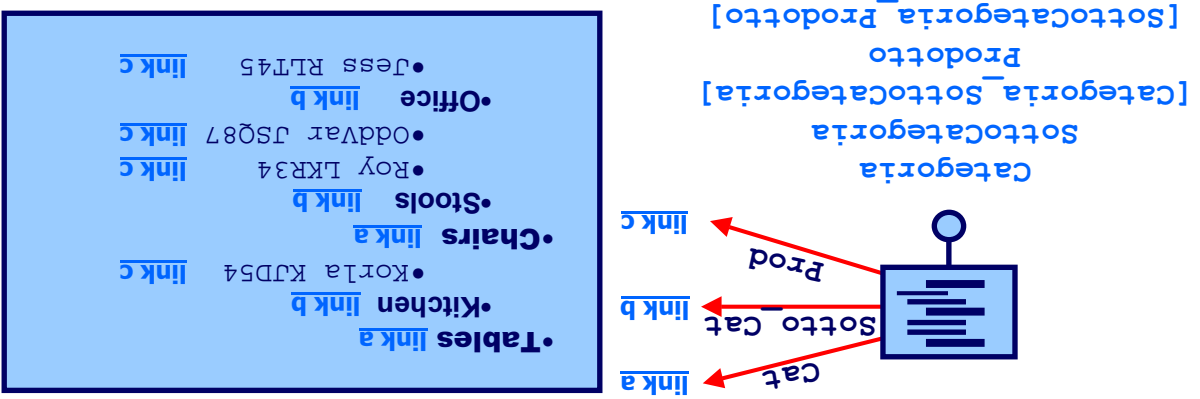
[Selettore (parametri) ]



## ScrollerUnit

MODELLO DI IPERTESTO

- Ogni link è visualizzato come un ancora ad un opportuno livello della gerarchia
- Il livello a cui il link è posizionato dipende dal tipo dei parametri sul link



## HierarchicalUnit: link uscenti

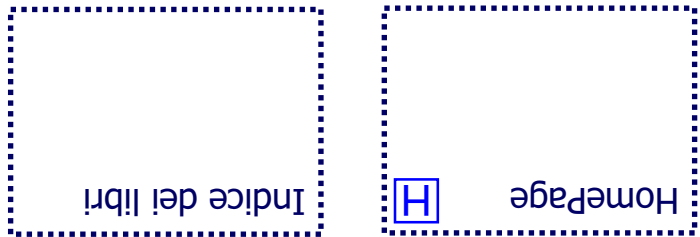
MODELLO DI IPERTESTO



- Una **pagina** è un contenitore di una o più unità di contenuto mostrate all'utente contemporaneamente

## Pagine

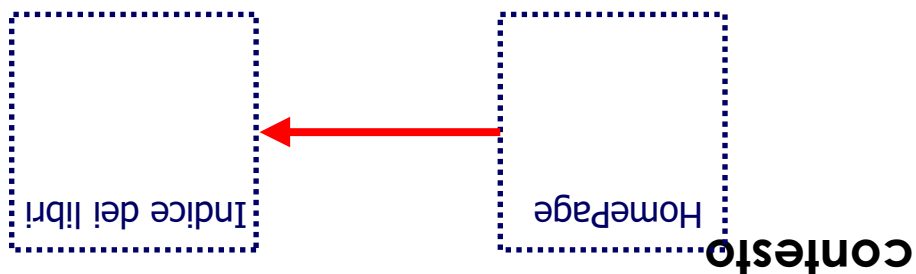
## 4. Pagine, Aree, Site view



- È la pagina principale di un sito
- La prima a cui l'utente accede
- Ogni site view deve contenere un pagina marcata come "Home"

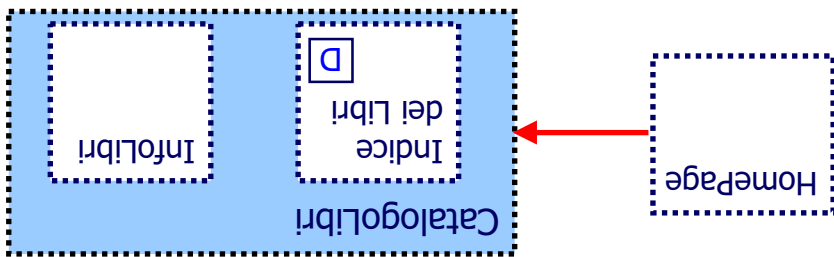
## Home Page

- L'utente naviga da una pagina all'altra per mezzo di un'ancora (es: [>>Indice dei Libri!](#))

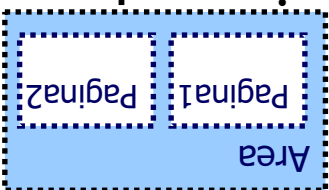


- Link definiti tra pagine
- Non trasportano alcuna informazione di contesto

## Link non contestuali

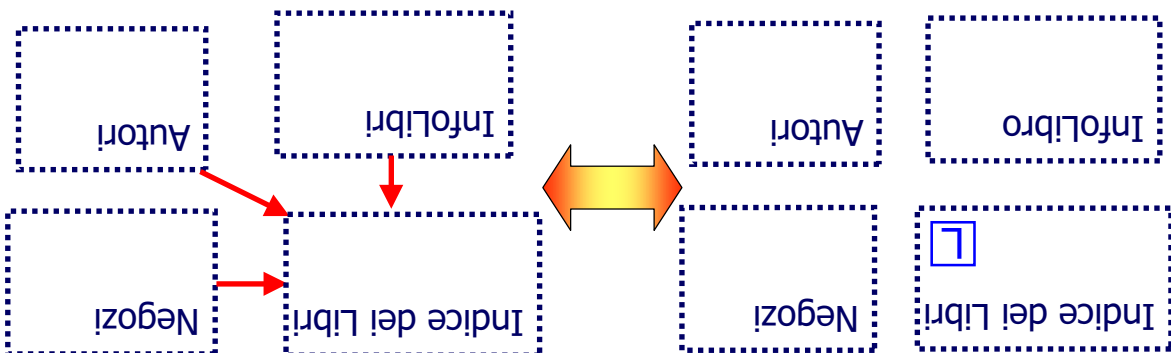


- **Insiami di pagine logicamente omogenee**
  - Sezioni di un portale: Sport, Musica, Tecnologia, ...
  - Le aree possono contenere sotto-aree annidate
  - Ogni area può includere una **pagina (o una sottopagina)** **di default**



## Areae

- **Pagine globalmente visibili.** L'utente può saltare ad esse da ogni altra pagina della site view
- **Equivalenti a link non contestuali definiti da ogni altra pagina della site view verso la pagina landmark**



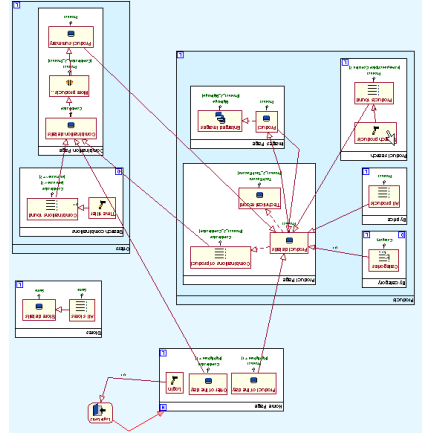
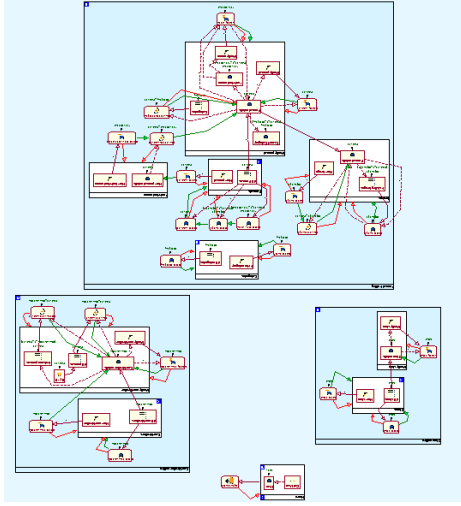
## Pagine Landmark

- Un insieme di pagine e/o aree che forniscono una vista coerente del sito
- Sullo stesso schema dei dati è possibile definire diverse site view
- Necessità di pubblicare ipertesti diversi per diversi tipi di utenti o per diversi tipi di dispositivi di output
- Es.:
  - Site view pubblica: accesso concesso ad ogni utente
  - Site view privata: accesso protetto tramite password

## Site View

## ACME Modellazione delle site view

- Due site view sullo stesso schema dei dati
- Customer: pubblica, per i clienti
- Admin: privata, per amministratori e gestori di contenuto

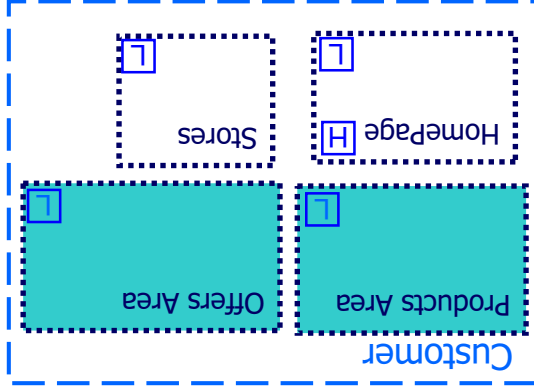


# ACME

## Site View Customer

- Scopo: permettere al cliente di accedere ai contenuti pubblicati tramite il sito
- Struttura:

- Due aree principali: Products e Offers
- Due pagine Landamark: HomePage, Store



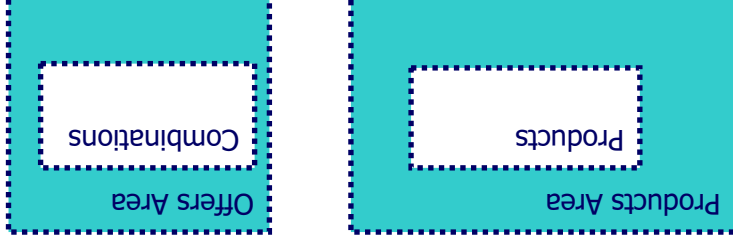
67

# ACME

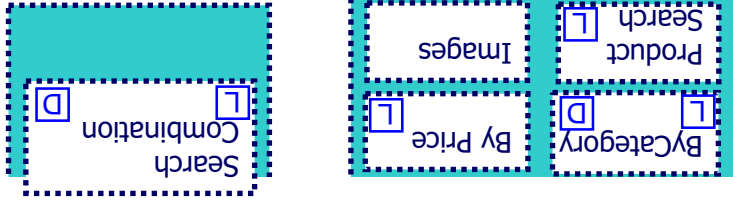
CASO DI STUDIO

## Organizzazione delle Aree

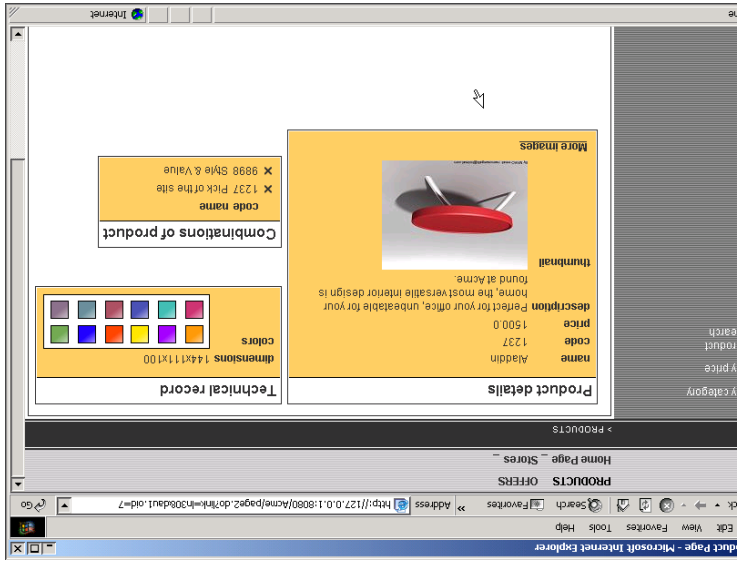
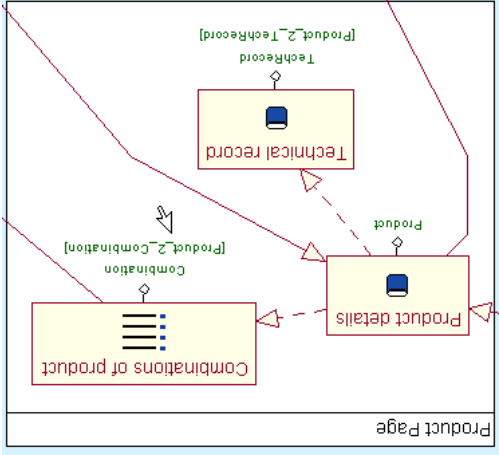
- D: quali sono le pagine principali?



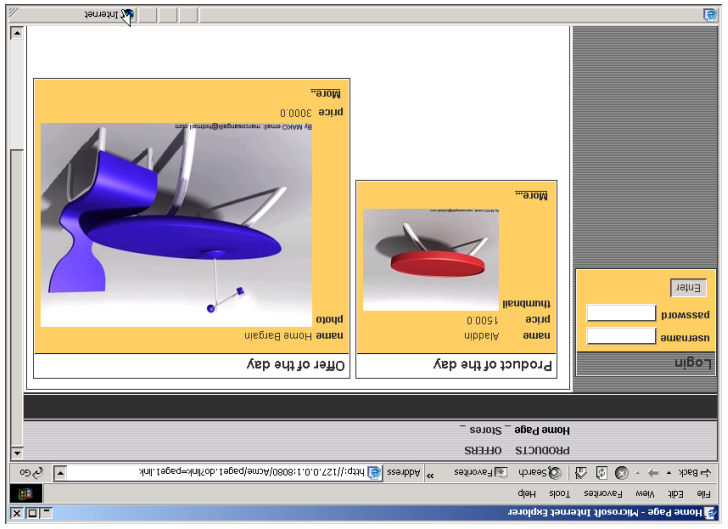
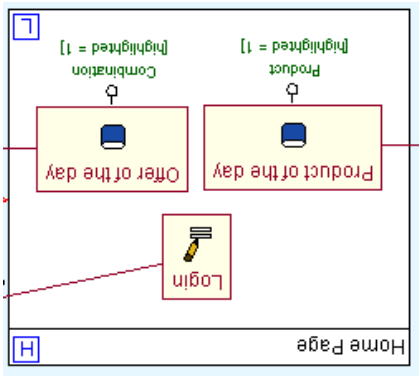
Q: quali altre pagine sono necessarie?



68

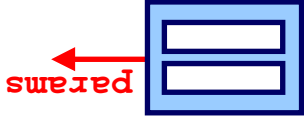


# ACME Pagina dei Prodotti



# ACME Composizione della Home Page

## Entry Unit



- Descrive moduli di immissione dati per l'invio di informazioni all'applicazione da parte dell'utente
- I contenuti sono forniti alle altre unit attraverso parametri sui link uscenti
- Corrispondono al costruito form HTML, descritto dal tag <form>, con relativo bottone di submit

72

MODELLO

## 5. Entry unit e ricerca

74

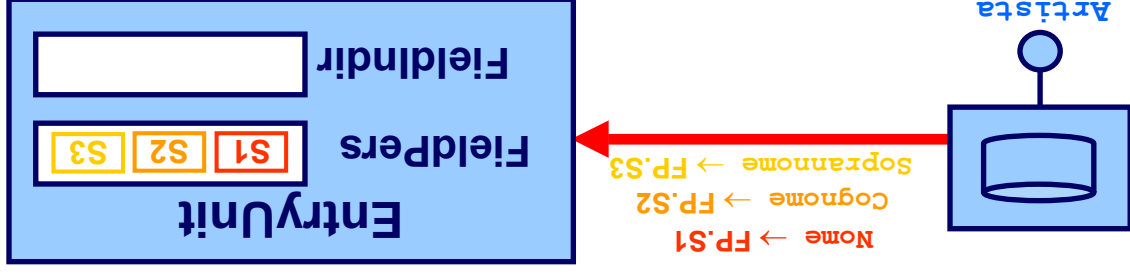
## Fields e Selection Fields

- Le entry units possono contenere 2 tipi di campi:
  - Fields per inserire nuovi valori
  - Selection fields per selezionare un valore da una lista
- I Fields hanno le seguenti proprietà:
  - Preloading: precaricamento di un valore iniziale nel field
  - Modifiability (Y/N): valore modificabile dall'utente o meno
  - Visibility (hidden/shown): nascosto o visibile
  - Data type (string, integer, text, etc.): tipo di dato
  - Validation rules: regole per la verifica di validità del dato fornito dall'utente

73

## Precaricamento di campi (Fields preloading)

- Un field può essere precaricato con uno o più valori:
  - Un field può essere associato a più slot
  - Gli slots consentono il concatenamento di più valori nello stesso field
  - E' possibile riempire ogni slot con un valore dinamico (preso dal DB) o statico



Bruce Springsteen "the boss"

Fields

Risultato

MODEL

MODEL

## 6. Parametri globali e personalizzazione

### Pre caricamento di campi (Selection Fields Preloading)

- Un selection field DEVF essere pre caricato con un insieme di istanze
- L'utente sceglie una tra queste
- Anche per i selection fields esiste il meccanismo degli slots per ogni istanza è possibile pre caricare uno o più campi
- Ogni slot può essere usato come:
  - **LABEL:** il valore caricato nello slot viene mostrato nell'elenco
  - **OUTPUT:** il valore caricato nello slot viene reso disponibile come parametro di output quando l'utente sceglie l'elemento
- **Esempio:** Selection Field Autore.
  - slots: Nome (Label), Cognome(Label,Output), Codice (Output)
  - menu a tendina con Nome e Cognome di autori; dell'autore scelto sono disponibili come parametri Cognome e Codice

## Informazioni globali

- **PROBLEMA:** Spesso è necessario:
  - impostare il valore di un parametro (ad esempio, nazione o lingua dell'utente)
  - Disporre di tale valore in tutte le pagine della siteview, senza doverlo trasportare come parametro attraverso i link
- **SOLUZIONE:**
  - Concetto di **PARAMETRO GLOBALE**, memorizzato nella sessione di navigazione dell'utente
  - Modalità di accesso in scrittura e in lettura a tali parametri globali
- Sostanzialmente: variabili che possono essere valorizzate o recuperate in qualsiasi momento

77

## Parametri globali (Global Parameters)

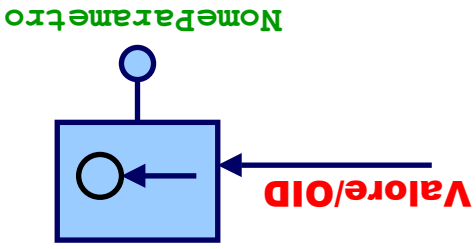
- Descrivono informazioni di interesse globale, per l'intera applicazione o per la sessione dell'utente corrente
- Proprietà di un parametro globale:
  - Name: nome
  - ID: identificativo univoco
  - Duration (User session or Application): durata
  - Value type: tipo di dato, che può essere:
    - Valore "stampabile" (integer, string, ...)
    - Un'entità (in tal caso, il parametro assumerà come valore un OID di questa entità)
  - Default value: valore statico di default (opzionale)

- **NOTA:** differenza nell'uso di link parameter e global parameter

78

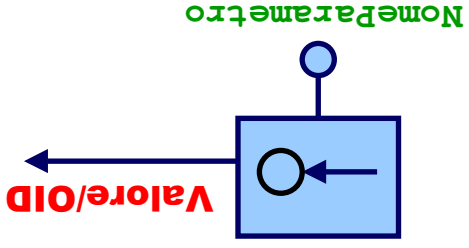
- E' sempre posizionata fuori da pagine
- Ha un solo link entrante
- Il valore da assegnare al parametro globale viene trasportato attraverso il link
- Non ha link uscenti!

- Permette di impostare il valore di un parametro



## Set unit

- E' sempre posizionata dentro una pagina
- Ha un solo link uscente
- Non ha link entranti
- Il valore recuperato viene fornito alle unit a valle attraverso il link



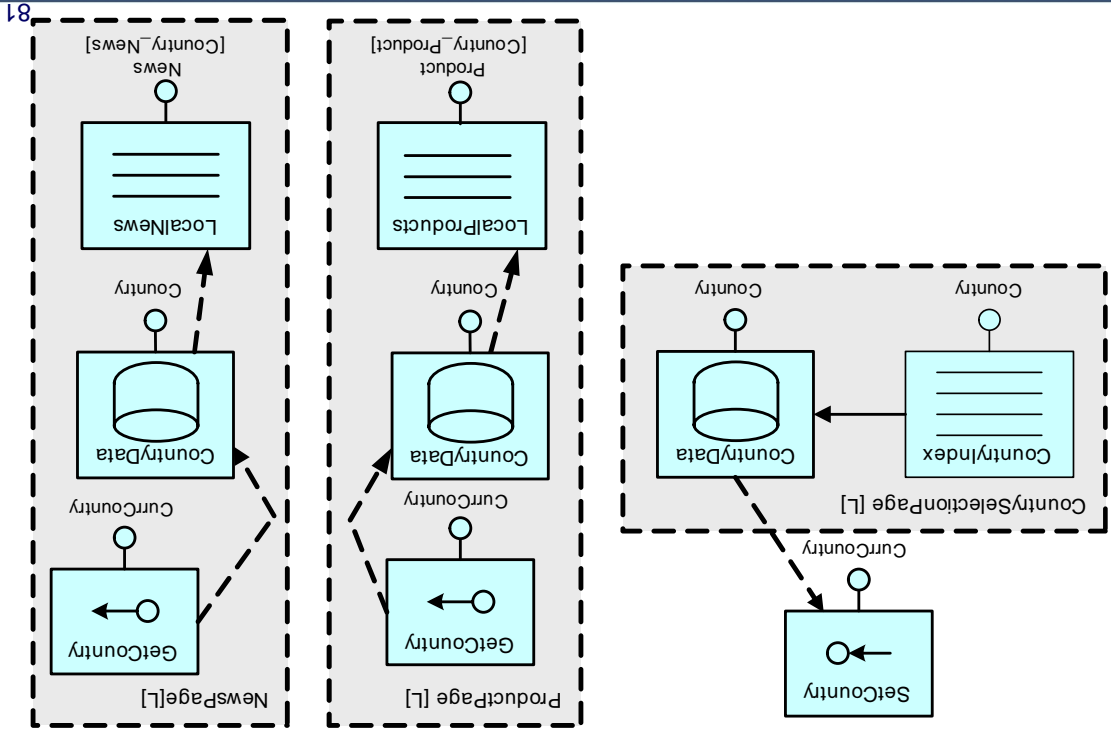
- Permette di recuperare il valore di un parametro globale

## Get unit

# Personalizzazione

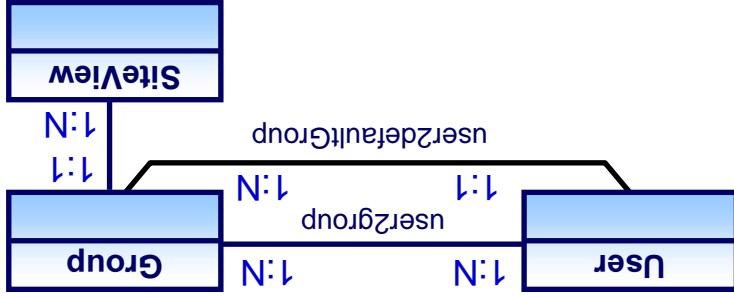
- **2 aspetti**
  - **Controllo degli accessi:** operazioni di login/logout per il riconoscimento di utenti.
  - **Accessibilità alle site-view basata sul gruppo di appartenenza**
    - ES: gruppo Admin accede alle siteview: PublicSV e AdminSV
    - **Personalizzazione dei contenuti:** pubblicazioni di contenuti dipendenti dall'utente o dal gruppo di appartenenza.
      - ES: libri preferiti dall'utente
      - ES: informazioni più rilevanti per un certo gruppo

# Esempio



## Modello utente/gruppo

- Entità predefinite nel modello dei dati per cui:
  - Ogni utente può appartenere ad uno o più gruppi
  - Ogni utente ha uno e un solo gruppo di default
  - Ogni gruppo è associato ad una siteview

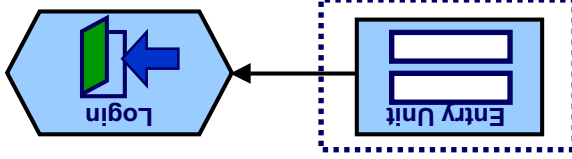


- Utente Everyone, Gruppo Everyone garantiscono l'accesso non registrato

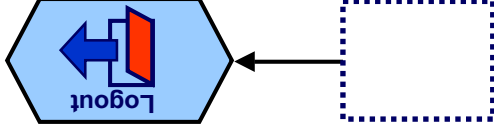
83

## Operazioni di Login/ Logout

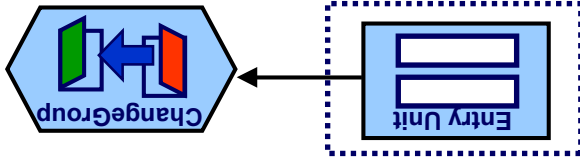
- Una siteview può contenere una form per consentire login



- Ogni siteview privata dovrebbe consentire logout



- E' consentito per un utente il cambio di ruolo durante la navigazione



84

## CurrentUser e CurrentGroup

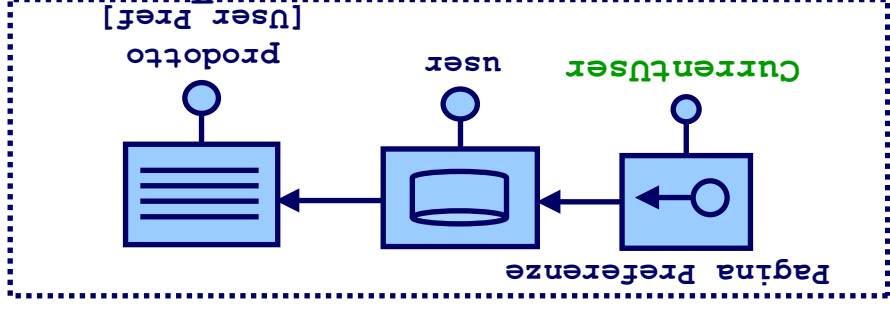
- WebML fornisce due global parameters predefiniti:
  - **CurrentUser**: OID dell'utente corrente
  - **CurrentGroup**: OID del gruppo di appartenenza dell'utente corrente
- L'utente deve avere fatto login (altrimenti è everyone)
- Le operazioni di Login, Logout e ChangeGroup impostano automaticamente questi due parametri

## Personalizzazione (user-level)

- Personalizzazione dei contenuti ottenibile attraverso appropriata progettazione dei dati



- L'ipertesto riflette i dati:



Dopo il login **CurrentUser** è identificato, quindi l'indice mostra i prodotti preferiti dall'utente

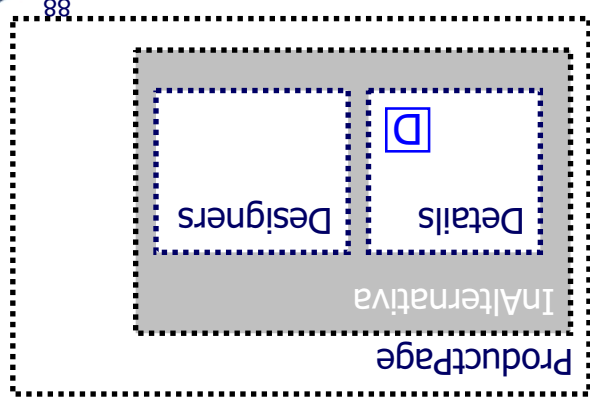
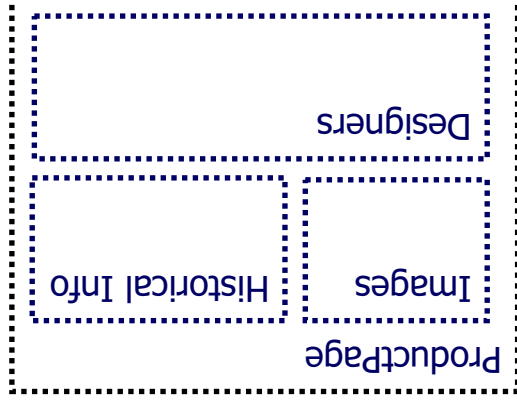
## 7. Pagine annidate e Patterns

87

## Pagine annidate

- E' possibile definire pagine annidate (sottopagine)
- Pagine annidate possono essere tra loro:

- **In AND** (presenza contemporanea nella pagina di tutte le sottopagine)
- **In OR**: Si inserisce un elemento ALTERNATIVE, in cui pagine figlie compaiono in mutua esclusione. Una di esse è marcata come Default

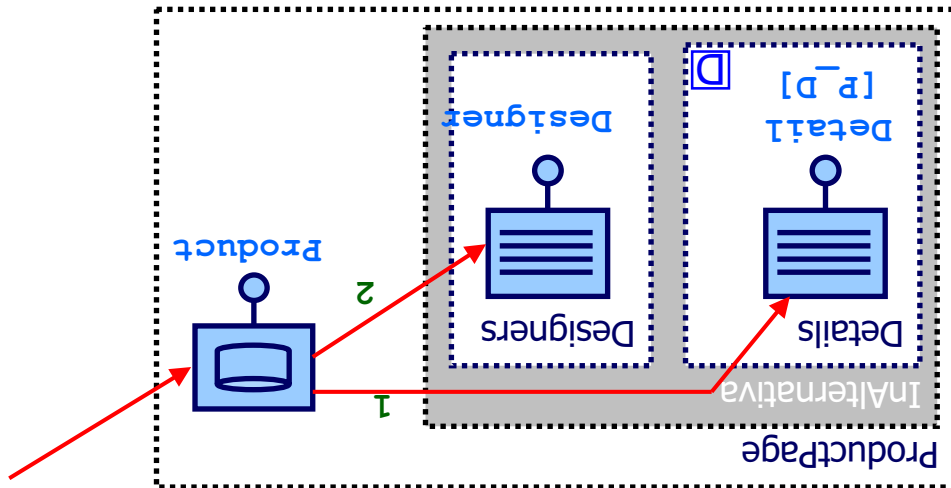


88

MODEL

- ## Pattern di ipertesto
- Dall'ingegneria del software:
  - Pattern = soluzione nota per problemi ricorrenti
  - Iper testo: strutture e soluzioni spesso ripetute
  - Ogni pattern è adatto per diversi contesti applicativi, ma per analoghi problemi e limitazioni
  - Vedremo 3 esempi di pattern per l'ipertesto

- NB. Ogni ipertesto può essere riportato a non avere sottopagine

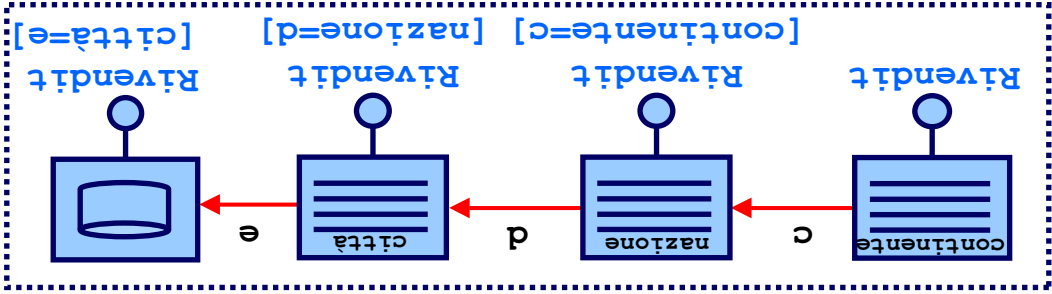


- Pagina di prodotto che mostra l'elenco dei dettagli oppure, a scelta dell'utente, l'elenco dei designer

## Esempio di pagine annidate in OR

## Patterns(1): indice multi-step

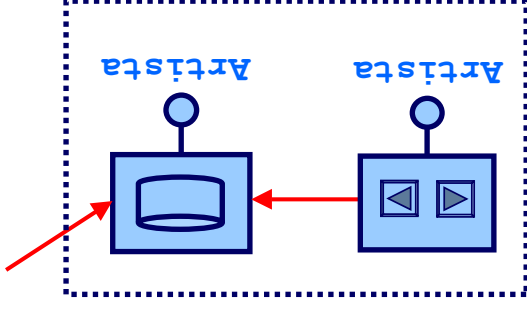
- Cascata di indici per restringere sempre più il sottoinsieme di ricerca:



- L'utente sceglie sequenzialmente un continente, una delle nazioni del continente scelto e una delle città della nazione scelta.
- Alla fine viene mostrato il rivenditore della città

91

## Patterns(2): Scroller + data unit



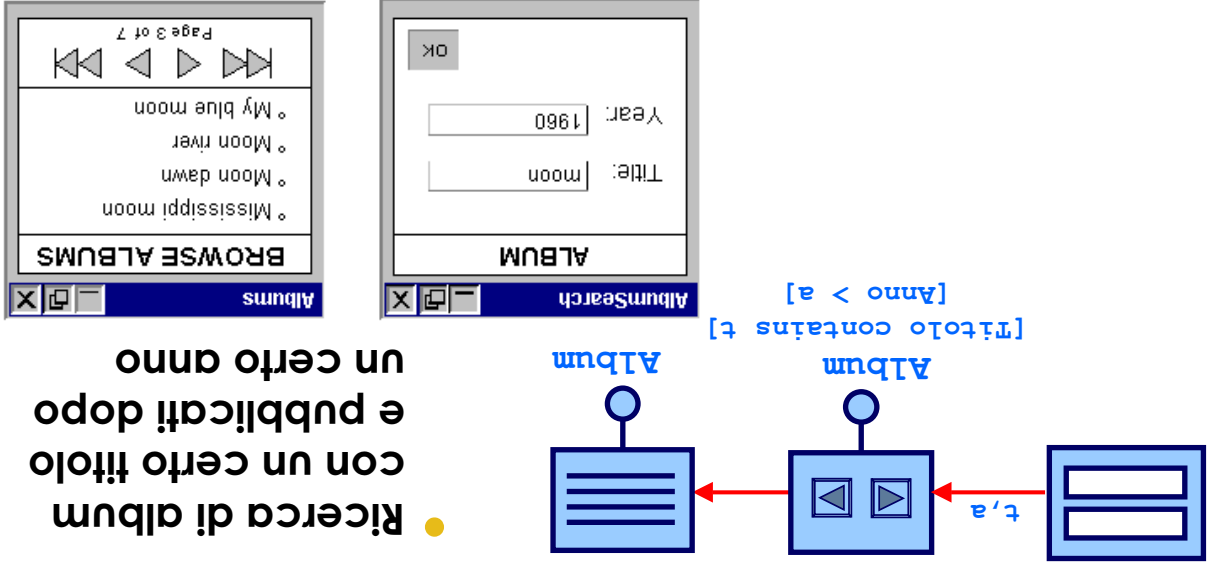
- Browsing tra le singole istanze dell'insieme
- Entità uguale per le due unit
- E' possibile arrivare alla data unit anche da un'altra pagina; la scroller unit è automaticamente sincronizzata
- Adatto per insiemi piccoli

92

# Patterns(3): Ricerca + Scroller + index

- Paginazione dei risultati di una ricerca

- Ricerca di album con un certo titolo e pubblicati dopo un certo anno



# Modello delle operazioni!

A cura del team WebML  
Politecnico di Milano

# Integrazione di ipertesti e logica applicativa

- Esempi di siti che fanno uso di servizi back-end o remoti:
  - Reservation Web site: richiesta di prenotazione, annullamento o modifica di prenotazione
  - Altri esempi: gestione contenuti, carrello della spesa di siti e-commerce, gestione del profilo personale, ...
- **PROBLEMA:**
  - Come modellare la chiamata di operazioni?
- **SOLUZIONE:**
  - Inserendo le chiamate di operazioni come nuove primitive di ipertesto

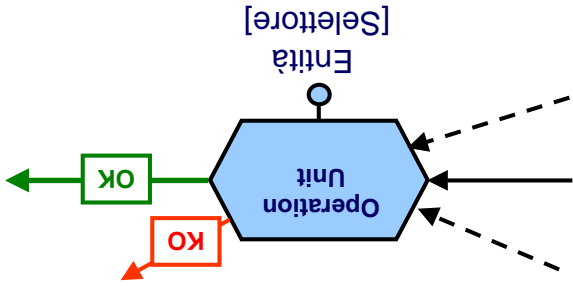
## 1. Operazioni!

## Operazioni predefinite (Built-in)

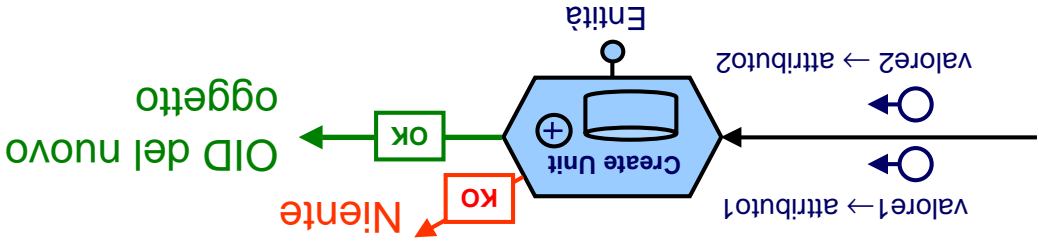
- WebML fornisce un insieme di operazioni predefinite per la gestione dei contenuti (istanze dei dati pubblicati sul sito)
- Operazioni tradizionali di gestione di basi di dati: create, delete, modify, create relationship, delete relationship
- Il loro comportamento è predefinito, e consente di gestire le istanze di entità e relazioni dichiarate nel modello dei dati
- E' possibile arricchire tale insieme con altre operazioni definite dall'utente

## Operation Unit

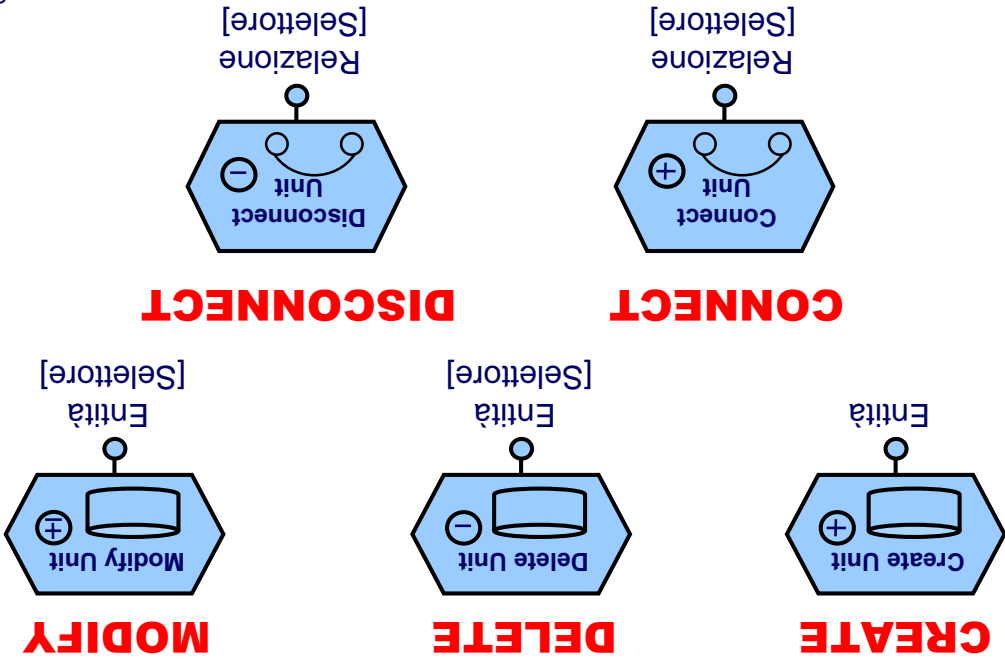
- Descrive una generica operazione esterna o predefinita
- Riceve input da uno o più link entranti (di cui uno deve essere un **normal link**, mentre gli altri saranno **transport links**)
- Due tipi di link di uscita:
  - **OK link**: seguito nel caso in cui l'operazione si conclude con successo
  - **KO link**: seguito nel caso in cui l'operazione fallisce
- Le operazioni non hanno scopo di pubblicare informazione → sono posizionate all'esterno delle pagine



- Riceve i valori da assegnare agli attributi del nuovo oggetto creato
- Restituisce l'identificativo dell'oggetto creato (se la creazione ha successo)

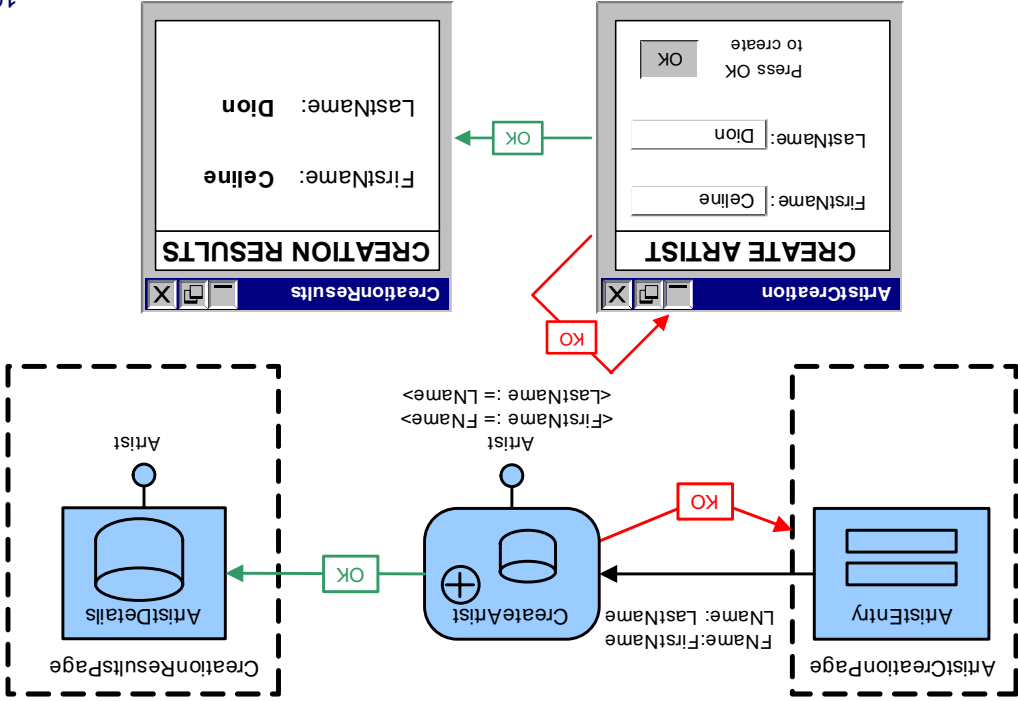


## Create Unit



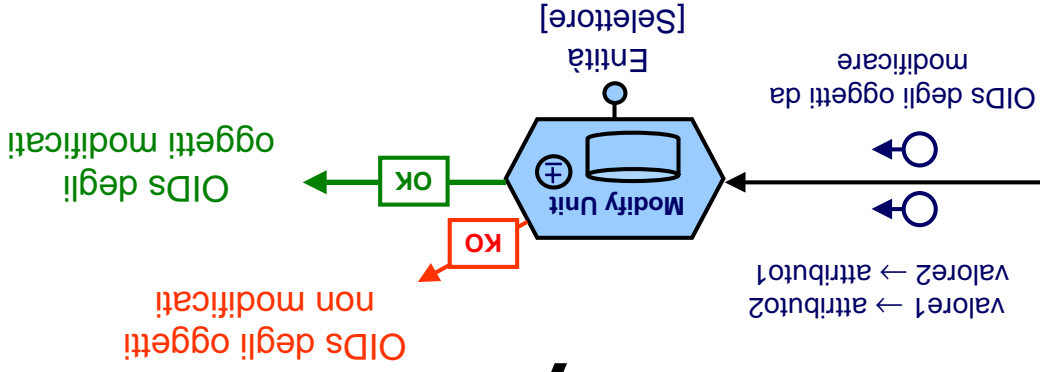
## Operazioni predefinite per la gestione dei contenuti

# Esempio di creazione



101

# Modify unit

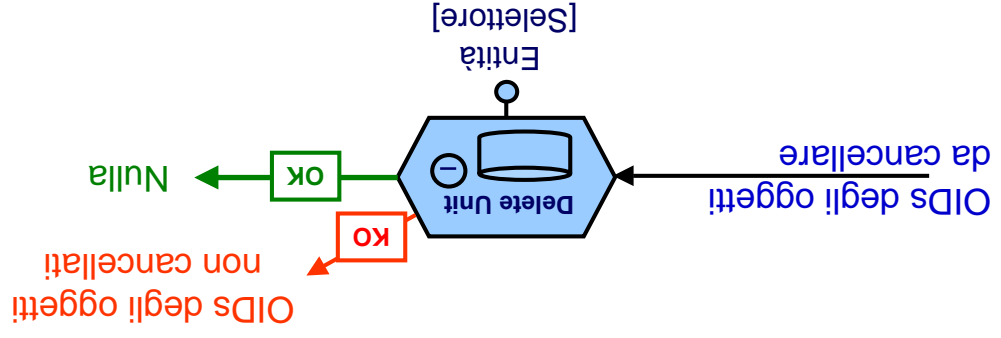


102

- Riceve i valori da assegnare agli attributi da modificare
- Nota: le istanze da modificare possono essere selezionate per mezzo di OID o per mezzo di selettore
- OK: Restituisce l'identificativo degli oggetti modificati (tutti quelli che erano stati richiesti)
- KO: Restituisce l'identificativo degli oggetti che non si è riusciti a modificare

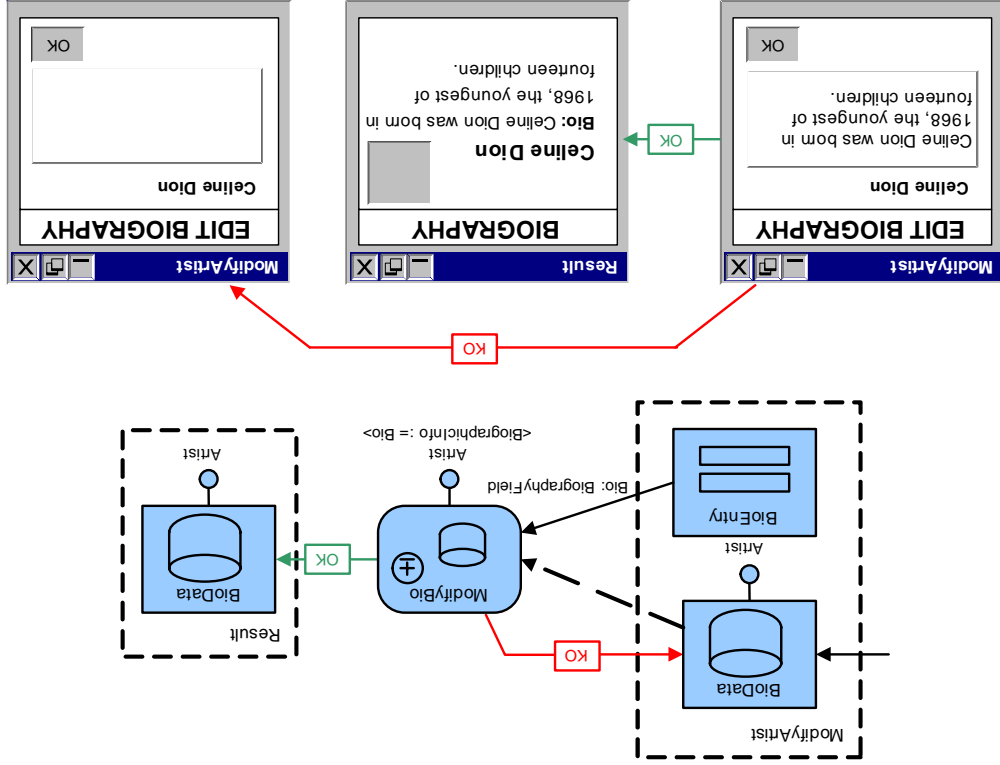
MODEL

- Riceve l'elenco degli oggetti da cancellare
- Nota: le istanze da cancellare possono essere selezionate per mezzo di OID o per mezzo di selettore
- OK: non restituisce nulla perché gli oggetti non esistono più
- KO: Restituisce l'identificativo degli oggetti non cancellati

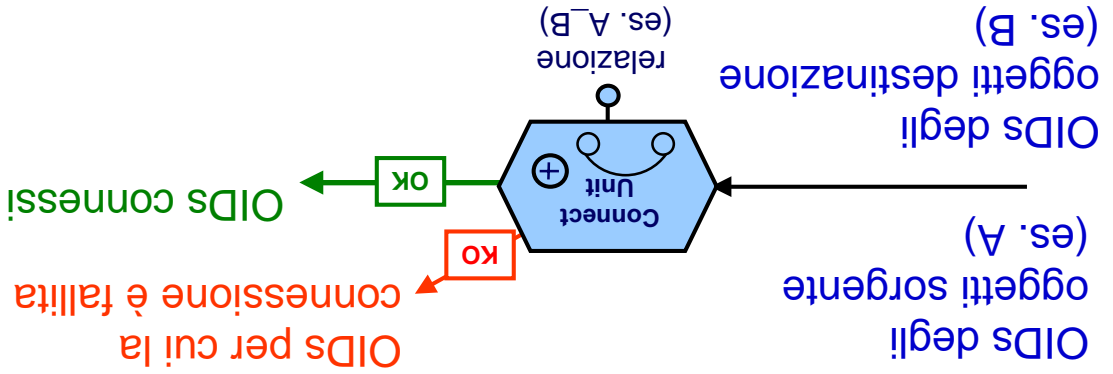


## Delete unit

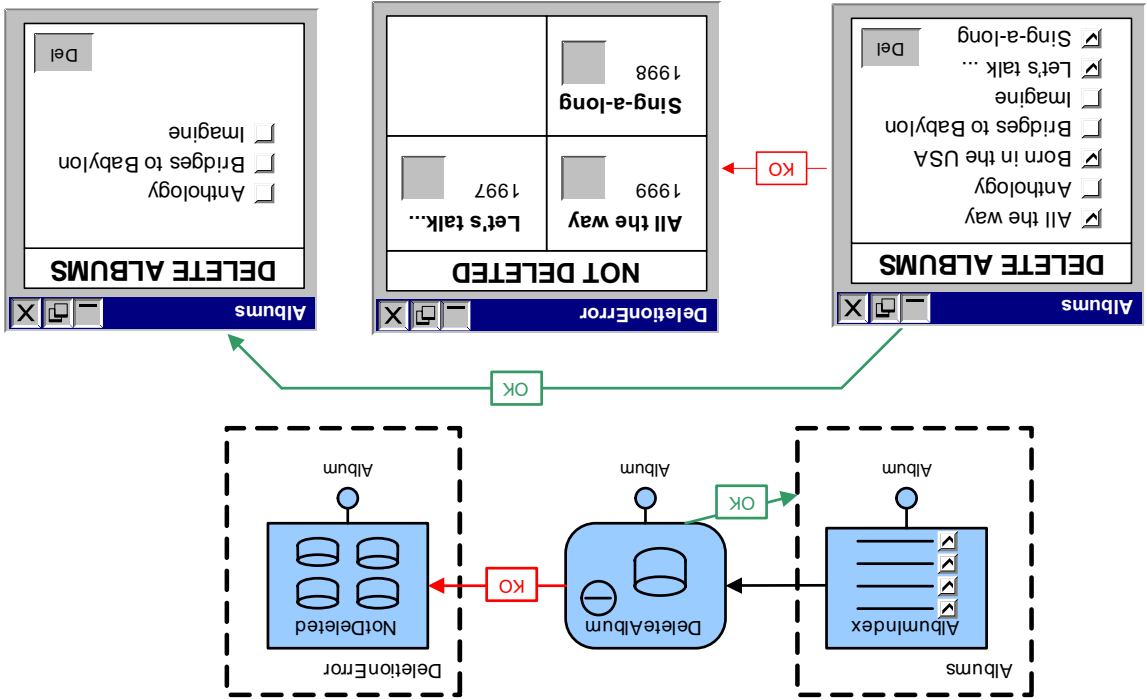
## Esempio di modifica



- Riceve l'elenco degli oggetti da collegare (sorgente e destinazione della relazione)
- OK: restituisce gli oggetti collegati
- KO: restituisce gli oggetti non collegati
- Se uno dei parametri di input è un insieme, tutti gli oggetti vengono collegati



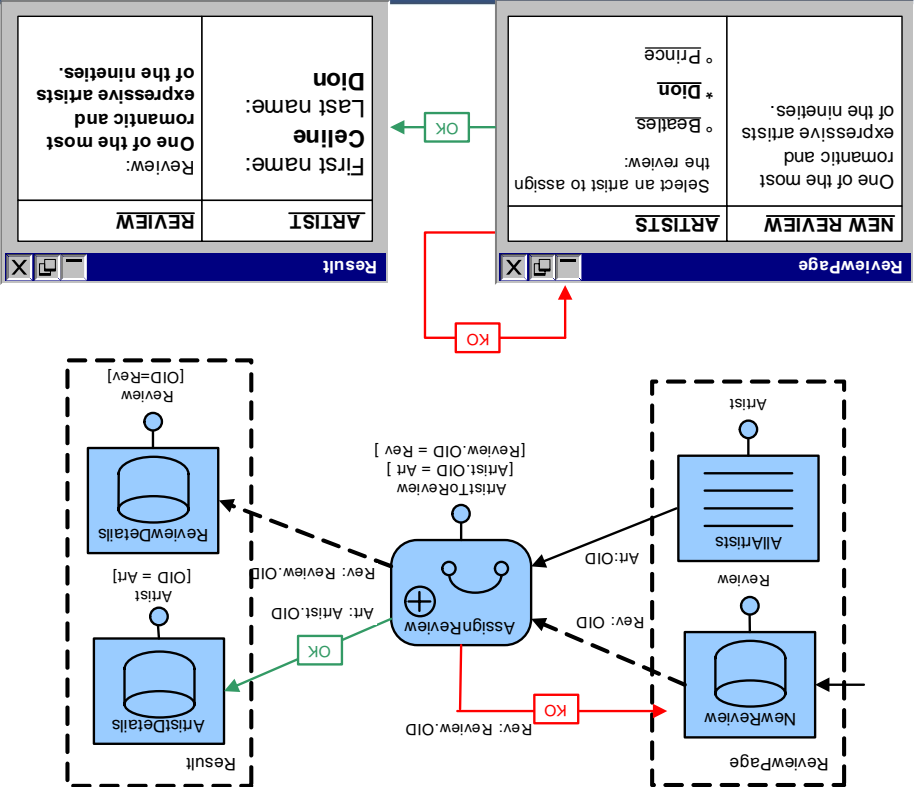
# Connect unit

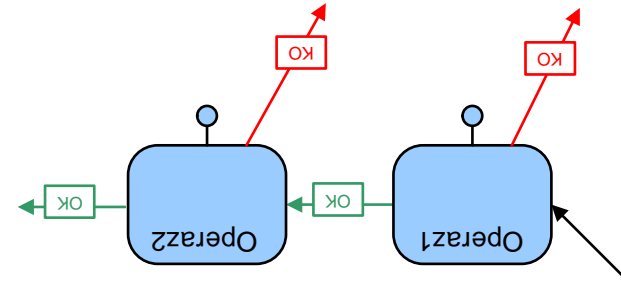


# Esempio di cancellazione

## 2. Operazioni: aspetti avanzati ed esempi!

### Esempio di connect





- E' possibile definire catene di operazioni da eseguire una dopo l'altra
- Esempio: creazione di un Album e collegamento con l'autore (un artista esistente)
- Collegamento diretto attraverso OK-link

## Catene di operazioni

110

MODEL

- E' possibile arricchire l'insieme di operazioni predefinite con altre operazioni definite dall'utente
- Esempi:
  - SendMail operation
  - VerifyCreditCard operation
  - ...
- Per ogni operazione l'utente deve fornire:
  - interfaccia di input / output (parametri attesi in ingresso e forniti in uscita)
  - Descrizione dell'operazione (icona, eventuali proprietà, ...)
  - implementazione
- Nota: anche login e logout sono operazioni!!!

## Operazioni esterne (custom operations)

109

MODEL

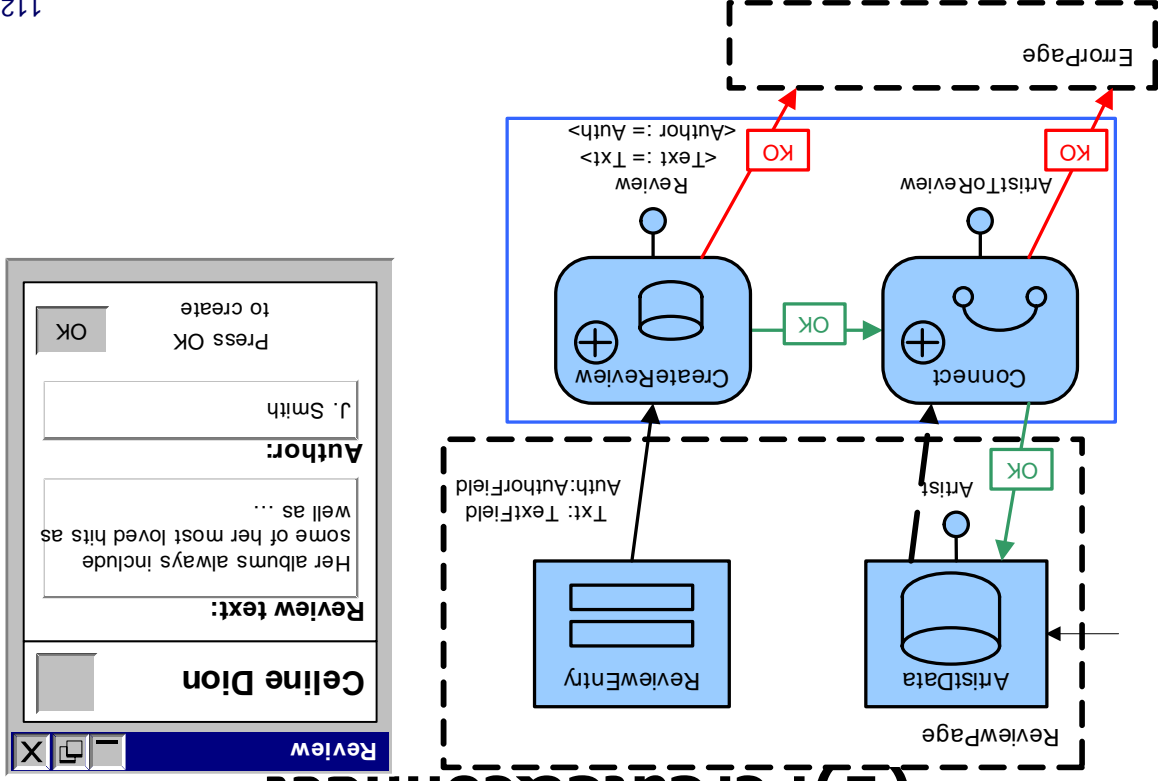
# Transazioni

- **PROBLEMA:** garantire che una catena di operazioni non venga mai eseguita in modo parziale
- **SOLUZIONE:** introduzione del concetto di **TRANSAZIONE**
- **TRANSAZIONE** = sequenza di operazioni eseguita in modo atomico, cioè in modo che o tutte le operazioni sono eseguite con successo, o l'intera sequenza di operazioni viene annullata.
- (concetto di basi di dati, per cui valgono le proprietà **ACID**)
- **Rappresentazione:** un rettangolo attorno alla catena atomica di operazioni (con eventuale KO link)

111

MODEL

# Pattern di content management (1): create&connect



112

MODEL

