

# UML & WEB

## - Approccio WAE: Conallen -

Henry Muccini  
Università degli Studi dell'Aquila  
muccini@di.univaq.it



Engineering | IgTechnology | Imola  
Informatica | Maggioli Informatica |  
Micron Technology | Neta | Nous  
Informatica | ObjectWay SED |  
TechnoLabs | Taipora

## Master in Web Technology

IV Edizione 2007/08

Dipartimento di Informatica

Università degli Studi dell'Aquila

## Copyright Notice

» Il materiale riportato in queste slide puo' essere riutilizzato, parziale o totalmente, a patto che le fonti e gli autori vengano citati

Henry Muccini



## Conallen

- » Conallen presenta un processo di sviluppo di applicazioni Web basato su
  - > RUP e ICONIX
  - > Architecture Centric
- » Vedremo nel seguito una
  - > introduzione al RUP
  - > una descrizione di cosa sia una Architettura per applicazioni Web
  - > Approccio di Conallen per la modellazione di applicazioni Web tramite UML
    - Uso dei Meccanismi di estensione di UML
  - > Tool support
  - > Esempio



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Approccio di Conallen per il Web



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

# Conallen

» Considera tutto il processo:

- > Requisiti
- > Analisi
- > Design
- > Implementazione

» Implementa il RUP per applicazioni Web



## Conallen

### Core Workflows

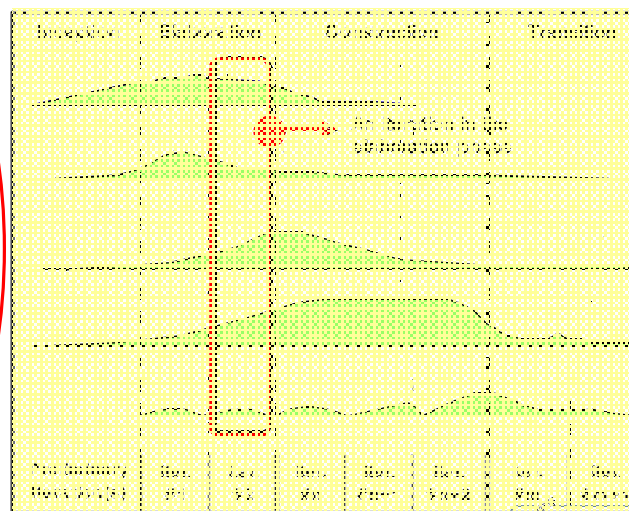
Requirements

Analysis

Design

Implementation

Test



Iterations



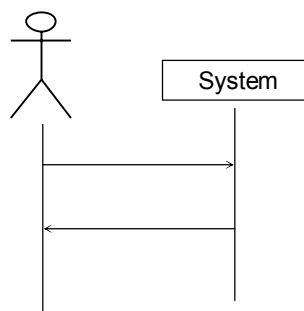
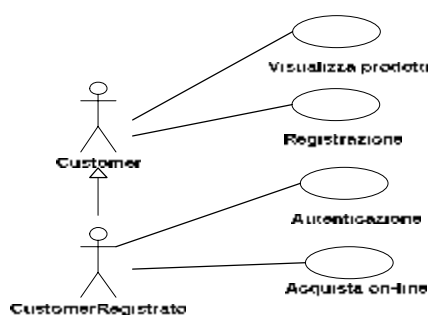
## Requisiti

- » Requisiti funzionali di applicazioni Web sono catturati e modellati tramite Use Cases;
- » Come nello sviluppo tradizionale, Use Case D. hanno l'obiettivo di presentare graficamente ed esplicitare tramite templates, gli obiettivi funzionali da raggiungere
- » Use Case diagram sono usati per guidare i seguenti passi nel processo di sviluppo Web
  - > **Web actors identification**
    - WebAdministrator, Browser, OnlineCustomer, Registered (unregistered) User
  - > **Web-related Use cases**
- » **Uso di Sequence Diagram**
  - > Ad altissimo livello



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Requisiti



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

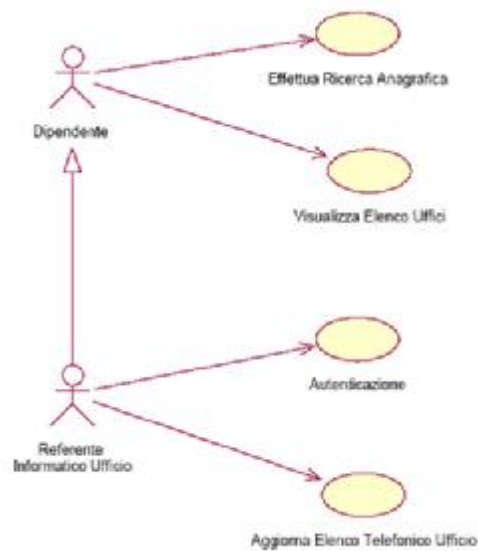
## Perche' usare Use Case diagrams?

- » Gli use case permettono di **separare la "logica" dall'implementazione**:
  - > lo stesso Use case puo' essere implementato in diversi modi...
- » Uno use case che descrive il **browsing** del catalogo di un negozio potra' infatti essere implementato come:
  - > **una applicazione client/server distribuita**
  - > **come un applicativo web**

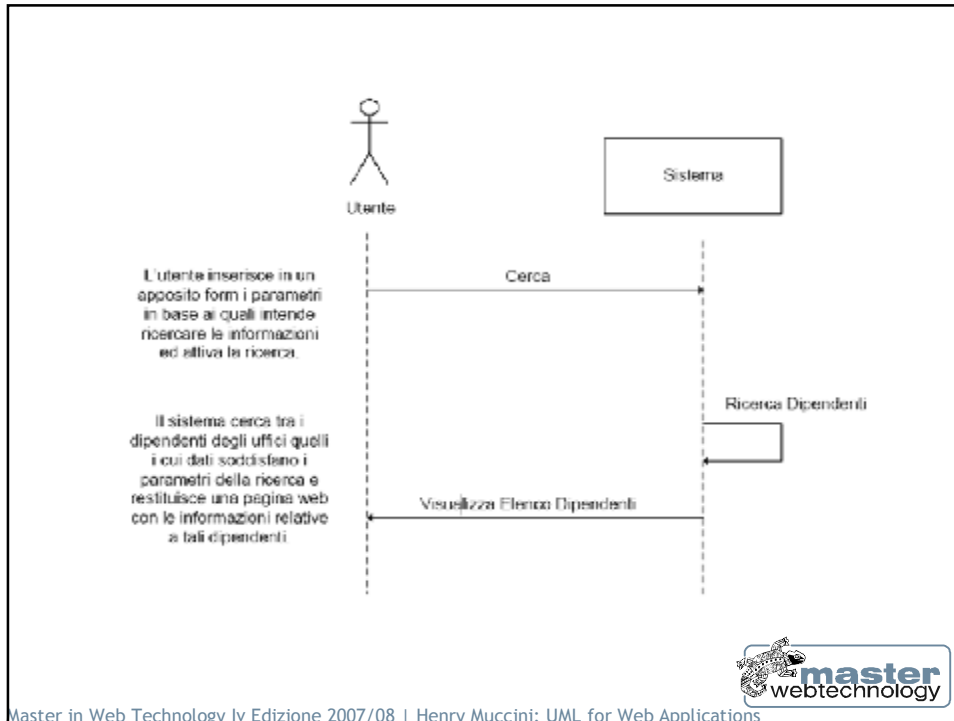


Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Piccolo Esempio



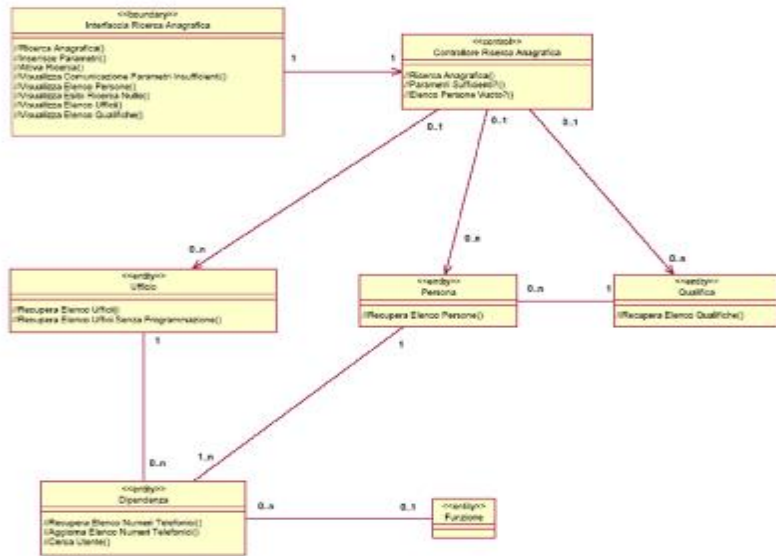
Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications



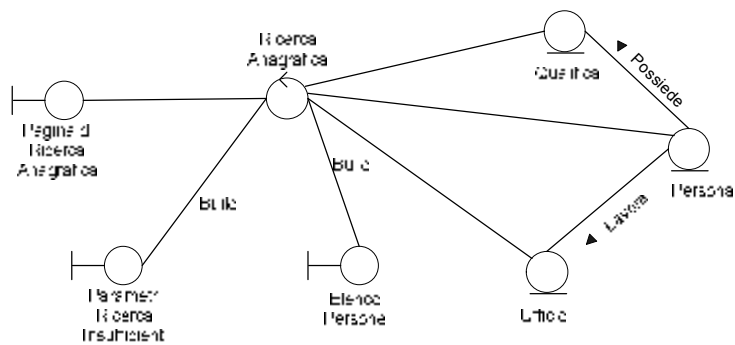
## Analysis

- » L'attività di analisi consente di trasformare i requisiti del sistema in un progetto
- » Gli oggetti dell'analisi degli use case sono:
  - > Identificare le **classi** e gli oggetti che esibiscono il comportamento descritto negli use case.
  - > Identificare le **responsabilità**, gli **attributi** e le **associazioni** delle classi.
  - > Rilevare l'utilizzo dei **meccanismi architetturali**
- » A questo livello si identificano delle "analysis class", che sono delle classi concettuali astratte.

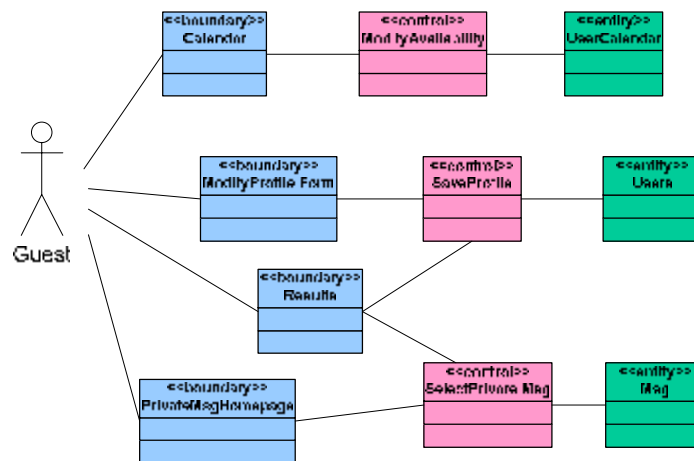
# Look and Feel



# Look and Feel



## Look and Feel



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

» Le analysis class possono essere stereotipate in tre tipi: boundary, entity e controller



> **Boundary object:**

- rappresentano l'**interfaccia** tra l'attore ed il sistema.
- Istanze di questi oggetti sono tipicamente **maschere di input** o controlli all'interfaccia utente.
- Nelle applicazioni Web, i boundary object possono anche rappresentare delle **interne pagine Web**.



> **Entity object:**

- esempi di entity object, le cui istanze possono apparire in più invocazioni di use case, sono **ordini**, **clienti**, **prodotti**, ecc.
- Nelle applicazioni Web, rappresentano i **dati**



> **Controller object:**




- rappresentano **processi**.
- Questi oggetti rappresentano delle **attività di sistema** alle quali può essere attribuito un nome.
- I controller object tipicamente **dirigono** le attività degli entity e dei boundary object.



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

# Analysis

## » Elementi dell'Analysis model:

- > Boundary: interfaccia grafica 
- > Entity: dati 
- > Control: attivita' 

## » Rules:

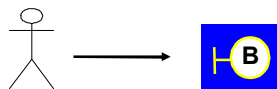
R1: Actors interagiscono solo con boundary obj

R2: Entity interagiscono solo con controller obj

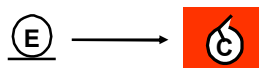
R3: Controller interagiscono con tutti



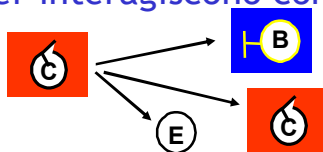
## » R1: Actors interagiscono solo con boundary obj



## » R2: Entity interagiscono solo con controller obj



## » R3: Controller interagiscono con tutti

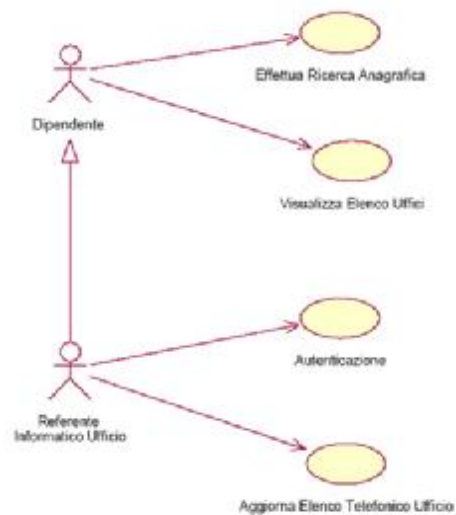


## Dai Requisiti all'Analysis

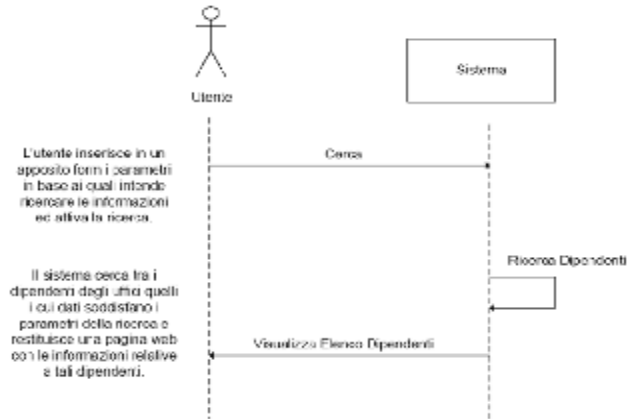
- » Per passare dai Requisiti all'Analysis, si seguono i seguenti passi:
  - > Per ciascun Use Case, si identificano le Analysis class che le possono realizzare
    - Per ciascuna classe, si identifica il tipo (boundary, control, entity)
  - > I sequence ad altissimo livello realizzati prima, vengono trasformati in sequence dettagliati



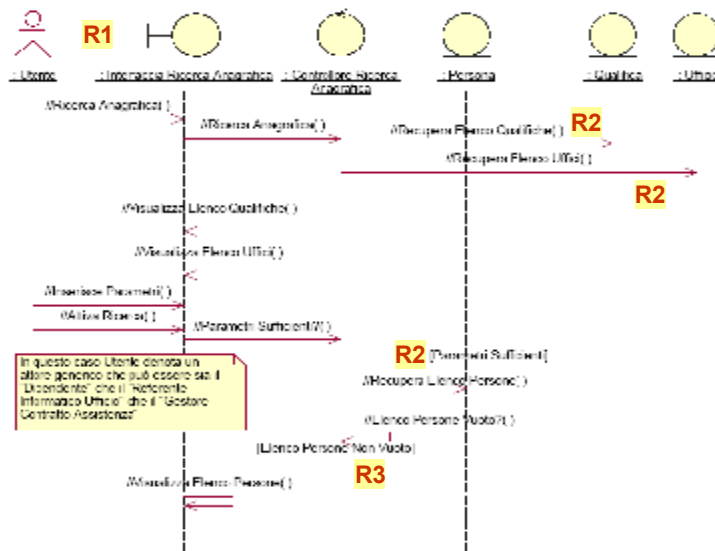
## Dagli Use Case alle Analysis class

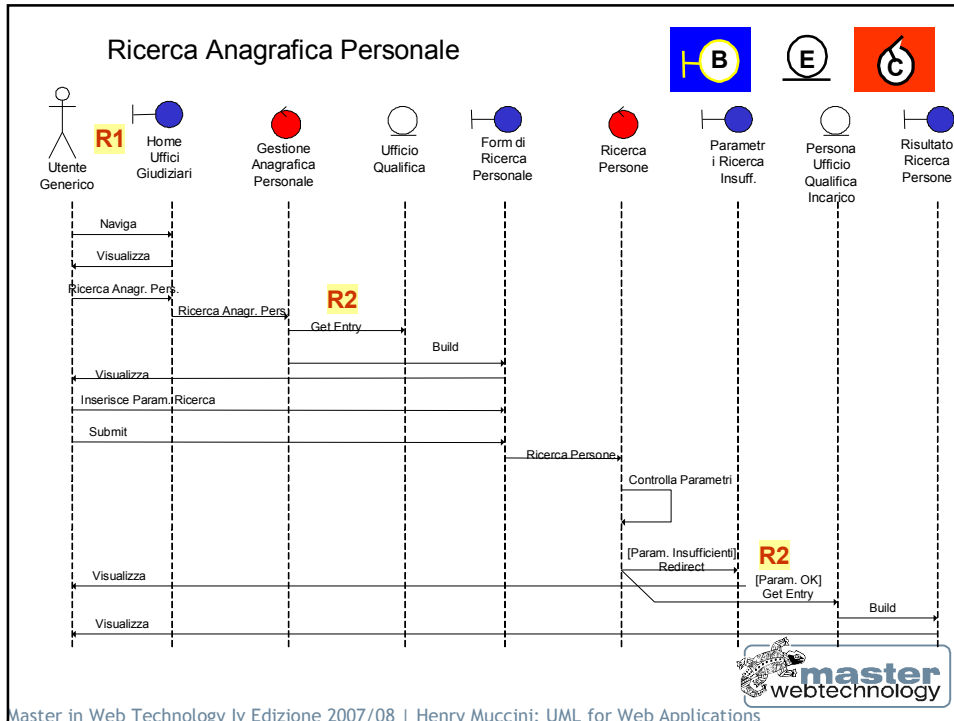


# Dai Sequence dei Requisiti ai Sequence di Analysis



- R1: Actors interagiscono solo con boundary obj
- R2: Entity interagiscono solo con controller obj
- R3: Controller interagiscono con tutti





## Altri diagrammi nella fase Analysis

- » Altri diagrammi:
  - > Packages
  - > Sequence diagram
  - > Activity diagram



## Lessons Learned

- » Analysis model classes can be elicited starting from:
  - > Single Use Cases
    - Merge is required
  - > Use Case Diagrams
- » **Traceability** among Use Cases and Analysis classes is a must
- » The **boundary-control-entity pattern** provides a very useful schema for modeling modern Web applications (E.g., MVC patterns)

## Esempio da fare in classe

- » Alumni
- » **Agenda di eventi via Web:**
  - > consente di selezionare una data in una pagina Web. L'applicativo Web visualizza, in un'apposita finestra della pagina, l'elenco delle attività programmate per quella data.
  - > La funzionalità può essere implementata attraverso un frameset, contenente una client page a partire dalla quale può essere selezionata la data ed un target nel quale verrà visualizzato l'elenco delle attività programmate per la data selezionata.
  - > Il meccanismo di scelta della data è realizzato attraverso il client object Calendario (implementato per mezzo di un ActiveX control), contenuto nella client page.
  - > Ogni volta che l'utente seleziona una data, l'elenco delle attività visualizzate nel target viene aggiornato, attraverso un link ad un'apposita server page incaricata di creare la pagina con il dettaglio delle attività.





## Design

- » Passi:
  - > La definizione delle interfacce utente, o **pagine Web**
  - > La suddivisione degli oggetti in **tier**, come client tier, server tier e così via.
- » Per poter partizionare gli oggetti nei vari tier, bisogna sapere quali tier si hanno a disposizione. Questo dipende dalle specifiche dell'architettura
- » Tutti quanti i pattern architetturali delle applicazioni Web prevedono l'uso delle **pagine Web**



## Pagine Web

- » Il primo problema di modellazione che si presenta è come modellare pagine Web:
  - > pagina Web **puo' contenere degli script** che devono essere **eseguiti sul server**.
  - > Per complicare ulteriormente le faccende, la stessa pagina può contenere una serie di **script che vengono eseguiti sul client**.
- » I modelli messi a disposizione da UML, da soli, non sono sufficienti ad esprimere tutte le sottigliezze che caratterizzano una pagina Web contenente degli script, per consentirne la rappresentazione in un class diagram.



## Pagina Client e Pagina Server

» **Nota:** Il comportamento di una pagina Web sul Server e' differente dal comportamento della stessa sul Client:

- > Sul Server la pagina deve accedere le risorse del Server (dbase, file system, ...)
- > Sul Client, la pagina deve essere invece collegata al browser, alle Applet, ActiveX controls, ...



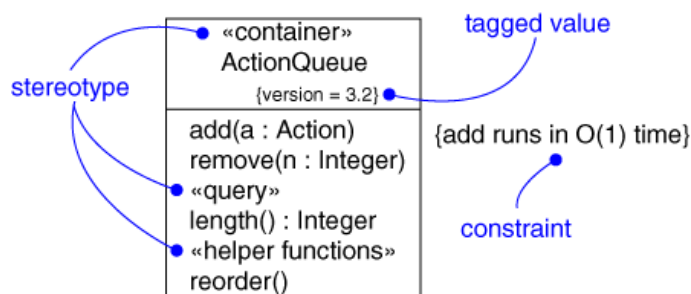
» Modelliamo gli aspetti Server-side e Client-side come elementi diversi, da linkare



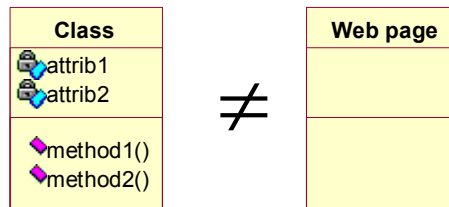
## Soluzione: Estendere UML

### Uml Extension Mechanisms

- Object Constraint Language
- Tagged Value
- Stereotypes



## Perche' estendere UML?



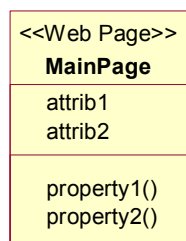
» Tali differenze sono importantissime quando:

- > Si usano **tool di modellazione**
  - **Controllori automatici** della sintassi dei modelli
- > Si usano **tool di analisi**
  - Significati diversi



## Idea

» Creare uno stereotipo che ci definisce quali siano i campi caratterizzanti una pagina Web, le sue proprietà, i suoi attributi, i suoi modi di interagire con altre pagine Web



» A seconda dell'architettura scelta, una Web page puo' avere delle diverse proprietà



## Cominciamo...

- » Vediamo ora come Conallen estende UML classico per modellare i tre tipi di architettura
  - > Thin Client
  - > Thick Client
  - > Web Delivery
  
- » Cominciamo con l'architettura Thin Client

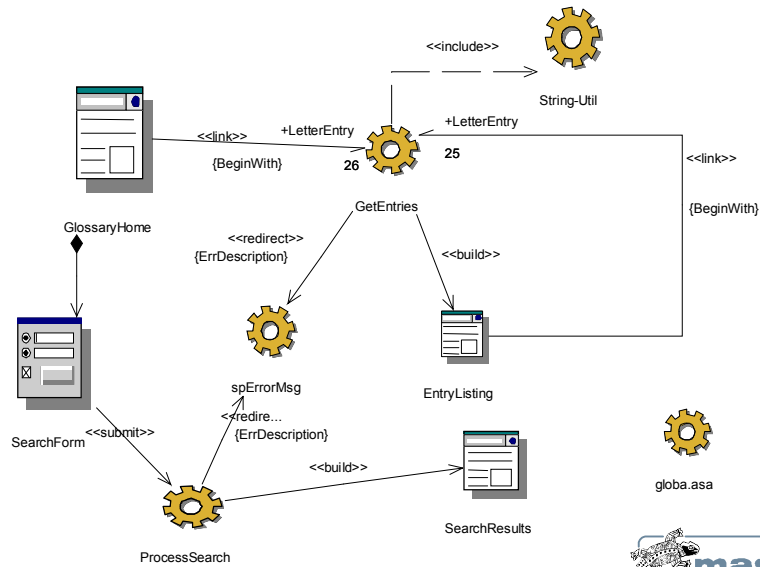


## Cosa otterremo

- » Aprire il file [16\\_Glossary.mdl](#)

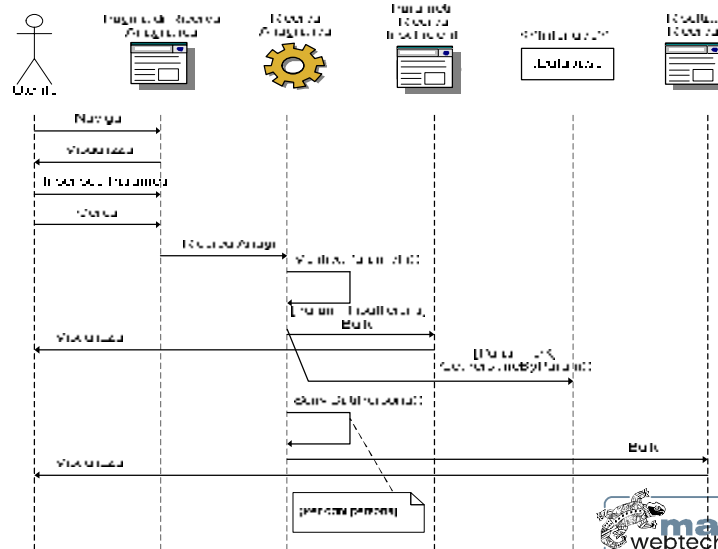


## Class Diagram - Design level



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Sequence Diagram - Design level



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Architettura Thin Client con Conallen

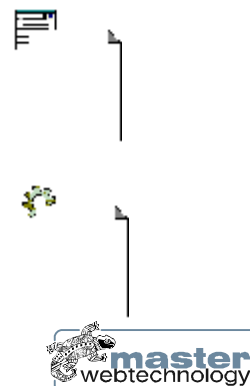
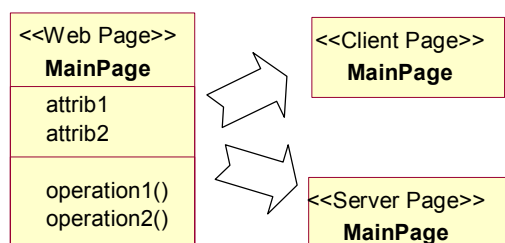
» In questa architettura, dobbiamo distinguere tra due diverse tipologie di pagine Web:

- > Client page, che possono interagire con dei **form**, per la raccolta di dati di input da inviare al server
- > Server page, che provvedono ad **elaborare i dati** (interagendo con altre risorse server-side) e costruiscono dinamicamente delle pagine con i risultati delle elaborazioni,
  - pagine che verranno inviate al client per la visualizzazione.



## Client e Server page

» Dobbiamo iniziare con il fare una distinzione tra “Client page” e “Server page”



## Da Analysis model a Design model

» Per passare da elementi dell'Analysis Model a quelli del Design model ci si può aiutare seguendo tali direttive:

- > **trasformare i boundary object in delle client page**
- > **trasformare i controller object in delle server page**
- > **gli entity object invece, verranno probabilmente mappati in una serie di server component**

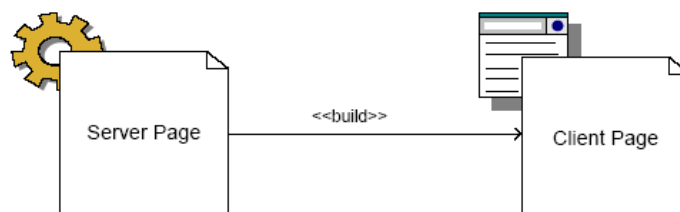


Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Relazioni tra Client e Server page

» Relazione fondamentale: **<<build>>**

- > La pagina Server, dopo avere elaborato i dati, esegue un build della pagina Client
- > E' una relazione **unidirezionale**

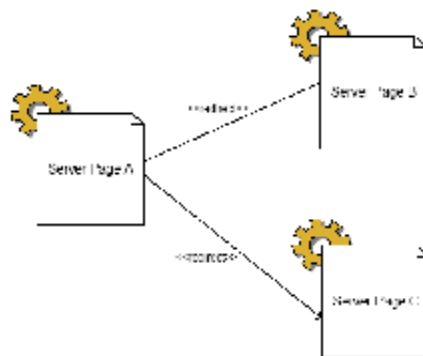


Master

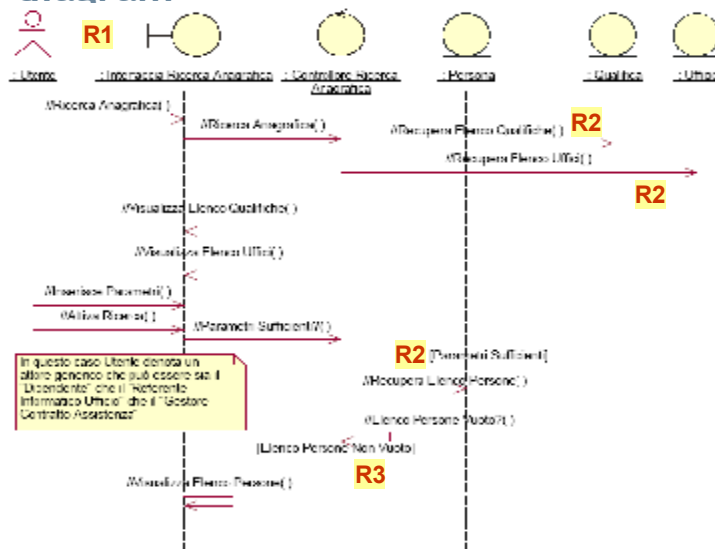
# Relazione di <<redirect>>

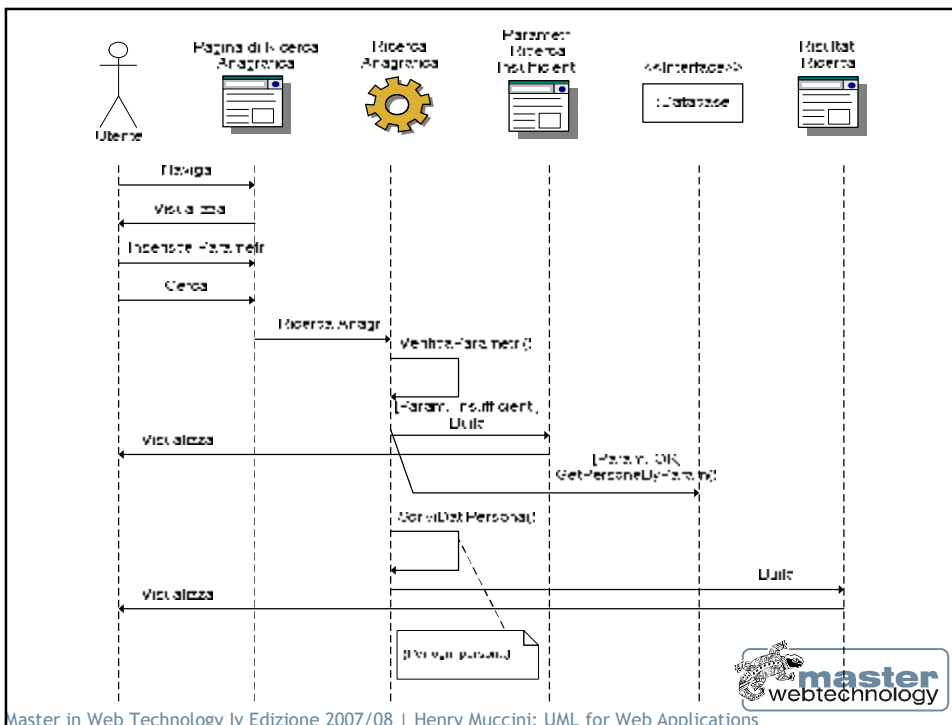
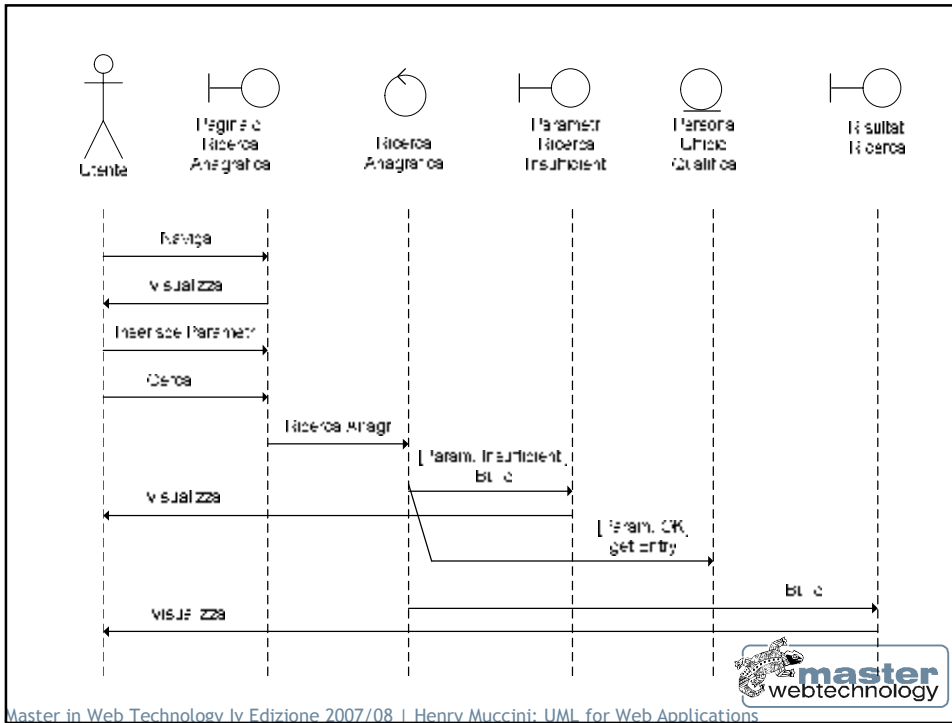
## » Relazione di <<redirect>>

- > consiste nella possibilità di **redirezionare** la gestione di determinate richieste da una server page ad un'altra



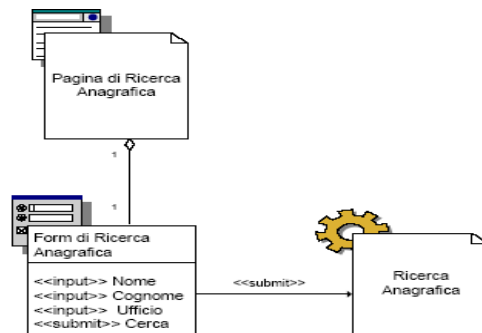
# Esempio: da Analysis class diagram a design class diagram





# Form

- » Un form è una raccolta di elementi di input
- » **Un form può esistere soltanto nel contesto di una client page**
- » La relazione esistente tra un form e la client page che lo contiene è la relazione di **aggregazione**
  - > Una client page puo' contenere piu' form
- » La relazione esistente tra un form e la pagina Web che lo elabora, viene espressa nel modello attraverso un'associazione con lo stereotype **<<submits>>**.

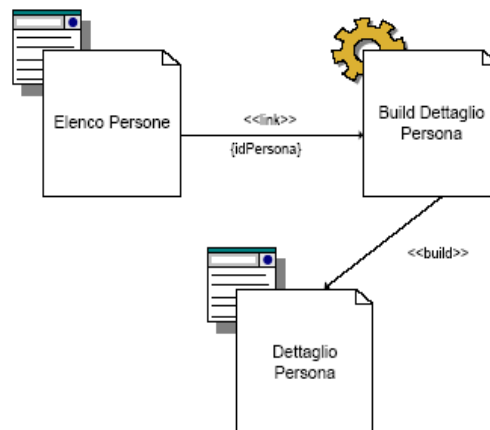


## Link

- » Un link in un'applicazione Web rappresenta un **percorso di navigazione** attraverso il sistema
- » Questa associazione ha **origine sempre da una client page** e punta ad un'altra client page oppure ad una server page



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications



- » I link possono anche contenere dei **parametri** (**<<parameter>>**), destinati ad essere utilizzati da parte della server page:

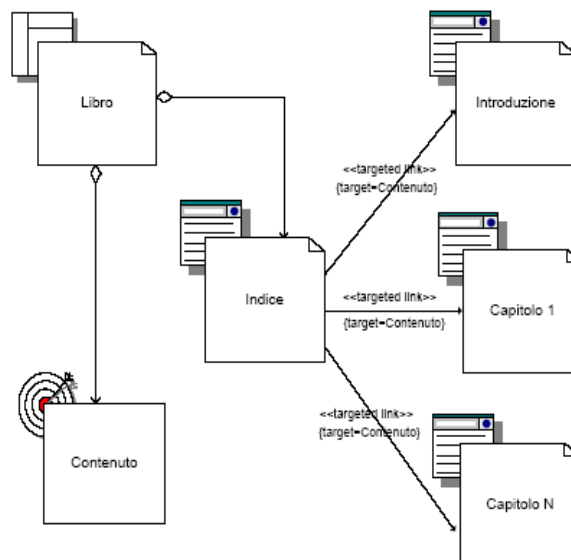
> Nell'esempio, il parametro identifica la persona di cui si vogliono i dettagli



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

# Frame

- » I frame sono implementati in HTML definendo un “frameset”
  - > I frameset sono rappresentati nel modello attraverso una client page con lo stereotype **<<frameset>>**
- » Il meccanismo utilizzato per referenziare frame in un frameset e' il “target”
  - > I target sono rappresentati tramite **<<target>>**
    - Un target non ha proprietà o attributi, è semplicemente un contenitore per una client page, referenziabile da parte di altre client page
- » In conclusione, un frameset può essere considerato come un'aggregazione di client page e di target



## Architettura Thick Web Client con Conallen

- » In questo pattern architetturale anche il Client puo' eseguire delle computazioni
- » In generale, si tratta di computazioni su risorse residenti esclusivamente al lato client
  - > E.g., validazione sui dati in input
- » Bisogna quindi modellare:
  - > le componenti dell'applicazione lato client
  - > le relazioni tra queste componenti e le altre risorse client-side
- » **client-side scripting** e **client object**



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Client-Side Scripting

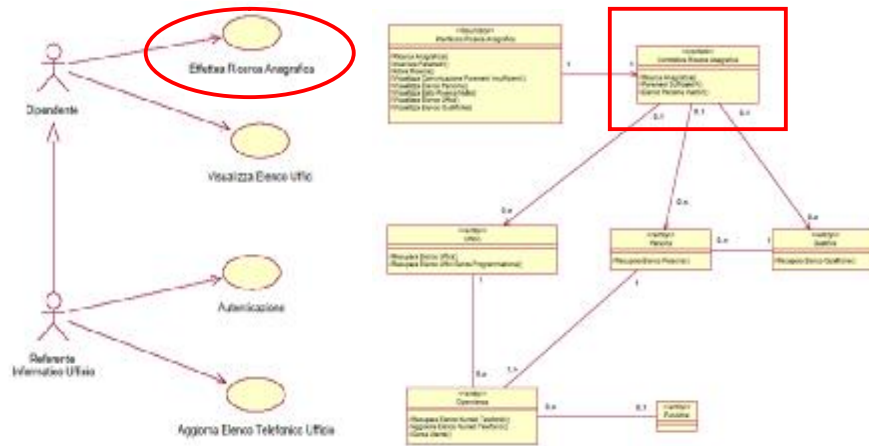
- » Con client-side scripting s'intende il codice (generalmente JavaScript o VBScript) che consente di arricchire le funzionalità offerte dalla pagina
- » Stereotype utilizzato:
  - > **«client-side script»** (o equivalentemente **«JavaScript»**, o **«VBScript»**), associata alla client page contenente lo script.

Nota: Nei Sequence diagram, l'esecuzione di un client-side script da parte di una client page viene rappresentato attraverso l'invocazione di un'operazione su client stesso



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Esempio: pagina ricerca anagrafica



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

### » Thin Web client:

- > Tutta la computazione viene eseguita sul Server



### » Thick Web client:

- > I controlli possono essere fatti sul client
  - Immissione parametro
  - Controlli sintattici



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Client Object

- » Generalmente ActiveX o applet Java
- » Stereotype «**ActiveX control**», o «**Java applet**».
- » La relazione esistente tra un client object e la client page che lo contiene è la relazione di aggregazione



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Esempio da fare in classe

- » Agenda di eventi via Web:
  - > consente di selezionare una data in una pagina Web. L'applicativo Web visualizza, in un'apposita finestra della pagina, l'elenco delle attività programmate per quella data.
  - > La funzionalità può essere implementata attraverso un frameset, contenente una client page a partire dalla quale può essere selezionata la data ed un target nel quale verrà visualizzato l'elenco delle attività programmate per la data selezionata.
  - > Il meccanismo di scelta della data è realizzato attraverso il client object Calendario (implementato per mezzo di un ActiveX control), contenuto nella client page.
  - > Ogni volta che l'utente seleziona una data, l'elenco delle attività visualizzate nel target viene aggiornato, attraverso un link ad un'apposita server page incaricata di creare la pagina con il dettaglio delle attività.



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Architettura Web Delivery con Conallen

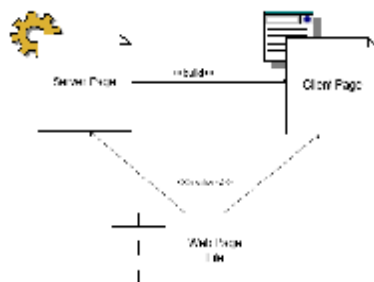
- » Il “Web Delivery Architectural Pattern” è essenzialmente un **sistema ad oggetti distribuito** che è basato su un sito Web.
- » La differenza principale tra il Web Delivery e gli altri pattern architetturali è il **metodo di comunicazione** tra client e server
  - > E.g., DCOM, IIOP, RMI
- » Per modellare un Web Delivery in UML, bisogna aggiungere delle estensioni per modellare **protocolli di comunicazione** tra client e server diversi da HTTP
  - > <<DCOM>>, <<IIOP>>, <<RMI>>



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Component Diagram

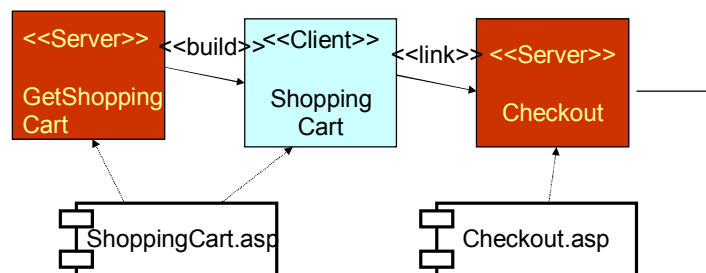
- » Una componente rappresenta a **tutti gli effetti un file** (generalmente un file ASP, JSP, PHP) che può essere richiesto dal Web server e che realizza almeno una server page o una client page
- » Una componente è qualcosa che **realizza un insieme di interfacce**.
- » Questa vista del modello esprime i moduli software e gli eseguibili, attraverso i quali il sistema verrà distribuito



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Component Diagram

» Associamo le classi a delle componenti implementative



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Lessons Learned

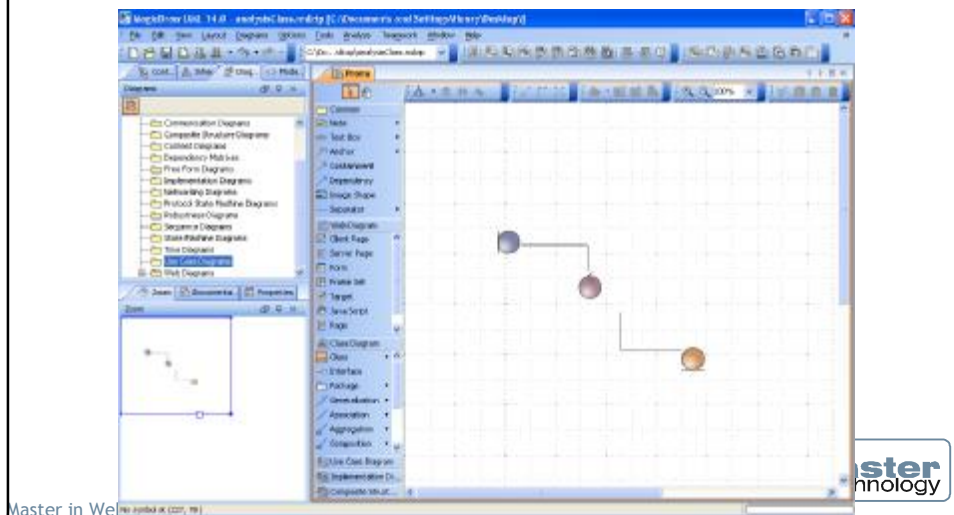
- » The design transformation rules (boundary to client pages, controller to server pages) works well for **ASP** components
- » In **MVC**, instead:
  - > There are at least **two server pages** for each **controller** (one S.P. for coordinating the business logic, another S.P. for creating the interface)
  - > The two S.P. are related via a **<<redirect>>** association
- » Data modeling is performed via J2EE specific patterns



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Da Fare in Classe

- » Uso di Magic Draw per modellare Analysis e Design class models di Conallen



## Implementation

- » Nella fase di implementazione
  - > Il design deve essere mappato sul codice e componenti
    - Tutti i diagrammi finora realizzati ci devono aiutare nella fase di implementazione
  - > Gestione delle modifiche tramite reverse engineering
    - Bisogna tenere allineati i modelli ed il codice

## Mapping del design sulle componenti

» Per implementare Server Pages possono essere adottate due strategie:

> **Compiled Page**

- Una compiled page ha generalmente il ruolo di implementare piu' server page
  - 1 componente → piu' server pages

> **Scripted Pages**

- Una componente realizza una server page
  - 1 componente → 1 server page



## Coding

» Per iniziare la fase di mapping design-codice, si:

- > Prende il modello del design
- > Prende il modello delle componenti
- > Identificano le componenti reali che implementeranno le pagine identificate



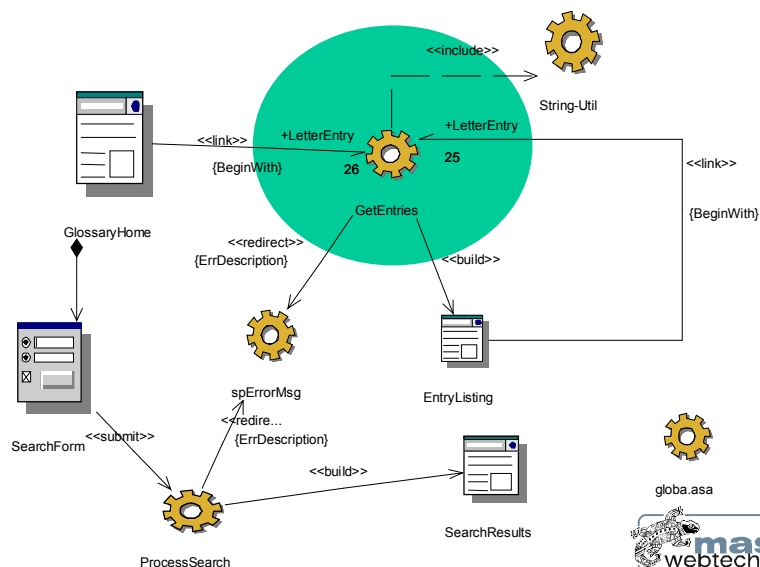
## Suddivisione in Package

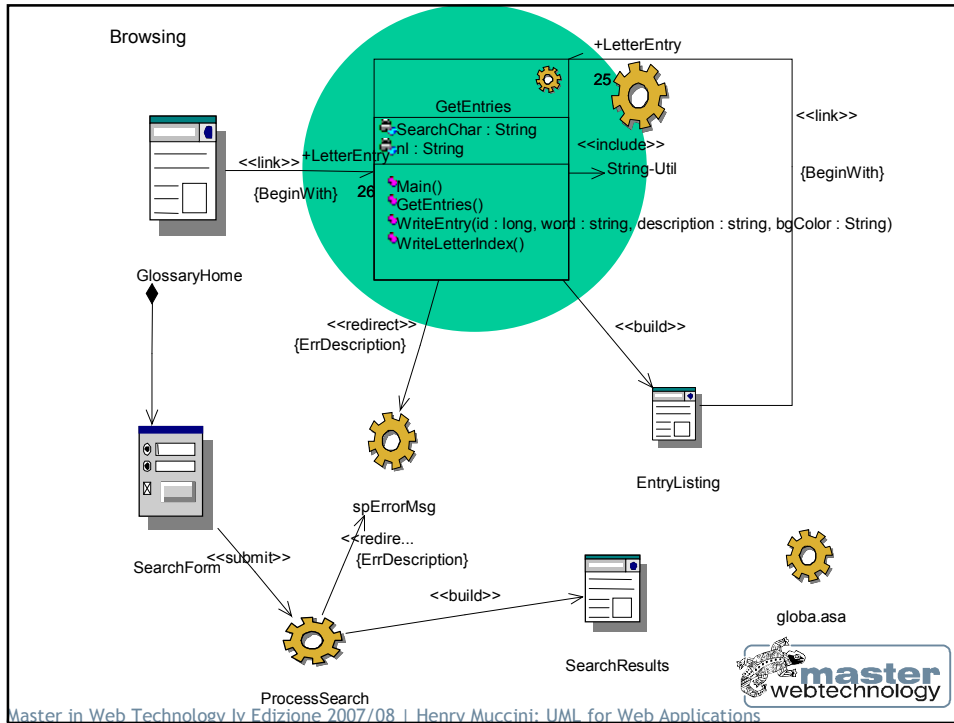
- » Web Pages tenute separate da Server Components
- » Web Page package:
  - > Astrazione di una directory sul Web Server
  - > Le informazioni sulla directory (nome e locazione) sono memorizzati nei tagged value del package



## Web pages package

Browsing





## Code

» Apri file [16\\_GetEntries.asp](#)

## NOTA: Business and Presentation Logic

- » Business e Presentation sono mischiate insieme?  
Non e' chiaro fino in fondo...
- > Le server pages dell'esempio sono implementate tramite scripted pages:
  - [Si ha un mapping 1 a 1 tra <<server page>> e file ASP. Vedi Fig. 11-5 e code in page180.](#)
  - [Si mischia HTML e Business Logic in JavaScript](#)
  - Lo stesso accade per Client Pages (page 186)



## NOTA: Business and Presentation Logic

- > **MA**, non si capisce se dipende SOLO dal fatto che si stia lavorando con ASP:
  - In page184, infatti, si dice “Despite our best attempts at separating presentation from business logic, it can't easily be separated in an ASP environment”
  - In page 175 e Fig. 11-3, si esegue una separazione fra “Server Components” e “Web Pages”
  - In page 177 e Fig. 11-4, si descrivono pagine con server-side processing
  - Concetto di Redirect, page 154 e Fig. 10-3, 10-4



# User eXperience Model



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## User eXperience

- » Rappresentano delle estensioni, apportate da Conallen nella seconda versione del libro
- » Catturano il “look and feel”:
  - > possibili scenari che l’utente attraverserà durante la navigazione
  - > diagrammare le possibili strade che sarà possibile imboccare durante la visita del sito
- » Tre concetti nuovi:
  - > **Screens and Content**
  - > **Storyboard scenarios**
  - > **Navigational paths** attraverso gli screens.



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

» **Screen:**

- > uno screen è precisamente ciò che è presentato all'utente
  - La combinazione dei contenuti statici e dinamici produce lo screen

» **Storyboard scenario:**

- > **Ordine di visita** degli screen
- > Lo scopo finale degli "Storyboard scenarios" è di esprimere un uso tipico del sistema attraverso gli occhi dell'utente

» **Navigational Path map:**

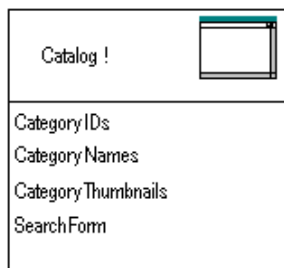
- > struttura degli screens di un'applicazione insieme alle **possibili "strade"** che possono essere percorse
- > mappa stradale degli screens dell'applicazione



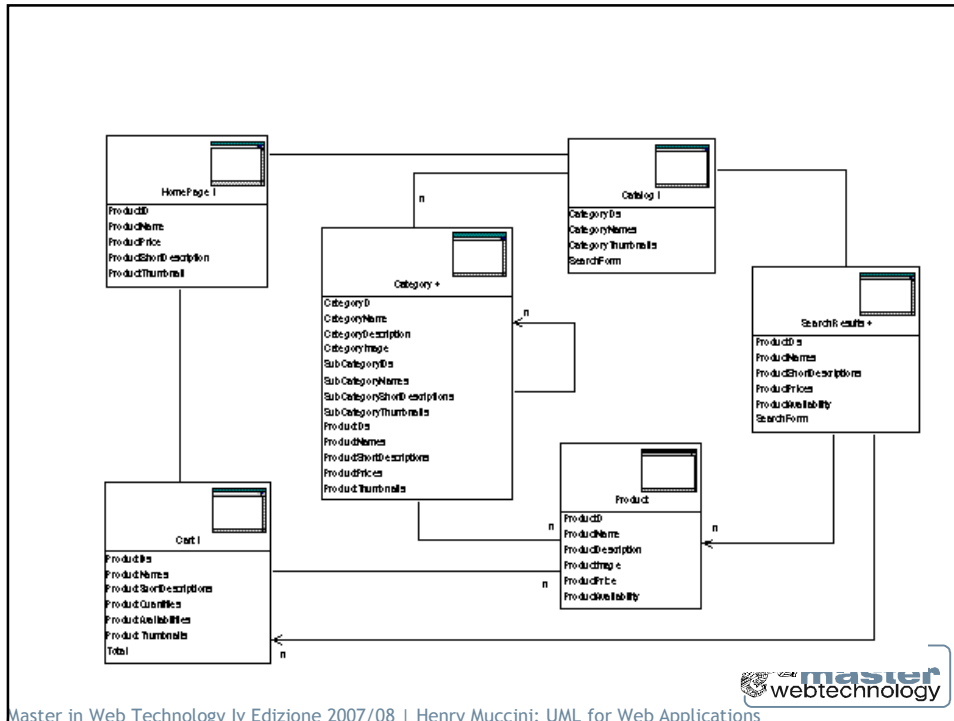
## Esempi

» **Screen:**

- È molto importante realizzare un mapping fra gli screens e gli elementi utilizzati in fase di analisi per mantenere la tracciabilità







## Tool Support and Bibliography

- » **IBM Rational** tools and Magic Draw contain the Conallen stereotypes
- » Visio Stencils
- » ...
- » **Building Web Applications with UML - 2nd edition. The Addison-Wesley Object Technology Series. Jim Conallen**
  - > [Con\_ACM99] Modeling Web Application Architectures with UML  
Jim conallen, Rational Software, June 1999  
In the October 1999 (volume 42, number 10) issue of Communications of the ACM. On line at <http://www.uml.com.cn/papers/webapps.htm>
  - > [ConExt] UML Extension for Web Applications 0.91  
Jim conallen.
  - > [UML2000] Modeling Web Applications in the UML  
UML2000 Workshop.  
Workshop home page: <http://www.txt.com/webuml/>



## Riassumendo



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Conallen

- » **Basics:**
  - > Basato sul RUP
  - > Guidato fortemente dall'architettura dei sistemi Web
- » **Fasi:**
  - > Requisiti, Analisi, Design, Implementazione
- » **Concetti base:**
  - > Pagina Web, che funge da legame tra Architettura e gli Analysis model, tramite partizionamento di oggetti in tiers
  - > Stereotype



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

» **Idea: Utilizzare UML stereotipato per rappresentare gli elementi di un Sito Web, cioè'...**

» **Oggetti**

**Page**

**Form**

**Frame**

**COM  
object**

» **Relazioni**

**Link**

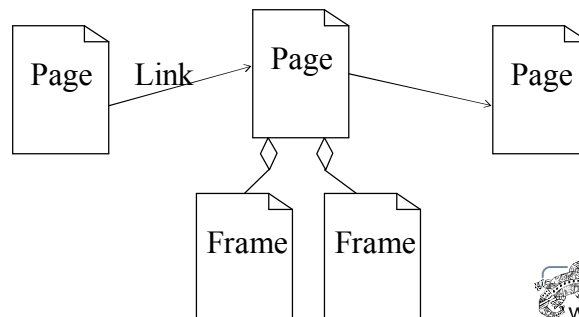
**Frame**

**Content**



## Intuizione

- Un Sito e' composto da pagine, che possono essere di tipo Client e Server.
- Una pagina puo' contenere delle Form e puo' far parte di un Frame Set.
- Piu' pagine possono essere linkate tra di loro



## **ServerPage:**

**Metamodel Class:**

**Descrizione**

**Icona**

**Attributi**

**Operazioni**

**Vincoli**

**Class nel Class Diagram**

**Rappresenta una pagina Web con degli Script. Tali script possono interagire con le risorse disponibili sul Server**



**Sono le variabili**

**funzioni degli script**

**Le Server Page possono solo essere collegate ad oggetti nel Server**



## **ClientPage:**

**Metamodel Class:**

**Descrizione**

**Icona**

**Attributi**

**Operazioni**

**Vincoli**

**Class nel Class Diagram**

**Rappresenta una pagina formattata in Html. E' un mix di dati, grafica e link. Possono essere linkate a ClientPage e ServerPage**



**Sono le variabili nei tag**

**Tag degli script**

**Nessuno**



» **Nota:** Il comportamento di una pagina Web sul Server e' differente dal comportamento della stessa sul Client:

- > Sul Server la pagina deve accedere le risorse del Server (dbase, file system, ...)
- > Sul Client, la pagina deve essere invece collegata al browser, alle Applet, ActiveX controls, ...



» Modelliamo gli aspetti Server-side con una classe e quelli Client-side con un'altra e leghiamo con il <<build>> le due pagine (Separation of Concerns)



## ***Form:***

### ***Metamodel Class:***

#### ***Descrizione***

### ***Class nel Class Diagram***

***Una form e' una collezione di campi di Input che fanno parte di una ClientPage***

#### ***Icona***



#### ***Attributi***

***Campi di Input***

#### ***Operazioni***

***Nessuna (non possono essere incapsulate in una form)***

#### ***Vincoli***

***Nessuno***



## **Frame Set:**

**Metamodel Class:**

**Descrizione**

**Icona**

**Attributi**

**Operazioni**

**Vincoli**

**Class nel Class Diagram**

**Un Frame set e' un contenitore di piu' pagine Web. Un Frame Set puo' essere una Page o un altro Frame Set**



**Quelli di una Page**

**Quelli di una Page**

**Nessuno**



## **Target:**

**Metamodel Class:**

**Descrizione**

**Icona**

**Attributi**

**Operazioni**

**Class nel Class Diagram**

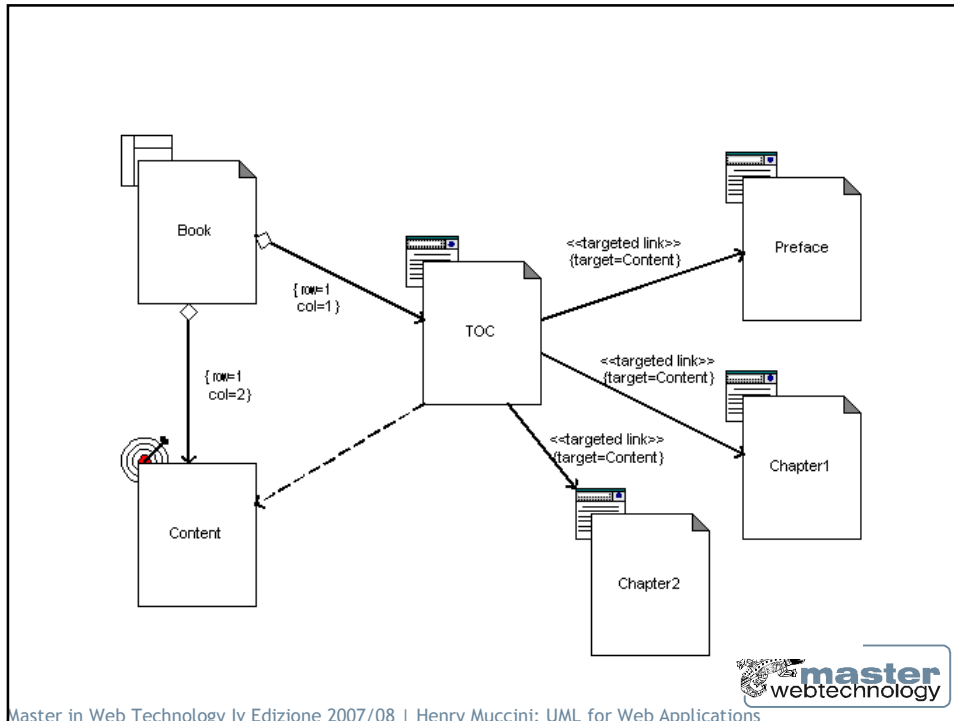
**E' un particolare Frame o una nuova finestra del Browser linkata da altre pagine client**



----

----





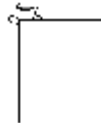
## **JavaScript:**

**Metamodel Class:  
Descrizione**

### ***Class nel Class Diagram***

***Su browser che abilitano i JavaScript e' possibile utilizzare tali oggetti. Possono esistere solo in ClientPage***

**Icona**



## **Link:**

**Metamodel Class:  
Descrizione**

### ***Association nel Class Diagram***

***Un Link e' un puntatore di una Client Page verso una Client o una Server Page. I parametri che possono essere inviati vengono rappresentati tramite Tagged Values***

**Icona**



## **Build:**

**Metamodel Class:**

**Descrizione**

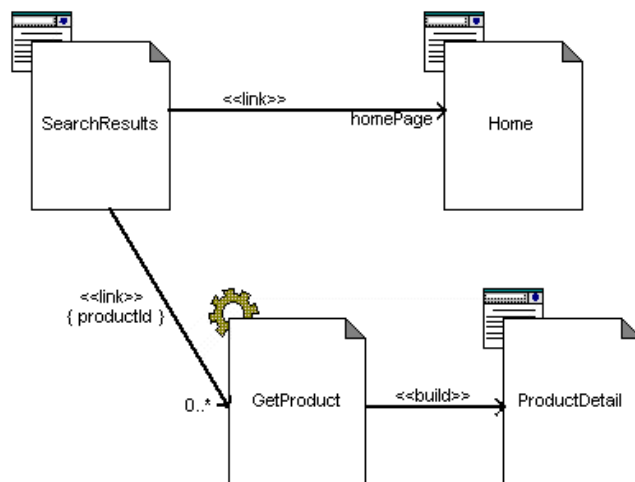
**Icona**

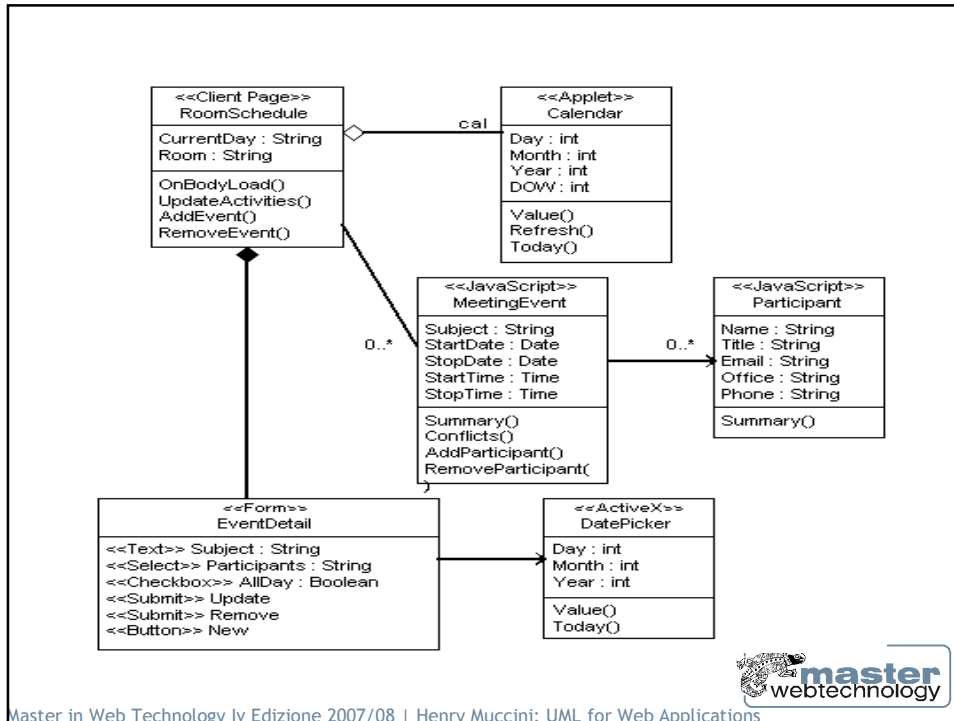
**Association nel Class Diagram**

**Una Server Page <<build>> una Client Page nel senso che, ad ogni Client page deve essere acciata la relativa Server Page.**

**Esempio: Server page = page dinamica, Client page = istanza della pagina dinamica**

**<< build >>**





## Page:

### Metamodel Class:

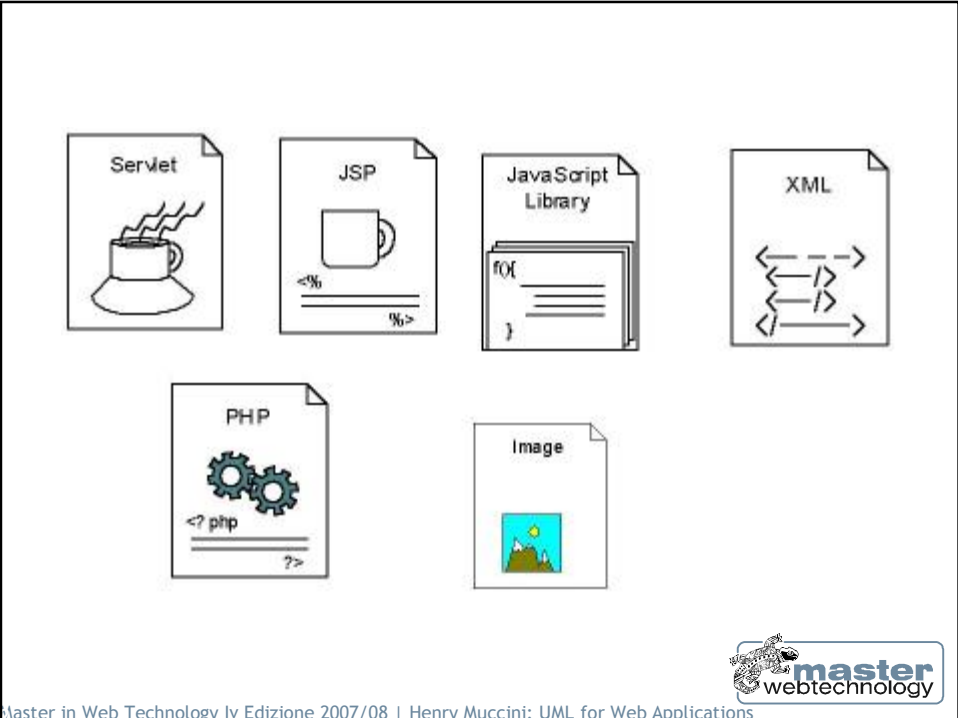
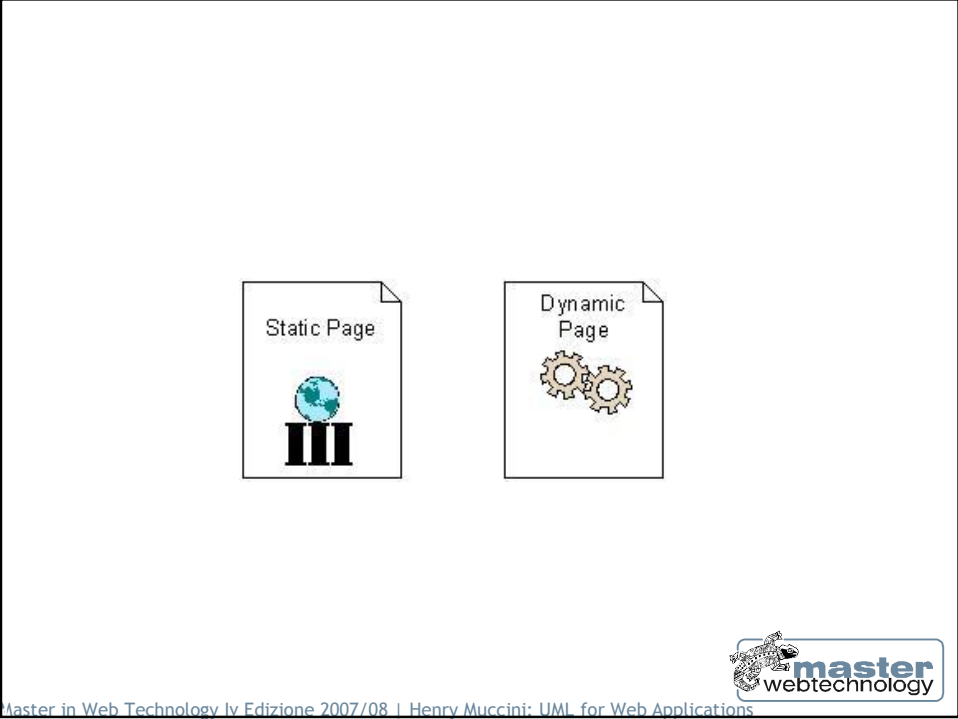
### Descrizione

### Component nel Component Diagram

Tramite questa vista descriviamo tutti i componenti (pagine Web) da sviluppare. Un component Diagram puo' essere visto come una mappa del sito. Le componenti collegate in un Component Diagram verranno rappresentate come Classi linkate in un Class Diagram

### Icona



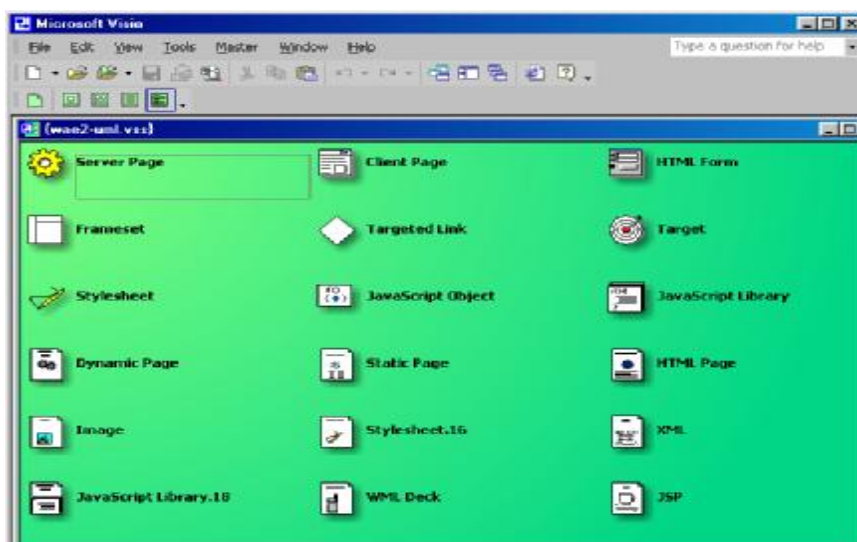


## Tool support



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Visio Stencils



Master

by

## Rational Rose

- » IBM Rational Rose include gli stereotipi di Conallen
- » Visual UML:
  - > <http://www.visualuml.com/products.htm>



## Esempio

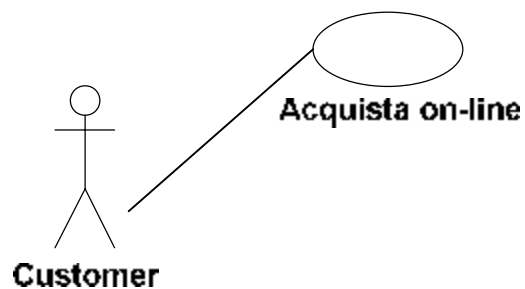


## Esempio

- » Prendendo ad esempio un sistema che permetta di eseguire acquisti on-line, tramite l'utilizzo di un carrello
- » Vedremo come tale sistema puo' essere modellato utilizzando le tre diverse architetture proposte precedentemente
- » Cercheremo di capire, quindi, quanto l'architetture influenza il design



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications



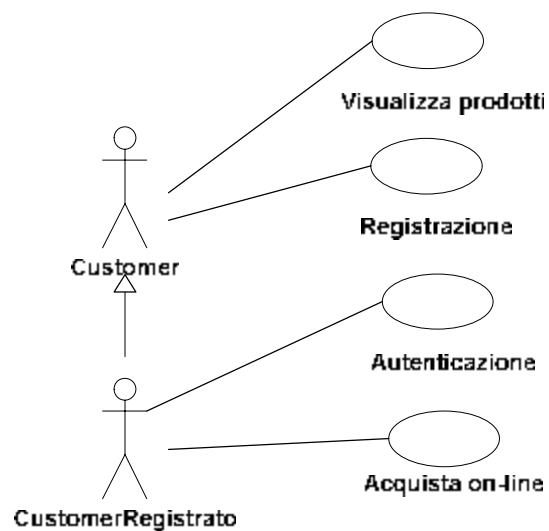
Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Richieste del committente

- » Un utente, collegandosi al sito, deve poter effettuare degli acquisti on-line.
- » Gli articoli possono essere visti direttamente sul sito, con una lista delle caratteristiche e del prezzo. Tutti gli utenti possono eseguire tale visita
- » Gli articoli possono essere acquistati solo da utenti registrati
- » Dopo la selezione di un prodotto, il carrello dovrà riportare la lista aggiornata dei prodotti da acquistare
- » Naturalmente, ci sono dei requisiti di sicurezza il sistema deve rispettare
- » Alla fine dell'operazione di acquisto, si esegue un logon
- » La sessione deve rimanere valida finché l'utente non esegue logon e comunque per non più di xxx minuti
- » ...



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Utilizzo di un Glossario

» Ha il compito di:

> identificare:

- Pagine Web
- Dati e Dbase
- Applicativi

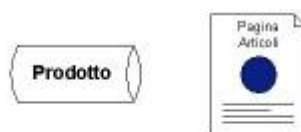
» Analizzare:

> l'architettura del sistema, al fine di distribuire componenti ed assegnare le giuste tecnologie di sviluppo



## Oggetti: glossario

» Gli articoli possono essere visti direttamente sul sito, con una lista delle caratteristiche e del prezzo. Tutti gli utenti possono eseguire tale visita



» Gli articoli possono essere acquistati solo da utenti registrati



» Dopo la selezione di un prodotto, il carrello dovrà riportare la lista aggiornata dei prodotti da acquistare



## Elementi del Modello

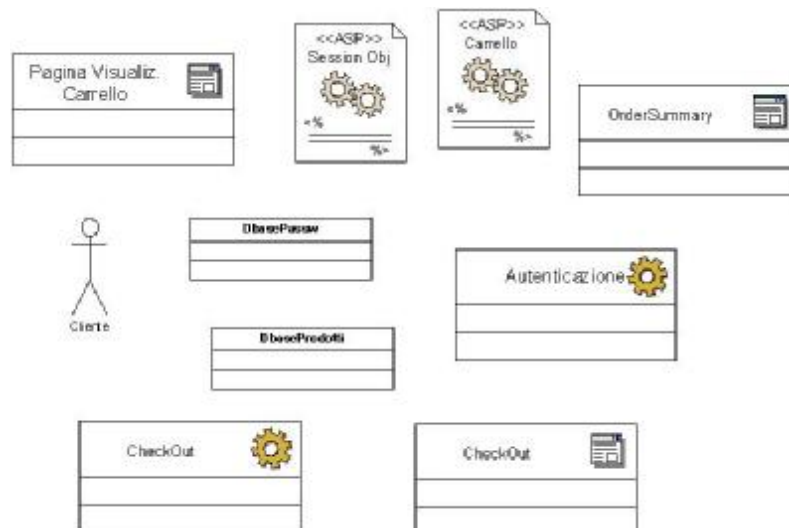
### » Pagine lato Client:

- > Pagina Iniziale con possibilita' di login
- > Pagina di Login (input)
- > Pagina di Errore nella Login (statica o dinamica)
- > Pagina con link ai prodotti (link statico o dinamico)
- > Pagina di ricerca sui prodotti (risultato dinamico)
- > Pagina di Logout



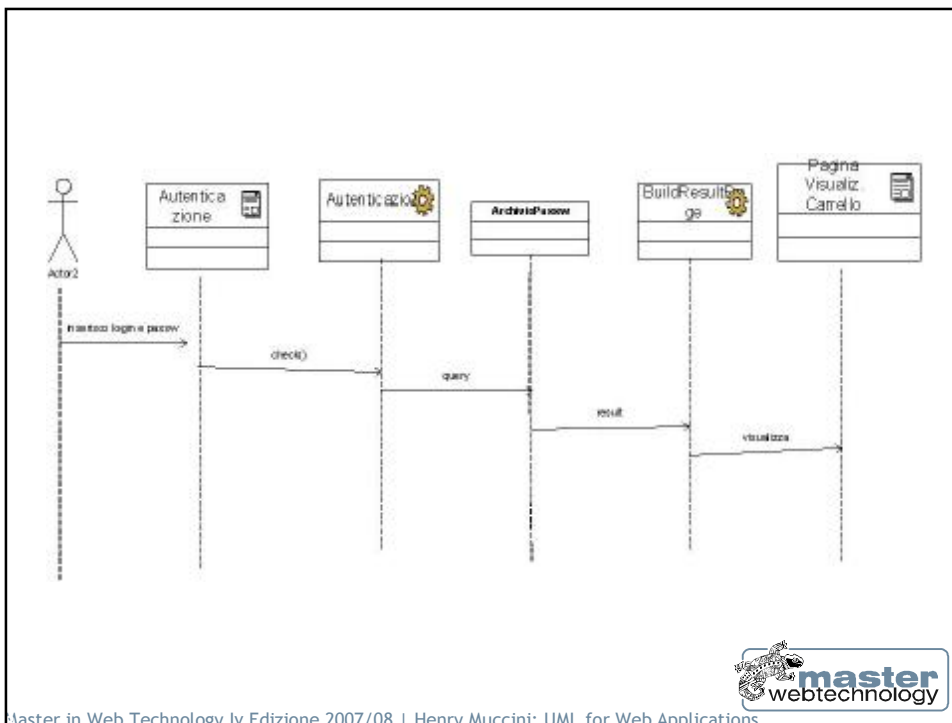
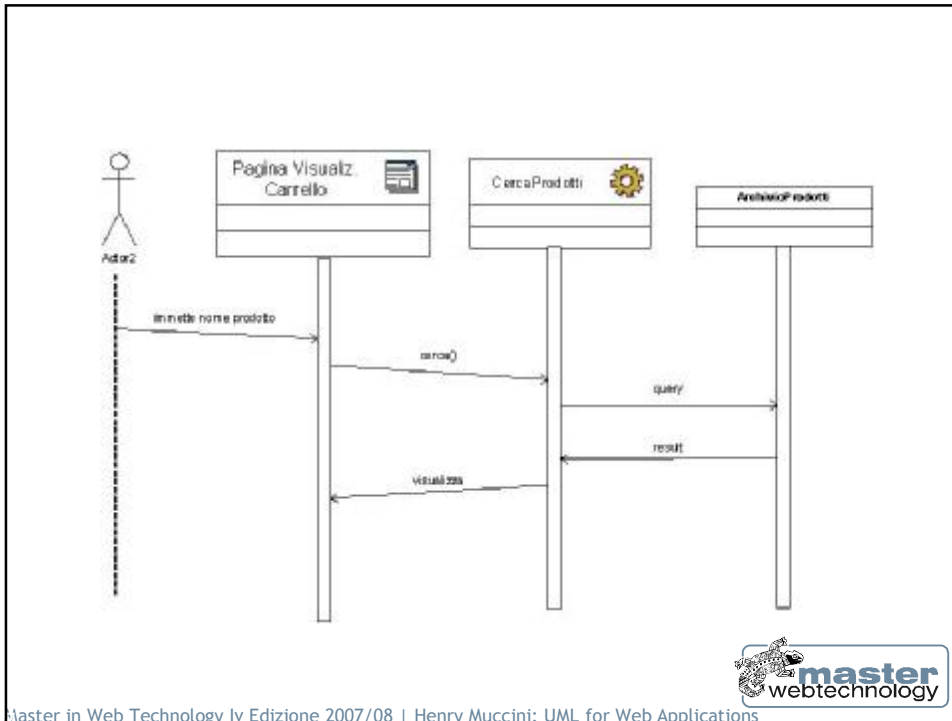
Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications

## Thin Web Client



Master in Web Technology Iv Edizione 2007/08 | Henry Muccini: UML for Web Applications





# Thick Web Client

