



MODELLAZIONE E COMPLESSITÀ DI RICH INTERNET APPLICATIONS

STUDENTE: ANTONIO FARALLI

RELATORE: PROF.SSA ANNA RITA FASOLINO

CORRELATORE: PROF. PORFIRIO TRAMONTANA

Università degli Studi "Federico II"

Corso di Laurea in Ingegneria Informatica

Anno Accademico 2006/2007

Introduzione (1)

- La rete internet cui siamo abituati oggi ha subito numerosi cambiamenti. All'inizio, tutte le pagine Web erano statiche; gli utenti richiedevano una risorsa e il server gliela restituiva. Tuttavia, dopo poco tempo gli utenti Web cominciarono a desiderare un'esperienza più dinamica. Iniziarono così a nascere nuovi linguaggi e tecniche, sia lato client che lato server

CGI, Servlet, ASP e PHP

Applet

JavaScript

Flash

Introduzione (2)

	Vantaggi	Svantaggi
CGI	Compilato in numerosi linguaggi, da Perl a Visual Basic	Poco sicuro. Si consente letteralmente alle persone di eseguire un programma sul proprio sistema
APPLET	Consentono di creare un <i>thick client</i> all'interno del proprio browser	È indispensabile utilizzare la versione Java adeguata al client
JavaScript	Agevola gli sviluppatori nella modifica dinamica dei tag delle loro pagine, con lo scopo di arricchire l'esperienza dell'utente. Sicuro poiché da interpretare	richiedono più tempo per la loro interpretazione a causa dei passaggi e delle trasformazioni che devono subire per essere compresi dal client
Servlet, ASP e PHP	Filosofia "scritto una volta ed eseguibile ovunque". Permettono agli sviluppatori di selezionare il server di applicazioni e gli ambienti server migliore.	Inclini a causare errori difficili da visualizzare. Appesantiscono il server
Flash	Consente ai progettisti di creare applicazioni straordinarie di natura considerevolmente dinamica. Non richiede particolari abilità di programmazione ed è facile comprenderne il funzionamento	Richiede il software sul client. Richiede una certa quantità di larghezza di banda di rete per essere eseguite in modo corretto

Introduzione (2)

La rivoluzione DHTML

VANTAGGI	SVANTAGGI
Combinazione tra HTML, Cascading Style Sheets (CSS), JavaScript e DOM. La combinazione di queste tecnologie consente agli sviluppatori di modificare al volo il contenuto e la struttura di una pagina Web	Nel momento in cui fu introdotto, richiedeva versioni browser che non erano ancora state largamente adottate. Spesso ciò comportava la creazione di script per controllare il tipo e la versione del browser. Mancanza di uno standard ufficiale

Introduzione (3)

Il problema principale

Molte persone rimasero convinte che il Web offriva applicazioni per utenti di classe B, ovvero applicazioni fin troppo semplici, con scarsa interattività, bassa qualità delle interfacce, e presentasse esperienze utente inferiori.

Dato che Internet era un sistema richiesta/risposta sincrono, l'intera pagina era costantemente aggiornata nel browser. All'origine, non era importante la semplicità della richiesta, se l'utente apportava una o due modifiche, l'intero documento doveva essere rinviato al server e la pagina completa veniva aggiornata. Anche se l'applicazione era utilizzabile, questa limitazione legata all'aggiornamento totale ne provocava un funzionamento piuttosto grossolano.

AJAX (I)

Asynchronous

JavaScript

XML

AJAX (2)

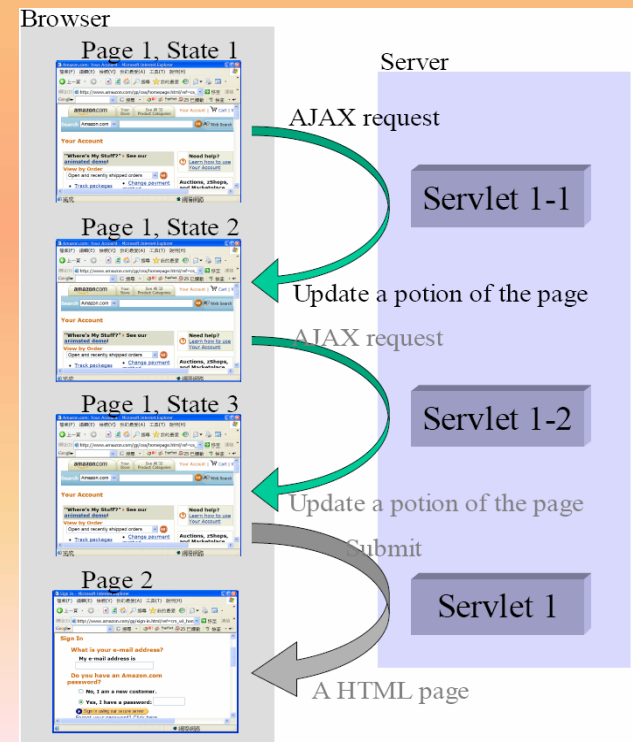
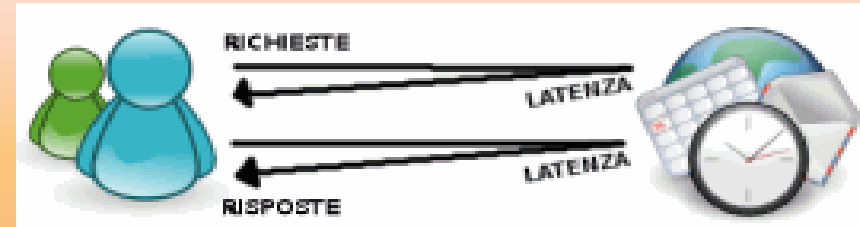
Caratteristiche di base principali:

- **TECNOLOGIA XMLHTTPREQUEST OBJECT;**
- **SCRIPTING REMOTO.**

AJAX (3)

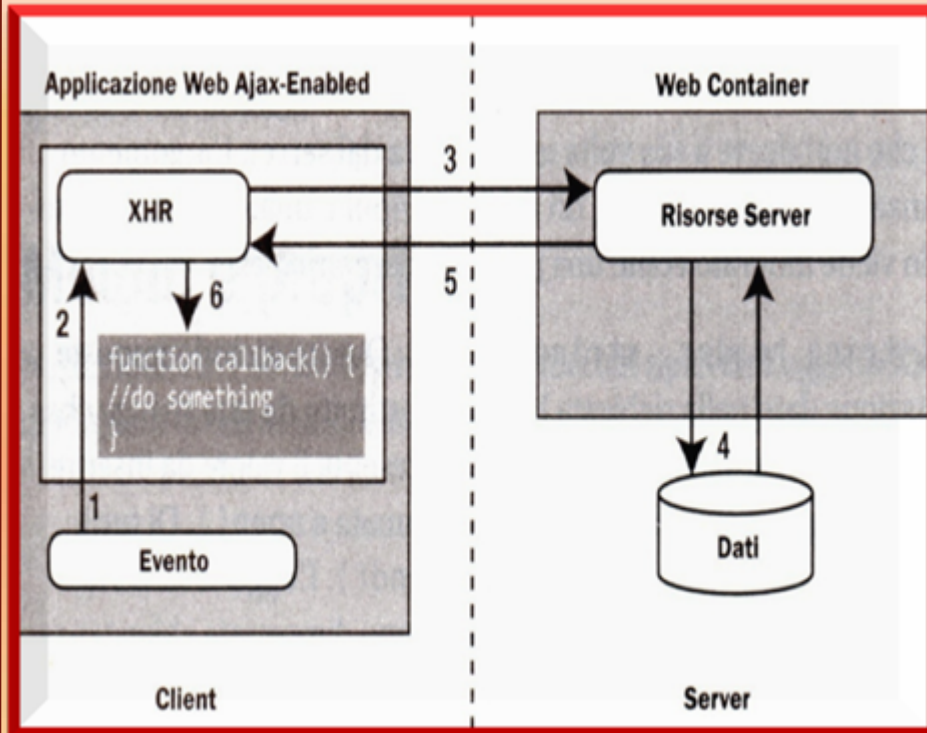
Teoricamente, è possibile effettuare decine di richieste simultanee al server per operazioni differenti con o senza controllo da parte del navigatore-

Non è necessario aggiornare completamente la pagina ma è possibile aggiornare soltanto porzioni di essa, riducendo al minimo l'utilizzo di banda.



AJAX - Interazione

1. Un evento dal lato client attiva un evento AJAX
2. Viene create un'istanza dell'oggetto XMLHttpRequest
3. Al server viene fatta una richiesta
4. Il server può eseguire qualsiasi operazione
5. La richiesta viene restituita al browser
6. In questo esempio si configura l'oggetto XHR affinché chiami la funzione callback() quando viene resa l'elaborazione. Questa funzione controlla la proprietà readystate nell'oggetto XHR e poi analizza il codice di stato reso dal server. Se tutto procede come previsto, la funzione callback() esegue operazioni sul client.



AJAX – Event-based Applications

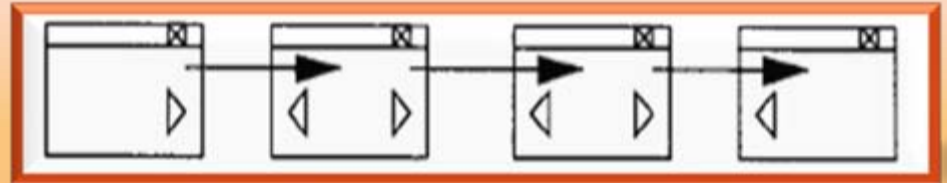
L'applicazione può generare differenti tipi di richiesta in dipendenza dello stato stesso della pagina e la richiesta generata da un qualunque evento può essere diversa dalla richiesta effettuata dallo stesso evento con premesse differenti

AJAX – Multithreading Applications

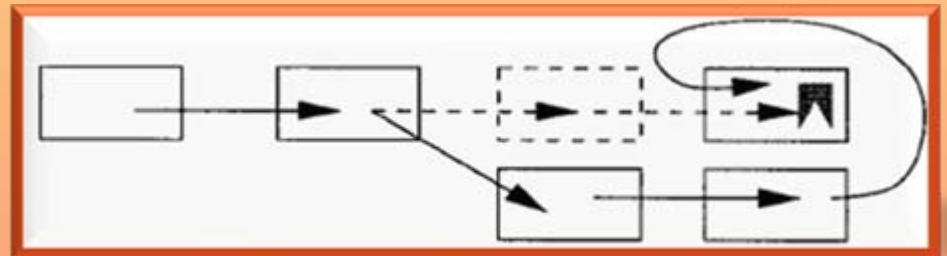
Una volta verificatosi un evento, l'esecuzione può proseguire su più flussi, cioè si crea una sorta di “fork” che dirama l'applicazione in più flussi separati e indipendenti

AJAX oriented Apps vs GUI and Classic WAs

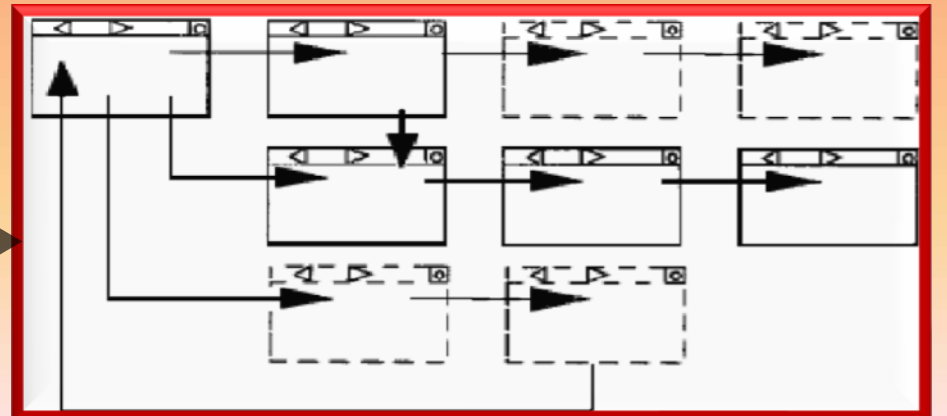
GUI



Classic
WAs



Was AJAX
Oriented



OBIETTIVI TESI

Ricavare informazioni circa la manutenibilità ed informazioni utili per un buon testing di una WA AJAX oriented



Proposta di un modello



**Proposte di metriche per
la valutazione della
complessità**

Reverse engineering delle WAs AJAX oriented

Eseguito in *white box*

Articolato in due fasi principali:

1. raccolta degli eventi
2. conoscenza profonda dei moduli JavaScript dell'applicazione che consenta, all'occorrenza di ogni evento JavaScript, una conoscenza dell'elaborazione prodotta e delle conseguenti modifiche sul DOM

Il modello proposto

Composto da due diagrammi:

- 1. Diagramma di stato:** descrive l'evoluzione in seguito al verificarsi di una successione di eventi;
- 2. Diagramma delle “Unità Funzionali”:** forniscono una conoscenza dell'elaborazione prodotta e delle conseguenti modifiche sul DOM (script richiamati, metodi di accesso al DOM) all'occorrenza di ogni evento JavaScript;

LO STATO (I)

Macrostatato dell'applicazione:

- numero e tipo di script presenti nella pagina corrente;
- struttura del DOM della pagina corrente;
- eventi gestiti;

Microstatato degli oggetti del DOM: l'unità di riferimento è il nodo, a cui si associano tre caratteristiche che ne definiscono lo stato:

- valori;
- attributi;
- gerarchie (*children*)

LO STATO (2)

Tabella di stato associata ad un oggetto del DOM

OGGETTO DOM						
Stato precedente			Evento + Unità Funziona le	Stato successivo		
Tag	Att.	Ger.		Tag	Att.	Ger.

IL PROCESSO DI MODELLAZIONE

Analisi statica: effettuata su pagine client costruite a tempo dinamico e poi analizzate staticamente; si tratta di un'analisi statica su componente ottenuto a tempo d'esecuzione

Analisi dinamica: analisi, svolta a tempo di esecuzione, di componenti che sono dinamicamente istanziati/modificati

ANALISI STATICA (I)

L'UNITÀ FUNZIONALE

```
<input type=button onClick="fun1()"> ◀ HTML Event Handler  
<script>  
  var x=1;  
  function fun1() { ◀ Triggered Script  
    fun2(x++);  
  }  
  function fun2(y) { ... } ◀ Triggered Subprocedure  
</script>
```

The diagram illustrates the execution flow of the provided code. It starts with an HTML event handler that calls the `fun1()` function. This function is identified as a "Triggered Script". Inside `fun1()`, the `fun2(x++)` function is called, which is identified as a "Triggered Subprocedure". The flow is indicated by arrows: a curved arrow from the `onClick="fun1()"` attribute to the `function fun1()` definition, and another curved arrow from the `fun2(x++)` call inside `fun1()` to the `function fun2(y)` definition. Straight arrows point from the labels "Triggered Script" and "Triggered Subprocedure" to their respective function definitions.

ANALISI STATICA (2)

I METODI DI RIFERIMENTO AL DOM

•GETS

Get reference to enclosing form for function
GET::INPUT.form

```
0 <input onclick="doIt(this.form)">
```

•SETS

Write on the window
CALL::DOCUMENT.CHILD.write

```
<script>  
1 function doIt (frm) {  
2   var newWin = window.open();  
3   newWin.document.write(frm.name);
```

•CREATES

```
4   frm.txt.value="ok";
```

Set value of input in form
SET::INPUT.value

```
5 }  
</script>
```

•CALLS

ANALISI DINAMICA (I)

- Si individuano tutti gli oggetti del DOM che, in seguito al verificarsi di un evento asincrono gestito con tecnica AJAX, vengono modificati e assoceremo ad ogni oggetto di questo tipo un microstato, descritto in questo modo

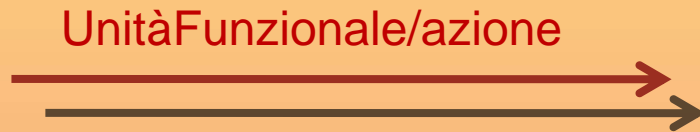
tag_id/stato

- Si individuano tutti gli eventi che determinano una variazione del macrostato e ogni macrostato sarà descritto come

Applicazione_Macrostato

ANALISI DINAMICA (2)

EVENTI E TRANSIZIONI



METRICHE DI COMPLESSITÀ PROPOSTE

STANDARD:

- Logical lines of code (LLOC);
- Number of attributes (NOA);
- Number of procedures (NOP);

DI STRUTTURA:

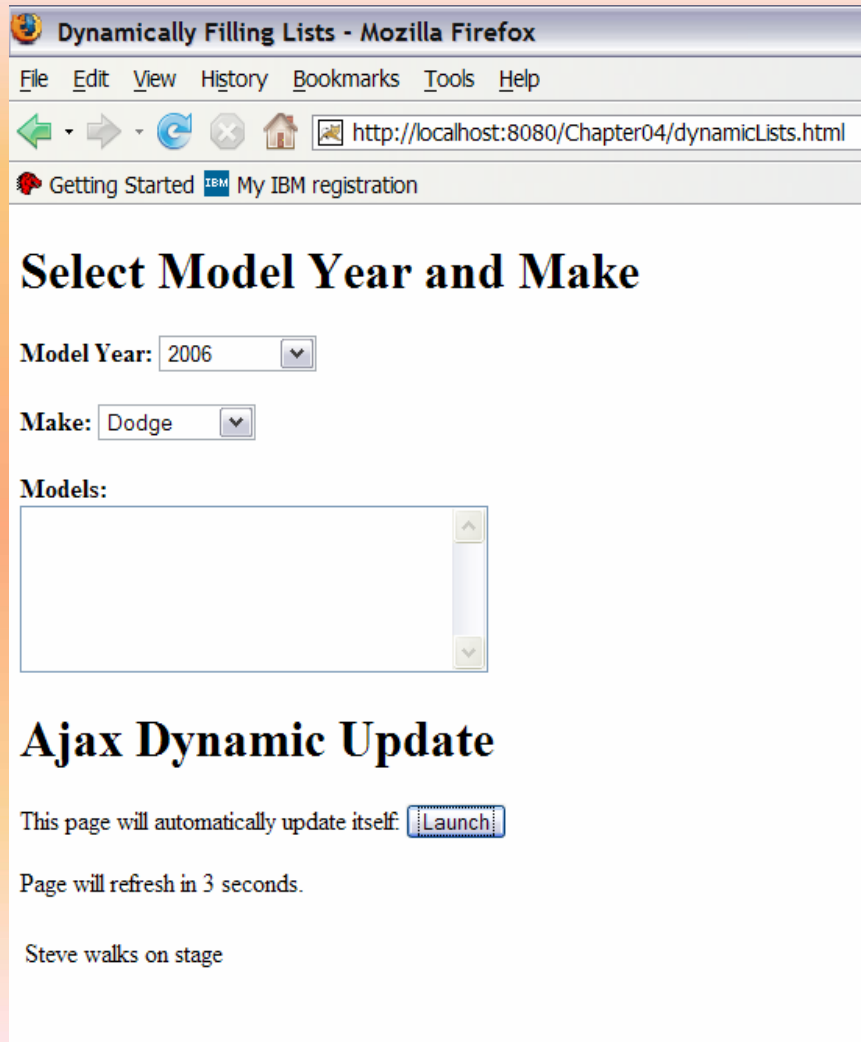
- Total cyclomatic complexity (TCC): $\text{Sum}(\text{CC}) - \text{Count}(\text{CC}) + 1$;
- Decision density (DECDENS): $\text{DECDENS} = \text{CC} / \text{LLOC}$;
- Depth of conditional nesting (DCOND);
- Number of calls (NOC);

SPECIFICHE:

- Number of functional units (NFU);
- Number of JavaScript Event Handlers (NJSEH);
- Number of Independent DOM Object (NIDO);

STUDIO DI UN'APPLICAZIONE (I)

L'APPLICAZIONE

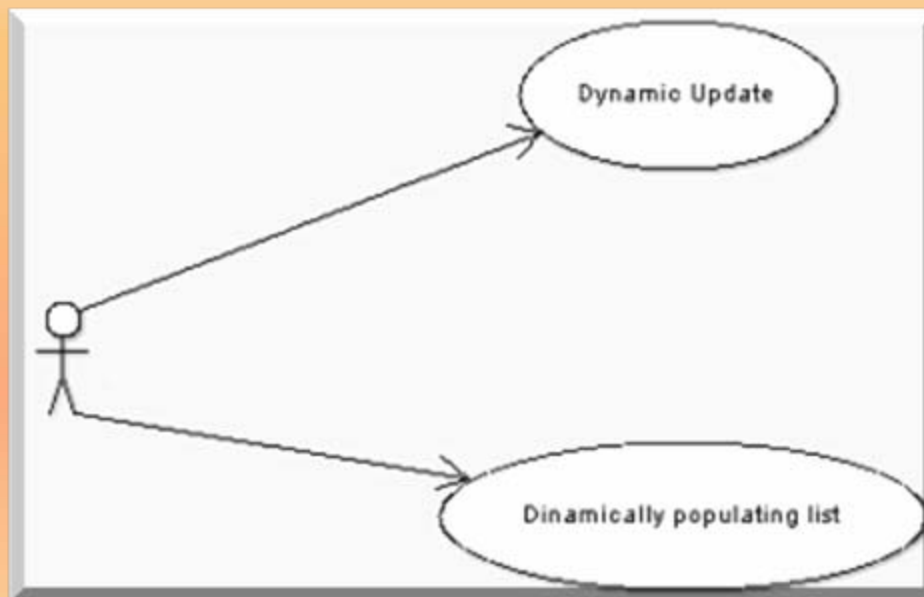


The screenshot shows a Mozilla Firefox browser window with the following elements:

- Title Bar:** Dynamically Filling Lists - Mozilla Firefox
- Menu Bar:** File Edit View History Bookmarks Tools Help
- Address Bar:** http://localhost:8080/Chapter04/dynamicLists.html
- Navigation Bar:** Back, Forward, Reload, Stop, Home, Search
- Page Content:**
 - Getting Started** IBM My IBM registration
 - ### Select Model Year and Make
 - Model Year:** 2006 (dropdown menu)
 - Make:** Dodge (dropdown menu)
 - Models:** An empty list box with up and down arrows.
 - ### Ajax Dynamic Update
 - This page will automatically update itself:
 - Page will refresh in 3 seconds.
 - Steve walks on stage

STUDIO DI UN'APPLICAZIONE (2)

CASI D'USO



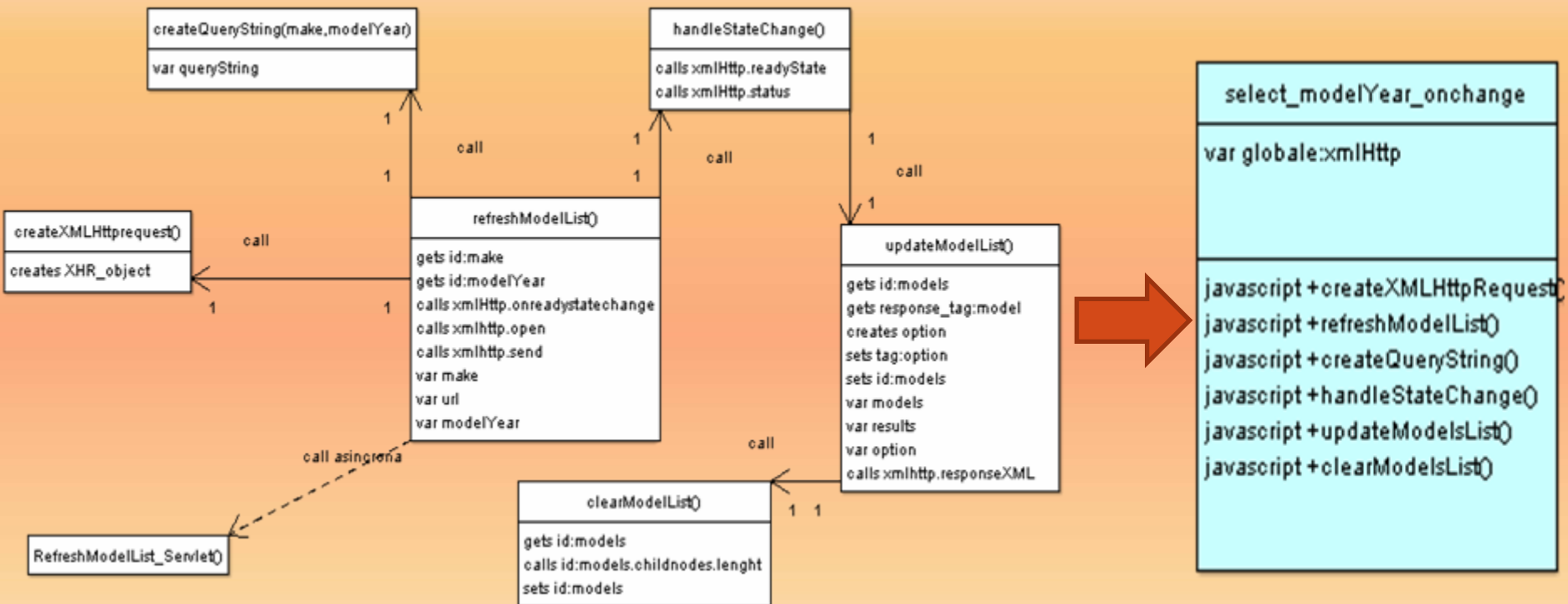
STUDIO DI UN'APPLICAZIONE (3)

GLI EVENTI GESTITI

DOM tag	DOM id	Gestore eventi associato	Script richiamato
select	modelYear	onchange	refreshModelList()
select	make	onchange	refreshModelList()
input	go	onclick	doStart()

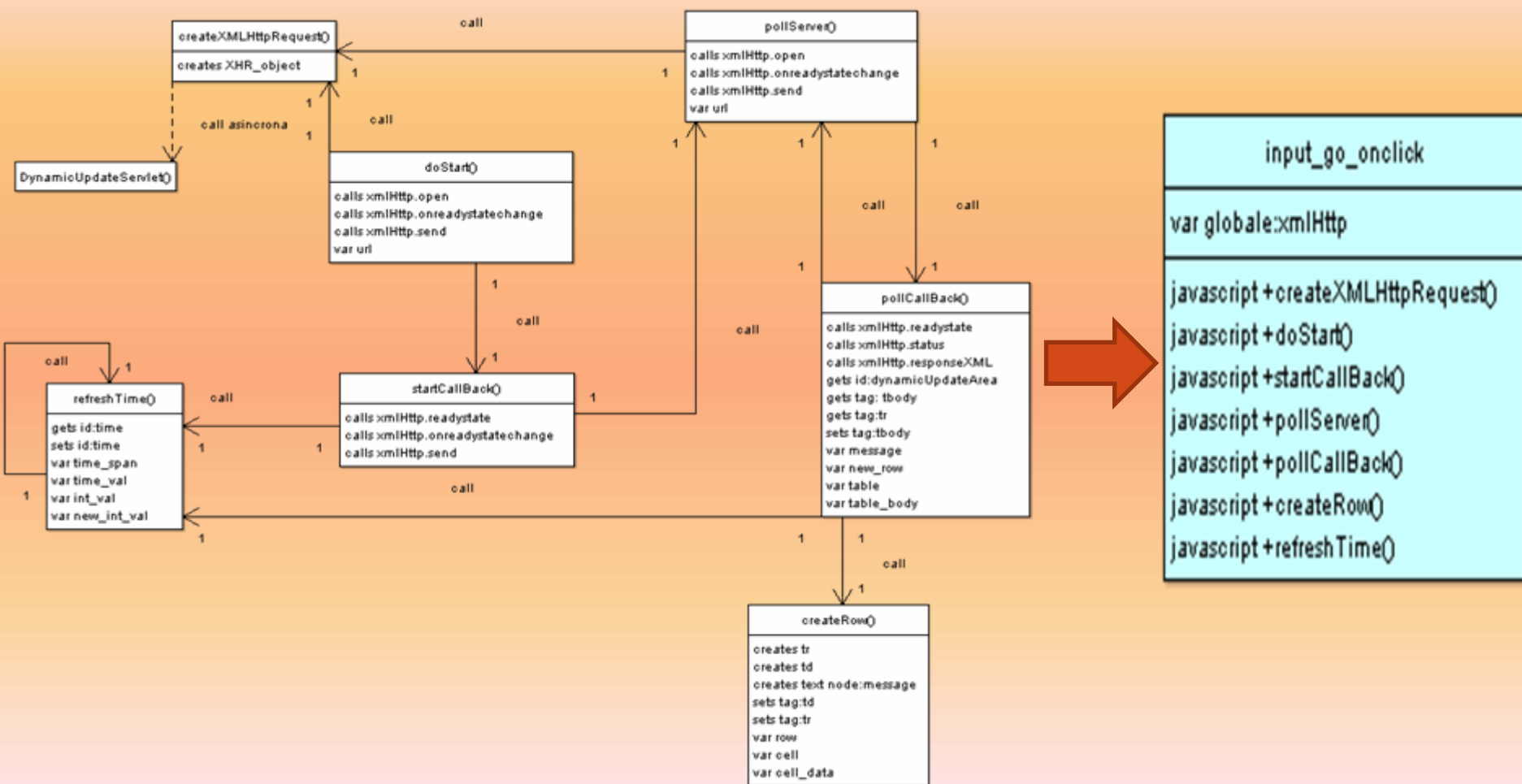
STUDIO DI UN'APPLICAZIONE (4)

L'UNITA FUNZIONALE `select_modelYear_onchange`



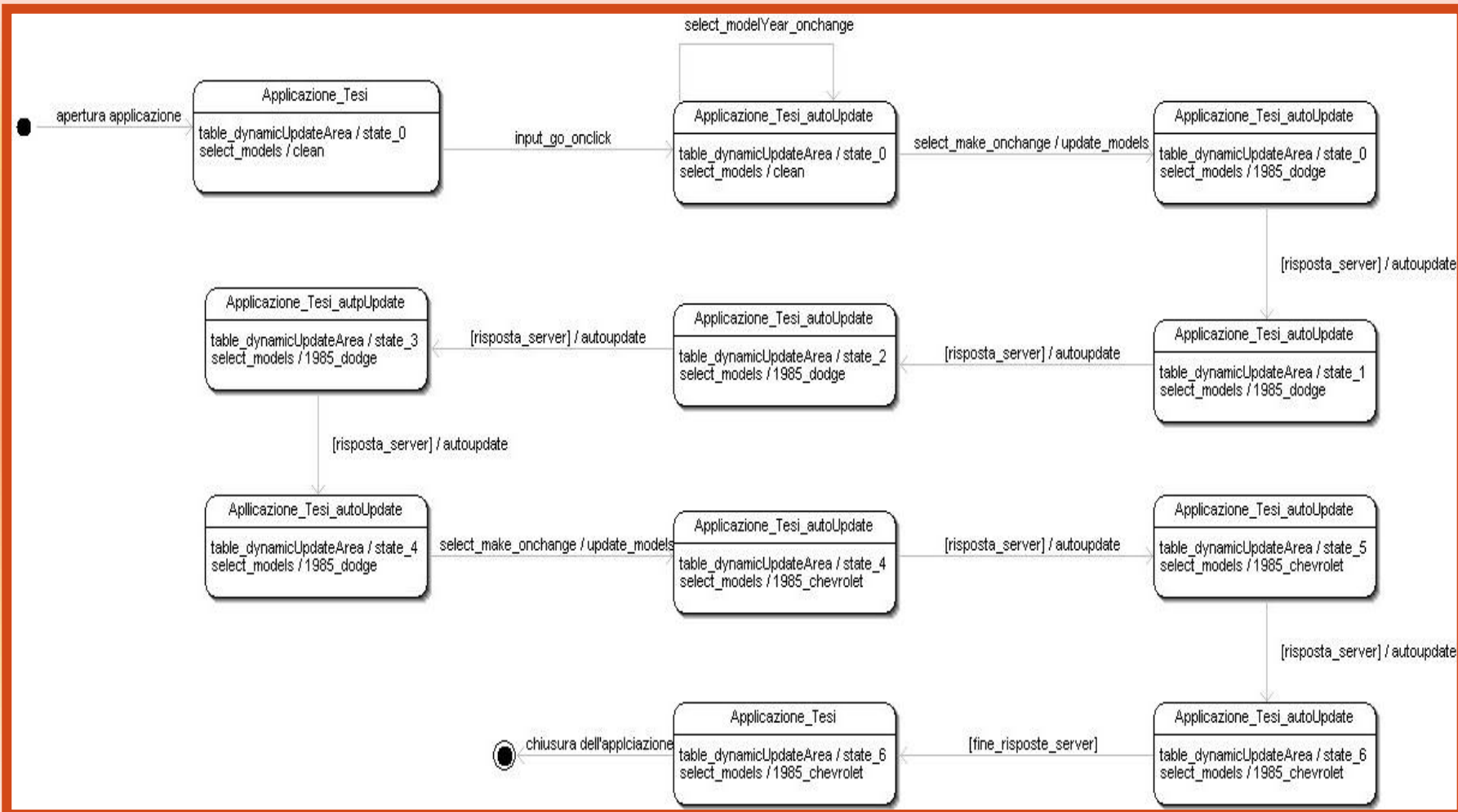
STUDIO DI UN'APPLICAZIONE (5)

L'UNITA FUNZIONALE `input_go_onclick`



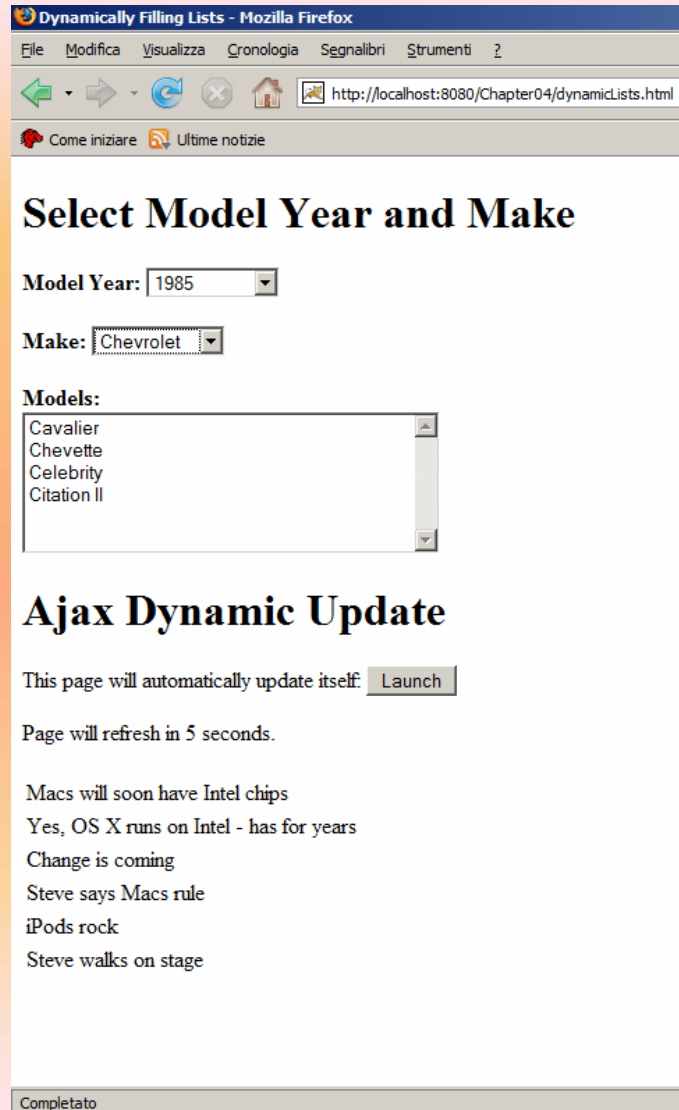
STUDIO DI UN'APPLICAZIONE (6)

IL DIAGRAMMA DI STATO



STUDIO DI UN'APPLICAZIONE (7)

APPLICAZIONE - SCHERMATA FINALE



Dynamically Filling Lists - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://localhost:8080/Chapter04/dynamicLists.html

Come iniziare Ultime notizie

Select Model Year and Make

Model Year: 1985

Make: Chevrolet

Models:

- Cavalier
- Chevette
- Celebrity
- Citation II

Ajax Dynamic Update

This page will automatically update itself:

Page will refresh in 5 seconds.

Macs will soon have Intel chips
Yes, OS X runs on Intel - has for years
Change is coming
Steve says Macs rule
iPods rock
Steve walks on stage

Completato

STUDIO DI UN'APPLICAZIONE (8)

VALUTAZIONE DELLE METRICHE

FUNZIONE	LLOC	NOA		CC	NOC	DLOOP
		var	DOM			
createXMLHttpRequest	6	-	1	3	2	-
refreshModelList	11	3	5	3	2	2
createQueryString	2	1	-	1	1	-
handleStateChange	5	-	2	3	3	1
updateModelsList	10	3	6	2	2	1
clearModelsList	4	-	3	2	2	-
doStart	3	1	3	1	1	1
startCallback	6	-	3	3	3	2
pollServer	5	1	3	1	1	2
refreshTime	10	4	2	2	2	1
pollCallback	14	4	7	4	4	3
createRow	6	3	5	1	1	-

STUDIO DI UN'APPLICAZINE (9)

UNITÀ FUNZIONALI	TOTAL LLOC	TOTAL NOA		NOP	TCC	DCOND max	NOC max	DECDE NS
		var	DOM					
select_modelYear_on change	38	7+1	17	6	9	3	2	0,24
select_make_onchang e	38	7+1	17	6	9	3	2	0,24
input_go_onclick	44	13+ 1	24	7	9	4	3	0,20

Numero di gestori eventi JS	Numero di unità funzionali	Num. Oggetti del DOM indipendenti
3	3	2

STUDIO DI UN'APPLICAZIONE (10)

VALUTAZIONI CONCLUSIVE