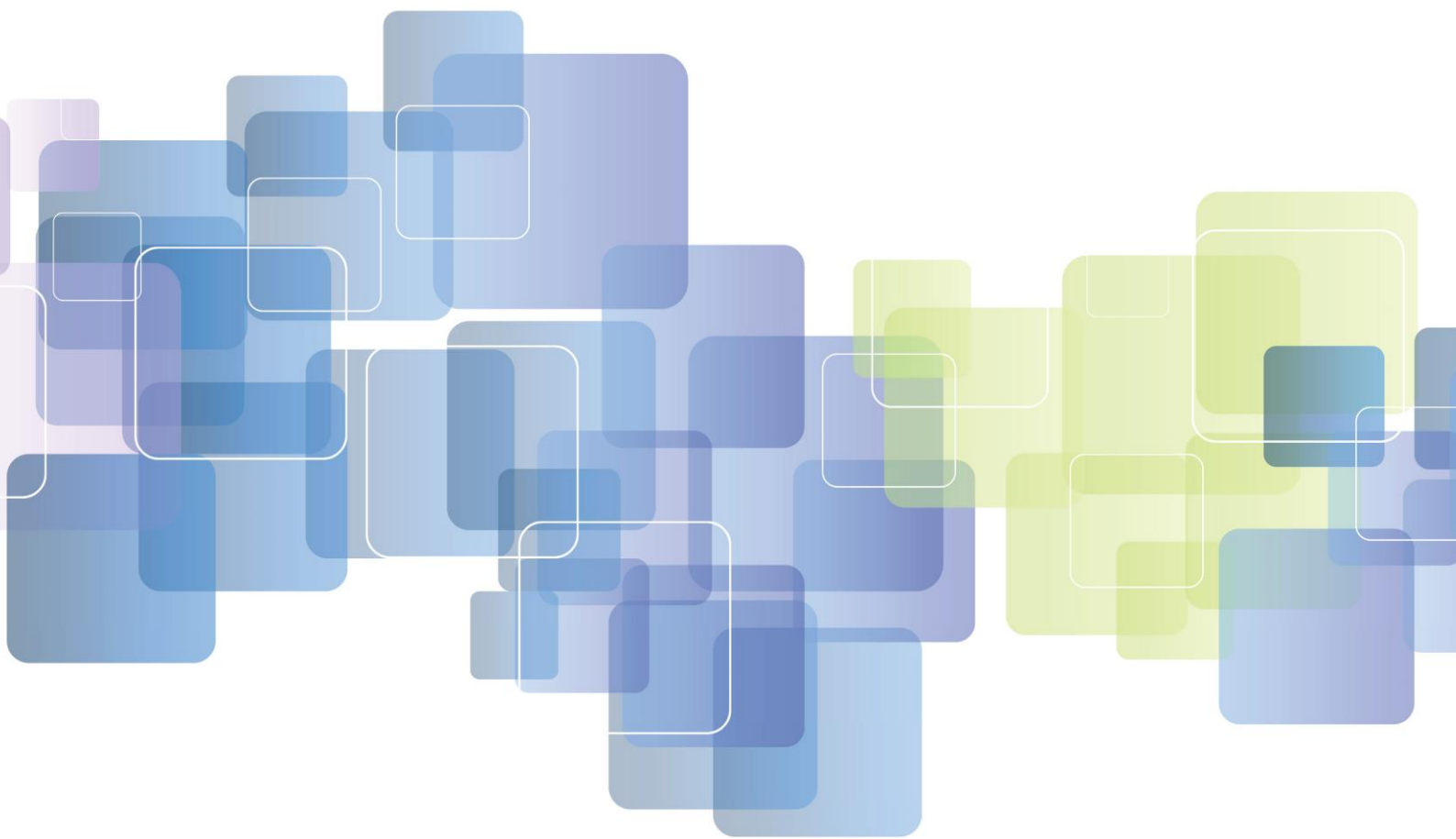


# Implementazione di MVC

**Gabriele Pellegrinetti**



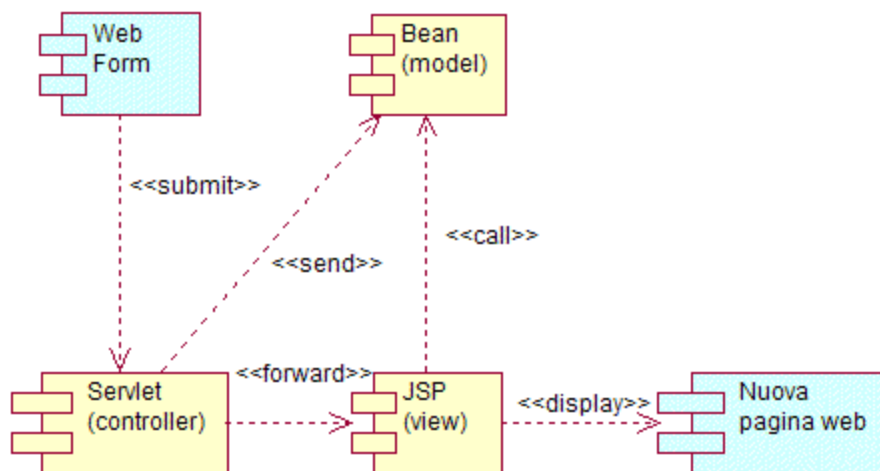
# Come implementare il pattern Model View Controller con le tecnologie JSP, ASP e XML

## Implementazione del pattern MVC in Java (JSP Model 2)

SUN è stato il primo produttore a suggerire di utilizzare il pattern MVC per lo sviluppo di applicazioni web tramite la piattaforma Java2 Enterprise Edition (J2EE).

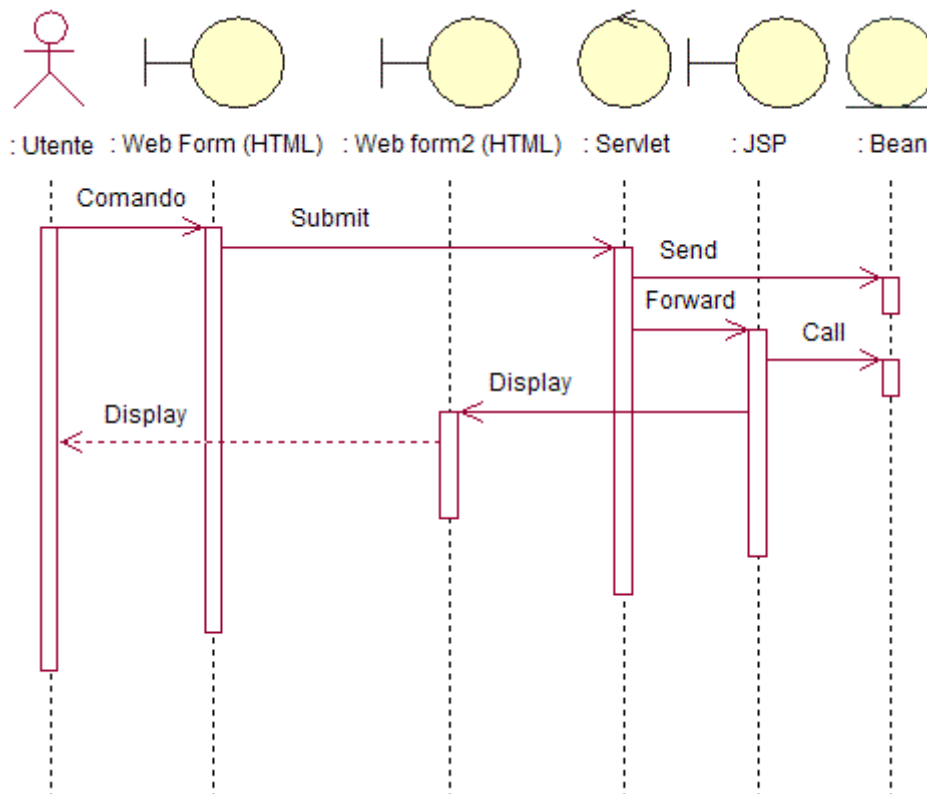
L'applicazione viene strutturata nel seguente modo:

- le view vengono implementate mediante l'utilizzo della tecnologia **Java Server Page** che consente di disegnare gli oggetti di presentation tramite il linguaggio HTML e di inserire all'interno di essi il codice Java da eseguire lato server. Questo codice non conterrà nessuna funzione di controllo o di accesso ai dati ma solo le funzionalità di composizione e dinamicizzazione delle pagine.
- il modello viene implementato tramite l'utilizzo di **Java Beans** quando le componenti sono locali all'application server che deve gestire l'applicazione o, tramite **Enterprise Java Beans** quando le componenti sono distribuite su più application server. All'interno dei bean verranno implementate tutte le funzioni di accesso ai dati e le regole di business per il loro trattamento. All'interno dei beans non vengono inserite funzioni di controllo o di presentation.
- gli oggetti di controllo vengono implementati tramite l'utilizzo delle **servlet** che avranno il compito di rimappare le richieste dell'utente, eseguite sulle mappe generate dalle JSP, in azioni che verranno eseguite dai beans.



il funzionamento è il seguente:

1. l'utente interagisce con una pagina web (web form) e specifica un comando. La pagina web invia al server il comando utente (<< submit >>);
2. sul server viene mandata in esecuzione la servlet di controllo che richiede ai bean (che rappresentano il modello) di eseguire le operazioni di business richieste dall'utente (<< send >>);
3. la servlet di controllo seleziona la view opportuna per presentare l'output all'utente e manda in esecuzione la rispettiva Java Server Page (<< forward >>);
4. la JSP recupera dal bean i dati aggiornati da visualizzare (<< call >>) quindi li elabora e genera come output una pagina HTML;
5. la pagina HTML generata dalla JSP viene inviata all'utente (<< display >>).



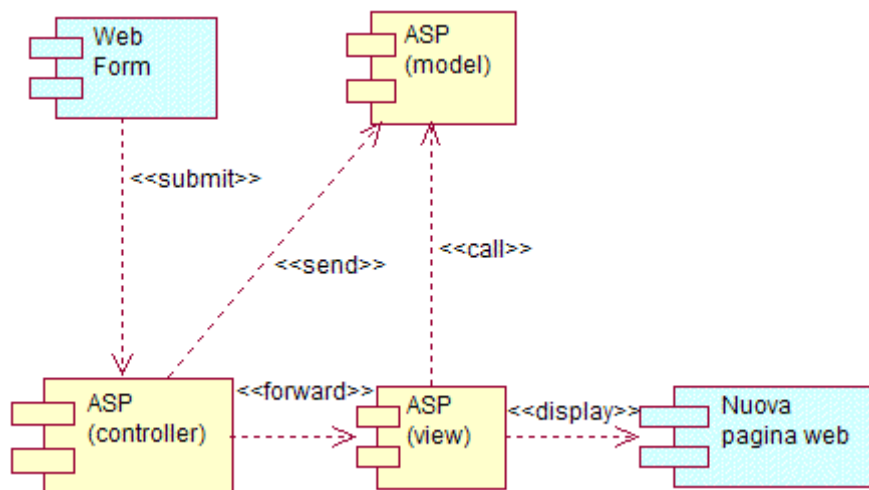
Nella comunità open source esistono diversi framework che consentono all'utente di implementare applicazioni Java tramite l'utilizzo di MVC. Tra di essi troviamo l'ottimo STRUTS del progetto Jakarta gestito dall'Apache Software Foundation (<http://jakarta.apache.org/struts/index.html>).

### Implementazione del pattern MVC in tecnologia Microsoft

Prima dell'uscita della nuova tecnologia Microsoft .NET, che integra al suo interno delle componenti che semplificano l'implementazione di applicazioni tramite questo pattern (come la nativa separazione, all'interno delle pagine ASP.NET, fra le componenti di

presentation e le componenti di controllo), l'utilizzo di MVC con la tecnologia di Redmond era abbastanza difficoltoso. Il modo più semplice per implementare MVC tramite Active Server Page era quello di implementare tutte le componenti tramite file ASP e file di testo contenenti classi e procedure in Visual Basic Script. Più in dettaglio:

- si implementano le view tramite l'utilizzo di pagine ASP prive di codice di controllo e di codice di business. Queste ASP conterranno solamente codice per la generazione della user interface;
- si implementa il controller tramite l'utilizzo di una pagina ASP priva di codice HTML. Questa pagina conterrà al suo interno una o più procedure in VbScript che avranno il compito di gestire la logica di controllo dell'applicazione e il mapping delle richieste utente in azioni del modello;
- si implementa il model tramite la scrittura di classi in VbScript che verranno richiamate dalle pagine ASP di controllo e di visualizzazione.



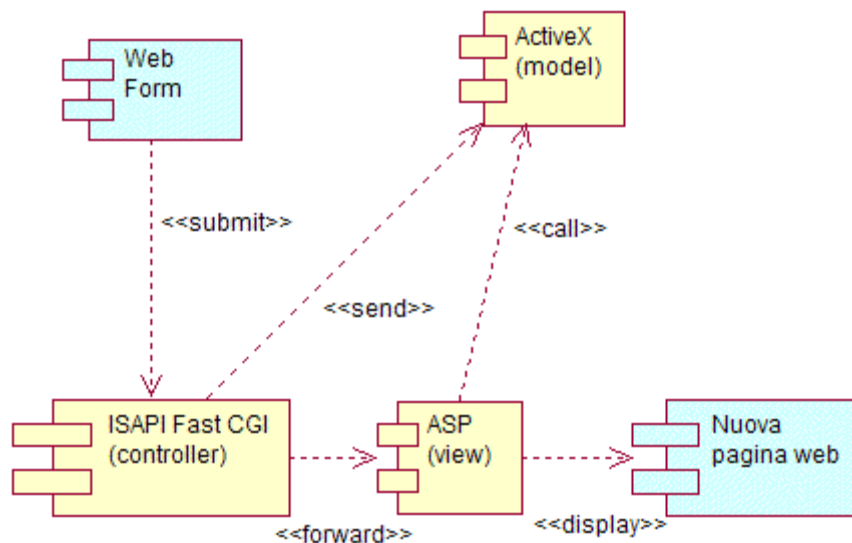
Il meccanismo di interazione fra le componenti è lo stesso di quello descritto per la tecnologia Java (cambia solo il nome delle componenti).

La soluzione precedente non è ottimale in quanto si utilizza una tecnologia nata per realizzare oggetti di presentation (le ASP) come supporto per l'implementazione di oggetti di controllo. Le classi VbScript, inoltre, non sono il meccanismo migliore per implementare oggetti persistenti. La soluzione consiste nell'utilizzare, per implementare gli oggetti di controllo e di accesso ai dati, i normali linguaggi di programmazione prodotti da Microsoft che producono oggetti compilati. In particolare:

- le view vengono implementate mediante l'utilizzo della tecnologia **Active Server Page** che consente di disegnare gli oggetti di presentation tramite il linguaggio HTML e di inserire all'interno di essi il codice VbScript da eseguire lato server. Questo codice non conterrà nessuna funzione di controllo o di accesso ai dati ma solo le funzionalità di composizione e dinamicizzazione delle pagine.
- il modello viene implementato tramite l'utilizzo di **ActiveX** costruite secondo la specifiche **COM** di Microsoft. All'interno degli ActiveX verranno implementate tutte le funzioni di accesso ai dati e le regole di business per il loro trattamento.

Il modo più semplice per implementare l'ActiveX è quello di utilizzare l'ambiente di sviluppo Visual Basic.

- gli oggetti di controllo vengono implementati tramite la creazione di fast CGI in tecnologia ISAPI che avranno il compito di rimappare le richieste dell'utente, eseguite sulle mappe generate dalle ASP, in azioni che verranno eseguite dagli ActiveX. L'utilizzo di questa tecnologia è più complessa rispetto ai corrispondenti servlet Java ma consente di ottenere performance superiori in quanto le componenti ISAPI sono compilate. Il modo più semplice per implementare la componente ISAPI è quello di utilizzare l'ambiente di sviluppo Visual Basic.

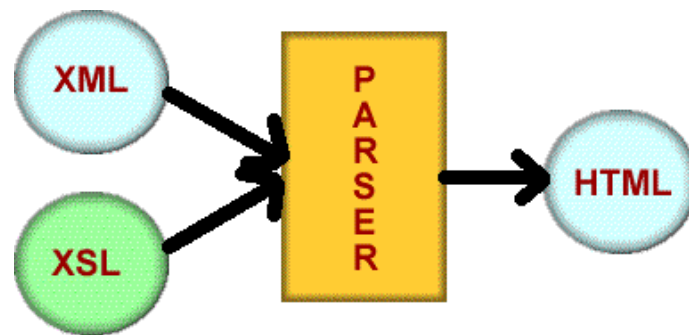


Il meccanismo di interazione fra le componenti è lo stesso di quello descritto per la tecnologia Java (cambia solo il nome delle componenti).

### MVC tramite XML

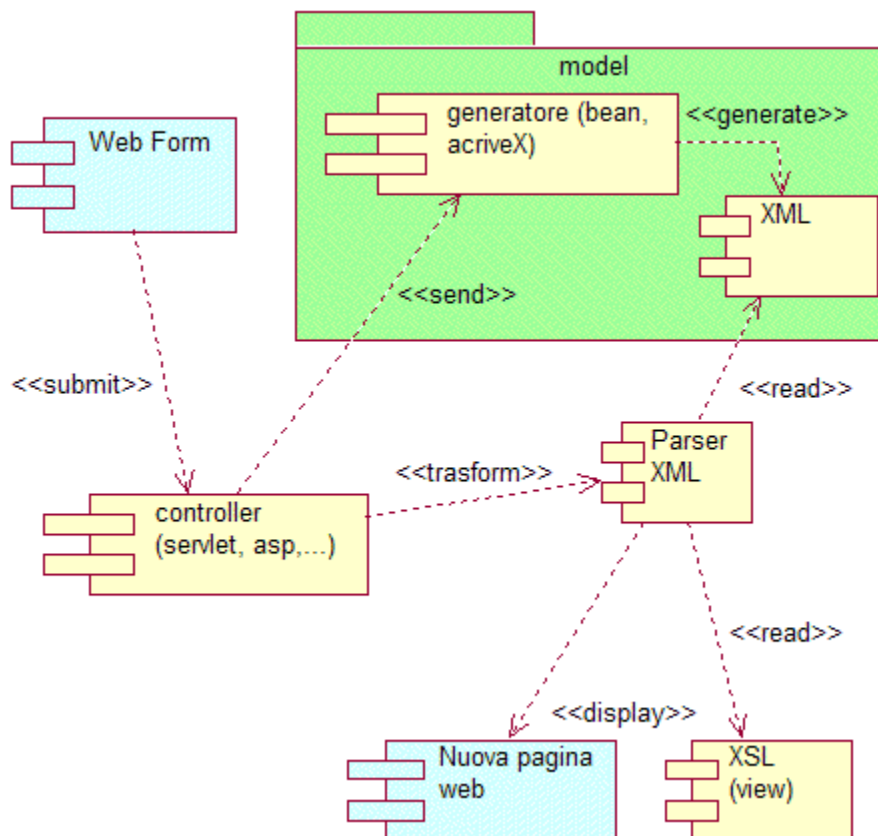
L'utilizzo della tecnologia basata su XML consente di creare applicazioni in grado di trasformare, tramite l'applicazione di un opportuno foglio di stile, un flusso dati, espresso in XML (eXtended Markup Language), in un qualsiasi altro formato. Per eseguire questa operazione le applicazioni si appoggiano principalmente su tre diverse tipologie di oggetti:

- i file XML che rappresentano i dati da proporre all'utente;
- i fogli di stile XSL che consentono di descrivere le regole di trasformazione che convertono il file XML in un altro formato (ad esempio in HTML)
- il parser che, presi in input il file XML contenente i dati e il foglio di stile XSL, esegue la trasformazione nel nuovo formato.



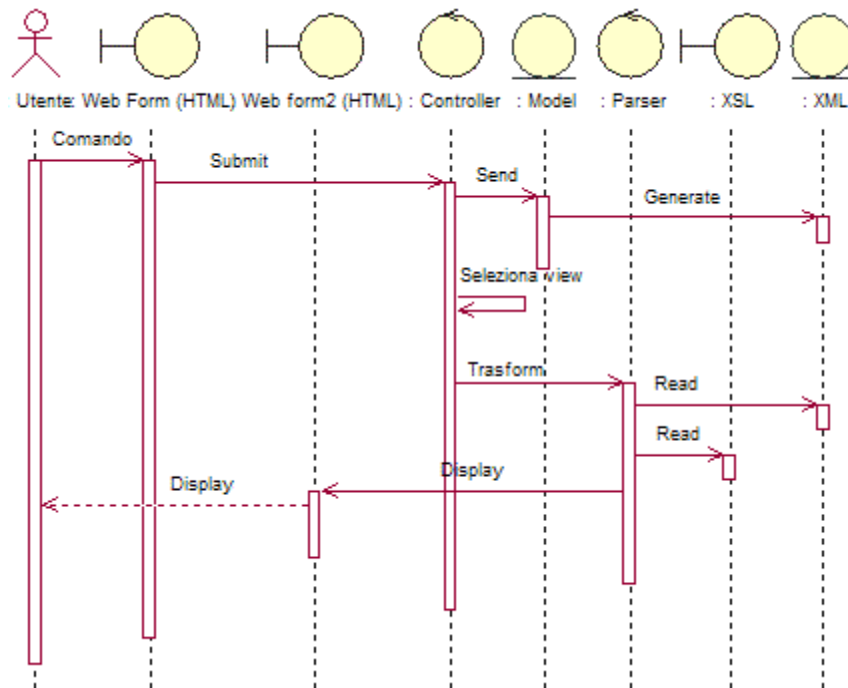
L'utilizzo di XML consente di implementare il pattern MVC strutturando l'applicazione nel seguente modo:

- le view vengono implementate mediante l'utilizzo di fogli di stile XSL. Visto che si vuole implementare una presentation in HTML si utilizzeranno le specifiche XSLT.
- il modello viene implementato tramite l'utilizzo di una componente (**ActiveX, bean,...**) che ha lo scopo di generare i file XML contenenti i dati. All'interno di queste componenti verranno implementate tutte le funzioni di accesso ai file XML e le regole di business per il loro trattamento.
- gli oggetti di controllo vengono implementati tramite le tecnologie tradizionali descritte in precedenza (servlet, ISAPI,...) e avranno il compito di rimappare le richieste dell'utente, eseguite sulle mappe generate dalle view, in azioni che verranno eseguite dagli dalle componenti del model. Sarà compito del controller invocare il parser per eseguire la trasformazione dell'XML in pagine web.



il funzionamento è il seguente:

1. l'utente interagisce con una pagina web (web form) e specifica un comando. La pagina web invia al server il comando utente (<< submit >>);
2. sul server viene mandato in esecuzione l'oggetto di controllo che richiede alle componenti che costituiscono il model di eseguire le operazioni di business richieste dall'utente (<< send >>);
3. le componenti che costituiscono il modello generano, se necessario, i dati in formato XML (<< generate >>);
4. la servlet di controllo seleziona il foglio di stile XSL (view) necessario per presentare l'output all'utente e chiede al parser di eseguire la trasformazione. Al parser vengono passati i riferimenti ai dati XML e al foglio di stile (<< trasform >>);
5. Il parser legge i dati XML e il foglio di stile XSL (<< read >>) e genera come output una pagina HTML;
6. la pagina HTML generata dalla parser viene inviata all'utente (<< display >>).





**Tecnet Dati s.r.l.**  
C.so Svizzera 185 -  
10149 - Torino (TO), Italia  
Tel.: +39 011 7718090 Fax.: +39 011 7718092  
P.I. 05793500017 C.F. 09205650154  
[www.tecnetdati.com](http://www.tecnetdati.com)

