

Una metodologia di progettazione di applicazioni web centrate sui dati



ALBERTO BELUSSI
ANNO ACCADEMICO 2009/2010

Progettazione logica di un sito centrato sui dati



Si propone la seguente metodologia di progetto.

FASI della metodologia

1. Organizzazione in pagine web del contenuto informativo
2. Navigazione tra le pagine web (definizione dei LINK)
3. Corrispondenza tra il contenuto delle pagine e la basi di dati (interrogazioni SQL)
4. Presentazione (aspetto grafico delle pagine: fogli di stile CSS)

Progettazione logica - FASE 1

La struttura delle pagine web e il loro contenuto informativo possono essere specificati definendo SCHEMI di PAGINA.

Linguaggio per la specifica di **schemi di pagina**:

```
page schema <nomeSchema> [unique] (  
    <nomeAttributo>: <TIPO>;  
    ...  
    <nomeAttributo>: <TIPO>;  
)  
<TIPO> ::= {string, integer, date, time,  
            real, <LISTA>}  
<LISTA> ::= list_of (<nomeAttr>: <TIPO>; ...  
                    <nomeAttr>: <TIPO>;)
```

Progettazione logica – FASE 2

La specifica dei legami tra pagine si ottiene aggiungendo i **link** tra gli schemi di pagina. In particolare, un link è un tipo specifico che si aggiunge agli altri possibili tipi di attributo.

```
<TIPO> ::= {string, integer, date, time, real,  
            <LISTA>, <LINK>}  
<LINK> ::= {link(<ETICHETTA>; *<nomeSchema>) |  
            link(<ETICHETTA>; url(<nomeURL>) |  
            link(<ETICHETTA>;  
                url(<nomeURL>, <nomeAttrib>)) }  
<ETICHETTA> ::= {<nomeEtichetta> |  
                <nomeAttrib>: {string | integer | date | time}}
```

Progettazione logica – FASE 3

La terza fase prevede la specifica della **corrispondenza** tra la base di dati e il contenuto informativo degli schemi di pagina. Nella sintassi proposta tale corrispondenza si precisa per ogni schema di pagina *<nomeSchema>* come segue:

DB to page schema *<nomeSchema>*

parameter (*<nomepar>*, ..., *<nomepar>*)

```
(  
    <nomeAttr>, ..., <nomeAttr>: <QUERY_SQL>;  
    ...  
    <nomeAttr>, ..., <nomeAttr>: <QUERY_SQL>;  
)
```

dove i parametri dovranno comparire nelle interrogazioni SQL con la sintassi *?nomepar?* e gli attributi *<nomeAttr>* devono essere tutti e soli gli attributi dello schema di pagina *<nomeSchema>*.

Risultato della progettazione logica

- **Schema logico della base di dati:**
 - schema relazionale: tabelle + vincoli di integrità
- **Elenco di schemi di pagina:**
 - schema del sito: page schema + link
- **Specifica della corrispondenza con la basi di dati:**
 - Interrogazioni SQL sullo schema relazionale mappati sul contenuto informativo degli schemi di pagina.

Implementazione?

- Quale tecnologia usare?
- Quale corrispondenza tra la progettazione logica e l'implementazione?

Tecnologie disponibili: **Servlet**, **JSP**, PHP, ASP, **postgresql**, Mysql, ecc...

Architetture software: approccio Model-View-Controller, MCV-2 servlet-centric.

Architetture software per applicazioni web

Approccio banale:

- Una servlet (o JSP) per ogni schema di pagina che si occupa di tutti gli aspetti:
 - ✦ Elaborazione della richiesta HTTP.
 - ✦ Connessione alla base di dati ed estrazione delle informazioni.
 - ✦ Generazione del codice HTML da restituire al browser.

Approccio più evoluto:

- Architettura MVC-2 (servlet centric): distribuire le varie funzionalità a diversi "attori" (moduli software = classi JAVA).