

PREMESSA

Nell'ambito del progetto Araneus, condotto dal gruppo di ricerca di Basi di Dati dell'Università di Roma Tre, è stato sviluppato il prototipo Penelope, che consente la generazione di siti Web a partire da dati memorizzati in una base di dati relazionale.

Oggetto principale di questa tesi è stato lo studio di estensioni e la reingegnerizzazione dell'intero prototipo Penelope, con particolare attenzione all'efficienza e alla portabilità.

Penelope si basa su un linguaggio dichiarativo (Penelope Language) che permette di definire la struttura di un sito e le relazioni con i dati contenuti nella base di dati; a partire da quanto specificato attraverso il linguaggio, il sistema genera tutte le pagine del sito in file, nei formato HTML o XML.

Il prototipo iniziale, sebbene molto robusto ed efficace per la generazione di siti statici, presenta numerose limitazioni:

1. il sistema è stato concepito come generatore di siti web statici, nel senso che esso implementa il sito generandone tutte le pagine in file di testo. Questo aspetto implica una inefficiente gestione del sito, in quanto, ogni qualvolta viene aggiornato il contenuto informativo della base di dati sottostante, occorre rigenerare l'intero sito (o quanto meno parte di esso);
2. la generazione delle pagine si basa su un algoritmo che richiede la materializzazione di numerose tabelle temporanee; questa scelta risulta essere molto inefficace in quanto richiede modifiche alla struttura della base di dati;
3. il sito viene generato esclusivamente attraverso dati memorizzati in una base di dati relazionale; può risultare molto utile prevedere che le informazioni possano essere reperite da dati generati dinamicamente da altri programmi o da file di testo;

4. infine, la generazione del sito richiede la disponibilità dell'intero prototipo nel caso di aggiornamenti della base di dati.

Per ovviare a queste limitazioni si è manifestata l'esigenza di creare un nuovo prototipo che permetta di generare non solo direttamente le pagine HTML o XML del sito, ma anche in alternativa programmi che, una volta eseguiti, consentano a loro volta di generare il sito. In particolare, per rispondere appieno alle limitazioni discusse precedentemente, questi programmi devono essere in grado di generare sia siti statici, che siti dinamici; per siti dinamici intendiamo siti le cui pagine vengono generate dinamicamente sul lato server, a seguito di una richiesta del client; non sono quindi richieste particolari funzionalità aggiuntive al browser, il quale deve semplicemente essere in grado di interpretare HTML/XML standard.

L'obiettivo perseguito nella realizzazione di questa tesi è stato quello di studiare e implementare nuove soluzioni che hanno portato ad una profonda reingegnerizzazione ed estensione del prototipo iniziale. In particolare è stato studiato e realizzato un nuovo approccio: Penelope non è più solo un generatore di siti web ma è anche un generatore di programmi.

Per la generazione dinamica di pagine web si è scelta la possibilità di creare Server Pages, ovvero pagine web all'interno delle quali è possibile inserire codice: questo codice verrà poi gestito dal server che provvederà alla sua compilazione ed esecuzione. Le Server Pages sono pagine che vengono interpretate da un server http, esteso attraverso opportune librerie. In particolare si è scelto di generare pagine JSP (Java Server Page) e ASP (Microsoft Active Server Page), per portabilità ed efficienza. In ogni caso l'algoritmo proposto, nella sua generalità, è applicabile per la generazione di qualsiasi tipo di pagine dinamiche (ad esempio PHP).

Al fine di rendere qualsiasi sito generato con Penelope portabile su qualsiasi piattaforma, per la generazione di siti a contenuto statico si è scelto di generare programmi Java che, una volta compilati ed eseguiti, permettono la costruzione dell'intero sito web.

La realizzazione del nuovo prototipo Penelope ha richiesto uno studio approfondito

e la rivisitazione delle strutture dati esistenti, e quindi l'introduzione di nuove strutture e nuovi algoritmi di generazione automatica del codice.

Nella generazione di siti dinamici si è inoltre dovuto tener conto di come gestire efficacemente le connessioni alla base di dati, situazione che può diventare critica per le prestazioni complessive del sistema, quando numerosi utenti accedono al sito contemporaneamente. Per gestire questo problema si è studiata ed implementata una soluzione che permette la gestione dinamica di un "pool" di connessioni, attraverso il quale è possibile specificare sia il massimo numero di connessioni che si vogliono per il sito, sia il numero di connessioni che si desidera utilizzare per gestire un elevato numero di richieste concorrenti.

Con il nuovo prototipo è possibile anche generare siti misti, siti cioè in parte statici e in parte dinamici; ciò è molto conveniente per siti che contengano pagine che sono aggiornate di frequente, e che quindi sono di tipo dinamico, e pagine che viceversa sono aggiornate a lungo termine, quindi statiche. La maggiore difficoltà nel generare siti misti è stata incontrata nella gestione dei link da una pagina di un tipo ad una pagina di un altro tipo, basti pensare ad una pagina statica in cui sono presenti link a pagine dinamiche o viceversa.

Un'ulteriore importante estensione che è stata studiata ed implementata per rendere il prototipo Penelope più flessibile, è stata quella di prevedere che i dati, oltre a provenire da basi di dati relazionali, possano provenire da sorgenti dati dinamiche, ad esempio generate su richiesta da programmi esterni; si fa notare che questa funzionalità diventa particolarmente interessante nel caso in cui questi programmi siano Servlet Java, in quanto questi possono essere invocati dal server http sul quale risiede il sito, pur trovandosi in un qualsiasi punto del web.

Al fine di integrare i dati provenienti da sorgenti esterne con le informazioni che Penelope estrae dalla base di dati è stato scelto un opportuno formato di scambio, in XML; l'implementazione di una opportuna classe permette di gestire con un elevato livello di astrazione questi dati in modo trasparente.

La tesi è articolata in tre parti.

Nella Prima Parte vengono presentati i concetti, proposti nell'ambito del progetto Araneus, su cui si basa la generazione di un sito web: nel Capitolo 1 viene discusso il modello dei dati ADM (Araneus Data Model), che è usato dal prototipo Penelope per definire la struttura di un sito web; il Capitolo 2 illustra il linguaggio PL (Penelope Language).

Nella Seconda Parte vengono presentate le strutture dati, le estensioni e gli algoritmi sviluppati che hanno portato alla definizione del nuovo prototipo Penelope: il Capitolo 3 illustra il nuovo prototipo Penelope; il Capitolo 4 presenta le strutture dati e gli algoritmi utilizzati per la generazione di programmi; il Capitolo 5 descrive il pattern per la gestione del "pool" di connessioni alla base di dati e come sia possibile utilizzare dati provenienti da sorgenti esterne. Vengono inoltre descritte due interfacce user-friendly utilizzate sia per generare programmi attraverso Penelope sia per generare siti statici a partire da programmi Java generati. Nel Capitolo 7 vengono mostrati alcuni esempi, attraverso i quali si è testato il nuovo prototipo Penelope.

Nella Terza Parte vengono riportate, nel Capitolo 8, le conclusioni e gli sviluppi futuri del prototipo.

Tre appendici completano la tesi. Le appendici A e B illustrano i concetti base relativi alle pagine JSP e ASP, rispettivamente. Nell'appendice C viene brevemente descritto il linguaggio di mark-up XML.