

ULC – Ultra Light Client

La libreria software che abilita Swing con la caratteristica “thin client”

Lo sviluppo e la distribuzione di applicazioni software deve oggi adeguarsi ad un contesto di business che si modifica repentinamente. Le applicazioni web risolvono i problemi di distribuzione e di partitioning. I browser web come front-end generici per le applicazioni J2EE sono l'interfaccia utente attuale per le applicazioni aziendali fruibili attraverso le Intranet e Internet.

Tuttavia, con HTML è troppo oneroso o addirittura impossibile sviluppare sofisticate interfacce utente di tipo “rich client”, cioè con un elevato grado di interattività. La libreria Swing di Java fornisce funzionalità di tipo “rich client”, ma il suo modello di programmazione client è di tipo “fat”, cioè richiede un'installazione sul client, e per tale motivo non può essere agevolmente integrato nell'architettura J2EE.

ULC facilita lo sviluppo di applicazioni di tipo “rich client” in grado di coprire il gap, ovvero di dare un supporto di tipo “thin client” a Swing e contemporaneamente mantenere una architettura J2EE lato server che consente di mantenere l'intera applicazione lato server, senza installazioni specifiche per ogni applicazione sul client.



Esigenza

L'esigenza è quella di propagare il modello di applicazione web, alle applicazioni client-server più in generale. Lo scopo è duplice:

- 1) una architettura lato server semplifica lo sviluppo, in quanto il tema complesso del “software partitioning” tra lato client e lato server è risolto in modo generico e trasparente, inoltre
- 2) ridurre il software sul lato client significa diminuire i costi di distribuzione dell'applicazione. I client di tipo “fat” aumentano anche i costi di sviluppo, perché lasciano l'incombenza del partitioning dell'architettura software agli sviluppatori. Questi ultimi devono tenere sotto controllo l'ottimizzazione del network, il caching, la sincronizzazione client-server ecc.. Inoltre, ad ogni nuova release il software lato client deve essere aggiornato.

I browser web sono i client di tipo J2EE più diffusi. I servlet e altri framework come Struts sono dotati di un modello di programmazione lato server che solleva gli sviluppatori dalla maggior parte dei problemi legati al partitioning come l'ottimizzazione del network e la sincronizzazione client-server. Questo funziona perfettamente per interfacce utente semplici e basate su form, ma si tratta di soluzioni poco scalabili per interfacce utente sofisticate. Aggiungere tecnologie come JavaScript e XML/XSLT può migliorare in qualche caso le interfacce grafiche. Ma, ancora una volta, queste tecnologie complicano il lavoro degli sviluppatori e aumentano i costi di sviluppo e manutenzione. Nelle situazioni peggiori, l'uso massiccio di JavaScript porta a client di tipo “fat”, con le tipiche dipendenze dal browser che sono ovviamente indesiderate.

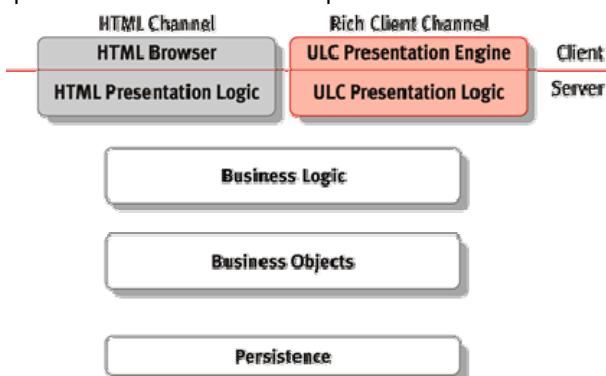
Soluzione

ULC estende le capacità di Swing a tal punto che le componenti di interfaccia grafica abilitate da ULC si integrano con il modello di programmazione lato server delle applicazioni web. ULC è basato interamente su standard come Java, Swing, J2EE, HTTP(S), o RMI/IIOP. Questo è un fatto molto rilevante per gli sviluppatori: possono infatti sviluppare client di tipo “rich” basandosi su API complete e omogenee, utilizzando un'unica tecnologia.

Le architetture basate sull'approccio della presentazione lato server, in cui anche la logica di presentazione viene gestita sul server, sono la scelta migliore per le nuove applicazioni sia per Internet sia per le Intranet. Le piattaforme basate su HTML che utilizzano i servlet o Java Server Pages sono in effetti la soluzione più utilizzata, nonostante i loro svantaggi rilevanti.

Questi stessi concetti hanno guidato allo sviluppo di ULC. ULC enfatizza i vantaggi dell'approccio di tipo “presentation server” e contemporaneamente consente di evitare gli svantaggi di HTML e JavaScript sul lato client.

La comunicazione tipicamente viene fruita tramite il protocollo HTTP(S), ma può essere sostituita all'occorrenza da altri protocolli per soddisfare altri requisiti di sicurezza o efficienza.



PRINCIPALI VANTAGGI

Aumento della produttività degli sviluppatori, migliore livello di interattività dell'applicazione, possibilità di riutilizzo della infrastruttura attuale, rapidità nello sviluppo e installazione di applicazioni web, compatibilità con tutti i sistemi operativi, costi di formazione ridotti sono solo alcuni degli elementi di vantaggio che contraddistinguono la soluzione ULC.

ULC: caratteristiche principali

ULC è una libreria che aggiunge funzionalità di tipo “rich client” ad applicazioni J2EE di tipo “thin client”, seguendo l’approccio di “presentation server”.

In particolare, ULC fornisce supporto di tipo “thin client” a Swing, la libreria che è un componente standard per i client Java di tipo “rich”. ULC supporta un modello di programmazione lato server omogeneo: gli sviluppatori possono utilizzare le componenti Swing abilitate da ULC come se risiedessero sul lato server. Il deployment è possibile in ogni servlet container o application-server compatibile con J2EE. Sul lato client, ULC utilizza un motore di presentazione indipendente basato su J2SE.

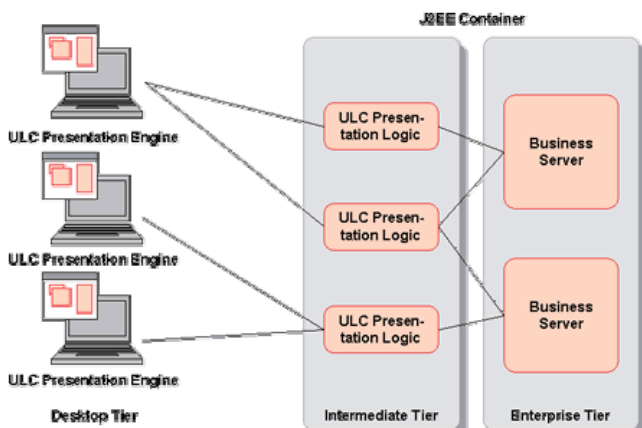
La distribuzione del layer di presentazione e le ottimizzazioni dell’utilizzo della rete sono trasparenti per gli sviluppatori software, che non devono quindi preoccuparsi della separazione client-server, gestita da ULC.

ULC influenza unicamente il layer di presentazione dell’applicazione e di conseguenza, in un eventuale scenario multi-canale, sarebbe in grado di integrare perfettamente HTML.

Un’applicazione basata su ULC può essere installata come una servlet in un servlet container, oppure, in alternativa, può essere installata anche come session bean, sfruttando le caratteristiche di gestione più avanzate dei contenitori EJB.

Il lato client di una applicazione basata su ULC è gestito attraverso un motore di presentazione generico, che ha una funzione simile a quella del browser per un’applicazione web: viene installato una sola volta e viene utilizzato con tutte le applicazioni.

ARCHITETTURA



ULC segue l’approccio descritto qui a fianco. Gestisce la logica di presentazione sul server e ha un motore di presentazione generico sul client.

Quest’ultimo è implementato in Java e utilizza la libreria Swing per presentare l’interfaccia utente e gestire gli input degli utenti. Il motore di presentazione è di piccole dimensioni (~500 KB) e può essere fatto girare come applicazione Java (sia standalone che utilizzando il meccanismo di Java Web Start) o come applet.

Lo stato della presentazione è mantenuto sul lato server, così come l’intera logica dell’applicazione.

Gli elementi grafici (widget) di ULC fanno da ponte tra l’applicazione (lato server) e Swing (lato client). Gestiscono al loro interno la comunicazione,

permettendo caching e minimizzando il traffico di rete e presentano al resto dell’applicazione una API di utilizzo molto simile a Swing. Ciò che ne facilita l’apprendimento: l’esperienza dimostra infatti che gli sviluppatori che hanno familiarità con Swing diventano produttivi in tempi rapidissimi.

REQUISITI TECNICI

ULC necessita di Java 2 (JRE o JDK 1.3 o successiva), ad esempio, J2SE per la UI Engine sul lato client e J2EE l’applicazione ULC lato server.

Di norma, ULC è compatibile con J2EE e può di conseguenza essere installata su ogni servlet compatibile con J2EE o container EJB.

Il Gruppo Tinext

Tinext SA è attiva nel campo dei servizi informatici e si rivolge alle imprese private e alle organizzazioni pubbliche. La società ha sede in Svizzera (sede centrale) e in Italia.

Tinext offre soluzioni software personalizzate basate su tecnologie all’avanguardia e strumenti come Java, J2EE, JBoss, Eclipse.

La metodologia dell’Extreme Programming e il suo approccio di pianificazione di tipo incrementale consente a Tinext di sviluppare soluzioni di successo e migliorare la propria capacità di soddisfare richieste che mutano rapidamente in base alle esigenze di business.

Gli ingegneri software di Tinext hanno acquisito una solida esperienza pluriennale nel campo della consulenza, dello sviluppo software e del project management di soluzioni personalizzate e di soluzioni di Enterprise Content Management (ECM).



Per maggiori informazioni: sales@tinext.com