

I WEB SERVICE IN LABVIEW

Questo articolo spiega i Web Service LabVIEW, una nuova opzione in LabVIEW 2010 aggiunta in risposta alle richieste dei clienti di un modo più aperto e standard per comunicare con VI attraverso il Web

Elijah Kerry

Le applicazioni spesso richiedono interfacce di rete per la comunicazione machine-to-machine, il monitoraggio remoto o il telecontrollo di un dispositivo embedded. La proliferazione di Internet e dell'infrastruttura di rete semplifica il compito di connettere hardware distribuito su reti fisiche esistenti. Il software LabVIEW vi offre una varietà di opzioni per stabilire una comunicazione fra applicazioni software.

VANTAGGI DEI WEB SERVICE LABVIEW

Consideriamo un'applicazione LabVIEW installata su un sistema distribuito. LabVIEW offre proprietà come le variabili condivise per stabilire la comunicazione, ma molti sviluppatori hanno bisogno di un modo per comunicare con queste applicazioni da dispositivi sui quali LabVIEW non usa la comunicazione Web-based standard. Con i Web service LabVIEW, potete:

- Comunicare con applicazioni LabVIEW embedded da qualsiasi dispositivo Web-enabled
- Stabilire una comunicazione machine-to-machine usando protocolli HTTP standard
- Monitorare e controllare da remoto applicazioni LabVIEW usando thin client custom

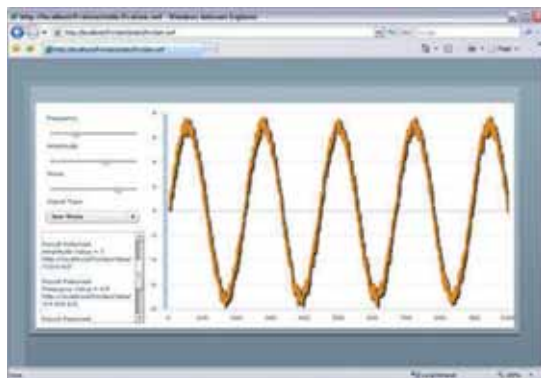


Fig. 1 - Questo schermo è un esempio di applicazione thin-client sviluppata usando Adobe Flex, che può legare i dati ricevuti da un VI LabVIEW a un'interfaccia utente Web-based

- Eseguire lo streaming di qualsiasi tipo di dati MIME standard, come testo, immagini e video
- Installare VI Web service su un target Windows o LabVIEW Real-Time

Usando i Web service, potete ora installare VI LabVIEW in un'applicazione distribuita, come quella descritta sopra, per la comunicazione sul Web. Potete facilmente configurare il Web server LabVIEW in modo da ricevere le richieste da thin client eseguiti su qualsiasi dispositivo in grado di operare su Web per eseguire e controllare VI da remoto. Tuttavia, capire che creare un'interfaccia thin-client richiede esperienza in un linguaggio basato su testo per sviluppare un'interfaccia utente in grado di interpretare e visualizzare informazioni.

CHE COS'È UN WEB SERVICE?

I Web service permettono di invocare un metodo su un target remoto usando protocolli basati su Web standard. Un client invia una richiesta ad un server remoto, che elabora la richiesta e replica con una risposta, la quale è quindi interpretata e visualizzata dall'applicazione client. Potete basarvi su questo metodo di comunicazione per attività di routine come la navigazione sul Web, il controllo di e-mail e la lettura di articoli online.

I punti seguenti sono tutti componenti di un Web service:

Server – Un'applicazione responsabile dell'evasione di una richiesta, eseguendo il metodo o l'azione appropriati e inviando una risposta al client

Client – Un'applicazione che invia una richiesta al server e aspetta di ricevere una risposta, che è quindi interpretata dal client

Protocolli standard – I protocolli Web-based come HTTP instradano i dati su reti fisiche dal client all'appropriato metodo server e quindi nuovamente al client

Rete – Lo strato fisico, come Ethernet o IEEE 802.11, sul quale vengono trasmessi i dati.

Con il nuovo Web server LabVIEW, potete installare VI come

Web service, che invocherete tramite una richiesta da un client usando HTTP standard. A differenza di opzioni come pannelli remoti e variabili condivise di LabVIEW, i client che costruite per comunicare con un VI installato non richiedono il motore di run-time LabVIEW. Ciò significa che potete usare qualsiasi tecnologia Web-based, inclusi linguaggi comuni come HTML, JavaScript e Adobe Flash.

METODO RICHIESTA/RISPOSTA

Per capire come funziona la comunicazione con un Web service, consideriamo l'esperienza di navigare un negozio online. Ogni volta che cliccate su un link o digitate un'URL, il vostro browser (il client) invia una richiesta a un server remoto, che elabora l'informazione e invia una risposta. Non occorre che il vostro browser sappia nulla dell'infrastruttura del server o dei database usati dal negozio online; infatti, esso si limita a inter-



Fig. 2 - Le interfacce thin-client standard possono invocare VI LabVIEW

pretare la risposta e a visualizzarla all'utente.

Quando comunicate con un target embedded, potreste trovare utile correlare questo all'esperienza di configurare un router. I router forniscono interfacce Web client a cui potete accedere per configurare e impostare parametri come la crittografia wireless e il port forwarding.

Questi dispositivi stand-alone ospitano un server a bordo per utilizzare l'interfaccia. Potete usare LabVIEW nello stesso modo su dispositivi embedded per installare un VI come Web service, a cui si può comunicare da un client come un Web browser.

Questo metodo di networking non è ideale per applicazioni di streaming dei dati ad alta velocità. Il client deve fare una nuova richiesta ogni volta che vuole dati aggiornati. Pertanto, è impossibile garantire la velocità di restituzione dei dati a causa di fattori come la qualità della connessione, la banda disponibile e la prossimità alla posizione del server.

LAYER DEI WEB SERVICE

La comunicazione Web-based è attualmente composta da diversi strati di protocollo, che spaziano dallo strato fisico sul quale i dati sono effettivamente inviati agli strati usati per codificare il messaggio e instradarlo al metodo appropriato. LabVIEW astrae ogni preoccupazione riguardante il funzionamento di questi strati, benché siano disponibili VI TCP e UDP a basso livello se volete sviluppare un'applicazione in cima allo strato di trasporto.

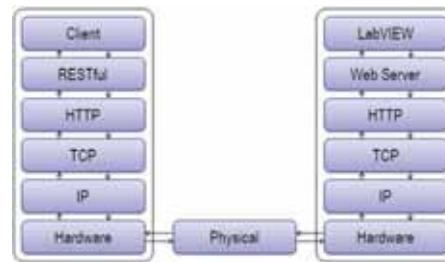


Fig. 3 - Questo diagramma mostra gli strati di protocollo standard usati per comunicare da un'applicazione client con un Web service LabVIEW.

I vari strati di protocollo usati nella comunicazione Web-based sono mostrati nella figura 3.

I Web service LabVIEW usano un'architettura RESTful in cima ai protocolli che sono ubiqui sul Web, come TCP/IP e HTTP, per offrire una comunicazione standard, amichevole nei confronti dei firewall.

NI ha scelto un'architettura RESTful perché è particolarmente utile per il software che richiede soluzioni leggere, perché involve un markup addizionale minimo.

WEB SERVER LABVIEW

In LabVIEW 2010, il Web server è stato sostituito con un HTTP Server Appweb di terza parte da Embedthis Software LLC. Questo nuovo Web server è ora usato per servire pannelli remoti e Web service. Embedthis è un fornitore leader di Web server embedded ed è stata selezionata sulla base della sua storia di fornitore di soluzioni robuste e leggere ideali per hardware embedded come NI CompactRIO e Compact FieldPoint.

ARCHITETTURA DEI WEB SERVICE

Potete usare Web service per ottenere un'interfaccia Web-based per comunicare con applicazioni LabVIEW esistenti. Notate che il VI di livello superiore di un'applicazione non è tipicamente installato come Web service; al contrario, un set addizionale di VI è responsabile delle comunicazioni con l'applicazione e del rinvio di dati alle richieste Web, come mostrato nella figura 4.

Le informazioni possono passare facilmente fra il contesto del-



Fig. 4 - I Web service possono comunicare con applicazioni esistenti attraverso variabili condivise e memoria condivisa per passare dati e parametri di configurazione a un'interfaccia Web remota

l'applicazione e il contesto del Web server usando variabili condivise o scrivendo dati in una locazione di memoria condivisa, come un file TDMS.

INVIO DI DATI DA VI WEB SERVICE

Vi sono numerose tecnologie per inviare e ricevere dati fra un'applicazione client e LabVIEW. Il modo più semplice è passare i dati attraverso il quadro connettore, che porta a dati XML standard o a semplice testo. Queste informazioni di ritorno possono essere facilmente analizzate da un client che può legare i dati a visualizzazioni più utili, come tabelle e grafici. Per comunicazioni più avanzate, una nuova palette di VI in LabVIEW 8.6 facilita:

- Lo streaming di dati
- Lo scripting sul lato server (ESP)
- Sessioni utente (cookie)

Potete trovare informazioni dettagliate su come usare questi metodi avanzati nel successivo paragrafo Lettura Avanzata o potete consultare l'Example Finder in LabVIEW.

INSTALLAZIONE DEL VOSTRO VI COME WEB SERVICE

Un VI Web service deve essere configurato, installato e gestito dalla sezione Build Specifications del LabVIEW Project Explorer. Cliccate con il tasto destro su 'Build Specifications' e selezionate New»Web Service (RESTful) per iniziare e configurare il vostro VI.

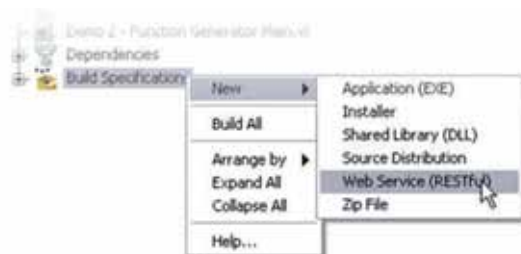


Fig. 5 - Configurate, costruite e installate VI Web service dal menu Build Specifications

Le impostazioni nel dialogo Build Specifications per Web service determinano come un utente remoto chiama e interagisce con il VI quando è stato installato. Qui sotto è riportata una panoramica delle informazioni e dei passi richiesti per comunicare con un VI che è stato installato come Web service.

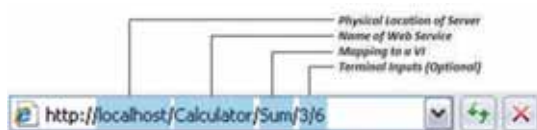


Fig. 6 - HTTP dirige la richiesta al server corretto, seguita dal server e dall'effettivo VI da invocare

1. Posizione fisica del server – L'HTTP usa URL per inviare una richiesta da un client a una specifica interfaccia server. La prima parte dell'indirizzo HTTP che dovete specificare in questo dialogo è la posizione fisica del Web server. Ciò è comunemente specificato usando un indirizzo IP, o dominio. Se state accedendo al Web da un client sullo stesso computer del ser-

ver, potete digitare 'localhost' per indicare che intendete comunicare con un server ospitato localmente.

2. Nome del Web service – Poiché potete configurare più servizi per una singola applicazione LabVIEW, il prossimo parametro che dovete inviare nel vostro URL è il nome del servizio, che è specificato nella finestra di dialogo illustrata nella figura 7.



Fig. 7 - Il nome del servizio è specificato dalla categoria Information della Build Specification

3. Mappatura ad un VI – Nel dialogo delle mappature URL, potete ora definire percorsi custom per ogni servizio mappato a VI o folder contenenti un contenuto statico. Dopo avere selezionato i VI da eseguire come Web service, inserite le estensioni URL che volete che puntino a ciascun VI.

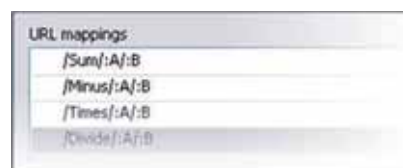


Fig. 8 - Nel dialogo di mappatura dell'URL, potete customizzare i percorsi a specifici VI e ingressi da terminale

4. Ingressi terminali – Gli ingressi possono essere passati al VI invocato attraverso il quadro connettori del VI. Potete mappare ingressi dall'URL all'ingresso terminale aggiungendo ":nome terminale" alla fine di un URL.

METODI DI COMUNICAZIONE

Per inviare dati a un VI, come un ingresso terminale, dovete determinare come passare i dati. GET e POST sono i metodi HTTP più comunemente usati; tuttavia, usando il Web server LabVIEW, avete anche l'opzione di PUT e DELETE. GET permette di introdurre i dati tramite variabili che estendono l'URL. POST è usato più comunemente per leggere dati da un modulo HTML che contiene una grande quantità di ingressi utente. Contenuto statico è qualsiasi documento diverso da un VI che volete rendere disponibile a un client che comunica con il Web server LabVIEW. Nella maggior parte dei casi, il contenuto statico include un'interfaccia utente progettata per invocare i metodi dei VI ed interpretare la risposta. Il contenuto statico deve essere posto in una directory di destinazione, che è collegata per nome nel dialogo di mappatura dell'URL.

PROGETTAZIONE DI INTERFACCE UTENTE WEB-BASED

Vi sono molti modi per sviluppare un'interfaccia utente Web

che visualizza il contenuto in un modo visivamente accattivante. Dagli albori di Internet, le pagine Web si sono basate in modo predominante su HTML e JavaScript, che rimangono ampiamente utilizzati e sono ideali per semplici layout, moduli d'ingresso utente e contenuto statico. Poiché esse richiedono un overhead minimo, possono essere servite facilmente e rapidamente attraverso l'attuale Web ad alta velocità.

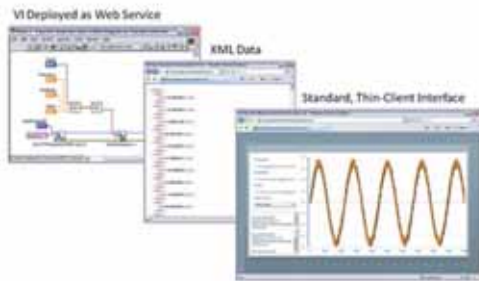


Fig. 9 - L'interfaccia thin-client interface sviluppata in Adobe Flex richiama un URL per eseguire il VI installato, che programmaticamente genera ed esegue lo streaming di dati XML in risposta

Le interfacce utente più ricche, come quelle mostrate nella figura 9, possono essere sviluppate per offrirvi grafica dinamica e un aspetto simile a quello di un'applicazione all'interno di una pagina Web, pur continuando a offrire soluzioni leggere per minimizzare la quantità di dati inviati dal server. Alcuni dei linguaggi più comuni per sviluppare queste applicazioni includono Adobe Flex, Adobe Flash, AJAX, Microsoft Silverlight e gli applet Java. Tutti richiedono una particolare esperienza nel linguaggio usato per la scrittura e lo script di comportamenti. Da Ottobre di quest'anno, dopo l'annuncio all'NI Week tenuto ad Austin, ad Agosto, National Instruments ha avviato il programma di valutazione beta di una nuova interfaccia utente basata su tecnologia Silverlight, per la realizzazione di thin client web. Il nuovo prodotto si chiamerà Web LabVIEW UI Builder e permetterà di creare facilmente interfacce web custom, utilizzando i tradizionali strumenti di programmazione grafica di LabVIEW.

CONCLUSIONI

I nuovi Web service in LabVIEW risolvono la crescente necessità di un modo aperto e standard per interagire con le applicazioni LabVIEW attraverso il Web. Selezionando il Web server Embedthis e il protocollo leggero RESTful, NI sta risolvendo la necessità di un robusto Web server che potete incorporare in applicazioni eseguite su sistemi LabVIEW Real-Time. La nuova esperienza di programmazione offerta da Web LabVIEW UI Builder permetterà di sviluppare thin web client con il classico approccio di programmazione grafica di LabVIEW.

Note sull'autore

Elijah Kerry è un product manager LabVIEW presso National Instruments, impegnato su grandi applicazioni di sviluppo mission-critical e su attività di ingegneria software. Ha una laurea breve in informatica conseguita presso l'University of Missouri, Columbia.