



Training per esperti

PROGRAMMAZIONE A EVENTI

I modelli di progettazione basati su eventi vi permettono di creare applicazioni più efficienti e flessibili

A cura di Matteo Foini

I modelli di progettazione basati su eventi utilizzano la struttura ad eventi per rispondere direttamente all'utente o ad altri eventi. Questa lezione descrive la programmazione guidata da eventi utilizzando la struttura ad eventi e i modelli di progettazione che impiegano la struttura ad eventi.

Argomenti

- A. Programmazione guidata da eventi
- B. Modelli di progettazione basati su eventi

A. Programmazione guidata da eventi

LabVIEW è un ambiente di programmazione a flusso di dati nel quale il flusso dei dati determina l'ordine di esecuzione degli elementi dello schema a blocchi. Le caratteristiche della programmazione guidata da eventi estendono le potenzialità dell'ambiente per consentire l'interazione diretta dell'utente con il pannello ed altre attività asincrone per influenzare ulteriormente l'esecuzione dello schema a blocchi.

Nota Le funzionalità della programmazione guidata da eventi sono disponibili solo nelle versioni Full e Professional di LabVIEW. Potete eseguire un VI costruito con queste caratteristiche anche in LabVIEW Base, ma non potete riconfigurare i componenti che gestiscono gli eventi.

Che cosa sono gli eventi?

Un evento è una notifica asincrona che qualcosa si è verificato. Gli eventi possono avere origine dall'interfaccia utente, da I/O esterno o da altre parti del programma. Gli eventi dell'interfaccia utente includono click del mouse, pressioni di tasti, e così via. Gli eventi di I/O esterni includono timer hardware o trigger che segnalano quando l'acquisizione dati è terminata o quando si verifica una condizione di errore. Altri tipi di eventi possono essere generati programmaticamente ed essere utilizzati per comunicare con parti differenti del programma stesso. LabVIEW supporta gli eventi di interfaccia utente e generati da programma, ma non supporta gli eventi di I/O esterni.

In un programma guidato da eventi, gli eventi che si verificano nel sistema influenzano direttamente il flusso di esecuzione. Al contrario, un programma procedurale viene

eseguito in un ordine sequenziale predeterminato. I programmi guidati da eventi includono solitamente un loop che attende il verificarsi di un evento, esegue codice per rispondere all'evento e si reitera in attesa dell'evento successivo. Come il programma risponde a ciascun evento dipende dal codice scritto per quello specifico evento. L'ordine nel quale un programma guidato da eventi viene eseguito dipende da quali eventi si verificano e dall'ordine nel quale essi si verificano. Alcune parti del programma potrebbero essere eseguite spesso perché gli eventi che gestiscono si manifestano frequentemente, mentre altre parti del programma potrebbero non essere eseguite affatto perché i relativi eventi non si verificano mai.

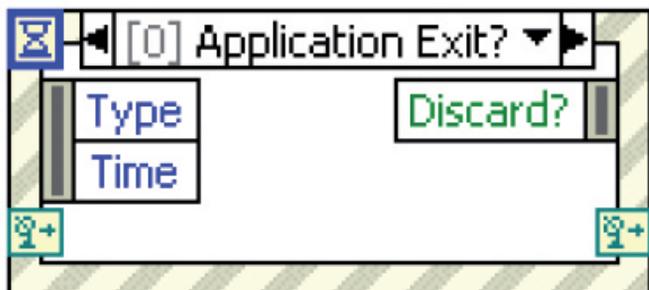
Perché usare gli eventi?

Usate eventi dell'interfaccia utente in LabVIEW per sincronizzare le azioni dell'utente sul pannello frontale con l'esecuzione dello schema a blocchi. Gli eventi vi permettono di eseguire una specifica sezione di gestione di evento ogni volta che un utente effettua una determinata azione. Senza eventi, lo schema a blocchi deve interrogare (polling) lo stato degli oggetti del pannello frontale in un loop, controllando se si è verificato qualche cambiamento. L'interrogazione del pannello frontale richiede tuttavia una significativa quantità di tempo della CPU e può fallire nel rilevamento di cambiamenti che si verificano troppo rapidamente. Utilizzando eventi per rispondere a specifiche azioni dell'utente, potete eliminare la necessità di interrogare il pannello frontale per determinare quali azioni siano state eseguite dall'utente. LabVIEW notifica attivamente allo schema a blocchi ogni volta che si verifica un'interazione che avete specificato. L'uso degli eventi riduce la richiesta di tempo macchina del programma, semplifica il codice dello schema a blocchi e garantisce la capacità del diagramma di rispondere a tutte le interazioni eseguite dall'utente.

Usate eventi generati da programma per comunicare fra parti differenti del programma che non dipendono dal flusso dei dati. Gli eventi generati da programma hanno molti degli stessi vantaggi degli eventi dell'interfaccia utente e possono condividere lo stesso codice di gestione degli eventi, facilitando l'implementazione di architetture avanzate, come le macchine a stati con gestione di code di eventi.

Componenti della struttura ad eventi

Usate la struttura ad eventi, illustrata in figura, per gestire gli eventi in un VI.



La struttura ad eventi opera come una struttura Case, con una funzione **Wait on Notification** incorporata. La struttura ad eventi può avere più casi, ciascuno dei quali è una routine separata di gestione degli eventi. Potete configurare ogni caso in modo da gestire uno o più eventi, ma può verificarsi uno solo di tali eventi per volta.

Quando la struttura ad eventi viene eseguita, attende fino al verificarsi di uno degli eventi configurati, quindi esegue il caso corrispondente a tale evento. La struttura termina l'esecuzione dopo avere gestito esattamente un evento. Essa non entra implicitamente in loop per gestire più eventi.

Analogamente a una funzione **Wait on Notification**, la struttura ad eventi può andare in timeout mentre è in attesa della notifica di un evento. Quando ciò si verifica, viene eseguito uno specifico caso Timeout.

L'etichetta del selettore eventi nella parte superiore della struttura ad eventi, illustrata in figura, indica quali eventi causano l'esecuzione del caso attualmente visualizzato.



Potete visualizzare altri casi di eventi cliccando la freccia verso il basso accanto al nome del caso e selezionando un altro caso dal menu abbreviato.

Il terminale Timeout nell'angolo superiore sinistro della struttura ad eventi, illustrato in figura, specifica il numero di millisecondi per i quali attendere un evento prima del timeout.

Il default è -1, che specifica un'attesa indefinita del verificarsi di un evento. Se cablate un valore al terminale Timeout, dovete fornire un caso Timeout.



L'Event Data Node, illustrato in figura, si comporta in modo simile alla funzione Unbundle By Name.



Il nodo è collegato al bordo interno sinistro di ogni caso di evento. Esso identifica i dati forniti da LabVIEW quando si verifica un evento. Potete ridimensionare questo nodo verticalmente per aggiungere altri elementi. Il nodo fornisce differenti elementi in ogni caso della struttura ad eventi, in base all'evento o agli eventi che il caso è configurato per gestire. Se configurate un singolo caso per gestire più eventi, l'Event Data Node fornisce solo gli elementi che sono comuni a tutti gli eventi configurati per quel caso.

L'Event Filter Node, illustrato in figura, è simile all'Event Data Node.



Il nodo è collegato al bordo interno destro dei casi di eventi di filtraggio. Esso identifica il subset di dati disponibili nell'Event Data Node che il caso di evento può modificare. Il nodo visualizza dati differenti in base all'evento o agli eventi che il caso è configurato per gestire. Per default, questi elementi sono al posto dei corrispondenti elementi nell'Event Data Node. Se non cablate un valore a un elemento dell'Event Filter Node, quell'elemento rimane invariato.

I terminali di eventi dinamici, illustrati in figura, sono disponibili cliccando il tasto destro sulla struttura ad eventi e selezionando **Show Dynamic Event Terminals** dal menu abbreviato.



Nota Analogamente ad una struttura Case, la struttura ad eventi supporta i tunnel. Tuttavia, di default, non dovete cablare i tunnel d'uscita della struttura ad eventi in ogni caso. Tutti i tunnel scablati usano il valore di default per il tipo di dato del tunnel. Cliccate con il tasto destro su un tunnel e deselezionate **Use Default If Unwired** dal menu abbreviato per tornare al comportamento di default della struttura Case quando i tunnel devono essere cablati in tutti i casi. Consultate il LabVIEW Help per informazioni sui valori di default per i tipi di dato.

Eventi di notifica e filtraggio

Vi sono due tipi di eventi di interfaccia utente: notifica e filtraggio.

Gli eventi di notifica sono un'indicazione che l'azione dell'utente si è già verificata, ad esempio quando l'utente ha

modificato il valore di un controllo. Usate gli eventi di notifica per rispondere ad un evento dopo che si è verificato e LabVIEW lo ha elaborato. Potete configurare la risposta allo stesso evento di notifica su uno specifico oggetto per un qualsiasi numero di strutture ad eventi. Quando l'evento si verifica, LabVIEW invia una copia dell'evento ad ogni struttura ad eventi configurata per gestire l'evento in parallelo.

Gli eventi di filtraggio vi informano che l'utente ha eseguito un'azione prima che LabVIEW la elabori, permettendovi di customizzare il modo con il quale il programma risponde alle interazioni con l'interfaccia utente. Usate gli eventi di filtraggio per partecipare alla gestione dell'evento, eventualmente escludendo il comportamento di default per l'evento stesso. In un caso di evento di filtraggio della struttura ad eventi, potete validare o modificare i dati dell'evento prima che LabVIEW ne completi l'elaborazione, o potete scartare interamente l'evento per evitare che il cambiamento abbia effetti sul VI. Per esempio, potete configurare una struttura ad eventi in modo da scartare l'evento Panel Close?, impedendo all'utente di chiudere interattivamente il pannello frontale del VI. Gli eventi di filtraggio hanno nomi che terminano con un punto interrogativo, come Panel Close?, per aiutarvi a distinguerli dagli eventi di notifica. La maggior parte degli eventi di filtraggio ha un evento di notifica associato con stesso nome, ma senza il punto interrogativo, che LabVIEW genera dopo l'evento di filtraggio se nessun caso di evento ha scartato l'evento stesso.

Come nel caso degli eventi di notifica, potete configurare la risposta di qualsiasi numero di strutture ad eventi allo stesso evento di filtraggio su uno specifico oggetto. Tuttavia, LabVIEW invia sequenzialmente gli eventi di filtraggio ad ogni struttura ad eventi configurata per l'evento. L'ordine nel quale LabVIEW invia gli eventi a ciascuna struttura ad eventi dipende dall'ordine con il quale gli eventi sono stati registrati. Ogni struttura ad eventi deve completare il suo caso per l'evento prima che LabVIEW possa notificare la successiva struttura ad eventi. Se un caso della struttura ad eventi cambia un qualsiasi dato dell'evento, LabVIEW passa il dato modificato alle successive strutture nella catena. Se una struttura ad eventi nella catena scarta l'evento, LabVIEW non passa l'evento ad alcuna struttura ad eventi rimanente nella catena. LabVIEW completa l'elaborazione dell'azione utente che ha triggerato l'evento solo quando tutte le strutture ad eventi configurate hanno gestito l'evento senza scartarlo.

Nota National Instruments raccomanda di utilizzare gli eventi di filtraggio solo quando si vuole prendere parte alla gestione dell'azione dell'utente, scartando l'evento o modificando i dati dell'evento. Se volete semplicemente sapere che l'utente ha effettuato una particolare azione,

usate gli eventi di notifica.

I casi della struttura ad eventi che gestiscono eventi di filtraggio hanno un Event Filter Node. Potete cambiare i dati dell'evento cablando nuovi valori a questi terminali. Se non cablate un elemento dati, l'elemento rimane invariato. Potete completamente scartare un evento cablando un valore TRUE al terminale Discard?

Nota Un singolo caso nella struttura ad eventi non può gestire eventi sia di notifica che di filtraggio. Un caso può gestire più eventi di notifica ma può gestire più eventi di filtraggio solo se gli elementi dell'evento sono identici per tutti gli eventi.

Consiglio Nell'area di dialogo Edit Events, gli eventi di notifica sono indicati da una freccia verde, mentre gli eventi di filtraggio sono indicati da una freccia rossa.

Uso di eventi in LabVIEW

LabVIEW può generare molti eventi differenti. Per evitare di generare eventi indesiderati, usate la registrazione degli eventi per specificare gli eventi che volete vi siano notificati da LabVIEW. LabVIEW supporta due modelli per la registrazione degli eventi: statica e dinamica.

La registrazione statica vi permette di specificare quali eventi sul pannello frontale di un VI volete gestire in ogni caso di una struttura ad eventi sullo schema a blocchi del VI stesso. LabVIEW registra automaticamente questi eventi durante l'esecuzione del VI, quindi la struttura ad eventi inizia ad attendere eventi non appena inizia l'esecuzione del VI. Ogni evento è associato a un controllo sul pannello frontale del VI, alla finestra del pannello frontale del VI nel suo complesso o all'applicazione LabVIEW.

Non potete configurare staticamente una struttura ad eventi in modo che gestisca gli eventi per il pannello frontale di un VI differente. La configurazione è statica perché non potete cambiare in fase di run time gli eventi gestiti dalla struttura ad eventi.

La registrazione dinamica degli eventi evita i limiti della registrazione statica integrando la registrazione degli eventi con il VI Server, cosa che vi permette di utilizzare Application, VI e control reference per specificare in fase di run time gli oggetti per i quali desiderate generare eventi. La registrazione dinamica offre una maggiore flessibilità nel controllare quali eventi sono generati da LabVIEW e quando deve avvenire la loro generazione. Tuttavia, la registrazione dinamica è più complessa della registrazione statica, perché richiede di registrare e deregistrare esplicitamente gli eventi tramite le funzioni del VI Server, anziché gestire automaticamente la registrazione usando le informazioni che avete configurato nella struttura ad eventi.

Nota In generale, LabVIEW genera eventi dell'interfaccia

utente solo a seguito di un'interazione diretta dell'utente con il pannello frontale attivo. LabVIEW non genera eventi, tipo Value Change, quando utilizzate variabili condivise, variabili globali, variabili locali, e così via.

Tuttavia, potete utilizzare la proprietà Value (Signaling) per generare un evento Value Change programmaticamente. In molti casi, potete utilizzare eventi generati da programma invece di code e notificatori.

I dati di evento forniti da un evento LabVIEW includono sempre un time stamp, un tipo Enum che indica quale evento si è verificato e una VI Server reference all'oggetto che ha triggered l'evento. Il time stamp è un contatore di millisecondi che potete utilizzare per calcolare il tempo trascorso fra due eventi o per determinare l'ordine di occorrenza. La reference all'oggetto che ha generato l'evento è strettamente tipizzata in base alla classe del VI Server di quell'oggetto.

Gli eventi sono raggruppati in classi in base al tipo di oggetto che genera l'evento, come Application, VI o Control. Se un singolo caso gestisce più eventi per oggetti di classi differenti del VI Server, il tipo di reference è la classe genitrice comune di tutti gli oggetti. Per esempio, se configurate un singolo caso nella struttura ad eventi per gestire eventi per un controllo numerico e un controllo di rampa colore, il tipo di reference della sorgente di eventi sarà Numeric, perché i controlli numerico e rampa colore sono nella classe Numeric. Se registrate lo stesso evento su entrambe le classi VI e Control, LabVIEW genera prima l'evento VI.

Nota I cluster sono gli unici oggetti contenitori per i quali potete generare eventi.

LabVIEW genera eventi Control per i cluster, prima di generare eventi per gli oggetti che contengono, ad eccezione del caso dell'evento Value Change. L'evento Value Change genera l'evento su un elemento nel cluster, quindi sul cluster stesso. Se il caso della struttura ad eventi per un evento VI o un evento Control su un oggetto contenitore scarta l'evento, LabVIEW non genera ulteriori eventi.

Ogni struttura ad eventi e funzione Register For Events sullo schema a blocchi possiede una coda che LabVIEW utilizza per memorizzare gli eventi. Quando un evento si verifica, LabVIEW mette una copia dell'evento in ogni coda registrata per quell'evento. Una struttura ad eventi gestisce tutti gli eventi nella sua coda e gli eventi nelle code di tutte le funzioni Register For Events che avete cablato ai terminali di evento dinamici della struttura ad eventi. LabVIEW usa queste code per assicurare che gli eventi siano consegnati in modo affidabile ad ogni struttura ad eventi registrata nell'ordine in cui gli eventi si verificano.

Per default, quando un evento entra in una coda, LabVIEW blocca il pannello frontale contenente l'oggetto che ha generato l'evento. LabVIEW tiene bloccato il pannello frontale finché tutte le strutture Event finiscono di gestire l'evento. Mentre il pannello frontale è bloccato, LabVIEW non

elabora l'attività del pannello frontale, ma mette quelle interazioni in un buffer e le gestisce quando il pannello frontale viene sbloccato.

Per esempio, un utente potrebbe prevedere che un caso di evento lanci un'applicazione richiedente inserimento di testo. Poiché l'utente sa già che è richiesto un inserimento di testo, egli potrebbe iniziare a digitarlo prima che l'applicazione appaia sul pannello frontale. Se l'opzione **Lock front panel until the event case for this event completes** è abilitata, quando l'applicazione viene lanciata ed appare sul pannello frontale, essa elabora le pressioni dei tasti nell'ordine in cui si sono verificate. Se l'opzione **Lock front panel until the event case for this event completes** è disabilitata, le pressioni dei tasti potrebbero essere elaborate altrove sul pannello frontale, perché LabVIEW non accoda la loro esecuzione in modo che dipenda dal completamento del caso di evento.

Il blocco del pannello frontale non influisce su certe azioni, come lo spostamento della finestra, l'interazione con le barre di scorrimento e il click del pulsante **Abort**.

LabVIEW può generare eventi anche quando nessuna struttura ad eventi è in attesa di gestirli. Poiché la struttura ad eventi gestisce solo un evento ogni volta che viene eseguita, mettete la struttura ad eventi in un While Loop per assicurare che la struttura ad eventi possa gestire tutti gli eventi che si verificano.

Attenzione Se non viene eseguita alcuna struttura ad eventi per gestire un evento e il blocco del pannello frontale è abilitato, l'interfaccia utente del VI diventa irresponsiva. In questo caso, cliccate il pulsante **Abort** per interrompere il VI. Potete disabilitare il blocco del pannello frontale cliccando con il tasto destro sulla struttura ad eventi e rimuovendo il segno di spunta dalla casella **Lock front panel until the event case for this event completes** nell'area di dialogo **Edit Events**. Non potete disattivare il blocco del pannello frontale per gli eventi di filtraggio.

Registrazione statica degli eventi

La registrazione statica degli eventi è disponibile solo per gli eventi dell'interfaccia utente. Usate l'area di dialogo **Edit Events** per configurare una struttura ad eventi in modo che gestisca un evento registrato staticamente. Selezionate la sorgente degli eventi, che può essere l'applicazione, il VI o un singolo controllo. Selezionate uno specifico evento che la sorgente di eventi può generare, come Panel Resize, Value Change e così via. Editate il caso per gestire i dati dell'evento in base alle esigenze dell'applicazione.

LabVIEW registra staticamente gli eventi in modo automatico e trasparente quando eseguite un VI che contiene una struttura ad eventi. LabVIEW genera eventi per un VI solo quando quel VI è in esecuzione o quando un altro VI in esecuzione richiama il VI come subVI.

Quando eseguite un VI, LabVIEW imposta il VI di livello

superiore e la gerarchia di subVI che il VI richiama nel suo schema a blocchi in uno stato di esecuzione denominato riservato. Non potete editare un VI o cliccare il pulsante **Run** mentre il VI è nello stato riservato perché il VI può essere richiamato come subVI in qualsiasi istante mentre il suo VI genitore è in esecuzione. Quando LabVIEW imposta un VI nello stato riservato, esso registra automaticamente gli eventi che avete configurato staticamente in tutte le strutture ad eventi presenti sullo schema a blocchi di quel VI. Quando l'esecuzione del VI di livello superiore termina, LabVIEW imposta tale VI e la sua gerarchia di subVI nello stato di esecuzione idle e deregistra automaticamente gli eventi.

Configurazione degli eventi

Completate i passi seguenti per configurare un caso della struttura ad eventi in modo che gestisca un evento.

1. (Opzionale) Se volete configurare la struttura ad eventi in modo che gestisca un evento di utente, un controllo booleano o un evento dell'interfaccia utente generato in base a una reference a un'applicazione, un VI o un controllo, dovette in primo luogo registrare dinamicamente quell'evento.
2. Cliccate con il tasto destro sul bordo della struttura ad eventi e selezionate **Edit Events Handled by This Case** dal menù abbreviato per visualizzare l'area di dialogo **Edit Events** per editare il caso corrente. Potete selezionare anche **Add Event Case** dal menù abbreviato per creare un nuovo caso.
3. Specificate una sorgente di eventi nella sezione **Event Sources**.
4. Selezionate l'evento che volete configurare per la sorgente di eventi, come **Key Down**, **Timeout** o **Value Change** dall'elenco **Events**. Quando selezionate una sorgente di eventi dinamici dall'elenco **Event Sources**, l'elenco **Events** visualizza quell'evento. Questo è lo stesso evento che avete selezionato quando avete registrato l'evento. Se avete registrato dinamicamente gli eventi e avete cablato **event reg refnum out** al terminale di eventi dinamici, le sorgenti appaiono nella sezione **Dynamic**.
5. Se volete aggiungere eventi addizionali da gestire nel caso corrente, cliccate il pulsante + e ripetete i passi 3 e 4 per specificare ogni evento addizionale.

La sezione **Event Specifiers** in cima all'area di dialogo elenca tutti gli eventi che il caso deve gestire. Quando cliccate su un elemento in questo elenco, la sezione **Event Sources** si aggiorna per evidenziare la sorgente di eventi che avete selezionato. Potete ripetere i passi 3 e 4 per ridefinire ciascun evento o cliccare il pulsante **X** per rimuovere l'evento selezionato.

6. Cliccate sul pulsante **OK** per salvare la configurazione e chiudere l'area di dialogo.

I casi di evento che avete configurato appaiono come opzioni di selezione nell'etichetta del selettore di eventi in cima alla struttura ad eventi e il nodo Event Data visualizza i dati

comuni a tutti gli eventi gestiti in quel caso.

7. (Opzionale) Potete usare un evento Timeout per configurare una struttura ad eventi in modo che attenda per una quantità di tempo specificata il verificarsi di un evento. Cablate un valore al terminale Timeout in alto a sinistra della struttura ad eventi per specificare il numero di millisecondi per il quale la struttura ad eventi deve attendere il verificarsi di un evento prima di generare un evento Timeout. Il valore di default del terminale Timeout è -1, che specifica di attendere indefinitamente il verificarsi di un evento.

8. Ripetete i passi da 1 a 6 per ciascun caso di evento che volete configurare.

Esempio di evento

La figura 1 illustra una struttura ad eventi configurata con l'evento Menu Selection (User). Questo VI usa la struttura ad eventi per catturare le selezioni di menu impiegando il menu definito dall'utente di nome sample.rtm. ItemTag restituisce l'elemento di menu che è stato selezionato, mentre MenuRef restituisce il refnum alla barra menù. Questa informazione è passata alla funzione Get Menu Item Info.

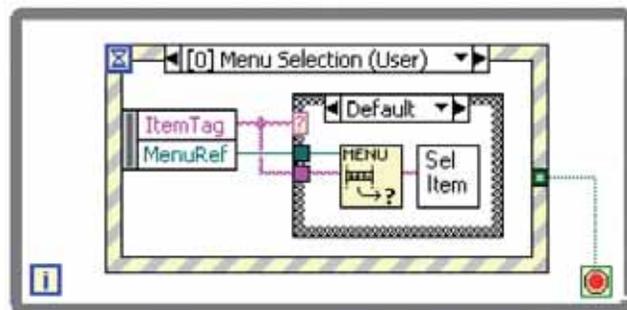


Figura 1. Evento Menu Selection (User)

Nota Se usate la funzione Get Menu Selection con una struttura ad eventi configurata per gestire lo stesso elemento di menu, la struttura ad eventi ha la precedenza e LabVIEW ignora la funzione Get Menu Selection. In qualsiasi VI, per gestire gli eventi di menù usate la struttura ad eventi o la funzione Get Menu Selection, non entrambe.

B. Modelli di progettazione basati su eventi

I modelli di progettazione basati su eventi offrono guadagni di efficienza perché essi rispondono solo al verificarsi di un evento. Quando LabVIEW esegue la struttura ad eventi, il VI che contiene la struttura ad eventi rimane dormiente finché si verifica o viene generato un evento registrato. Quando un evento registrato viene generato, la struttura ad eventi si risveglia automaticamente ed esegue il sottodiagramma appropriato per gestire l'evento.

Modello di progettazione gestore di eventi dell'interfaccia utente

Il modello di progettazione gestore di eventi dell'interfaccia utente mette a disposizione un'architettura potente ed efficiente per gestire le interazioni dell'utente con LabVIEW. Usate il gestore di eventi dell'interfaccia utente per rilevare quando un utente cambia il valore di un controllo, sposta o clicca il mouse o preme un tasto.

Il template standard del gestore di eventi dell'interfaccia utente consiste di una struttura ad eventi contenuta in un While Loop, come illustrato nella figura 2. Configurate la struttura ad eventi in modo da avere un caso per ogni categoria di evento che volete rilevare. Ogni caso di evento contiene il codice di gestione che viene eseguito immediatamente dopo il verificarsi di un evento.

Poiché il loop del gestore di eventi si risveglia esattamente al verificarsi di un evento e rimane dormiente fra gli eventi, non dovete fare polling o leggere ripetutamente i valori di un controllo per rilevare quando un utente clicca un pulsante. Il gestore di eventi dell'interfaccia utente vi permette di minimizzare l'uso del processore senza sacrificare l'interattività.

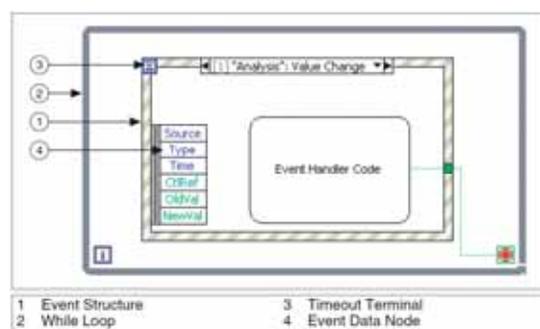


Figura 2. Modello di progettazione del gestore di eventi dell'interfaccia utente

Un problema comune quando si usa il gestore di eventi dell'interfaccia utente è che esso calcola il termine del While Loop prima che la struttura ad eventi sia eseguita. Ciò fa sì che il While Loop iteri una volta in più del previsto. Per evitare questa situazione, calcolate il termine del While Loop all'interno di tutto il vostro codice di gestione degli eventi.

Il codice del gestore di eventi deve essere eseguito rapidamente, generalmente entro 200 ms. Se è più lento, si ha la sensazione che l'interfaccia utente sia bloccata. Inoltre, se il codice del gestore di eventi richiede molto tempo per essere eseguito, la struttura ad eventi potrebbe bloccarsi. Di default, il pannello frontale si blocca mentre un evento viene gestito. Potete disabilitare il blocco del pannello frontale per ogni caso di evento in modo da rendere più responsiva l'interfaccia utente. Tuttavia, ogni nuovo evento generato mentre si sta gestendo un altro evento non viene gestito immediatamente. Pertanto, l'interfaccia utente sembra ancora poco responsiva.

Qualsiasi parte di codice presente all'interno di un caso di evento non può essere condivisa con un'altra struttura ad eventi. Quando impiegate la struttura ad eventi dovete utilizzare un codice ben progettato. Modularizzate il codice che verrà condiviso fra più casi della struttura ad eventi.

La struttura ad eventi include un evento Timeout, che vi permette di controllare quando l'evento Timeout viene eseguito. Per esempio, se impostate un Timeout di 200 ms, il caso di evento Timeout viene eseguito ogni 200 ms in assenza di altri eventi. Potete usare l'evento Timeout per eseguire una temporizzazione critica nel vostro codice.

Modello di progettazione produttore/consumatore (di eventi)

Uno dei modelli di progettazione più versatili e flessibili combina i modelli di progettazione produttore/consumatore e del gestore di eventi dell'interfaccia utente.

Un VI costruito utilizzando il modello produttore/consumatore (di eventi) risponde all'interfaccia utente in modo asincrono, permettendo all'interfaccia utente di rispondere continuamente all'utente. Il loop consumatore di questo modello risponde al verificarsi di eventi, analogamente al loop consumatore del modello di progettazione produttore/consumatore (di dati).

Il modello di progettazione produttore/consumatore (di eventi) usa la stessa implementazione del modello di progettazione produttore/consumatore (di dati), ad eccezione del fatto che il loop produttore usa una struttura ad eventi per rispondere agli eventi dell'interfaccia utente, come illustrato nella figura 3. La struttura ad eventi permette una risposta continua all'interazione dell'utente.

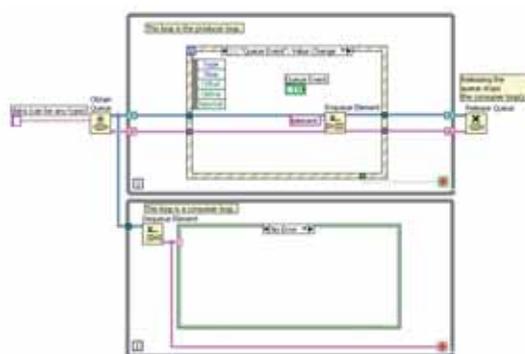


Figura 3. Modello di progettazione produttore/consumatore (di eventi)

La figura illustra come potete usare VI e funzioni di sincronizzazione per aggiungere funzionalità al modello di progettazione. Le code hanno la capacità di trasferire qualsiasi tipo di dato. Il tipo di dato trasferito nella figura 3 è una stringa. Una stringa non è il tipo di dato più efficiente per elaborare i dati nei modelli di progettazione. Un tipo di dato più efficiente per passare dati nei modelli di progettazione è un cluster contenente un controllo di tipo Enum e un Variant.

Readerservice.it n. 1126