Interfacce Utente avanzate con eventi dinamici.

Nicola Bavarone

L'introduzione degli eventi in LabVIEW ha fornito nuove possibilità nella realizzazione di interfacce avanzate.

E' possibile non solo gestire gli eventi che avvengono sul pannello ma anche eventi generati da **VI** esterni.

Se dovessimo realizzare un sistema distribuito, realizzeremo un interfaccia principale (**main**), e quindi vari servizi(task) che gestiscono il sottosistema.

Si può utilizzare la tecnologia **VIserver** per eseguire e controllare i task, i quali attivano l'evento definito programmaticamente nel pannello principale.

Un esempio aiuterà a comprendere il vantaggio degli eventi dinamici.

Si realizza un interfaccia utente in grado di controllare il pannello principale e dallo stesso poter controllare vi esterni all'applicazione.

La struttura a eventi può reagire non solo ai cambiamenti di valore, ma anche a eventi del mouse o tastiera, quindi si presta ad una gestione del pannello più completa.

Altro vantaggio, a meno di non impostare un timeout definito (che può essere utilizzato per l'esecuzione di codice in continuo), non esegue iterazioni se non viene attivato un evento.

I task vengono caricati dinamicamente e rimangono in esecuzione fino al termine del programma.

Per passare dati all'applicazione principale i task possono usufruire delle funzioni **Events** presenti nella palette ***"Dialog&User Interface"*** e generare una notifica di evento che verrà letto dall'interfaccia principale.

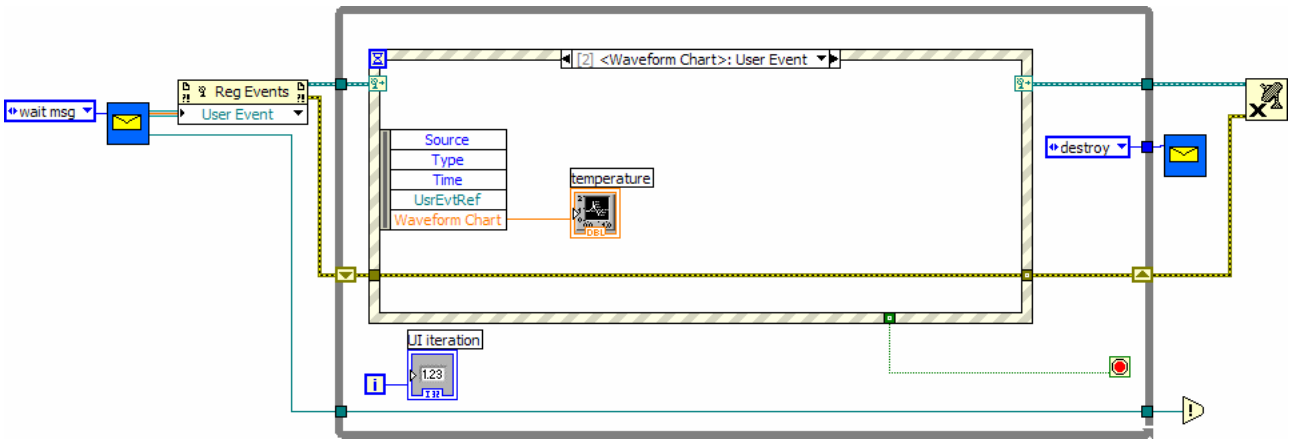

Figura 1 main.vi

Nel diagramma del main.vi, gli eventi definiti da utente devono essere registrati tramite il **dynamic terminal** (sul bordo della **event structure**) e mediante il **RegEvents**.

I vi di notifica evento sono inseriti in una subVI per utilizzarli come reentrant e inserirli nei vari task distribuiti. In questo modo i task possono comunicare con l'interfaccia principale aggiungendosi semplicemente agli eventi generati direttamente dal pannello.

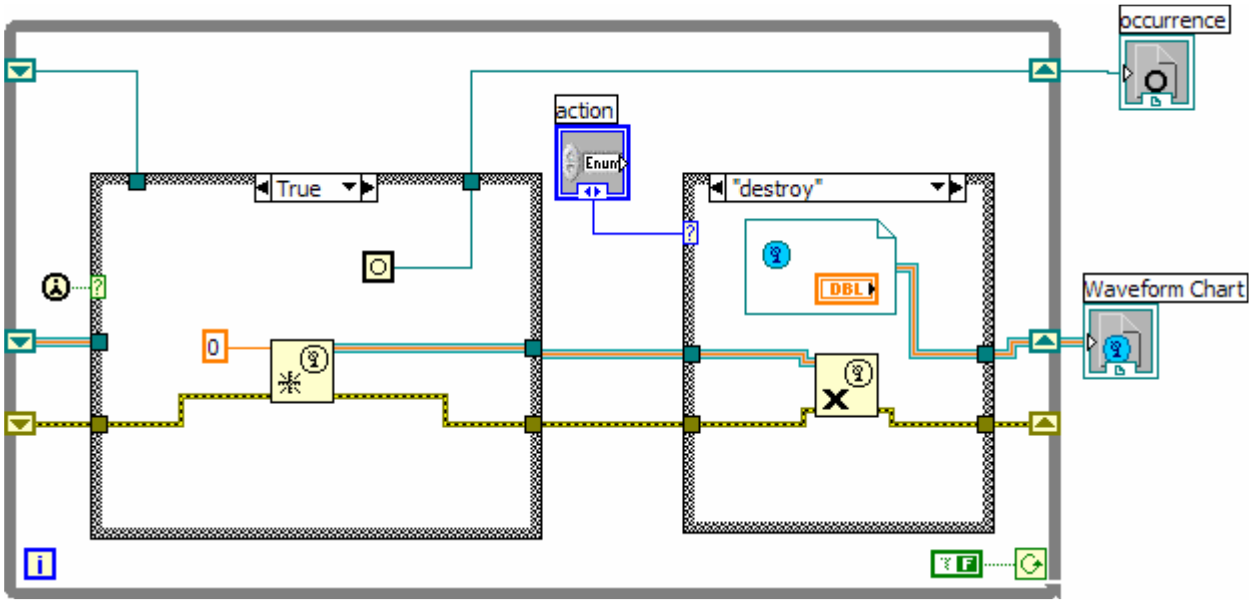

Figura 2 event_generator

Il subVI event-generator, crea l'evento alla prima esecuzione, quindi lo rende disponibile, e tramite il terminale esterno action, si possono selezionare le funzioni che si desidera utilizzare.
Attraverso il generate user event, viene richiamato l'evento sull'interfacci principale.

Si crea anche un refnum di occorrenza, in modo da sincronizzare i cicli dei task e l'interfaccia principale.

Un esempio di task potrebbe essere simile a questo:

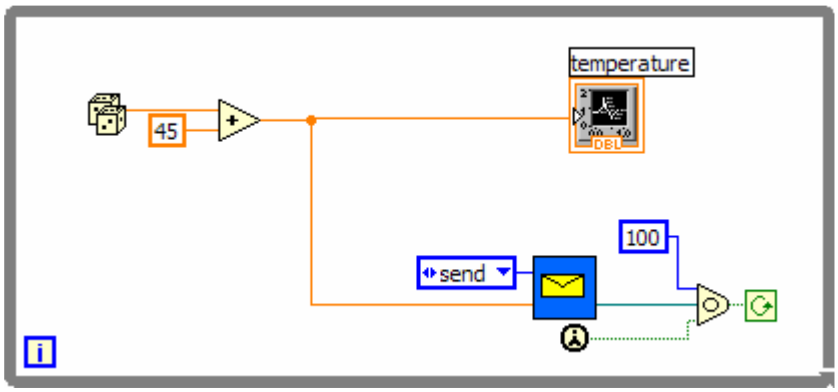

Figura 3 task

Il while loop viene controllato dall' occorrenza, che viene settata dal VI principale, in modo da chiudere tutti i task alla chiusura dell'applicazione.

Citazione:
Lo spunto di questo articolo è preso da questo articolo pubblicato su LTR qualche anno fa.
Articolo "Dynamic and user event handling"
Nicola Chiari
pubblicazione LTR Vol.11,Number 1