

SVILUPPO DI APPLICAZIONI CON LABVIEW STATECHART MODULE

Nell'ambito della release 8.5 di LabVIEW, National Instruments introduce la prima versione di LabVIEW Statechart Module. Questo modulo aggiunge un nuovo modello di programmazione a LabVIEW per lo sviluppo di sistemi di controllo e test basati su eventi

I modello di programmazione a diagrammi di stato completa il panorama dei modelli presenti in LabVIEW per la gestione del flusso di dati, la matematica testuale, la modellazione di sistemi dinamici e lo sviluppo basato su finestre di configurazione. Potete scegliere il modello o la combinazione di modelli più adatta per sviluppare il vostro sistema in base ai requisiti specifici dell'applicazione.

opera di sistemi. I diagrammi di stato di LabVIEW offrono uno strumento di progettazione ad alto livello di grande scalabilità grazie a concetti di programmazione quali gerarchia, concorrenza ed eventi.

Poiché i diagrammi di stato offrono una visione a livello di sistema, potete utilizzarli come specifiche dei programmi eseguibili. Il modello di programmazione a diagrammi di stato è parti-

STORIA DEI DIAGRAMMI DI STATO

Il diagramma di stato è stato inventato da David Harel del Weizmann Institute of Science negli anni '80. Secondo Harel, lo scopo del diagramma di stato era quello di "estendere i convenzionali diagrammi a stati-transizioni con ... le nozioni di gerarchia, concorrenza e comunicazione". Harel ha inventato il diagramma mentre contribuiva a progettare un sistema avionico complesso, presumibilmente evidenziando le carenze degli strumenti esistenti per un sistema di quel tipo. Negli anni '90, i diagrammi di stato sono stati adottati come diagrammi comportamentali nell'ambito della specifica UML (Unified Modeling Language), largamente utilizzata per la modellazione dei sistemi embedded.

FUNZIONAMENTO DEI DIAGRAMMI DI STATO

Per iniziare a capire i diagrammi di stato, è meglio partire dal classico diagramma a stati e aggiungere quindi le nozioni di gerarchia, concorrenza ed eventi. Il classico diagramma a stati consiste di due costrutti principali: gli stati e le transizioni. Nella figura 2, il diagramma a stati descrive un semplice distributore automatico di bibite con cinque stati e sette transizioni per illustrare il funzionamento della macchina. La macchina parte dallo stato di 'riposo' e passa nello stato 'conteggio monete' quando si inseriscono delle monete. Il diagramma a stati contiene stati e transizioni aggiuntivi dove la macchina è in attesa di una selezione, dispensa una bibita e fornisce il resto.

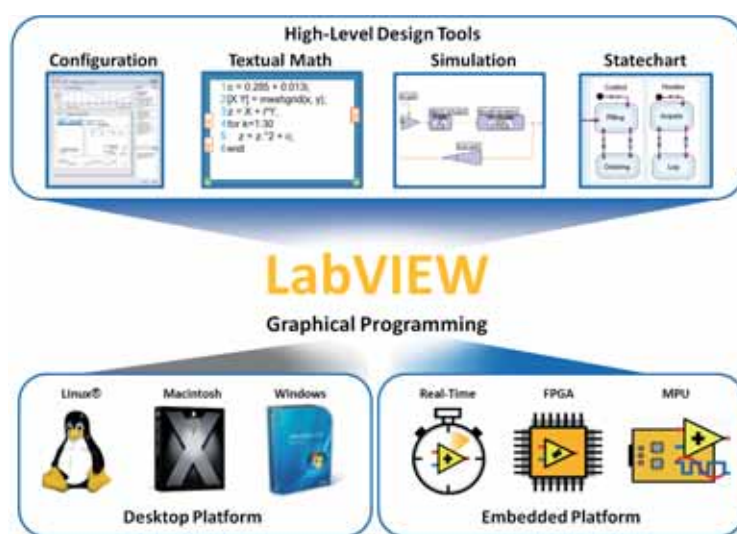


Figura 1 - La piattaforma grafica di sviluppo NI LabVIEW

La piattaforma di progettazione grafica di sistemi di National Instruments combina i modelli di programmazione di LabVIEW con I/O di misura e controller commerciali desktop ed embedded. Grazie a questa combinazione, potete avere una catena di strumenti di sviluppo integrata per la progettazione, la prototipazione e la messa in

colamente utile per sviluppare sistemi complessi che devono rispondere ad una varietà di eventi, come sistemi di controllo embedded e sistemi di telecomunicazione.

Con lo Statechart Module potete distribuire progetti su un'ampia gamma di piattaforme hardware che spazia dai PC desktop ai chip FPGA.

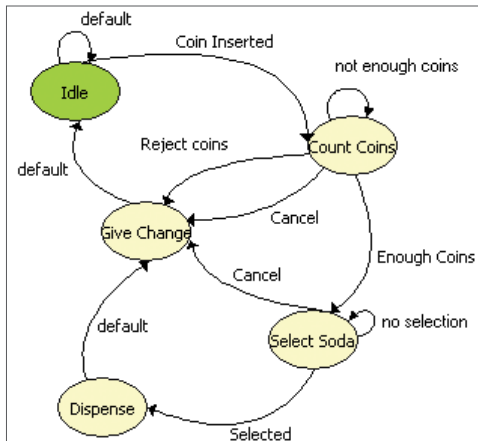


Figura 2 - Diagramma a stati classico

La figura 3 illustra un diagramma di stato che descrive il comportamento della stessa macchina.

Notate come la nozione di gerarchia ed eventi riduca il numero di stati e transizioni. Nel diagramma di stato, potete annidare gli stati 'conteggio delle monete' e 'dispensazione' all'interno di un superstato.

Dovete definire semplicemente una transizione (T3) da uno di questi due stati allo stato 'dare il resto'.

Potete configurare la transizione T3 per rispondere a tre eventi: bibita erogata, resto richiesto o monete rifiutate. Inoltre, potete eliminare lo stato 'selezione della bibita' nel classico diagramma a stati introducendo una condizione di 'guardia' sulla transizione T2.

Le condizioni di guardia devono diventare 'vere' perché la transizione possa avvenire.

Se il risultato della condizione di guardia è 'falso', l'evento viene ignorato e la transizione non si verifica.

A questo punto, potete espandere il diagramma di stato per dimostrare la nozione di concorrenza

aggiungendo un elemento di controllo della temperatura al software dentro il distributore automatico. La figura 4 spiega come potete incapsulare la logica di dispensazione ed il controllo della temperatura in un and state.

Gli and-state descrivono un sistema che è simultaneamente in due stati indipendenti fra loro.

La transizione T7 descrive come i diagrammi di stato possono definire

un'uscita che vale per entrambi i sotto-diagrammi di stato. Oltre a gerarchia e concorrenza, i diagrammi di stato hanno altre caratteristiche che li rendono utili per i sistemi complessi. I diagrammi di stato hanno un concetto di storia, che permette ad un superstato di 'ricordare' quale sottostato al suo interno è stato attivo in precedenza.

Consideriamo, per esempio, un superstato che descrive una macchina che versa una sostanza e quindi la scalda. Un evento di arresto potrebbe mandare in pausa l'esecuzione della macchina mentre essa sta versando. Quando si verifica un evento di ripresa, la macchina ricorda di riprendere a versare.

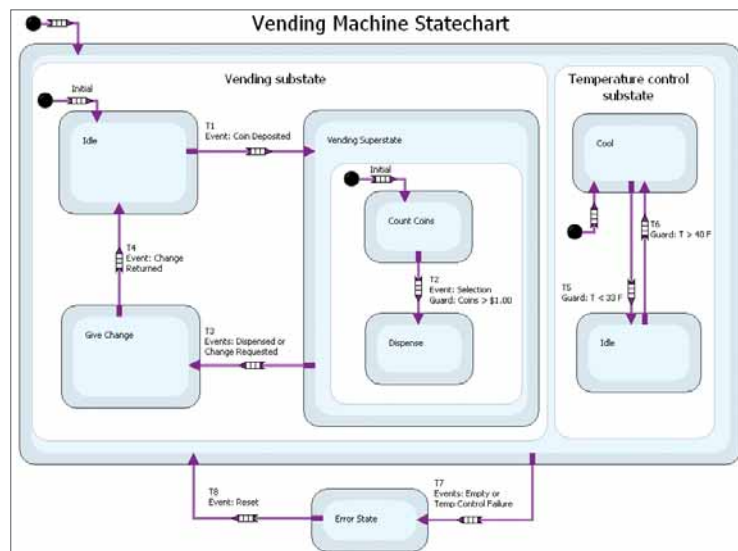


Figura 4 - Implementazione di un and state

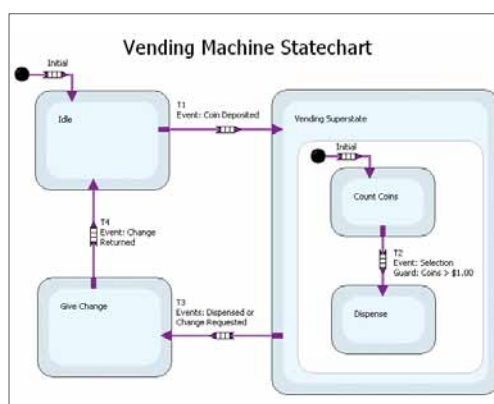


Figura 3 - Diagramma di stato, realizzato tramite statechart s, equivalente alla versione di figura 2

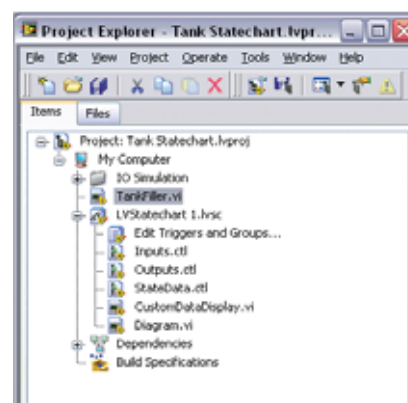


Figura 5 - Progetto di LabVIEW con definizione dello statechart LVStatechart 1.lvsc

USO DEI DIAGRAMMI DI STATO DI LABVIEW

Con lo Statechart Module di LabVIEW potete progettare componenti software con un diagramma di stato e definire il comportamento degli stati e la logica di transizione con la programmazione grafica a flusso di dati. Utilizzate il Project Explorer di LabVIEW per integrare pienamente i diagrammi di stato nell'ambiente di sviluppo. Ogni diagramma di stato di LabVIEW ha diversi componenti che potete utilizzare per configurare il contesto del progetto. La figura 5 descrive un diagramma di stato esemplificativo denominato LVStatechart 1.lvsc.

Potete creare trigger che corrispondono a transizioni e a risposte degli stati, nonché editare l'elenco delle variabili di ingresso e uscita utilizzate dal diagramma di stato.

Il file Diagram.vi contiene l'effettivo diagramma di stato.

All'interno di tale diagramma, potete creare gli stati del sistema e le transizioni fra loro.

Uno dei principali benefici dei diagrammi di stato è come rappresentano visivamente il comportamento del sistema e, quindi, il modo in cui essi autodocumentano il software. La figura 6 illustra un diagramma di stato che

descrive una macchina per imballaggio. Potete facilmente vedere i diversi stati della macchina e le transizioni fra gli stati. I diagrammi di stato sono molto utili anche per descrivere sistemi reattivi. Potete progettare ciascuno stato con più risposte associate, che corrispondono ad una varietà di trigger o eventi, inviati al diagramma di stato da un dispositivo hardware o da un'interfaccia utente.

Le reazioni sono implementate con la programmazione grafica di LabVIEW.

La figura 6 mostra il codice LabVIEW che viene eseguito quando il sistema è nello stato 'Produzione' e si verifica il trigger 'Livello materiale basso'. I trigger possono anche causare transizioni da eseguire fra gli stati.

Un modo alternativo per determinare le transizioni è utilizzare codice LabVIEW per valutare una guardia.

Una guardia descrive le condizioni che devono essere soddisfatte per potere eseguire una transizione.

La figura 7 illustra il codice di guardia per la transizione denominata 'Materiali esauriti'. Il codice LabVIEW assicura che il livello sia minore di 35,5 per eseguire la transizione dallo stato 'Produzione' allo stato 'Standby'.

Per soddisfare le esigenze associate a casi di utilizzo differenti, i diagrammi

di stato di LabVIEW generano codice secondo due modalità di esecuzione: modalità sincrona e asincrona.

Il modo sincrono è progettato per descrivere il comportamento di un controllore con diversi stati che reagiscono ad un set di ingressi di I/O aggiornati ad una frequenza costante.

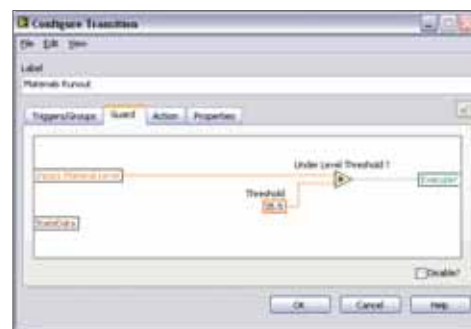


Figura 7 - Codice di guardia di una transizione

Questo modo può essere applicato a sistemi di controllo embedded come le unità di controllo motori (ECU), motion controller e controllori ambientali.

Il modo asincrono è progettato invece per applicazioni che coinvolgono eventi provenienti da un'applicazione esterna. Esso è utile per programmare interfacce uomo-macchina (HMI) e per modellare sistemi ed algoritmi basati su eventi.

Quando avete determinato il corretto modo di esecuzione per il vostro diagramma di stato, potete generare codice eseguibile sotto forma di un subVI modulare, o chiamata di funzione. Potete quindi chiamare il subVI dall'interno di uno schema a blocchi di LabVIEW come illustrato nella figura 7. Potete eseguire visivamente il debug del diagramma di stato sia attraverso la modalità di Highlight Execution presente in LabVIEW, sia attraverso elementi di debugging standard come breakpoint, sonde (finestre mobili di osservazione) e l'esecuzione passo-passo. Potete generare codice basato su diagrammi di stato per una varietà di piattaforme hardware, inclusi sistemi desktop, interfacce uomo-macchina (HMI), programmable automation con-

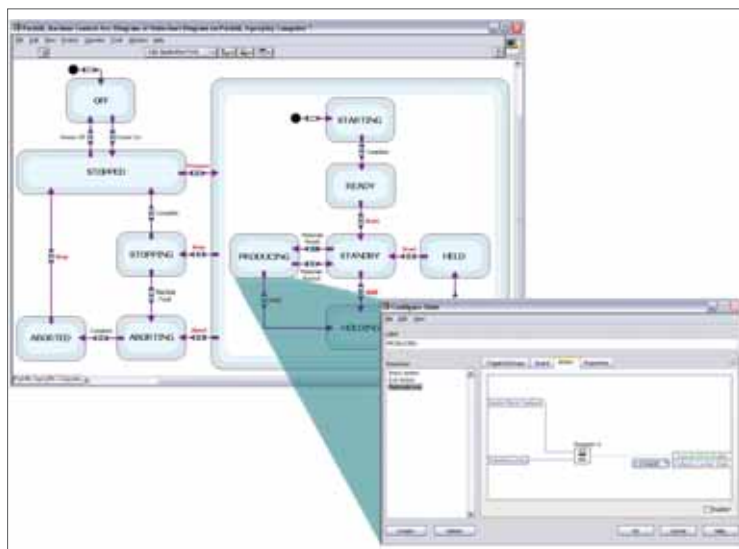


Figura 6 - Diagramma di stato che descrive una macchina per imballaggio

troller (PAC) come NI CompactRIO e PXI, field-programmable gate array (FPGA) presenti su hardware NI e qualsiasi microprocessore a 32 bit. La possibilità di distribuire applicazioni basate su diagrammi di stato su varie piattaforme hardware rende lo Statechart Module di LabVIEW un eccellente strumento di progettazione per lo sviluppo e la messa in opera di sistemi embedded. Potete anche utilizzare i diagrammi di stato con il Control Design and Simulation Module di LabVIEW per modellare e valutare sistemi ibridi attraverso la simulazione di sistemi dinamici.

BENEFICI DEI DIAGRAMMI DI STATO

L'uso dei diagrammi di stato di LabVIEW per la progettazione di sistemi offre numerosi benefici agli sviluppatori software. In particolare, i diagrammi di stato offrono una visione a livello di sistema che descrive la funzione complessiva di un sistema o applicazione, perché un diagramma di stato cattura ogni possibile stato del sistema. Pertanto, l'uso dei diagrammi di stato aiuta a ridurre la possibilità di "blocchi" software ed altri comportamenti inaspettati, perché siete costretti a considerare ogni alternativa alla quale il software deve rispondere. Come è stato evidenziato nell'articolo, il modello di programmazione a diagrammi di stato è particolarmente utile per i sistemi reattivi, che sono caratterizzati proprio dal loro modo di rispondere agli ingressi.

Potete quindi progettare un sistema in modo scalabile per gestire molteplici risposte dello stato e transizioni sulla base di qualsiasi combinazione di eventi. I diagrammi di stato sono simili ai programmi grafici a flusso di dati, perché sono autodocumentanti e promuovono il facile trasferimento di conoscenza fra gli sviluppatori.

Un nuovo membro di un team di progettazione può osservare un diagramma di stato ed afferrare rapidamente gli elementi del sistema.

CONCLUSIONE

Il modello di calcolo a diagramma di stato offre un metodo sofisticato per affrontare lo sviluppo di applicazioni complesse. I diagrammi di stato sono particolarmente utili per programmare applicazioni che rispondono ad eventi, come intricate interfacce utente e macchine a stati avanzate, utilizzate per implementare controllori di sistemi dinamici, logica di controllo macchine e protocolli digitali di telecomunicazione. Con il nuovo Statechart Module di LabVIEW, potete beneficiare dei brevi tempi di sviluppo e della stretta integrazione hardware offerti dalla piattaforma LabVIEW. Ora potete aggiungere anche i diagrammi di stato alla lista dei vostri strumenti per la programmazione di applicazioni complesse.

Readerservice.it n. 403

Note sull'autore

Laureato in ingegneria nucleare al Politecnico di Milano, Matteo Foini lavora in qualità di Technical Marketing Engineer presso National Instruments Italy

BREVI

Il modulo Real-Time di LabVIEW 8.5 per il multiprocessing simmetrico

L'ultima versione del modulo Real-Time di LabVIEW aggiunge il supporto per il multiprocessing simmetrico (SMP), al fine di estendere il determinismo ai sistemi embedded multicore. Oggi potete utilizzare LabVIEW per progettare rapidamente applicazioni real-time embedded parallele, in grado di sfruttare le prestazioni dei processori multicore.

Con questa nuova funzionalità, i sistemi embedded possono migliorare i tempi-ciclo ed elaborare set di dati di maggiori dimensioni per sviluppare sistemi di controllo ad alte prestazioni, come simulazioni hardware-in-the-loop e sistemi di motion control ad alta velocità.

Inoltre, la nuova versione di Real-Time Execution Trace Toolkit permette il debug di applicazioni LabVIEW multithreaded su sistemi multicore embedded.

Il modulo Real-Time di LabVIEW estende l'approccio di sviluppo grafico di LabVIEW ad hardware NI e di terze parti con sistema operativo real-time, ed offre interfacce verso un'ampia varietà di I/O commerciali, per aiutarvi a sviluppare rapidamente sistemi real-time embedded. Il modulo Real-Time di LabVIEW fa parte della piattaforma embedded di LabVIEW — una famiglia di prodotti National Instruments che comprende i moduli LabVIEW FPGA, LabVIEW PDA, LabVIEW Touch Panel e LabVIEW Microprocessor SDK (Software Development Kit) — che offre un ambiente completo di programmazione per sistemi embedded.

Per maggiori informazioni sul modulo Real-Time di LabVIEW, visitate www.ni.com/realtime.

Readerservice.it n. 404

SEI UN FAN DI LABVIEW?

ENTRA NELLA NOSTRA COMUNITÀ!

Caro lettore, se ritieni che questa nuova iniziativa LabVIEW World sia un utile strumento per la tua attività e desideri continuare a rimanere aggiornato sul mondo LabVIEW, collegati alla pagina

http://www.ilb2b.it/labview/lb_world.asp