



Training per esperti

ANALISI DEL PROGETTO

a cura di Matteo Foini

Analizzare un progetto prima di iniziare a costruire un VI può aiutarvi a sviluppare VI in modo più efficiente e ad evitare banchi nelle funzionalità. Questa lezione descrive i passi essenziali del processo di analisi di un progetto: valutazione del documento delle specifiche, creazione del documento dei requisiti e definizione dell'applicazione. Alla fine della lezione, analizzerete un documento delle specifiche ed un documento dei requisiti. Il documento dei requisiti definisce le caratteristiche richieste all'applicazione affinché funzioni secondo le specifiche

A. VALUTAZIONE DELLE ESIGENZE DEL CLIENTE

Tipicamente, si sviluppa del software perché qualcuno ha evidenziato la necessità del software stesso. Per esempio, un cliente potrebbe identificare un'esigenza ed assumervi per creare software in grado di soddisfarla. Prima che possiate sviluppare l'applicazione, dovete analizzare il progetto per essere sicuri di avere compreso a fondo le esigenze del cliente. Il cliente potrebbe fornirvi un documento delle specifiche che definisce la funzione dell'applicazione per la cui creazione siete stati assunti.

Il cliente non sempre produce il documento delle specifiche. Spesso, dovete raccogliere le specifiche sulla base delle informazioni fornite dal cliente. Il documento delle specifiche dovrebbe permettere una comprensione dettagliata del software da produrre.

Tuttavia, il documento delle specifiche non è necessariamente l'architettura del software. Il cliente potrebbe non avere compreso a fondo il problema che desidera farvi risolvere. In questo caso, dovete determinare i dettagli mancanti. Un buon progetto inizia sempre da una buona comprensione delle specifiche software.

E, per capire le specifiche, è necessario leggerle attentamente e sviluppare domande da porre al cliente se l'intento di qualche specifica non è chiaro.

La fase delle specifiche del processo di progettazione permette anche di identificare eventuali aree mancanti nel documento delle specifiche. Questo è il primo passo nella produzione di un documento dei requisiti. Le specifiche possono essere di due tipi: funzionali e non funzionali.

- **Specifiche funzionali** - Definiscono le funzioni che il software deve eseguire. Gli esempi seguenti sono specifiche funzionali:

- Campionare i dati a 100 kS/s
- Eseguire una Trasformata Veloce di Fourier
- Memorizzare i dati in un database

- **Specifiche non funzionali** - Definiscono le caratteristiche o gli attributi di un'implementazione software. Gli esempi seguenti sono specifiche non funzionali:

- Il sistema deve essere affidabile fino a 10 utenti simultanei

- Il sistema deve essere implementato entro il 10 Ottobre
- I VI devono essere scalabili

Le specifiche funzionali e non funzionali sono legate fra loro. Separare le specifiche nelle categorie funzionale e non funzionale può facilitare l'implementazione di un'applicazione. Le specifiche funzionali diventano requisiti nel documento dei requisiti.

Successivamente, quando implementerete le specifiche, le specifiche funzionali diventeranno moduli dell'applicazione. Le specifiche non funzionali definiscono gli attributi delle specifiche funzionali.

Le specifiche definiscono ciò che il cliente desidera ottenere con il software o quali funzioni devono essere eseguite dall'applicazione.

Un'analisi dettagliata delle specifiche produce le esigenze partendo dai desideri. Molte volte, un documento delle specifiche diventa un elenco dei desideri anziché un documento ben definito. Eliminando le specifiche non essenziali dal documento delle specifiche si favorisce la prevenzione di un errato funzionamento delle caratteristiche.

Potete ottenere le esigenze dai desideri analizzando le parole chiave nella specifica.

Parole come "faremo", "dovremo" e "potremo" definiscono le specifiche che il sistema deve implementare.

Parole come "faremmo", "dovremmo", "potremmo" indicano specifiche che il cliente desidera ma di cui potrebbe non avere bisogno.

Potete organizzare le specifiche che sono semplici desideri in un elenco dei desideri da utilizzare in futuro.

L'analisi delle specifiche richiede una buona comprensione dello scopo generale dell'applicazione.

Anche se il cliente ha fornito un documento delle specifiche completo, è importante esaminarlo in modo dettagliato e determinare se nelle specifiche manca qualche elemento. Fate domande e raccogliete tutti i requisiti che non sono menzionati nel documento.

ELENCO DI VERIFICA DELLE SPECIFICHE

Potete utilizzare il seguente elenco di controllo per identificare le aree d'interesse in un documento delle specifiche.

Non tutte le voci dell'elenco sono pertinenti ad ogni specifica di progetto.

- ☐ Esiste già un documento delle specifiche?
- ☐ Le operazioni eseguite dall'utente sono indicate nel documento delle specifiche?
- ☐ Le specifiche sono definite chiaramente?
- ☐ Il documento specifica chiaramente i requisiti tecnici del progetto?
- ☐ Il documento delle specifiche è privo di conflitti?
- ☐ Sono state elencate tutte le specifiche?
- ☐ I costi e i tempi sono indicati chiaramente?
- ☐ Tutte le specifiche sono implementabili?
- ☐ Dal documento delle specifiche sono stati esclusi dettagli del progetto?
- ☐ Ogni specifica è verificabile mediante collaudo?

Per un documento delle specifiche completo e dettagliato, è necessario verificare tutti gli elementi applicabili dell'elenco di controllo. Tutti gli elementi che non potete verificare rappresentano domande da porre al vostro cliente.

B. COMUNICARE CON IL CLIENTE

Il processo di finalizzazione di un documento delle specifiche richiede la raccolta di informazioni e la comunicazione con il cliente per chiarire le specifiche.

Dovete interagire strettamente con il cliente per essere sicuri che il documento delle specifiche contenga tutte le informazioni necessarie. Se manca qualche informazione, dovete determinare che cosa manca e aggiungerlo al documento delle specifiche.

Le aspettative del cliente e del programmatore sono diverse. Dovete quindi determinare le aspettative del cliente ed il modo per relazionarvi ad esse.

ASPETTATIVE DEL CLIENTE NEI CONFRONTI DEL PROGRAMMATORE

Considerate le seguenti aspettative che il cliente può avere nei vostri confronti come programmatore:

- Capisce che cosa gli è stato detto
- Comunica le sue idee di progettazione
- Comunica eventuali problemi
- Rispetta i tempi
- Mantiene le promesse
- Pone le domande corrette
- Dirige il progetto

ASPETTATIVE DEL PROGRAMMATORE NEI CONFRONTI DEL CLIENTE

Considerate le seguenti aspettative che potete avere nei confronti del cliente:

- Sa che cosa vuole
- Comunica le sue esigenze

- Conosce eventuali problemi
- E' coerente nell'esposizione del suo punto di vista.

Poiché le loro aspettative reciproche sono diverse, il cliente ed il programmatore parlano lingue differenti. E' importante capire che il cliente si aspetta che risolvi molti degli elementi riguardanti i requisiti finali del progetto. Potete usare questa aspettativa a vostro vantaggio, perché potete raccomandare la tecnologia migliore per risolvere un particolare problema. Per esempio, il cliente potrebbe richiedere la visualizzazione di dati in formato tabulare, ma dopo avere valutato i dati, voi potreste raccomandare la visualizzazione in un tracciato grafico, in modo che i dati siano disponibili visivamente. E' proprio nella risoluzione dei problemi riguardanti i requisiti finali del progetto che potete aggiungere valore al processo di analisi del progetto stesso. Comunicare con il vostro cliente può anche contribuire a ridurre i costi. Se le specifiche cambiano durante la fase di implementazione, potreste essere costretti a scartare del codice. Non è facile aggiungere cambiamenti ad un'applicazione esistente.

Studi software compiuti da aziende come IBM, TRW e GTE hanno messo in luce i significativi incrementi di costo associati al cambiamento delle specifiche durante le varie fasi del processo di sviluppo del software.

Se si verifica un cambiamento durante la fase di progettazione, esso è cinque volte più costoso. Se si verifica un cambiamento nella fase di implementazione, esso è dieci volte più costoso. Se si verifica un cambiamento nella fase di test, esso è venti volte più costoso.

Se il cambiamento si verifica durante la messa in opera, esso è 50 volte più costoso. E' quindi importante accertarsi che il documento delle specifiche sia il più completo possibile, in modo che eventuali cambiamenti possano verificarsi solo nella fase di progettazione.

E' inoltre importante fare in modo di comunicare con le persone giuste. Comunicate sempre con la persona che ha un'autorità sulla progettazione. Per evitare frustrazioni, formalizzate il processo decisionale in modo che la persona che ha la parola finale sul progetto sia identificata e sia consapevole dei requisiti del progetto stesso.

Per comunicare efficacemente con il vostro cliente, accertatevi innanzitutto di potere verificare tutte le voci applicabili nell'Elenco del Contenuto delle Specifiche.

Tutti gli elementi che non potete verificare rappresentano domande da porre al vostro cliente.

ELENCO DI VERIFICA DELLE COMUNICAZIONI CON IL CLIENTE

Quando comunicate con il vostro cliente utilizzate il seguente elenco di controllo per accertarvi di essere concordi sui singoli elementi delle specifiche:

- ☐ Sapete perché ciascuna specifica deve essere implementata?

- ☐ I metodi di gestione dei dati e reportistica sono stati definiti?
- ☐ Le esigenze di interfaccia utente sono state definite?
- ☐ L'hardware è stato definito?
- ☐ Tutte le funzioni che il software deve eseguire sono state definite?

C. SVILUPPO DEL DOCUMENTO DEI REQUISITI

I documenti dei requisiti forniscono un terreno comune per voi ed il cliente sul quale lavorare. Dopo le riunioni sulle specifiche e le riunioni sui requisiti con il cliente, dovreste avere tutte le informazioni che vi occorrono per produrre un documento dei requisiti. Il documento dei requisiti può essere creato da voi o dal cliente. Se un progetto richiede che siate voi a creare il documento dei requisiti, usate le linee guida in questo paragrafo per determinare più facilmente gli elementi da includere.

Ricordate che, a questo punto del processo di sviluppo, il progetto del software non è ancora iniziato, ma avverrà più tardi. Scrivete il documento dei requisiti in un linguaggio comprensibile sia a voi che al vostro cliente, in modo da potere completare il progetto del software.

L'IEEE (Institute of Electrical and Electronic Engineers) definisce gli standard per l'ingegneria del software, inclusi gli standard per i documenti dei requisiti. In questo articolo, per il documento dei requisiti utilizziamo un formato simile ai documenti dei requisiti inclusi negli esami NI Certified LabVIEW Developer.

ELENCO DI CONTROLLO DEL DOCUMENTO DEI REQUISITI

Utilizzate il seguente elenco di controllo per essere sicuri che il documento dei requisiti sia completo e adeguato. L'elenco non è completo, ma offre un punto di partenza per stabilire se il documento dei requisiti è appropriato:

- ☐ Ogni requisito deve essere chiaro e comprensibile.
- ☐ Ogni requisito ha un solo significato.
- ☐ I requisiti spiegano il comportamento funzionale del sistema software.
- ☐ I requisiti non si contraddicono fra loro.
- ☐ I requisiti non specificano un comportamento non valido.
- ☐ Ogni requisito può essere testato.

D. DEFINIZIONE DELL'APPLICAZIONE

I requisiti del progetto descrivono ciò che il sistema software deve fare. Avere un set di requisiti del progetto è il prossimo passo verso lo sviluppo dell'applicazione.

Dopo avere comunicato con il cliente ed avere finalizzato le specifiche del progetto, potete determinare i requisiti del progetto. I requisiti sono potenti strumenti che vi dicono esattamente che cosa il software dovrebbe fare.

Quindi, i requisiti devono contenere un dettaglio sufficiente. Se i requisiti non sono abbastanza dettagliati, potreste perdere di vista l'intento del cliente ed implementare codi-

ce che non soddisfa le sue esigenze. Determinare i requisiti del progetto è pertanto essenziale per sviluppare un documento dei requisiti. Prima di sviluppare un progetto dettagliato di un sistema, definite chiaramente gli obiettivi. Iniziate componendo un elenco dei requisiti.

Alcuni requisiti sono specifici, come i tipi di I/O, le velocità di campionamento o la necessità di analisi real-time.

Fate qualche ricerca in questa fase iniziale per essere sicuri di riuscire a soddisfare le specifiche.

Altri requisiti dipendono invece dalle preferenze dell'utente, come i formati dei file o gli stili grafici. Potete ricavare l'elenco dei requisiti direttamente dalle specifiche.

Cercate di distinguere fra i requisiti assoluti e quelli desiderati. Anche se potete cercare di soddisfare tutte le richieste, è meglio avere un'idea circa le caratteristiche che potete sacrificare se il tempo non vi basta.

Fate attenzione a non rendere talmente dettagliati i requisiti da vincolare il progetto. Per esempio, quando progettate un sistema di I/O, il cliente probabilmente ha determinato requisiti in termini di velocità di campionamento e precisione. Tuttavia, anche il costo è un vincolo. Includete tutti questi elementi nei requisiti.

Se potete evitare di specificare l'hardware, potete semplicemente mettere a punto il progetto dopo avere iniziato la prototipazione ed il benchmarking dei vari componenti. Fintantoché i costi rientrano nel budget specificato e gli aspetti di timing e precisione sono rispettati, il cliente può non fare caso al fatto che il sistema utilizzi un particolare tipo di scheda plug-in o altro hardware.

Un altro esempio di vincolo globale di un progetto è quello di essere troppo specifici circa il formato di visualizzazione utilizzato nelle varie schermate con le quali il cliente interagisce. Un'immagine di un display può essere utile per spiegare i requisiti, ma dovete essere chiari sul fatto che l'immagine sia un requisito o una linea guida.

Alcuni progettisti attraversano notevoli difficoltà per produrre un sistema che si comporti in un modo specifico perché un certo comportamento faceva parte dei requisiti. In questo caso, esiste probabilmente una soluzione più semplice che produce gli stessi risultati ad un costo minore e in un periodo di tempo più breve. Un modo per limitare la quantità d'informazione che dovete scrivere quando analizzate un progetto è quello di disegnare uno schema dell'applicazione. Creare uno schema aiuta a migliorare la propria capacità di progettare l'applicazione e trasferire idee al vostro cliente. Il primo passo nella creazione di uno schema è determinare i componenti astratti di ogni requisito.

ASTRAZIONE DEI COMPONENTI DAI REQUISITI

Lo sviluppo di applicazioni complesse richiede tecniche di progettazione per realizzare elementi complessi a livello software. L'astrazione descrive intuitivamente un'applicazione senza spiegare come scrivere l'applicazione

stessa. L'astrazione generalizza l'applicazione ed elimina i dettagli. In generale, l'astrazione può essere di due categorie: l'astrazione procedurale e l'astrazione dei dati.

Astrazione procedurale

L'astrazione procedurale separa ciò che una procedura deve ottenere da come la procedura è implementata.

Per esempio, consideriamo la seguente descrizione di un'applicazione.

Procedura	Implementazione
Aprire il file datalog.	
Ottenere l'informazione di frequenza per i dati.	
Visualizzare tutti i dati che soddisfano il parametro di ricerca.	

Con questo esempio di astrazione procedurale, l'implementazione della procedura è separata dalla funzione della procedura stessa.

Astrazione dei dati

L'astrazione dei dati separa i dati che desiderate memorizzare dal mezzo fisico di memorizzazione dei dati stessi. L'astrazione dei dati offre una visione più logica dei dati, rispetto ai bit e ai byte che formano i dati. Un esempio di astrazione dei dati è un cluster. Un cluster può rappresentare i dati in modo più logico, senza richiedere all'utente di interessarsi ai dettagli della sua implementazione. Consideriamo il seguente estratto dal documento dei requisiti di un sistema di controllo delle luci teatrali. Per analizzare il documento dei requisiti di questo esempio, estraiamo tutti i nomi dal documento ed utilizziamo i nomi per la definizione dei componenti.

Inizio dell'estratto del documento dei requisiti

OBIETTIVO

Progettare, implementare, collaudare e mettere in opera un sistema di controllo delle luci teatrali che permetta a un tecnico delle luci di controllare e programmare facilmente le luci teatrali per qualsiasi produzione. Comandi sul pannello frontale controllano il funzionamento del software di controllo delle luci teatrali. Indicatori sul pannello frontale indicano lo stato corrente del software di controllo delle luci teatrali.

FUNZIONAMENTO GENERALE

Il controllore memorizza l'intensità del canale, il colore del canale, il tempo di attesa del canale, il tempo di attenuazione del canale, il tempo di attuazione del canale ed il nome dell'effetto quando l'utente clicca il pulsante

Record. Quando l'utente clicca il pulsante **Play**, il controllore attua ogni effetto contenuto nel Controllo degli Effetti ciclando attraverso gli effetti registrati, a partire dal primo effetto nel Controllo degli Effetti. Un effetto riprodotto deve aspettare il tempo di attesa specificato, quindi attenuare i canali al colore e all'intensità desiderati entro il tempo di attenuazione specificato ed infine aspettare per il tempo di attuazione specificato. L'utente può interrompere l'effetto attualmente in riproduzione cliccando il pulsante **Stop**. L'utente può spostare verso l'alto un effetto nel Controllo degli Effetti cliccando il pulsante **Up**. L'utente può spostare verso il basso un effetto nel Controllo degli Effetti cliccando il pulsante **Down**. L'utente può cancellare un effetto nel Controllo degli Effetti cliccando il pulsante **Delete**, che cancella l'effetto attualmente selezionato. Il controllore si ferma quando l'utente seleziona **File»Exit**.

Fine dell'estratto del documento dei requisiti

L'elenco seguente indica i nomi ottenuti dal precedente documento dei requisiti: sistema di controllo delle luci teatrali, ingegnere delle luci teatrali, luci teatrali, controlli di produzione, pannello frontale, indicatori, stato, controllore, intensità del canale, colore del canale, tempo di attesa del canale, tempo di attenuazione del canale, tempo di attuazione del canale, nome, effetto, utente, pulsante di registrazione, pulsante di riproduzione, controllo degli effetti, pulsante di interruzione, pulsante Up, pulsante Down, pulsante di cancellazione, effetto selezionato.

Quando avete ottenuto un set di nomi, potete raggrupparli formando componenti effettivi del vostro sistema. La tabella seguente organizza i componenti in componenti più astratti:

Nomi	Nomi astratti
Luci teatrali	Hardware
Indicatori	Display
Stato	
Pulsante di registrazione	
Pulsante di riproduzione	
Pulsante di interruzione	
Pulsante Up	
Pulsante Down	
Pulsante di cancellazione	
Controllo degli effetti	
Effetto selezionato	
Intensità del canale	Effetto
Nome	
Effetto	
Colore del canale	
Tempo di attesa del canale	Temporizzazione
Tempo di attenuazione del canale	
Tempo di attuazione del canale	

Applicando la tecnica di astrazione all'intero documento dei requisiti si ha una soluzione per determinare un set di componenti astratti. Il sistema di controllo delle luci teatrali ha i seguenti componenti: Hardware, Display, Effetto, Temporizzazione, Errore e File. I componenti diventano i moduli del sistema.



Nota I componenti astratti sono un set di componenti che vi permettono di modularizzare il sistema. Potete astrarre un numero infinito di componenti. L'esperienza e la conoscenza del problema vi permetteranno di astrarre il set migliore di componenti per l'applicazione.

L'uso di documenti dei requisiti correttamente formattati facilita la determinazione delle azioni per ogni modulo. Il documento dei requisiti contiene molti verbi ed espressioni verbali che rappresentano il comportamento del sistema e sono tipicamente correlati ai VI che definirete. Potete fare corrispondere i verbi e le espressioni verbali con i componenti astratti che eseguono le azioni. Consideriamo il seguente estratto dal documento dei requisiti di un sistema di controllo delle luci teatrali.

Inizio dell'estratto del documento dei requisiti

RIPRODUZIONE

Cliccate il pulsante Play per riprodurre gli effetti registrati. Quando la riproduzione inizia, il controllore dovrebbe disabilitare i pulsanti di spostamento verso l'alto degli effetti, spostamento verso il basso degli effetti, cancellazione, registrazione e riproduzione sul pannello frontale. I valori del primo effetto nell'elenco degli effetti vengono caricati in memoria. Il controllore attende in base al numero di secondi specificato per il tempo di attesa dell'effetto corrente. Il controllore quindi smorza o amplifica il canale in base all'intensità corrente del canale stesso ed all'intensità desiderata per il canale. Il software scrive il colore e l'intensità nel sistema di controllo hardware delle luci teatrali ed aggiorna i canali sul pannello frontale. Il controllore deve finire lo smorzamento entro il tempo di smorzamento specificato. Il controllore finisce l'elaborazione del tono aspettando il numero di secondi specificato per il tempo di attuazione dell'effetto corrente. Quando la riproduzione è terminata, il controllore deve abilitare i pulsanti di spostamento verso l'alto degli effetti, spostamento verso il basso degli effetti, cancellazione, registrazione e riproduzione sul pannello frontale.

Fine dell'estratto del documento dei requisiti

Estraete i verbi dall'estratto del documento dei requisiti. Considerate il contesto dei verbi per stabilire le funzioni eseguite dai componenti astratti. L'elenco seguente indica i verbi dal precedente estratto del documento dei requisiti: cliccare, disabilitare, caricare, attendere, smorzare, scrivere, abilitare, aggiornare.

Quando avete ottenuto un set di verbi, potete organizzare i verbi per determinare le funzioni eseguite dai componenti astratti. Potete ignorare i verbi che non migliorano la vostra conoscenza del sistema, come *cliccare* nell'elenco precedente. Dovete anche cercare di ricordare il contesto dei verbi. La tabella seguente organizza i verbi in componenti astratti.

Verbo	Componente astratto
Disabilitare	Display
Abilitare	
Aggiornare	
Caricare	Effetto
Scrivere	Hardware
Attendere	Temporizzazione
Smorzare	

Creando una frase per ogni verbo si migliora la leggibilità e si ottiene la seguente lista di funzioni.

Espressione verbale	Componente astratto
Disabilita il pannello frontale	Display
Aggiorna il pannello frontale	
Otteni l'effetto	Effetto
Scrivi colore e intensità	Hardware
Attendi, esegui	Temporizzazione
Smorza	

Eseguendo l'analisi verbale sull'intero documento dei requisiti si può generare un elenco delle singole azioni eseguite da ciascun componente.

La tabella seguente elenca ciascun componente e le singole azioni eseguite dal modulo.

Modulo	Azioni
Hardware	Scrivi colore e intensità
Display	Inizializza il pannello frontale
	Aggiorna il pannello frontale
	Seleziona un effetto
	Abilita/disabilita il pannello frontale
	Aggiorna la lista degli effetti
Effetto	Aggiungi effetto
	Cancella effetto
	Preleva i valori dell'effetto
	Imposta i valori dell'effetto
	Scambia
	Preleva il numero degli effetti
	Preleva effetto vuoto
Temporizzazione	Attendi
	Smorza
	Attua
Errore	Crea report errori
	Gestisci errori
File	Salva effetti
	Leggi effetti

Dopo avere identificato le azioni ad alto livello per ogni modulo, potete dividere ciascuna azione in azioni più piccole. Questa tecnica vi aiuterà ad analizzare più a fondo l'applicazione e ad assicurarvi che ciascun modulo esegua un solo compito. Se un modulo esegue più di un compito, infatti, è difficile testare il modulo e determinare se funziona correttamente.

Se determinate che un modulo esegue più di un compito, dividetelo in più moduli. Con questo approccio, ogni modulo esegue un solo compito specifico. Creare un progetto software con questo livello di dettaglio vi aiuta a sviluppare VI scalabili, leggibili e consistenti di codice riutilizzabile.

ACCOPIAMENTO

Un modulo raramente vive da solo. Un VI viene richiamato da altri VI. I VI di livello superiore possono diventare dei plug-in in altri VI o utilità per il sistema operativo. L'accoppiamento indica le connessioni esistenti fra i moduli. Qualsiasi modulo che si basa su un altro modulo per eseguire qualche funzione è accoppiato a quel modulo. Se un modulo non funziona correttamente, anche gli altri moduli a cui esso è accoppiato non funzionano correttamente. Più sono le connessioni fra i moduli, più è difficile determinare quale modulo causa un problema.

L'accoppiamento misura quante dipendenze sono contenute in un sistema di moduli. Meno sono le uscite esposte da un modulo, più potrete cambiare il codice all'interno del modulo senza bloccare altri moduli. Un basso accoppiamento contribuisce inoltre a rendere più efficiente il tracciamento degli errori software. Per esempio, un errore di acquisizione dati non può verificarsi in un VI che esegue solo un controllo di strumenti GPIB. Se in un VI GPIB appare un errore di acquisizione dati, sapete che il codice probabilmente ha un accoppiamento troppo stretto.

Il modello a flusso di dati di LabVIEW incoraggia automaticamente un basso accoppiamento fra i moduli. Modello a flusso di dati significa che ogni VI ha solo l'informazione che gli occorre per eseguire il suo compito. Un VI non può cambiare i dati in un altro VI.

Potete introdurre altri percorsi di comunicazione, come variabili globali, code, collegamenti di I/O, ecc. che aumentano l'accoppiamento, ma un tipico VI è autosufficiente e si accoppia solo con i SubVI che richiama. L'obiettivo di ogni architettura gerarchica è fare in modo che l'implementazione usi un accoppiamento limitato o nullo.

COESIONE

In un VI ben progettato, ogni modulo esegue un solo compito ed è quindi fortemente coesivo. Se un singolo modulo cerca di eseguire più di un compito, dovrete probabilmente dividerlo in più moduli. Quando un modulo cerca di eseguire più compiti, lo schema a blocchi diventa più difficile

da leggere e il codice per un compito può influire sul funzionamento del codice di un altro compito.

In questo caso, eliminare un errore in un compito potrebbe bloccare il codice di un altro compito.

Un VI con una forte coesione ha un compito per modulo.

La forte coesione diminuisce il tempo di sviluppo complessivo. Potete capire più facilmente un singolo modulo, se esso è focalizzato su un compito. Potete leggere rapidamente uno schema a blocchi pulito e senza complicazioni, eseguendo con sicurezza i cambiamenti deliberati.

I VI e le funzioni matematiche forniti con LabVIEW dimostrano una forte coesione. Per esempio, la funzione Seno esegue un singolo compito e lo esegue bene.

Essa è un perfetto esempio di una funzione che ha una forte coesione. Il nome della funzione indica chiaramente ciò che la funzione esegue.

Altri VI che hanno una forte coesione sono i VI e le funzioni File I/O. I VI e le funzioni File I/O mostrano una coesione sequenziale, perché deve essere eseguito un modulo prima che ne possa essere eseguito un altro.

Le funzioni Apri File, Leggi File, Scrivi File e Chiudi File eseguono un singolo compito ciascuna all'interno di ogni modulo. Dopo avere astratto i componenti, potete produrre uno schema che aiuta a visualizzare i componenti astratti. Usate lo schema per iniziare a progettare i singoli moduli nell'applicazione. Ogni componente astratto corrisponde ad un modulo all'interno dell'applicazione.

Al livello superiore, definirete le azioni per ogni componente astratto. La definizione delle azioni vi permetterà di determinare il compito specifico di ogni singolo modulo.

DISEGNO DEI COMPONENTI ASTRATTI

Vi sono due metodi per schematizzare le astrazioni dei componenti di livello superiore: i diagrammi di flusso e a flusso di dati. I paragrafi seguenti mostrano l'uso di ciascun metodo per schematizzare il sistema di controllo delle luci teatrali.

DIAGRAMMI DI FLUSSO

I diagrammi di flusso sono un modo potente per organizzare le idee relative a un blocco di codice. I diagrammi di flusso dovrebbero fornire una buona comprensione del flusso dell'applicazione. Il paradigma di programmazione basato su diagrammi a blocchi usato in LabVIEW è facile da capire e simile a un diagramma di flusso.

Molti tecnici usano già i diagrammi di flusso per descrivere i loro sistemi. Il diagramma di flusso semplifica infatti la conversione del progetto in codice eseguibile.

Lo scopo di un diagramma di flusso è dividere in modo logico un compito in singole parti. Iniziate a progettare la gerarchia dei VI dividendo il problema in parti logiche. La figura 1 illustra un diagramma di flusso per la funzionalità Riproduzione relativa al Controllore delle luci teatrali.

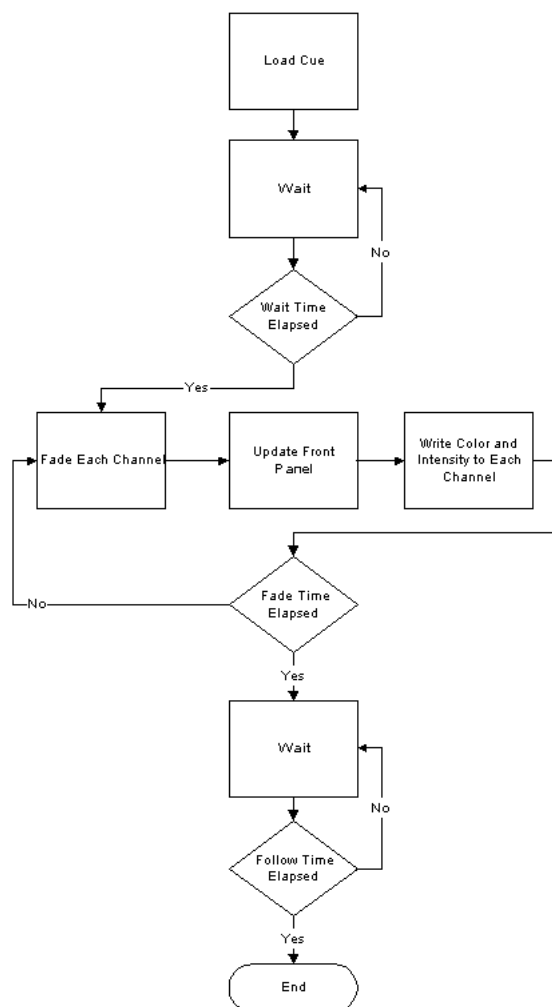


Figura 1. Schema di flusso della funzione di riproduzione nel sistema di controllo delle luci teatrali

DIAGRAMMI A FLUSSO DI DATI

I diagrammi a flusso di dati seguono il paradigma di LabVIEW, in quanto rappresentano il flusso dei dati attraverso un'applicazione.

La figura 2 descrive i simboli comuni utilizzati nei diagrammi a flusso di dati.

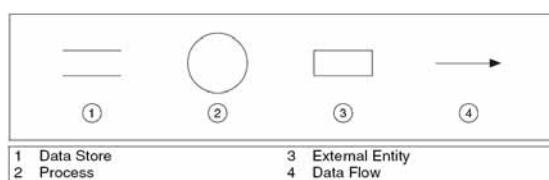


Figura 2. Simboli usati nei diagrammi a flusso di dati

- **Data Store** – Rappresenta file o archivi nel sistema.
- **Process** - Accetta dati in ingresso, elabora i dati e genera dati in uscita.

- **External Entity** – Rappresenta elementi esterni al sistema con i quali il sistema comunica.
- **Data Flow** – Indica il flusso dei dati.

La figura 3 illustra un diagramma a flusso di dati per la funzionalità Riproduzione nel Controllore delle luci teatrali. Notate come il focus di un diagramma a flusso di dati sia rappresentato proprio dai dati, concettualizzando il sistema in base al flusso dei dati attraverso il sistema, analogamente al paradigma a flusso di dati di LabVIEW.

Anche se LabVIEW è un linguaggio di programmazione a flusso di dati, è comunque importante dedicare del tempo lontano da LabVIEW per progettare un sistema appropriato.

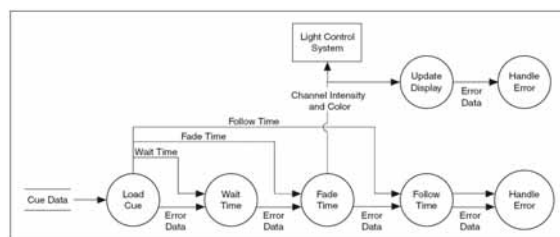


Figura 3. Diagramma a flusso di dati della funzione di riproduzione nel sistema di controllo delle luci teatrali

Come potete vedere dalle differenze esistenti tra il diagramma di flusso e il diagramma a flusso di dati della funzione di riproduzione nel sistema di controllo delle luci teatrali, i processi sono differenti. I nodi del diagramma di flusso rappresentano le funzioni da eseguire.

I nodi del diagramma a flusso di dati rappresentano invece i processi e l'attenzione è posta sul flusso dei dati attraverso il sistema. Potete tradurre un diagramma a flusso di dati in uno schema a blocchi di LabVIEW.

Note sull'autore

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