
Subject: Re: Anwendung des modularen Entwurfs in LabVIEW

Posted by [Brand](#) on Fri, 09 Mar 2007 16:40:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Peter,

LabVIEW ist im Prinzip eine Programmiersprache wie jede andere auch, nur halt graphisch und nicht Text-basiert, und erlaubt bedingt durch den Datenfluß das einfache Programmieren von parallelen Abläufen.

Der modulare Entwurf ist mit LabVIEW meiner Meinung nach auch gut möglich. Es bleibt aber natürlich die analytische Begabung des Entwicklers beim Programmmentwurf gefragt. Diese wird durch keine Programmiersprache der Welt ersetzt. Daran ändert auch die Methodik nichts, funktional oder objekt-orientiert. Die Analyse ist eine intellektuelle Leistung.

1. Man erzeuge ein neues VI und beschreibe seine Funktionalität/Verantwortlichkeit in der VI-Beschreibung (Kontexthilfe). Dieses VI kann ein UseCase-, ein Sequenz- oder Aktivitätsdiagramm darstellen. Das hängt von der aktuellen Abstraktionsebene, die gerade im Fokus ist, ab.
 - 1.1. Falls es sich um ein UseCase handelt, kann man in dieses weitere neue VIs einfügen, und damit das Blockdiagramm zu einem Sequenzdiagramm machen.
 - 1.2. VIs in den Sequenz-VIs beinhalten die Aktivitäten.
 - 1.3. Die Aktivitätsdiagramme beinhalten VIs mit konkreten Implementierungen, siehe Punkt 3.
2. Kontrollen und Indikatoren, die mit der Connector-Pane verbunden sind, definieren die Ein- und Ausgabe-Parameter. Ihre Kontexthilfe erläutern ihre Funktion. Cluster-Daten sollten als Typedef gespeichert werden.
3. Implementierung der Funktionalität.
 - 3.1. Im ersten Schritt kann der zu implementierende Algorithmus als interaktive Benutzerabfrage gestaltet werden.
 - 3.2. Im zweiten Schritt könnte eine Simulation implementiert werden. Die Möglichkeit einer Simulation erlaubt es auch, absichtlich Fehlerzustände zu erzeugen, die man in der realen Anwendung lieber nicht erleben möchte.
 - 3.3. Im dritten Schritt wird dann der echte Algorithmus implementiert.
 - 3.4. Die Umschaltung zwischen diesen drei Modi könnte über ein VI ermöglicht werden, das den zubenutzenden Modus als Ausgabeparameter liefert.

Private und öffentliche VIs können per Konvention in entsprechen gekennzeichneten Verzeichnissen oder .llb-Dateien gespeichert werden. Hier ist natürlich die Disziplin der Entwickler gefragt, sich an die Konventionen zu halten. In LabVIEW 8.20 bieten die Libraries die Möglichkeit diese Disziplin zu erzwingen.

Gruß Holger
