

---

Subject: Anwendung des modularen Entwurfs in LabVIEW  
Posted by [Peter Schieberle](#) on Fri, 09 Mar 2007 09:03:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo \*all,

ich würde hier gerne die Diskussion von  
[http://forum.gsi.de/index.php?t=tree&th=1242&start=0&rid=0&amp;S=8524bb595c8067ca73143f428b03a959#page\\_top](http://forum.gsi.de/index.php?t=tree&th=1242&start=0&rid=0&amp;S=8524bb595c8067ca73143f428b03a959#page_top) fortsetzen.

Erstmal Danke an Holger Brand für die schnelle und fundierte Antwort.

Ein objekt-orientierter Ansatz in LabVIEW hört sich ja sehr vielversprechend an. Da ich noch keine Kenntnisse bezüglich der Integration von OOP in LabVIEW habe, kann ich mir an der Stelle kein Urteil erlauben. Allerdings habe ich bemerkt, daß in anderen Bereichen der Begriff „objektorientiert“ ja inzwischen marketingstrategisch recht inflationär verwendet wird, ohne die erforderlichen Sprachelemente zur Verfügung zu stellen.

Mich interessiert vor allem eine analytische Zerlegung eines großen Projekts in kleinere Teilaufgaben. Dazu reicht mir im ersten Ansatz eigentlich der module Entwurf, wenn die notwendigen Sprachmittel zur Verfügung stehen. An eine Objektorientierung möchte ich mich noch gar nicht wagen.

Ich muß auch gestehen, daß ich noch einige Defizite bezüglich Projektverwaltung (LabVIEW 8.x) und Objektmanagement habe (LabVIEW 8.20), da das aktuelle Kundenprojekt in LabVIEW 7.1 umgesetzt wird. Da ich in dieser Version das Attribut "privat" für ein Vi einer LLB nicht gefunden habe, befürchte ich, daß dies erst ab LabVIEW 8.x möglich ist.

Bye,  
Peter

---

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 ein 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 Programmentwurf 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 Aktivitätsdigramm 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, dass den zubenutzenden Modus als Ausgabeparameter liefert.

Private und öffentliche VIs können per Konvention in entsprechen gekennzeichneten Verzeichnissen oder .lib-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

---

---

Subject: Re: Anwendung des modularen Entwurfs in LabVIEW

Posted by [Brand](#) on Sat, 10 Mar 2007 09:18:00 GMT

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

---

Hi Peter,

ich möchte Dich noch auf folgende Seite der LVUG hinweisen:

<http://www-win.gsi.de/lvug/techniken.htm>.

Hier werden einige grundlegende LabVIEW Programmier Techniken vorgestellt. Ein Schwerpunkt liegt auf Anwendungen, in denen verschiedenen Aufgaben in parallelen Prozessen bearbeitet werden, und wie man diese synchronisiert, sowie ein Beispiel zur Benutzung der VI-Server Methoden, um zur Laufzeit aus VI-Templates neue Prozesse zu starten.

Gruß Holger

---

---

Subject: Re: Anwendung des modularen Entwurfs in LabVIEW

Posted by [Peter Schieberle](#) on Mon, 12 Mar 2007 10:04:48 GMT

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

---

Hallo Holger,

Danke für die beiden Antworten. Ich habe mich am Wochenende mit den verschiedenen Informationen auseinandergesetzt und werde diese Ansätze in mein Projekt einfließen lassen. Bei Bedarf würde ich mich wieder an die Forumsgemeinde wenden.

Eine Frage noch: Inwieweit ist ein Upgrade von LabVIEW 7.0/7.1 auf 8.20 als kritisch zu bewerten. Welchen Aufwand bedeutet es ein mittleres Projekt (ca. 200 Vis.) in die

Projektverwaltung zu integrieren.

Danke !

Bye,  
Peter

---