

EXAMPLE PROGRAM DRAFTS

[Register \(https://lumen.ni.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://forums.ni.com/auth/saml&cifaction=create&isoLang=en&target=https%3A%2F%2Forums.ni.com%2Ft5%2FExample-Program-Drafts%2FApplying-Common-Object-Oriented-OO-Design-Patterns-to-LabVIEW%2Fta-p%2F3510571%3Fprofile.language%3Den\)](https://lumen.ni.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://forums.ni.com/auth/saml&cifaction=create&isoLang=en&target=https%3A%2F%2Forums.ni.com%2Ft5%2FExample-Program-Drafts%2FApplying-Common-Object-Oriented-OO-Design-Patterns-to-LabVIEW%2Fta-p%2F3510571%3Fprofile.language%3Den)

|
[Login \(https://lumen.ni.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://forums.ni.com/auth/saml&cifaction=login&isoLang=en&target=https%3A%2F%2Forums.ni.com%2Ft5%2FExample-Program-Drafts%2FApplying-Common-Object-Oriented-OO-Design-Patterns-to-LabVIEW%2Fta-p%2F3510571%3Fprofile.language%3Den\)](https://lumen.ni.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://forums.ni.com/auth/saml&cifaction=login&isoLang=en&target=https%3A%2F%2Forums.ni.com%2Ft5%2FExample-Program-Drafts%2FApplying-Common-Object-Oriented-OO-Design-Patterns-to-LabVIEW%2Fta-p%2F3510571%3Fprofile.language%3Den)

| Help (/t5/help/faqpage)

Documents

▼ Search the community

Document Options

Applying Common Object-Oriented (OO) Design Patterns to LabVIEW

by  Elijah_K (/t5/user/viewprofilepage/user-id/71463) on 12-07-2008 08:30 PM - edited on 01-30-2017 08:57 AM by

 Elijah_K (https://forums.ni.com/t5/user/viewprofilepage/user-id/71463) 

When talking about computer programming, a design pattern is a standard correct way to organize your code. When trying to achieve some particular result, you look to the standard design patterns first to see if a solution already exists. This sounds a lot like an algorithm. An algorithm is a specific sequence of steps to take to calculate some result from a set of data. Generally algorithms can be written once in any given programming language and then reused over and over again. Design patterns are rewritten over and over again. For example, in house cleaning, consider this algorithm for vacuuming a carpet: "Start in one corner, walk horizontally across the floor, then move to the side and continue making adjacent parallel stripes until the whole floor is vacuumed." Compare this with the design pattern for cleaning a room: "Start cleaning at the ceiling and generally work your way down so that dirt and dust from above settle on the still unclean areas below." Design patterns are less specific than algorithms. You use the patterns as a starting point when writing the specific algorithm.

Every language develops its own design patterns. LabVIEW has patterns such as "Consumer/Producer Loops" or "Queued State Machine." These are not particular VIs. Many VIs are implementations of queued state machines. When a LabVIEW programmer describes a problem, another programmer may answer back, "Oh. You need a queued state machine to solve that problem." Starting in LabVIEW 7.0, LabVIEW has had VI templates for many LabVIEW design patterns available through the File>>New... dialog. With the release of LabVIEW Object-Oriented Programming, we need to find the new patterns that arise when objects mix with dataflow.

The seminal text on design patterns appeared in 1995: Design Patterns by Gamma, Helm, Johnson and Vlissides. This text is colloquially known as "the Gang of Four book." This text focused on object-oriented design patterns, specifically those that could be implemented using C++. Many of those patterns are predicated on a by-reference model for objects. As such they apply very well to classes made with the GOOP Toolkit (or any of the several other GOOP implementations for LabVIEW). LabVIEW classes, added in LV8.2, use by-value syntax in order to be dataflow safe. This means that the known patterns of object-oriented programming must be adapted, and new patterns must be identified. This document describes the LabVOOP-specific patterns I have identified thus far.

Stephen Mercer,

LabVIEW R&D

Design Patterns

- [Template Method \(aka Channeling\) Pattern \(/docs/DOC-13725\)](#) - To provide a guaranteed pre-processing/post-processing around some dynamic central functionality. Allows child classes ability to override some steps of an algorithm without giving them the ability to change out the entire algorithm.
- [Factory Pattern \(/docs/DOC-13724\)](#) - Provide a way to initialize the value on a parent wire with data from many different child classes based on some input value, such as a selector ring value, enum, or string input.
- [Hierarchy Composition Pattern \(/docs/DOC-13723\)](#) - To represent a single object as a tree of smaller instances of that same object type.
- [Delegation Pattern \(/docs/DOC-13722\)](#) - To have two independent classes share common functionality without putting that functionality into a common parent class.
- [Visitor Pattern \(/docs/DOC-13711\)](#) - To write a traversal algorithm of a set of data such that the traversal can be reused for many different operations.
- [Aggregation Pattern \(/docs/DOC-13710\)](#) - To treat an array of objects as a single object and define special behavior for that particular type of array.

This site uses cookies to offer you a better browsing experience. Learn more about our privacy policy. (//www.ni.com/legal/privacy/unitedstates/us/)

- [Specification Pattern \(/docs/DOC-13709\)](#) - To have different functionality for a class depending upon some value of the class without updating case structures throughout your VI hierarchy when you add another value.
- [Singleton Pattern \(/docs/DOC-13462\)](#) - Guarantee that a given class has only a single instance in memory

OK

- [Decorator \(Wrapper\) Pattern](#) (/docs/DOC-13332) - Extend a concrete object's responsibilities dynamically. Extend the features of an existing class without refactoring tested code.
- [Strategy \(Policy\) Pattern](#) (/docs/DOC-13311) - Change the behavior of a class's method without having to edit the method's implementation.

Application Examples

- [State Pattern, Factory Pattern](#) (/docs/DOC-15014) - this is an example of a UI Plugin Framework that illustrates the use of both patterns
- [Hardware Abstraction Layer](#) (<http://zone.ni.com/devzone/cda/epd/p/id/6307>) - Illustrates the use of classes and a factory pattern

Those are the design patterns thus far. I hope to update the list as time goes by, both with more patterns and with better examples for the existing patterns. Pay attention to the code you write – when you find yourself following the same routine over and over, try giving a name to the pattern. See if you can clearly identify when it is useful and how best to do it.

Exploration of these patterns helps us design better code. Using the patterns is using the learning of previous developers who have already found good solutions. In studying the patterns, we can learn what good code “feels” like, and we are then more aware when we create an unmaintainable hack in our own programs.

When we follow the standard designs it helps others looking at our code understand what they're looking at. If a programmer knows the general pattern being used, he or she can ignore the framework and focus on the non-standard pieces. Those points are where the program is doing its real work. It is akin to looking at a painting in a museum – the painting is the interesting part. The picture frame and the wall are just there to hold up the art. If either the frame or wall is too decorative, it pulls the viewer's focus.

I do not talk about any anti-patterns in this document. Anti-patterns are common attempts that look like good ideas but have some generally subtle problem that will arise late in development. Some anti-patterns in other languages are sufficiently detectable that a compiler could even be taught to recognize them and report an error. Perhaps these exist in LabVIEW. We should be cataloguing these as well.

*Let wire and node
gently flow into a world
of class and method.
– LabVOOP*

Reference:

Design Patterns. Gamma, Erich, et al. 1995.
Addison Wesley Longman, Inc.

Elijah Kerry
Chief Product Manager, Software Platform

Follow my [Software Engineering for LabVIEW Blog](#) (<http://ekerry.wordpress.com/>)

architecture (/t5/tag/architecture/tg-p/board-id/3046) design (/t5/tag/design/tg-p/board-id/3046) design_patterns (/t5/tag/design_patterns/tg-p/board-id/3046)
framework (/t5/tag/framework/tg-p/board-id/3046) LabVIEW (/t5/tag/LabVIEW/tg-p/board-id/3046) LabVOOP (/t5/tag/LabVOOP/tg-p/board-id/3046)
LVOOP (/t5/tag/LVOOP/tg-p/board-id/3046) Object (/t5/tag/Object/tg-p/board-id/3046) OOP (/t5/tag/OOP/tg-p/board-id/3046)
oriented (/t5/tag/oriented/tg-p/board-id/3046) Patterns (/t5/tag/Patterns/tg-p/board-id/3046) tutorial (/t5/tag/tutorial/tg-p/board-id/3046)

View All (12) (/t5forums/tagdetailpage/tag-cloud-grouping/tag/tag-cloud-style/frequent/message-scope/single/message-uid/3510571/user-scope/all/tag-scope/all/timerange/all/tag-visibility-scope/all)

★ ([HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510571?T:AC=BLOG-ID/3046/ARTICLE](https://forums.ni.com/t5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510571?T:AC=BLOG-ID/3046/ARTICLE)

(/t5/Example-Program-Drafts/Applying-Common-Object-Oriented-OO-Design-Patterns-to-LabVIEW/ta-p/3510571) (/t5/Example-Program-Drafts/A

2FTA-P%2F3510571&TITLE=APPLYING+COMMON+OBJECT-ORIENTED+%2800%29+DESIGN+PATTERNS+TO+LABVIEW&USERNAME=POWEREDBYLITHIUM)

COMMENTS

 (/t5/user/viewprofilepage/user-id/67927)  Daklu (/t5/user/viewprofilepage/user-id/67927)  ACTIVE PARTICIPANT 12-15-2008 05:44 PM Options▼

After reviewing the Rabbit Squad class, I now understand why NI password protects many of their block diagrams--self documenting code at its finest!



This site uses cookies to offer you a better browsing experience. Learn more about [our privacy policy](#). (<http://www.ni.com/legal/privacy/unitedstates/us/>)

★ ([HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510572?T:AC=BLOG-ID/3046/ARTICLE](https://forums.ni.com/t5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510572?T:AC=BLOG-ID/3046/ARTICLE)



(/t5/user/viewprofilepage/user-id/271138) dostini (/t5/user/viewprofilepage/user-id/271138) MEMBER 07-15-2012 07:34 PM

Options▼

What happened to the Observer Pattern? I'm sure it used to be listed here....

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510573?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/20831) AristosQueue (/t5/user/viewprofilepage/user-id/20831) PROVEN ZEALOT 08-15-2012 12:58 PM

Options▼

I removed it. The implementation given was one that I would never recommend anyone use. Someday I'd like to get back to that project, working with the engineer who created it, but it fell off my radar.

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510574?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



David_Staab (/t5/user/viewprofilepage/user-id/81768) ACTIVE PARTICIPANT 09-10-2012 02:41 PM

Options▼

Why was the link to Observer removed in rev 24?

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510575?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/71463) Elijah_K (/t5/user/viewprofilepage/user-id/71463) ACTIVE PARTICIPANT 09-10-2012 09:19 PM

Options▼

See previous comment



Elijah Kerry
Chief Product Manager, Software Platform

Follow my [Software Engineering for LabVIEW Blog](http://ekerry.wordpress.com/) (<http://ekerry.wordpress.com/>)

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510576?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/199098) dpnsw (/t5/user/viewprofilepage/user-id/199098) MEMBER 09-20-2012 10:11 PM

Options▼

Great central point for aggregating the various design patterns.

Just noticed this page elsewhere on the forums.

<https://decibel.ni.com/content/docs/DOC-21531> (/docs/DOC-21531) for a take on the MVC (Model View Controller) Pattern

This site uses cookies to offer you a better browsing experience. Learn more about our [privacy policy](#). ([//www.ni.com/legal/privacy/unitedstates/us/](http://www.ni.com/legal/privacy/unitedstates/us/))

OK

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510577?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/181528)



Jpina (/t5/user/viewprofilepage/user-id/181528)



MEMBER 11-10-2014 05:30 AM

Options ▾

AristosQueue wrote:

I removed it. The implementation given was one that I would never recommend anyone use. Someday I'd like to get back to that project, working with the engineer who created it, but it fell off my radar.

Hi Aristos,

when you say that you would not recommend the implementation, are you referring to this one?

<https://decibel.ni.com/content/docs/DOC-20182> (/docs/DOC-20182)

Could you tell us why would you not recommend that implementation or even the one that Michael suggested? I am working on a project where the observer pattern would really suit, but I'd like to know why it is not recommendable before I get into troubles.

Thank you

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510578?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/100152)



SebastienM (/t5/user/viewprofilepage/user-id/100152)

MEMBER 03-21-2015 01:02 PM

Options ▾

Hi all,

Maybe it's not the good place to talk about that ...

My concern is about object based framework.

LabVIEW give us since ... I don't remember when it appeared ... the Actor Framework, which is certainly powerfull, but just unreadable.

When I conduct Core3 training, I'm tempted to show what could be object based framework in LabVIEW, then I remember how the actor framework looks, then I decide to make my trainees continue to use LabVIEW, and finally I don't show the actor framework.

When will NI provide a SIMPLE object based framework in the templates ?

Something like the command patern exposed here by Elijah Kerr :

<https://ekerry.wordpress.com/2012/01/16/oriented-design-patterns-for-labview-that-can-make-your-life...>

[\(https://ekerry.wordpress.com/2012/01/16/oriented-design-patterns-for-labview-that-can-make-your-life-easier/\)](https://ekerry.wordpress.com/2012/01/16/oriented-design-patterns-for-labview-that-can-make-your-life-easier/)

or in a the test executive example of the PTP sequencer :

<http://www.ptpartners.co.uk/ptp-sequencer/> (<http://www.ptpartners.co.uk/ptp-sequencer/>)

I personnaly use the "old" QSM (please Daklu, don't bite me), for sure it is not an ideal solution...

I love objects, I used them in my code, but not in for my high level code.

Any links ?

Thanks,

SebastienM

Sébastien
CLA, CTA

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510579?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/20831)



AristosQueue (/t5/user/viewprofilepage/user-id/20831)



PROVEN ZEALOT 03-23-2015 10:21 AM

Options ▾

This site uses cookies to offer you a better browsing experience. Learn more about our privacy policy. ([//www.ni.com/legal/privacy/unitedstates/us/](http://www.ni.com/legal/privacy/unitedstates/us/))

> When will NI provide a SIMPLE object based framework in the templates ?

Your question is confusing to me. I'm going to try to lay out the reason for my confusion... perhaps you can then post a response to clarify your request.

OK

To me, it sounds the same as asking, "When will NI provide a simple polyVI-based framework in the templates?" You don't have a template to demonstrate the language features. You have a template to solve a particular kind of problem. We might have a polyVI in a template if it were appropriate, but we wouldn't have a template for using a polyVI.

We have examples in the Example Finder for OO. Examples are where we demonstrate features. Check out the board inspection LVOOP example.

Do you just want a template that has a project with nothing but an empty class in it? That was something I thought should be in the templates list but got overruled.

But beyond that, I don't know what you're asking for. Frameworks are meant to solve problems, generally large problems. You don't build a framework for small stuff. The AF tackles a particularly hard problem in LabVIEW (or any programming language). If you are teaching in Core 3 how to build applications that include multiple parallel and intercommunicating processes, you should be teaching the AF or one of the other user-developed-but-freely-available frameworks, as they are all far simpler than trying to code on your own. But last I checked, you're not doing that in Core 3, so you probably shouldn't be teaching AF.

But template projects for "OO" generally doesn't make sense to me. You don't template language features. You template solutions to problems.

As for the command pattern that you mentioned from Eli -- that's *exactly* what the Actor Framework is an instance of. No more, no less. The command pattern as a pattern is not something you can template -- patterns are things that recur in software and you build from scratch each time and they follow a particular form of development but the details are totally different each time. If they could be formalized, they would become a framework or library. The trick is that to make a generic command pattern into a standalone thing, you need the environment that gets the commanded thing running and the controls for stopping it.

So, instead of asking for an OO-based framework in the templates, tell me a problem that you think has a generic solution worth templating, and then we can discuss using classes to solve that problem. That would be a template that uses OO. But we need a simple problem first.

Do you see the difference? Do you see why even asking the question is confusing to me? Have I missed the point of what you were hoping to see?

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510580?T:AC=BLOG-ID/3046/ARTICLE-ID/438



(/t5/user/viewprofilepage/user-id/71463)



Elijah_K (/t5/user/viewprofilepage/user-id/71463) ACTIVE PARTICIPANT 03-23-2015 10:26 AM Options▼

I agree with and echo Aristos' comments.

Elijah Kerry

Chief Product Manager, Software Platform

Follow my [Software Engineering for LabVIEW Blog](http://ekerry.wordpress.com/) (<http://ekerry.wordpress.com/>)

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510581?T:AC=BLOG-ID/3046/ARTICLE-ID/438



(/t5/user/viewprofilepage/user-id/67445)



BeCeGa (/t5/user/viewprofilepage/user-id/67445) TRUSTED ENTHUSIAST 03-23-2015 10:56 AM Options▼

Hi SebastienM

as Aristos mentiones, it makes not that much sense of just having a object framework in the template without a problem to solve just to show the features. But you can find some libraries like the Configurator Editor Framework (<https://decibel.ni.com/content/docs/DOC-37225> (/docs/DOC-37225)) or the Tag Bus Framework (<https://decibel.ni.com/content/docs/DOC-41721> (/docs/DOC-41721).) which are object based frameworks, for solving specific problems. When you install the VIPMs they will add templates of this implementations to LabVIEW.

Best Regards

Benjamin C

Principal Systems Engineer // CLA // CLED

This site uses cookies to offer you a better browsing experience. Learn more about our privacy policy. (<https://www.ni.com/legal/privacy/unitedstates/us/>)

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510582?T:AC=BLOG-ID/3046/ARTICLE-ID/438

OK



(/t5/user/viewprofilepage/user-id/100152) SebastienM (/t5/user/viewprofilepage/user-id/100152) MEMBER 03-23-2015 12:19 PM

Options▼

Hi,

First, thank you all for your time, I didn't expect that LabVIEW stars like you would answer to that

Second : unfortunately I did a cross post so the conversation was split in half :

<http://forums.ni.com/t5/LabVIEW/SIMPLE-Object-Based-Framework/td-p/3107317> (<http://forums.ni.com/t5/LabVIEW/SIMPLE-Object-Based-Framework/td-p/3107317>)

Sébastien
CLA, CTA

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510583?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



(/t5/user/viewprofilepage/user-id/100152) SebastienM (/t5/user/viewprofilepage/user-id/100152) MEMBER 03-23-2015 12:44 PM

Options▼

Up, now it's here :

<http://forums.ni.com/t5/LabVIEW/SIMPLE-Object-Based-Framework/td-p/3107317> (<http://forums.ni.com/t5/LabVIEW/SIMPLE-Object-Based-Framework/td-p/3107317>)

Sébastien
CLA, CTA

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510584?T:AC=BLOG-ID/3046/ARTICLE-ID/438)



NOT APPLICABLE 01-11-2016 08:28 AM

Options▼

Sorry about my OOO msg... deleting it

★ (HTTPS://FORUMS.NI.COM/T5/TKB/V2_4/ARTICLEPAGE.KUDOSBUTTONV2.KUDOENTITY:KUDOENTITY/KUDOSABLE-GID/3510585?T:AC=BLOG-ID/3046/ARTICLE-ID/438)

CONTRIBUTORS



(/t5/user/viewprofilepage/user-id/71463) Elijah_K (/t5/user/viewprofilepage/user-id/71463)

Powered by



(<https://khoros.com/powerd-by-khoros>)

This site uses cookies to offer you a better browsing experience. Learn more about our privacy policy. (<http://www.ni.com/legal/privacy/unitedstates/us/>)

+ PRODUCT

OK

 SUPPORT

 COMPANY

[\(http://www.facebook.com/NationalInstruments\)](http://www.facebook.com/NationalInstruments)

[\(//twitter.com/niglobal\)](http://twitter.com/niglobal)

[\(https://www.linkedin.com/company/3433?trk=tyah\)](https://www.linkedin.com/company/3433?trk=tyah)

[\(//www.youtube.com/nationalinstruments\)](http://www.youtube.com/nationalinstruments)

MISSION

NI equips engineers and scientists with systems that accelerate productivity, innovation, and discovery.

South Africa ▾

or Call 011 805 8197

Legal (/www.ni.com/legal/) | Privacy (/www.ni.com/en-us/about-ni/legal/privacy/privacy-statement.html) © 2019 National Instruments. All rights reserved.

This site uses cookies to offer you a better browsing experience. Learn more about our privacy policy. ([//www.ni.com/legal/privacy/unitedstates/us/](http://www.ni.com/legal/privacy/unitedstates/us/))

OK