Introduction to
Basic LabVIEW Design Patterns

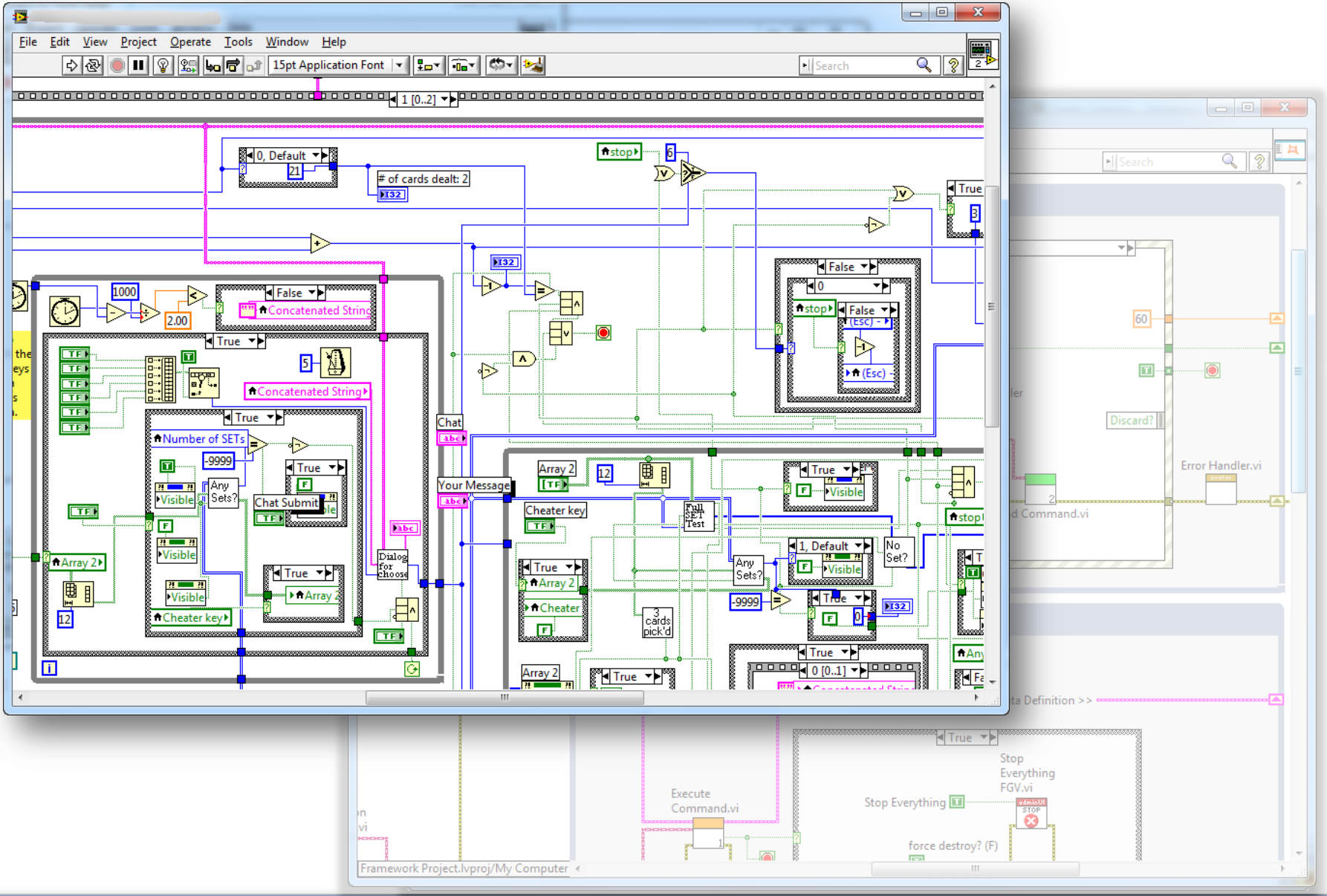
What is a Design Pattern?

Definition: A well-established solution to a common problem.

Why Should I Use One?

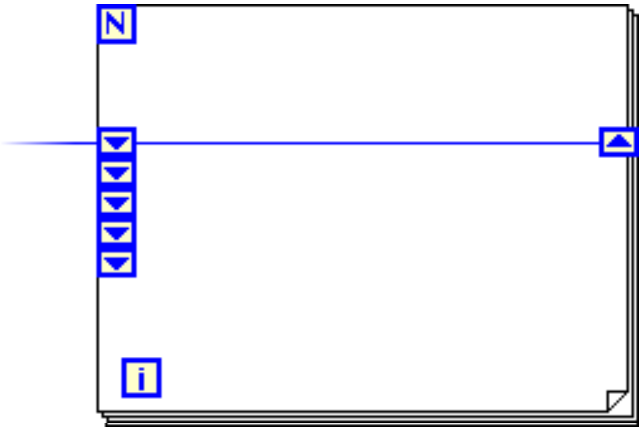
Save time and improve the longevity and readability of your code.

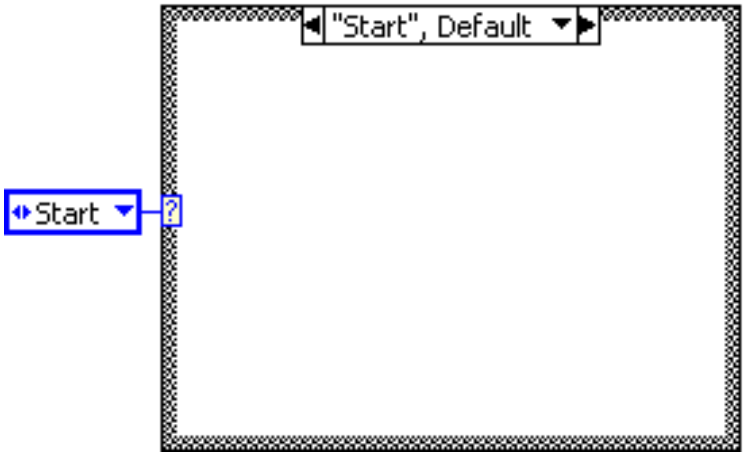
… or else…

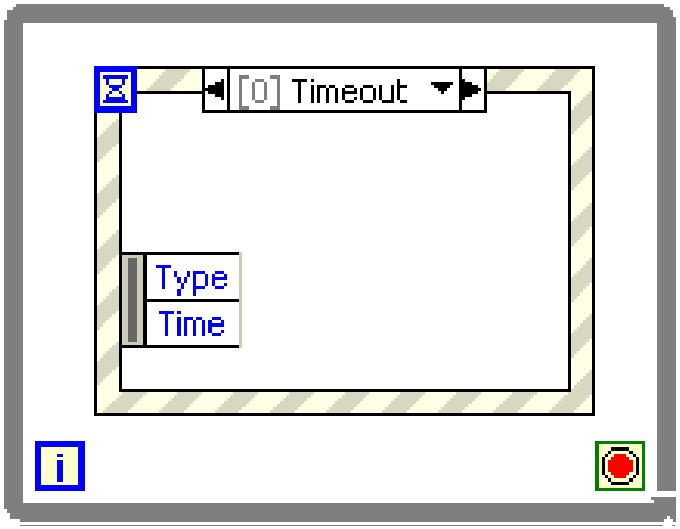

File Edit View Project Operate Tools Window Help
15pt Application Font
Search
# of cards dealt: 2
Concatenated String
Number of SETs
Any Sets?
Chat Submit
Chat
Your Message
Cheater key
Full SET Test
No Set?
3 cards pick'd
Dialog for choose
Framework Project.lvproj/My Computer
Execute Command.vi
Stop Everything
Stop Everything FGV.vi
force destroy? (F)
Discard?
Error Handler.vi

You Should Already Be Familiar With..

- Loops
- Shift Registers
- Case Structures
- Enumerated Constants







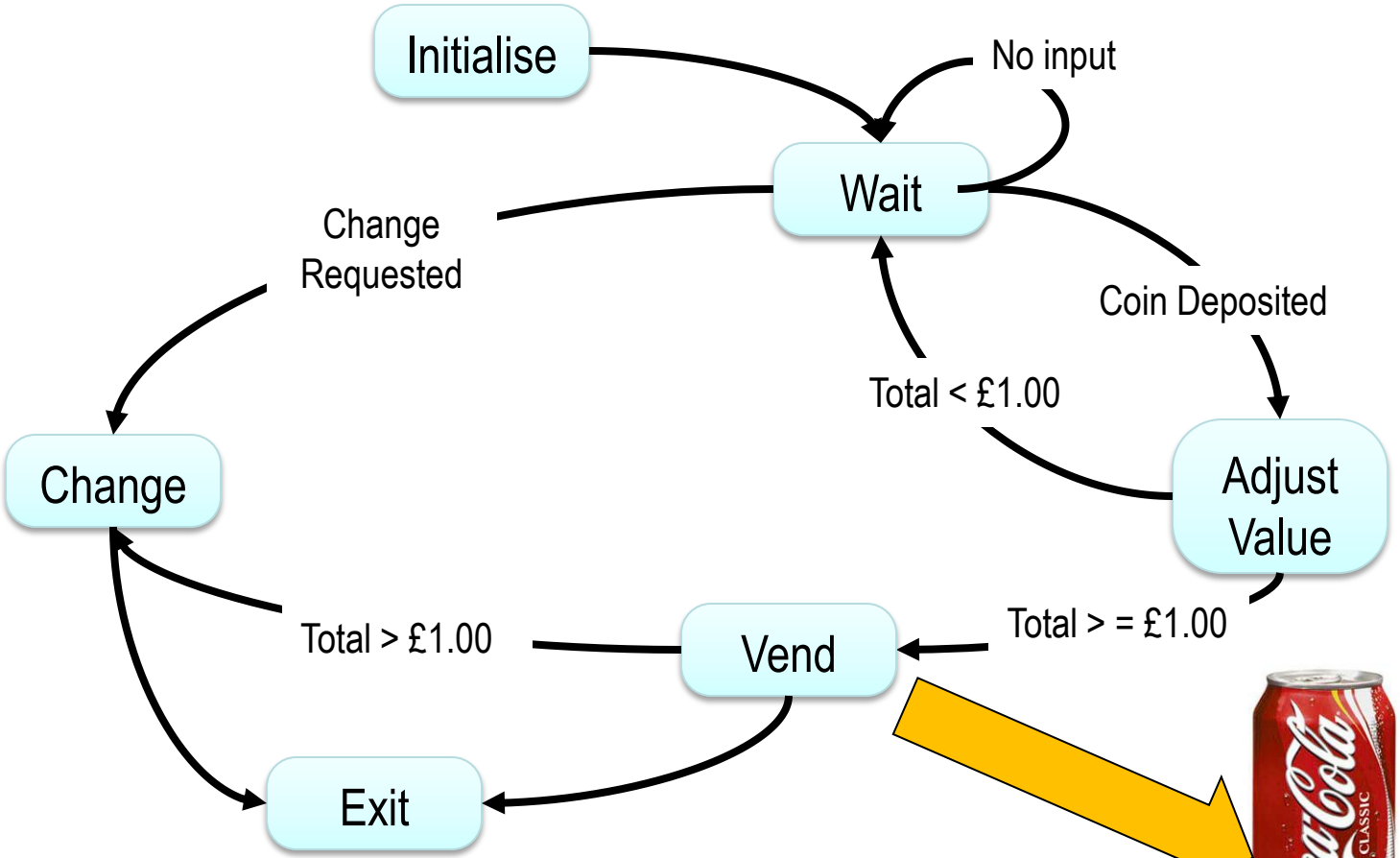
Design Patterns

- State Machine / Statecharts
- Producer / Consumer

State Machine

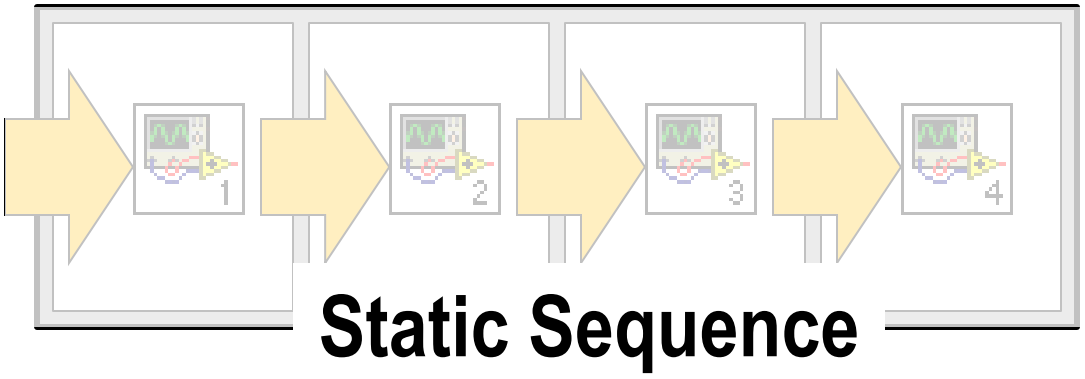
I need to execute a sequence of events, but the order is determined programmatically

Vending Machine

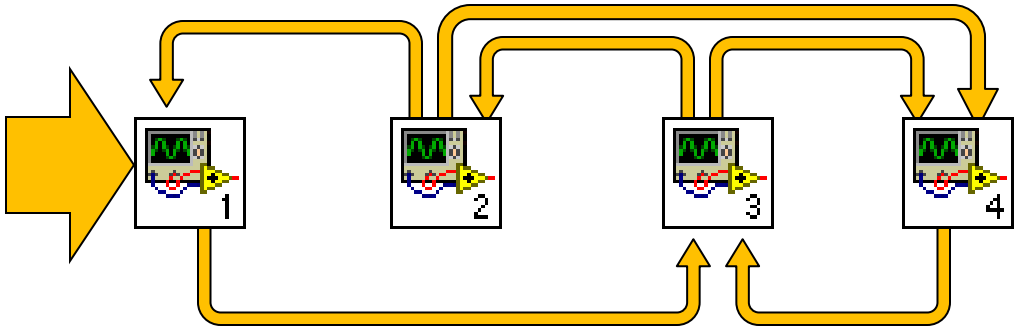


Drink costs £1.00

Background



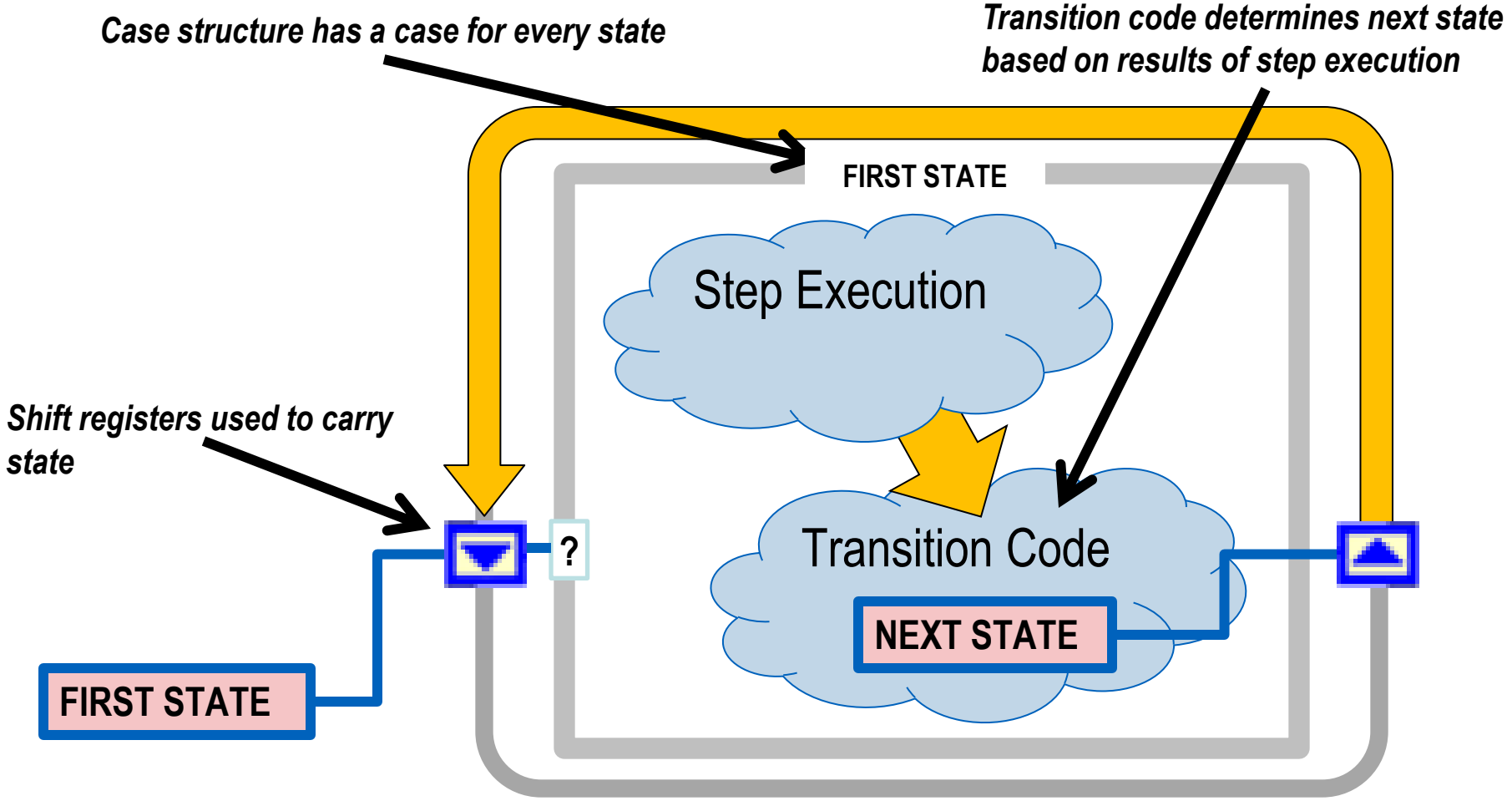
Dynamic Sequence: Allows distinct states to operate in a programmatically determined sequence



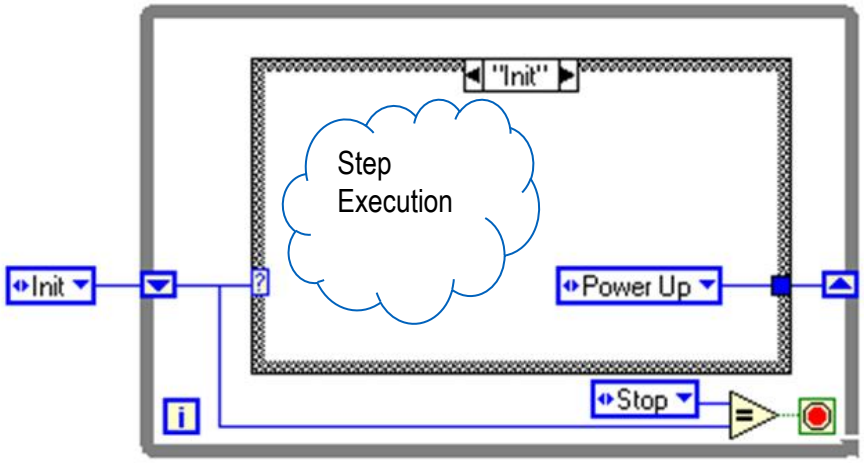
Breaking Down the Design Pattern

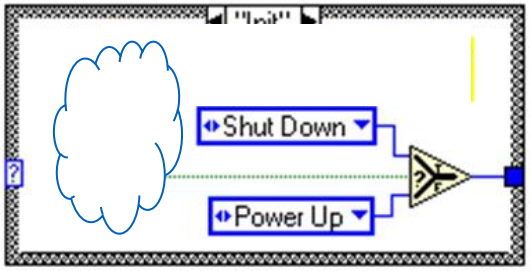
- Case Structure inside of a While Loop
- Each case is a state
- Current state has decision making code that determines next state
- Use enumerated constants to pass value of next state to shift registers

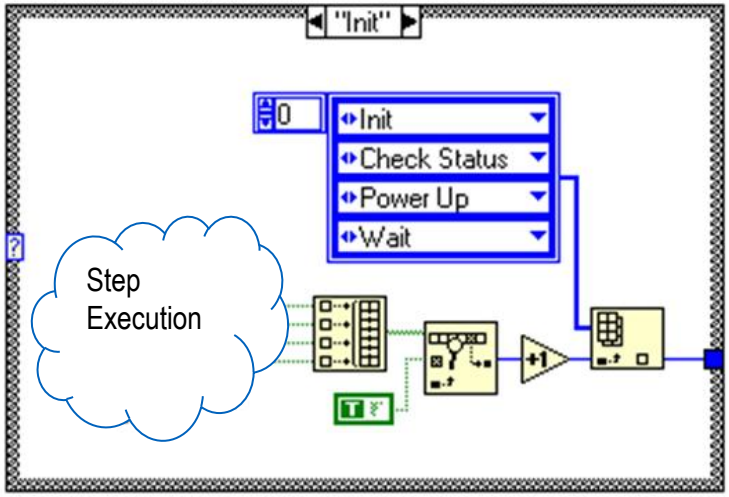
The Anatomy of a State Machine

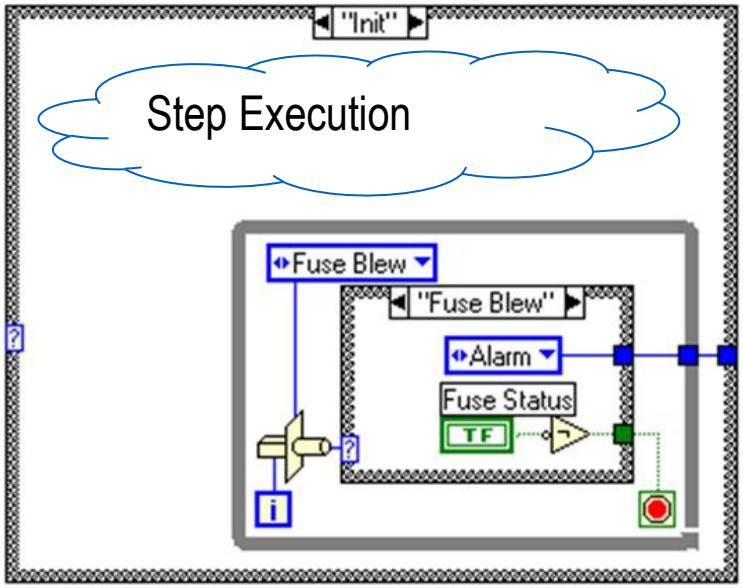


Transition Code Options









State Machine

DEMO

Recommendations

Use Cases

- User interfaces
- Data determines next routine

Considerations

- Creating an effective State Machine requires the designer to make a table of possible states.

Producer / Consumer

I have two processes that need to execute at the same time, and I need to make sure one can't slow the other down

Background

I want to execute code in parallel and at asynchronous rates, but I need to communicate between them!

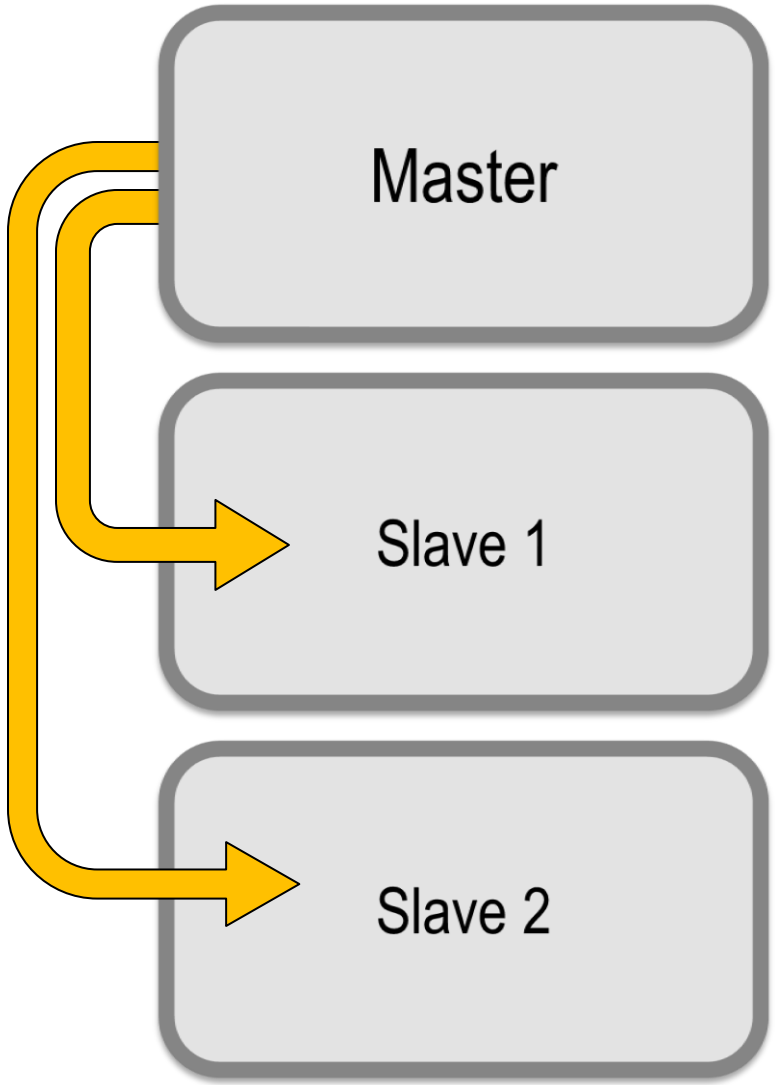
I have two processes that need to execute at the same time, but I want them to be independent of one another, and I need to make sure one can't slow the other down

Breaking Down the Design Pattern

- Data independent loops
- Master / slave relationship
- Communication and synchronisation between loops

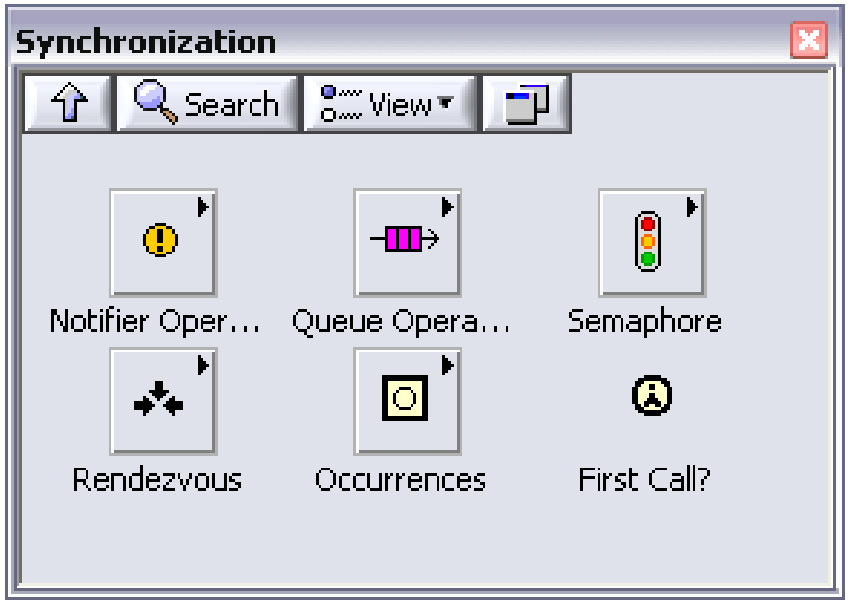
How It Works

- One or more slave loops are told by a master loop when they can run
- Allows for asynchronous execution of loops
- Data-independence breaks dataflow and allows multi-threading
- De-couples processes



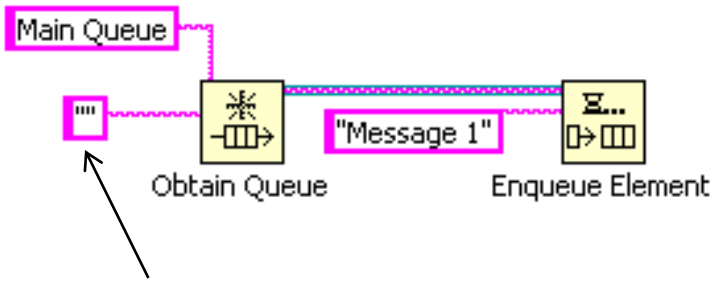
Master / Slave: Loop Communication

- Variables
- Occurrences
- Notifier
- Queues
- Semaphores
- Rendezvous



Queues

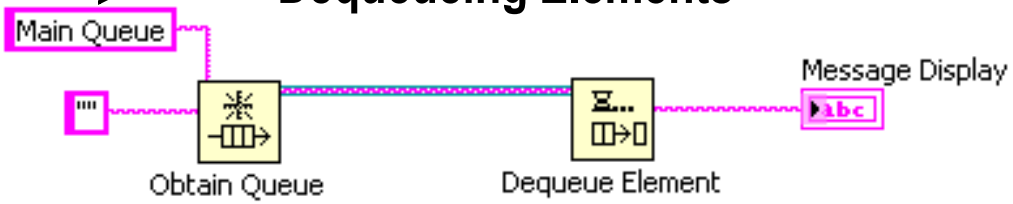
Adding Elements to the Queue



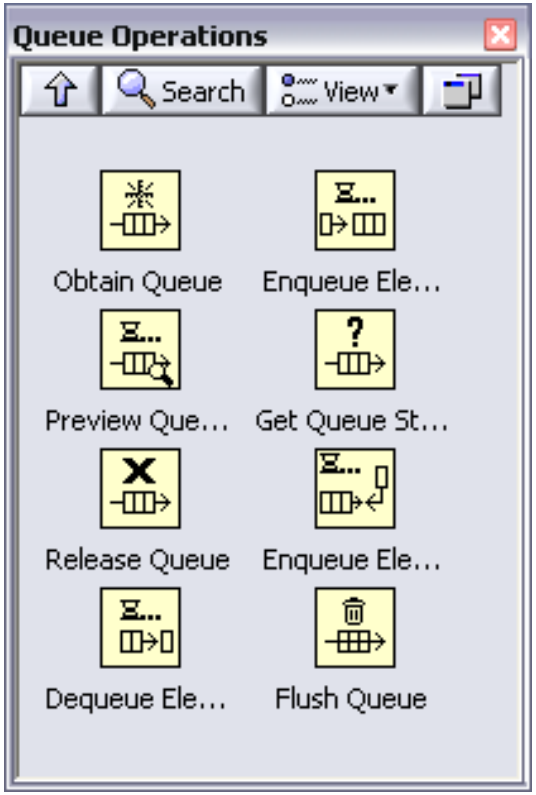
Select the data-type the queue will hold

Reference to existing queue in memory

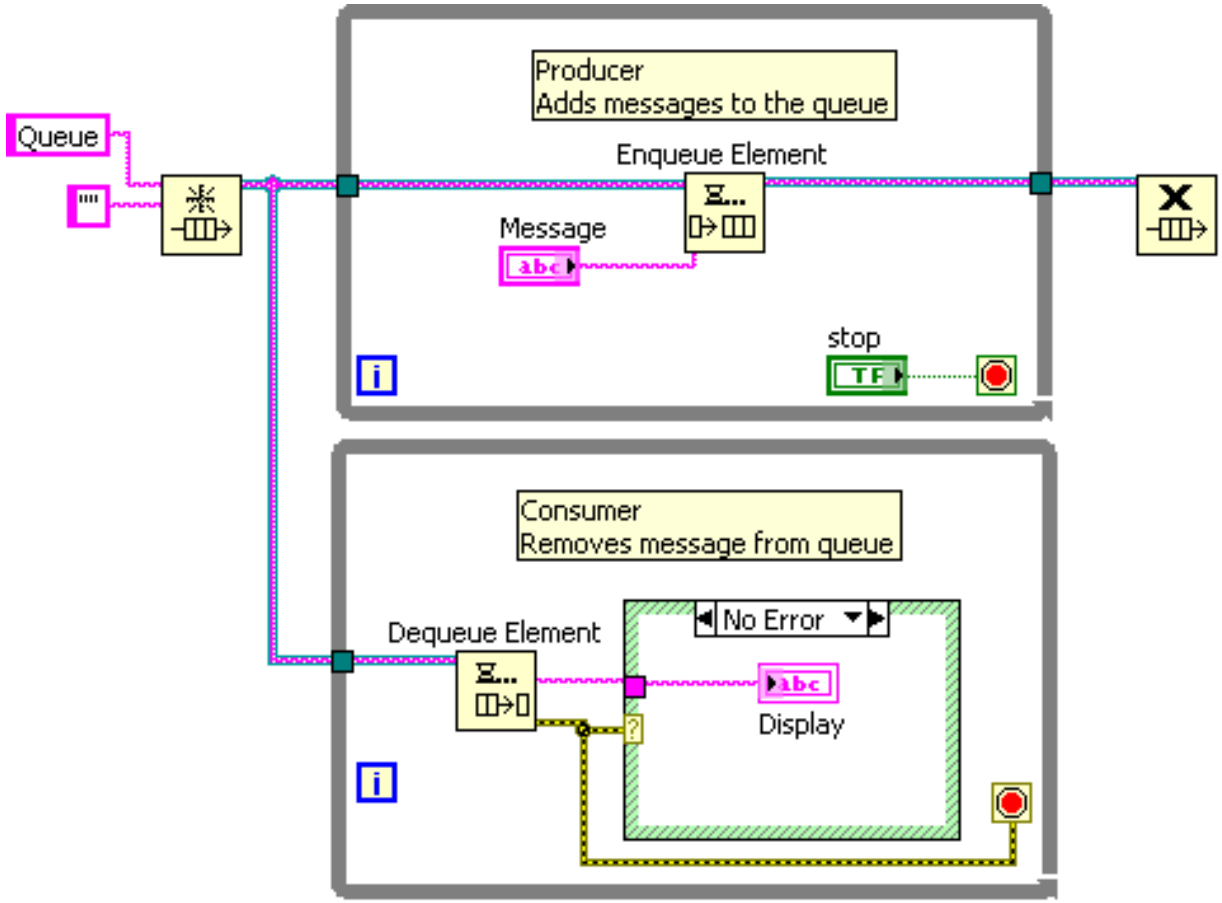
Dequeueing Elements



Dequeue will wait for data or timeout (defaults to -1)



Producer / Consumer



Producer / Consumer

DEMO

Recommendations

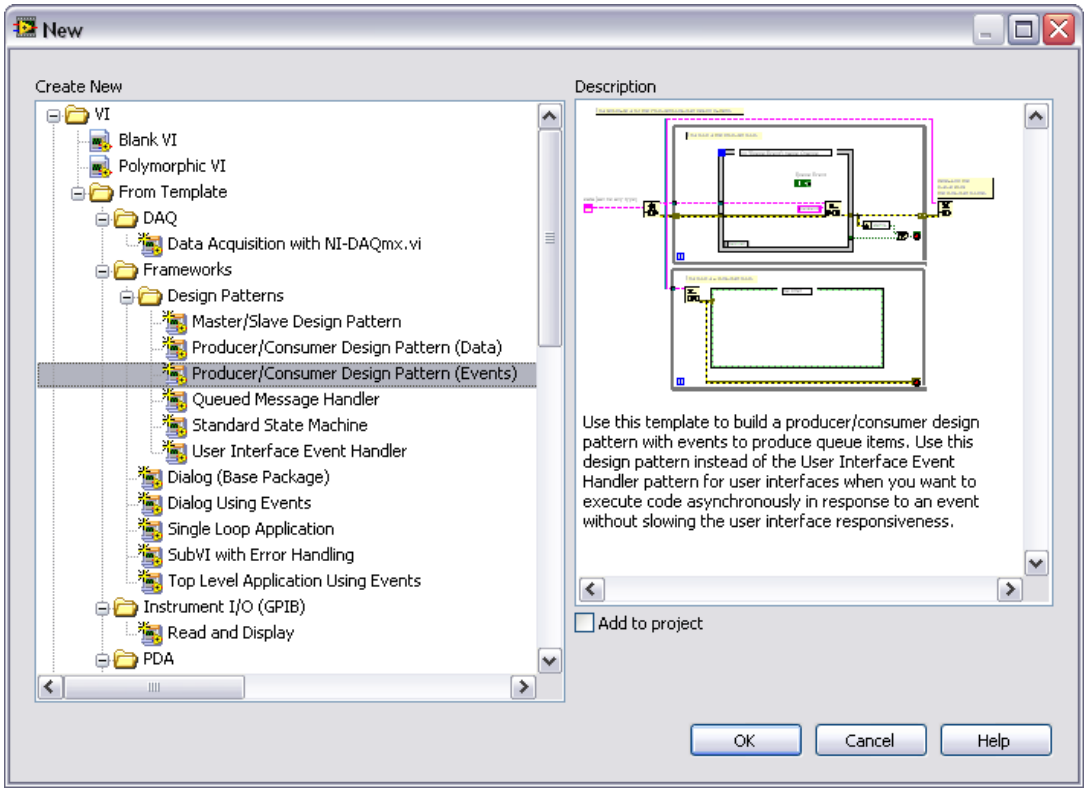
Use cases

- Handling multiple processes simultaneously
- Asynchronous operation of loops

Considerations

- Multiple producers → One consumer
- One queue per consumer
- If order of execution of parallel loop is critical, use occurrences

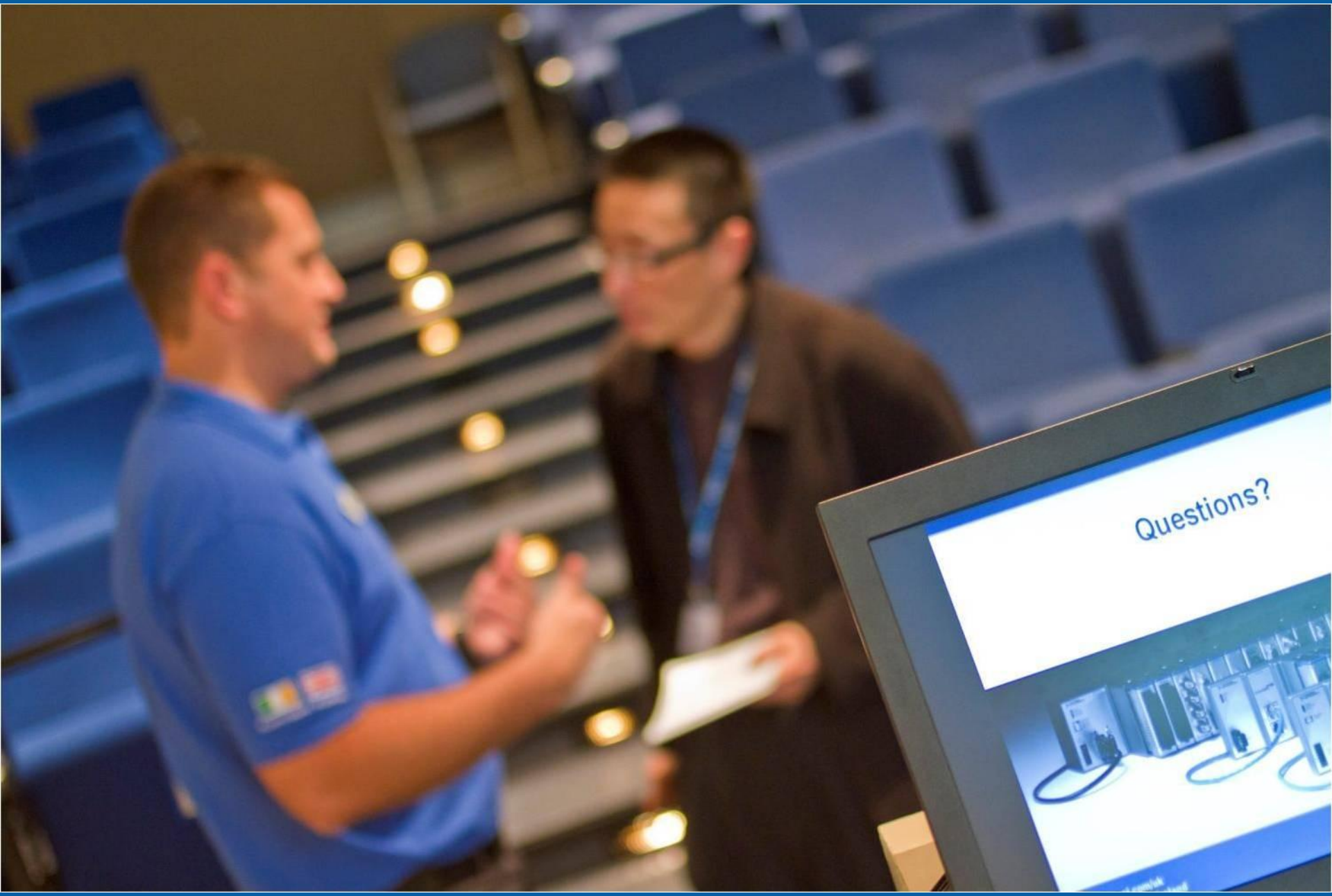
Adding Your Own Design Patterns



C:\Program Files\National Instruments\LabVIEW 2011\templates\Frameworks\DesignPatterns

How do I go further?

- Training
  - LabVIEW Core 3
- Reading
  - ni.com/largeapps
- Talk to your
  - Local Engineer
  - Local Alliance Member


Questions?

Coming up:

08:45	**Registration**	
09:15	**Changing the World with Graphical System Design**	
	Presentation Track	**Hands-On Track**
09:45	Basic LabVIEW Design Patterns	Hands-On: Introduction to LabVIEW & Data Acquisition with NI CompactDAQ
10:15	LabVIEW Performance Tips and Tricks	
10:45	**Break**	
11:00	5 RF Measurements Every Engineer Should Know	Hands-On: Introduction to LabVIEW Real-Time with CompactRIO
11.30	Implementing Hardware-in-the-Loop & Real-Time Testing Techniques with NI VeriStand	
12:00	**Lunch**	
12:45	LabVIEW User Presentation: Graphical System Design in Practice	
13:15	Changing Your Mindset for LabVIEW Real-Time & LabVIEW FPGA Programming	Hands-On: Create an Automated Test System with PXI Modular Instrumentation, LabVIEW & NI TestStand
14:00	Power Measurements and Smart Grid Technologies	
14:30	**Break**	
14:45	Web Services for Data Acquisition Applications	
15:15	Designing Vision Systems Including 3D Vision and the Microsoft Kinect	
16:00	**Wrap-up and Prize Draw**	
16:15	**Optional CLAD Examination and CLD Exam Preparation Class**	