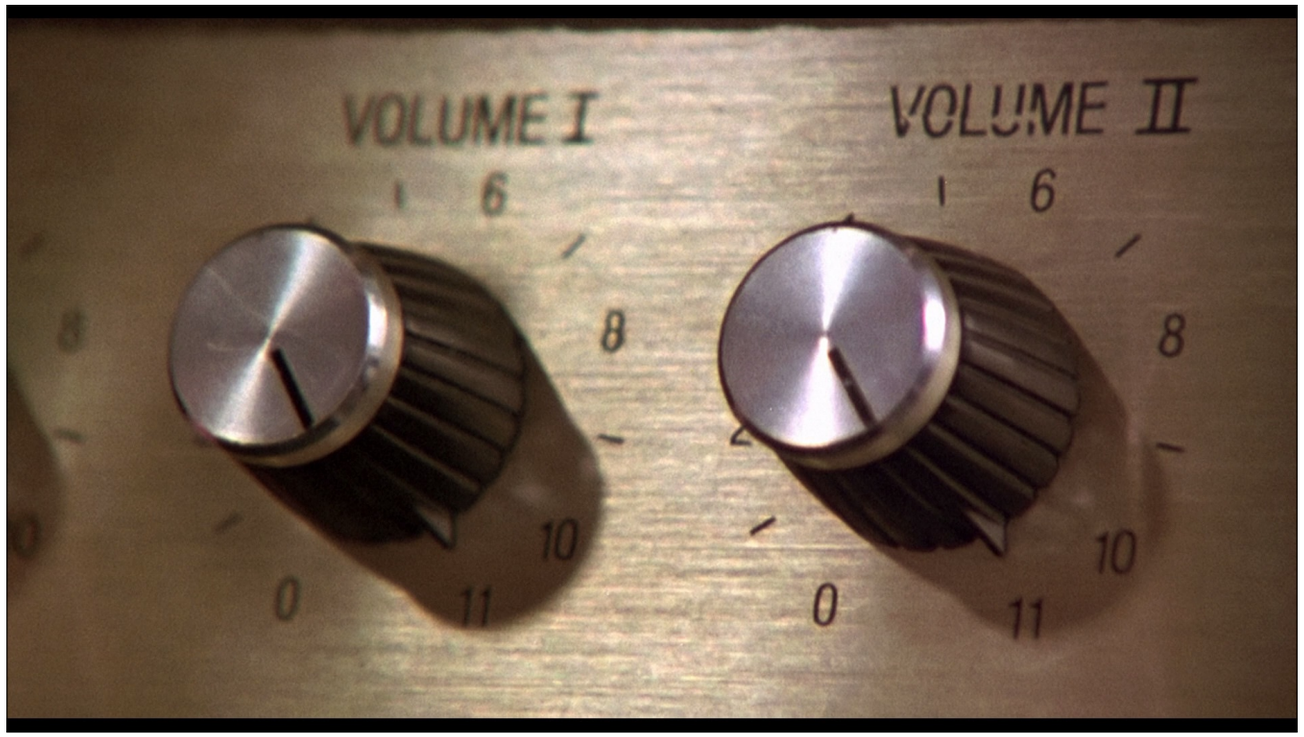
ReactiveUI

Turning MVVM up to 11

Brendan Forster
@shiftkey


VOLUME I
VOLUME II
1
6
8
10
0
11
2

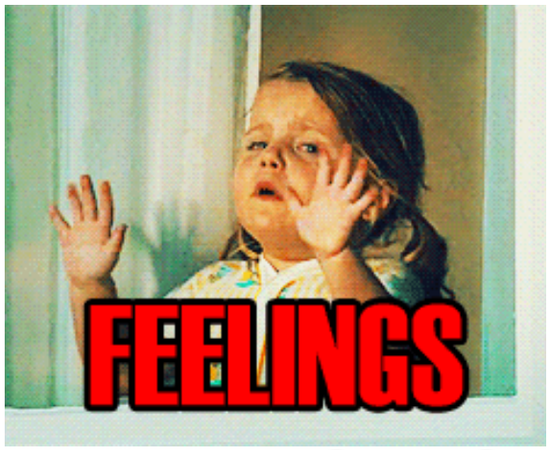
MVVM

MVC

MVP

MVYOLO

MVVM



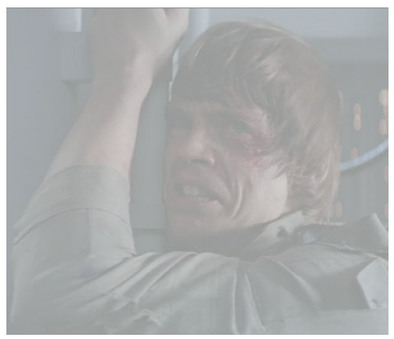
MVYOLO



Part 1: discover this lovely new pattern called MVVM and start out on a journey to save the world



Part 2: drop you in a situation where everything is terrible and you start to question everything around you

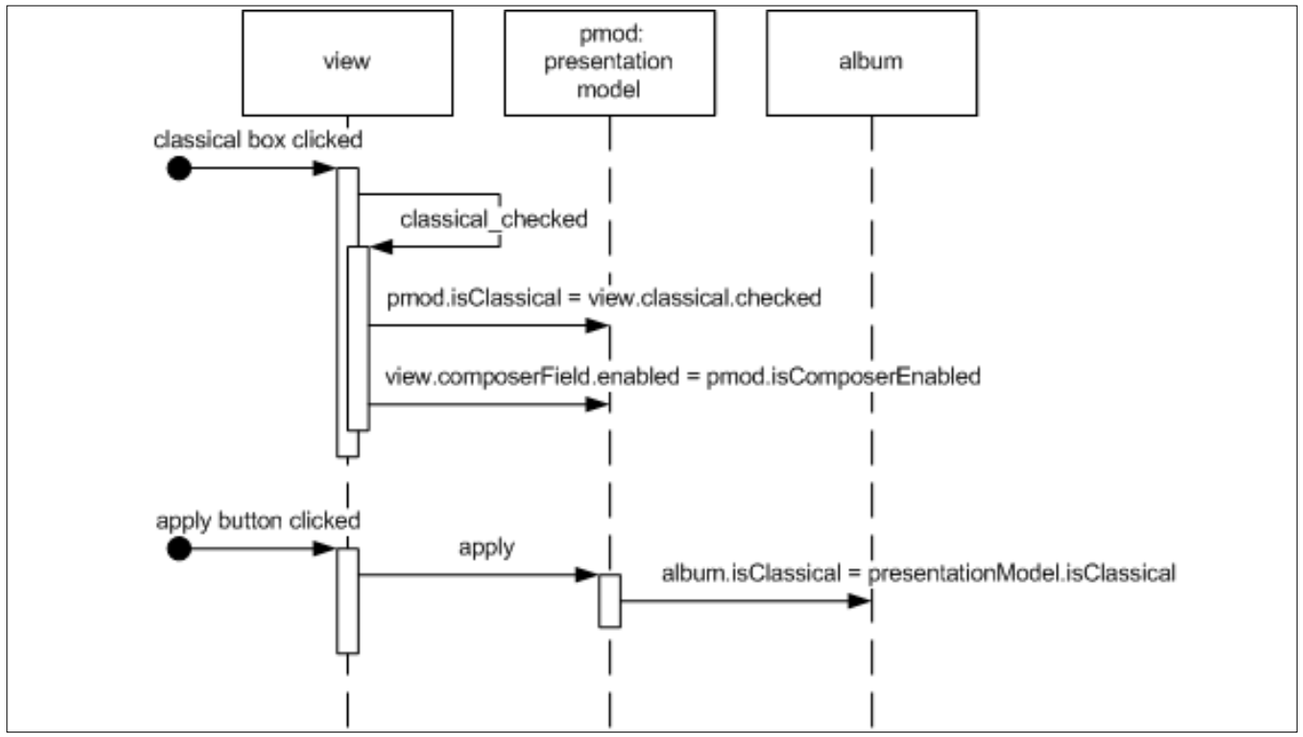

Part 3: reflect, meditate and return to the fight armed with a better tool for the job

Presentation Model - July 2004

http://martinfowler.com/eaaDev/PresentationModel.html

“as an abstract of the view that is not dependent on a specific GUI framework.”

"The essence of a **Presentation Model** is of a fully self-contained class that represents all the **data** and **behavior** of the UI window, but without any of the controls used to render that UI on the screen"



Hints of things to come

"Probably the most annoying part of Presentation Model is the **synchronization between Presentation Model and view**. It's simple code to write, but I always like to minimize this kind of boring repetitive code."

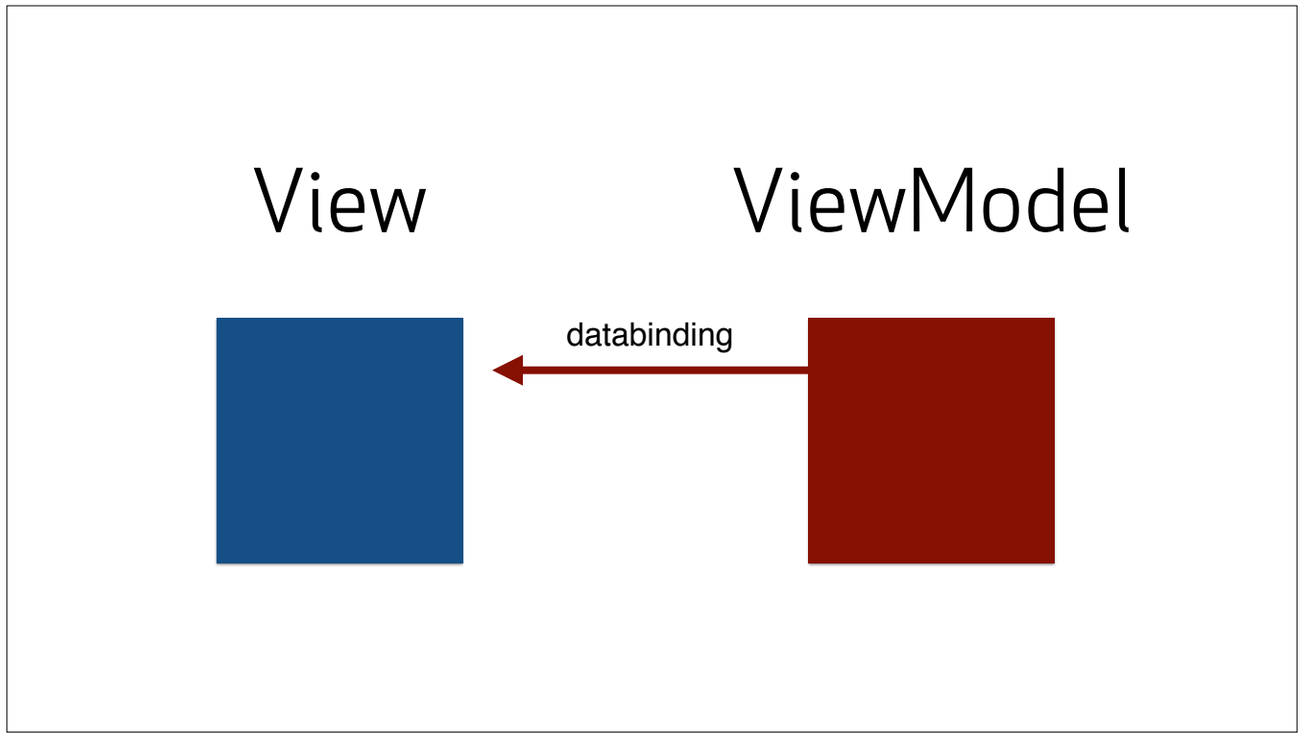
Model-View-ViewModel - September 2005

John Gossman was the first to use the name. The name and pattern as we know it was birthed in the WPF team

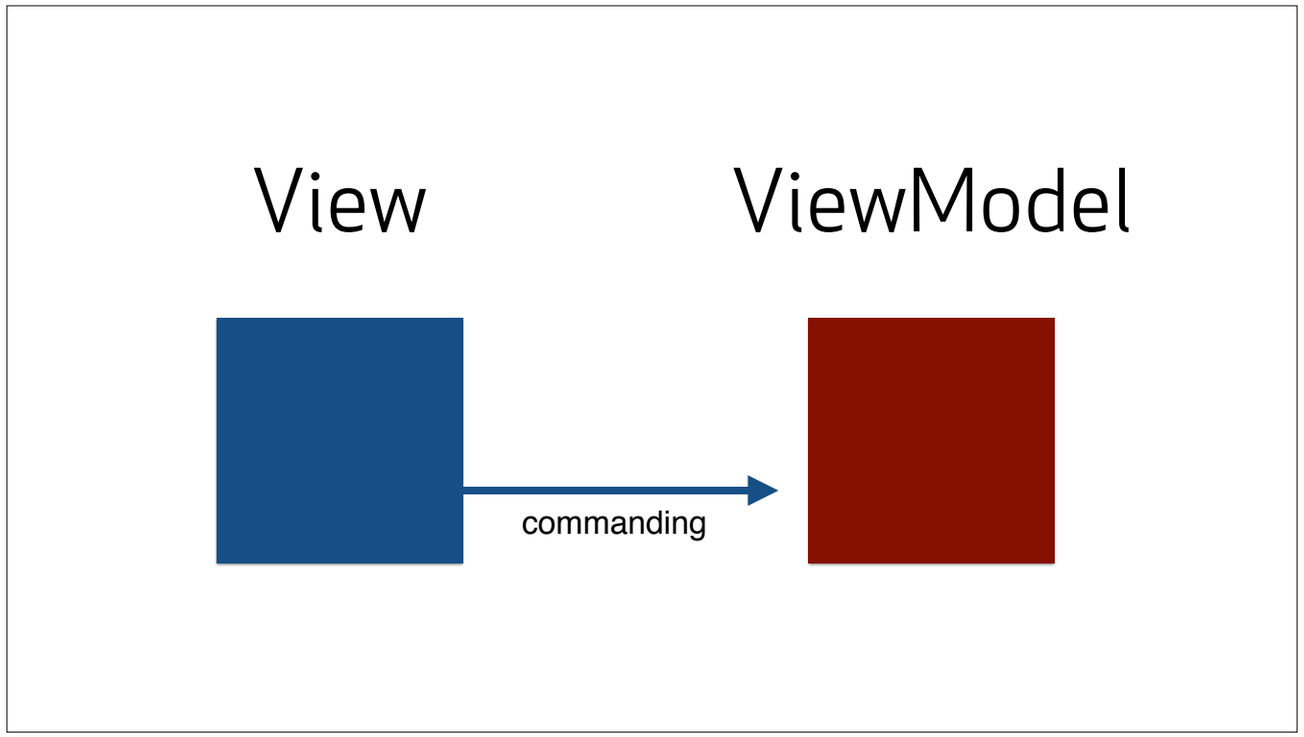
http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx

WPF 3.0 - November 2006

The innovative bit at the time was the two interactions which were baked into WPF.

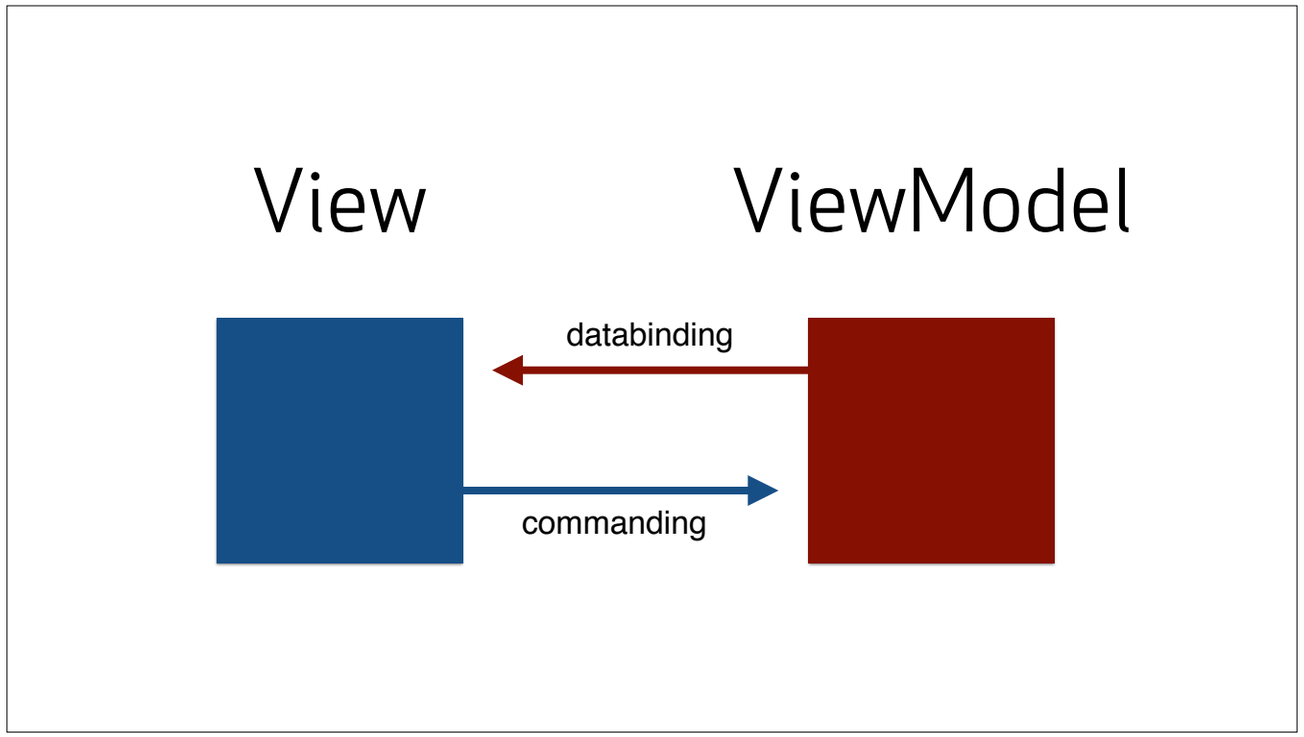

View
ViewModel
databinding

INotifyPropertyChanged
INotifyCollectionChanged


View
ViewModel
commanding

ICommand

```
bool CanExecute(object o)
void Execute(object o)
```


View
ViewModel
databinding
commanding

Separation of Concerns!

Testability!

Composition!

MvvmLight

Caliburn.Micro

MvvmCross

Catel

Cinch

KnockoutJS

DurandalJS

…

Lots of other frameworks out there

ReactiveUI

reactiveui.net

@paulcbetts - BDFL

54 contributors (PRs merged)

.NET 4.5

WP8

Windows Runtime

Monotouch/MonoMac

MonoAndroid

Universal Apps (v6)

ReactiveUI

Reactive Extensions

Functional Reactive Programming and Bacon

Mikael Brevik

Wednesday 16:20 - 17:20

Scaling Event Processing at all scales and all languages with the Reactive Extensions

Matthew Podwysocki

Wednesday 16:20 - 17:20

Reactive Extensions

in 10*ish* mins

```
foreach(var i in items) {
   // do something with i
}
```

```
public interface IEnumerable<T> {
  IEnumerator<T> GetEnumerator();
}
```

```
public interface IEnumerator<T> {
  bool MoveNext();
  T Current { get; }
}
```

What if the data isn't present at the time MoveNext is called?

What if MoveNext
throws an exception?

What about events?

```
public interface IObservable<T> {
    IDisposable Subscribe(
        IObserver<T> observer);
}
```

```
public interface IObserver<T> {
    void OnNext(T item);
    void OnException(Exception ex);
    void OnCompleted();
}
```

```
public interface IEnumerable<T> {
  IEnumerator<T> GetEnumerator();
}

public interface IObservable<T> {
    IDisposable Subscribe(
        IObserver<T> observer);
}
```

```
public interface IEnumerator<T> {
  bool MoveNext(); // throws
  T Current { get; }
}

public interface IObserver<T> {
    void OnNext(T item);
    void OnException(Exception ex);
    void OnCompleted();
}
```

Duality

http://en.wikipedia.org/wiki/Dual_(category_theory)
http://csl.stanford.edu/~christos/pldi2010.fit/meijer.duality.pdf

[demo]

cold observables:
inactive when no
observers subscribed

hot observables:

always active, even when no observers subscribed

schedulers:

control where work runs

TaskPoolScheduler

ThreadScheduler

DispatcherScheduler

...

[the app]

```
<TextBox Grid.Row="1"
         Text="{Binding SearchText, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}">
  <i:Interaction.Triggers>
     <i:EventTrigger EventName="KeyUp">
       <cal:ActionMessage MethodName="UpdateSearchResults" />
     </i:EventTrigger>
  </i:Interaction.Triggers>
</TextBox>
```

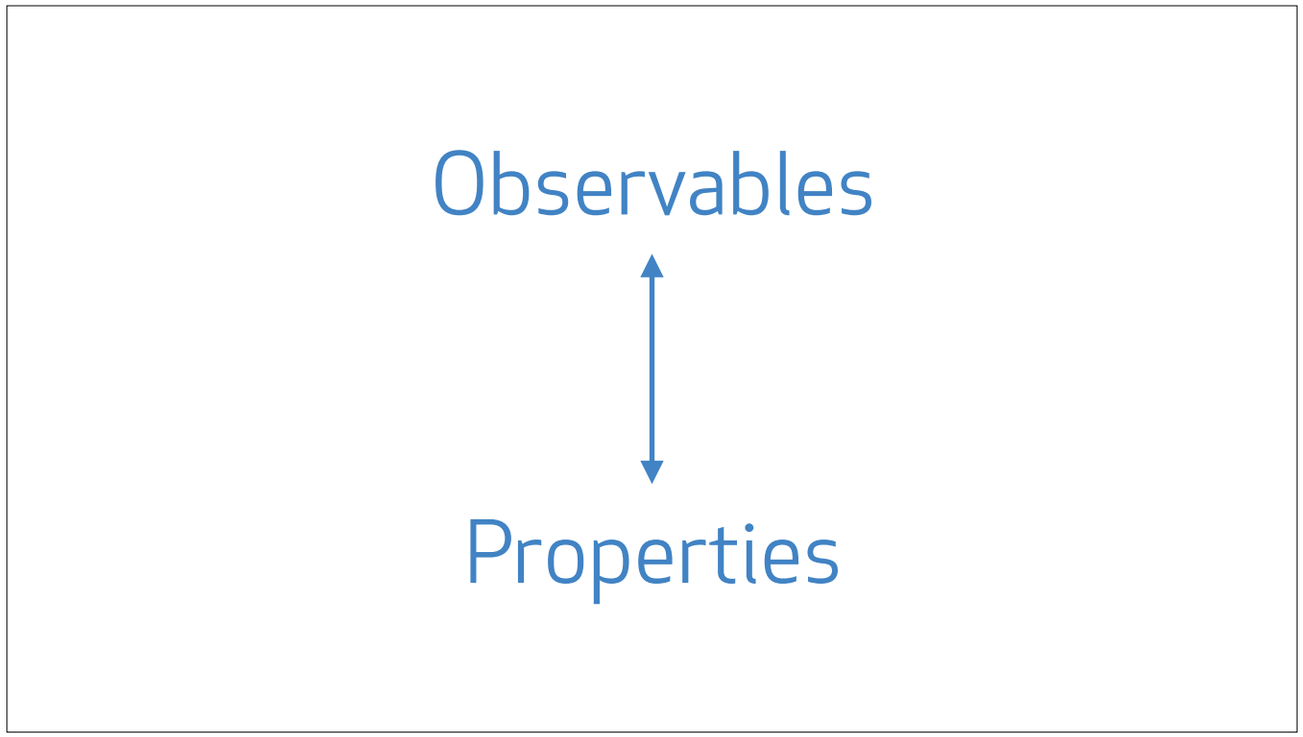
fix behaviour

[refactor]

ReactiveObject

this.WhenAny

eliminate ActionMessage


Observables
Properties

decompose complexity

[refactor]

ReactiveCommand

.Create()

.CreateAsyncTask()

ReactiveCommand

.ThrownExceptions

.IsExecuting

ObservableAsPropertyHelper

eliminate state

declarative ViewModels

simpler bindings

[refactor]

IViewFor<T>

.Bind()

.OneWayBind()

.BindTo()

type-safe bindings

compile-time validation

advanced selectors

code-behind is fine

if you don't need/want

to test the code

real

observable

collections

[refactor]

ReactiveList

.CountChanged

.ItemAdded

.ItemRemoved


4GIFs.com

Navigation

Mobile

Service Location

https://github.com/paulcbetts/splat

what have I learned from using RxUI?

limitedness of the interface

ReactiveCommand

[refactor]

declarative ViewModels

async everywhere

is amazing

tests and the concept of time

what is challenging
about RxUI?

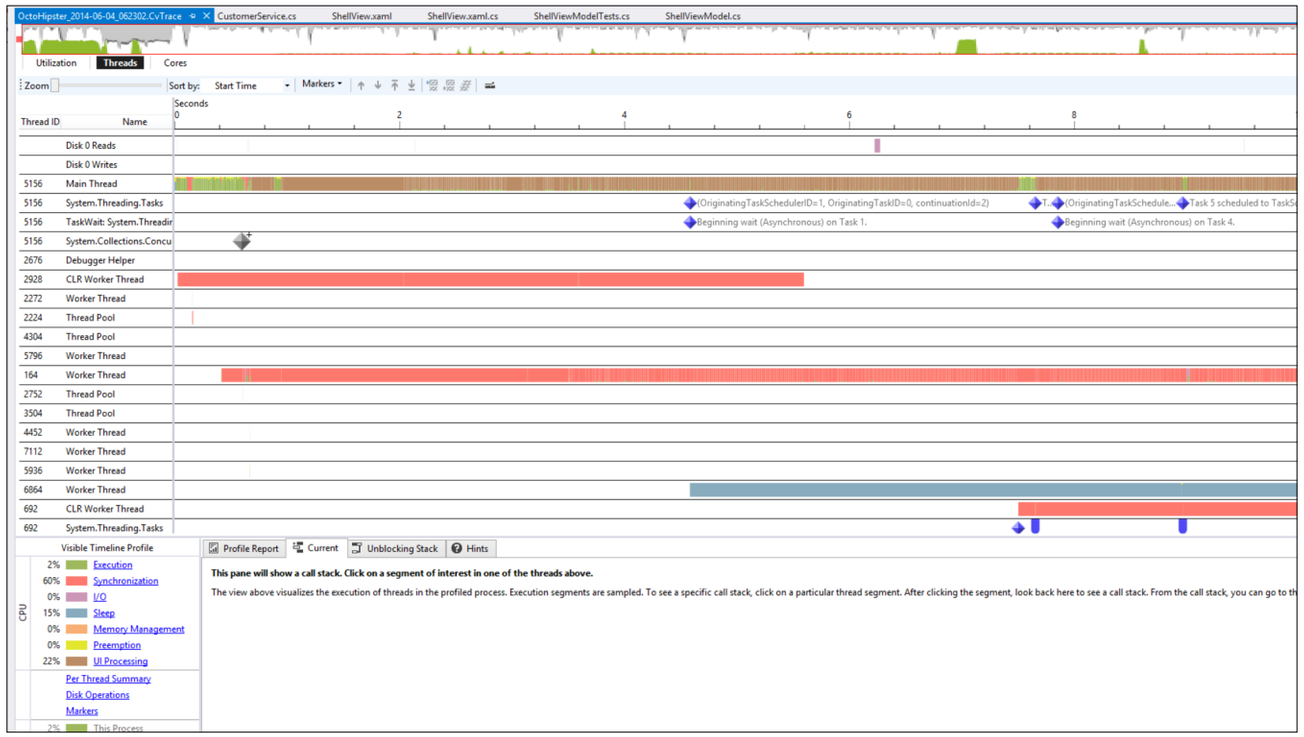
dude, where's my event?

get off my ~~lawn~~

main thread

Concurrency Visualizer

http://visualstudiogallery.msdn.microsoft.com/24b56e51-fcc2-423f-b811-f16f3fa3af7a


OctoHipster_2014-06-04_062302.CvTrace
CustomerService.cs
ShellView.xaml
ShellView.xaml.cs
ShellViewModelTests.cs
ShellViewModel.cs
Utilization
Threads
Cores
Zoom
Sort by:
Start Time
Markers
Seconds
0
2
4
6
8
Thread ID
Name
Disk 0 Reads
Disk 0 Writes
5156 Main Thread
5156 System.Threading.Tasks
5156 TaskWait: System.Threadir
5156 System.Collections.Concu
2676 Debugger Helper
2928 CLR Worker Thread
2272 Worker Thread
2224 Thread Pool
4304 Thread Pool
5796 Worker Thread
164 Worker Thread
2752 Thread Pool
3504 Thread Pool
4452 Worker Thread
7112 Worker Thread
5936 Worker Thread
6864 Worker Thread
692 CLR Worker Thread
692 System.Threading.Tasks
(OriginatingTaskSchedulerID=1, OriginatingTaskID=0, continuationId=2)
Beginning wait (Asynchronous) on Task 1.
(OriginatingTaskSchedule...
Task 5 scheduled to TaskS
Beginning wait (Asynchronous) on Task 4.
Visible Timeline Profile
CPU
2% Execution
60% Synchronization
0% I/O
15% Sleep
0% Memory Management
0% Preemption
22% UI Processing
Per Thread Summary
Disk Operations
Markers
Profile Report
Current
Unblocking Stack
Hints
This pane will show a call stack. Click on a segment of interest in one of the threads above.
The view above visualizes the execution of threads in the profiled process. Execution segments are sampled. To see a specific call stack, click on a particular thread segment. After clicking the segment, look back here to see a call stack. From the call stack, you can go to th

async everywhere

warps the mind

?