Microsoft.Activities.StateMachine

# overview

This project is designed to produce a code based experience for creating state machines that a non-workflow developer would find easy and friendly.

The implementation of the state machine would be provided by a Workflow Foundation (WF4) state machine activity.

# scenarios

**State Machine for the Non-Workflow Developer**

As a C# Developer  
I want to create a state machine for my application in code  
So that I do not have to use the Workflow Designer and learn Workflow Foundation

**State Machine for a WPF Client UI**

As a C# Developer building a rich-client application with WPF  
I want to use a state machine to implement a M-V-VM architecture   
So that my code is easy to read and maintain

**State Machine for a Web Client UI (Silverlight / Javascript)**

As a C# Developer building a RIA application   
I want to use a state machine to implement a M-V-VM architecture   
So that my code is easy to read and maintain

**State Machine for a Long Running Business Process**

As a C# developer building a business process (Insurance application for example)  
I want to use a code-based State Machine to implement a long-running process  
So that I can leverage persistence and I don't have to use a Workflow Designer

# features

|  |  |
| --- | --- |
| Priority | Feature |
| 1 | Fluent interface for creating a state machine |
| 1 | Use standard C# constructs such as enums, Func<T>, Action<T> |
| 1 | Provide events for state changes |
| 1 | Support Workflow tracking, extensions and persistence |
| 1 | Works well with async/await |
| 1 | Works well with reactive framework |
| 2 | Allow me to use activities created by Workflow Developers |
| 2 | Allow me to create a Workflow Service with this |
| 2 | Works well with LINQ |
| 1 | Strongly typed |
| 1 | Use type inference wherever possible |

## Tasks

#### Create a State Machine

* 1. Creating the state machine should be simple with a Fluent interface experience.

#### Run the State Machine

Q: What does it mean to run a state machine?

A: Because a state machine is reactive you create it and then it reacts to things which happen to it. In a synchronous model you could start it and wait for it to complete or run to an idle point.

In an asynchronous model you would start it and it would immediately return responding with actions invoked on other threads.

In a rich client application the async model is preferred

In a console app, web service or test environment the sync model is preferred.

#### In/Out Arguments & Variables

I don't want to force people to use IDictionary<string,object> to pass arguments.

You should be able to think of arguments as properties of the state machine. Perhaps a dynamic property is the right approach.

Some people who want a better intellisense experience will provide strongly typed property wrappers. They need a way to set the in/out arguments.

## Design issues

#### Typical Pattern

Most people who create a state machine will create a class that provides a domain specific layer. For example the ATM Machine from the hands on lab will include a class that provides methods meant to be called from the WPF host to start the state machine, wait for it to reach a certain state or

Should state machine class be sealed or not?

Not unless you want to force containment as the only model to use it.

At this point I'm in favor of designing for both containment and inheritance. Will have to consider which virtual methods need to be supported.

#### Triggers

* 1. A trigger in the code state machine is a named signal. The type of TTrigger can be anything but it is ultimately converted to a string and used as a bookmark name.
  2. Resuming a bookmark with the named trigger simply resumes the bookmark.

#### Messaging Triggers

* 1. TBD

#### Code based State Machines and State

* 1. How should we think about state in the code state machine?
  2. Some code based state machines will store their state entirely outside of the state machine. State stored in this way will not be persisted unless we provide some kind of API to get the state into the WF context
  3. If you want to use WF activities in your state machine you have to have a way to provide data to arguments (variable references etx).
  4. You cannot use variables from the code outside of an activity which declares the variable.
  5. You might want to put values into a property bag which would be persisted along with the state machine
  6. Values in the property bag could also be used as variables in a workflow (maybe?)