Documentación de MVVM en DevExpress

Interface ISupportViewModel

object ViewModel { get; }

void ParentViewModelAttached();

MainForm implementa ISupportViewModel

Setea el propio ViewModel a ViewModelSource.Create( () => new MainViewModel(this) );

ViewModelSource es una clase que permite agregar a un ViewModel una serie de comportamientos en runtime. Por ejemplo luego de llamar a Create, el campo viewModel de MainForm tiene una referencia a un tipo MainViewModel\_362d761b\_b227\_4362\_9601\_1d1dd0c7152c y no a MainViewModel como es de esperarse.

ViewModelSource realiza lo siguiente

* Crear bindable properties (lanzan notificaciones) para cada propiedad publica virtual, o bien para cada propiedad automatica que tenga un get y un set (minimo protected)
* Crear comandos para cada método publico que tenga cero o un parametro (no dice nada si debe retornar void)

Ambas caracteristicas pueden ser controladas con atributos o mediante fluent API

Que tengo que ver:

* como se enlazan las propiedades desde la vista al view model (en terminos de esta aplicación, desde un Modulo a un ViewModel)
* como presenta un dialogo modal con toda una vista completa (editar nuevo empleado)
* como valida datos en este dialogo
* como transfiere datos entre diferentes view models
* como se arma la ribbon para cada view