MVIEWVM

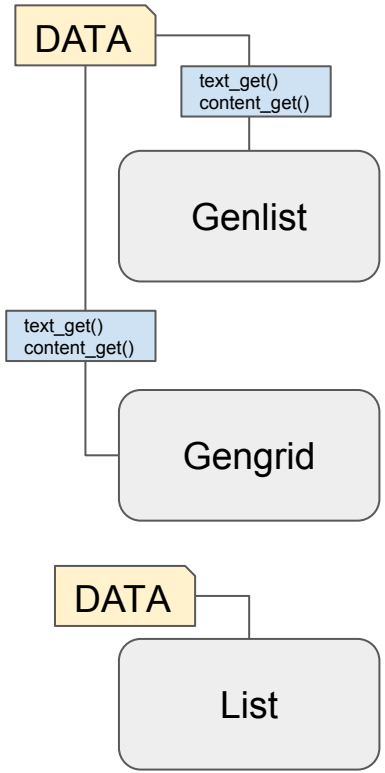
SangHyeon Jade Lee | SAMSUNG Research

Enlightenment Developer Days 2019

Pompeu Fabra University, Barcelona

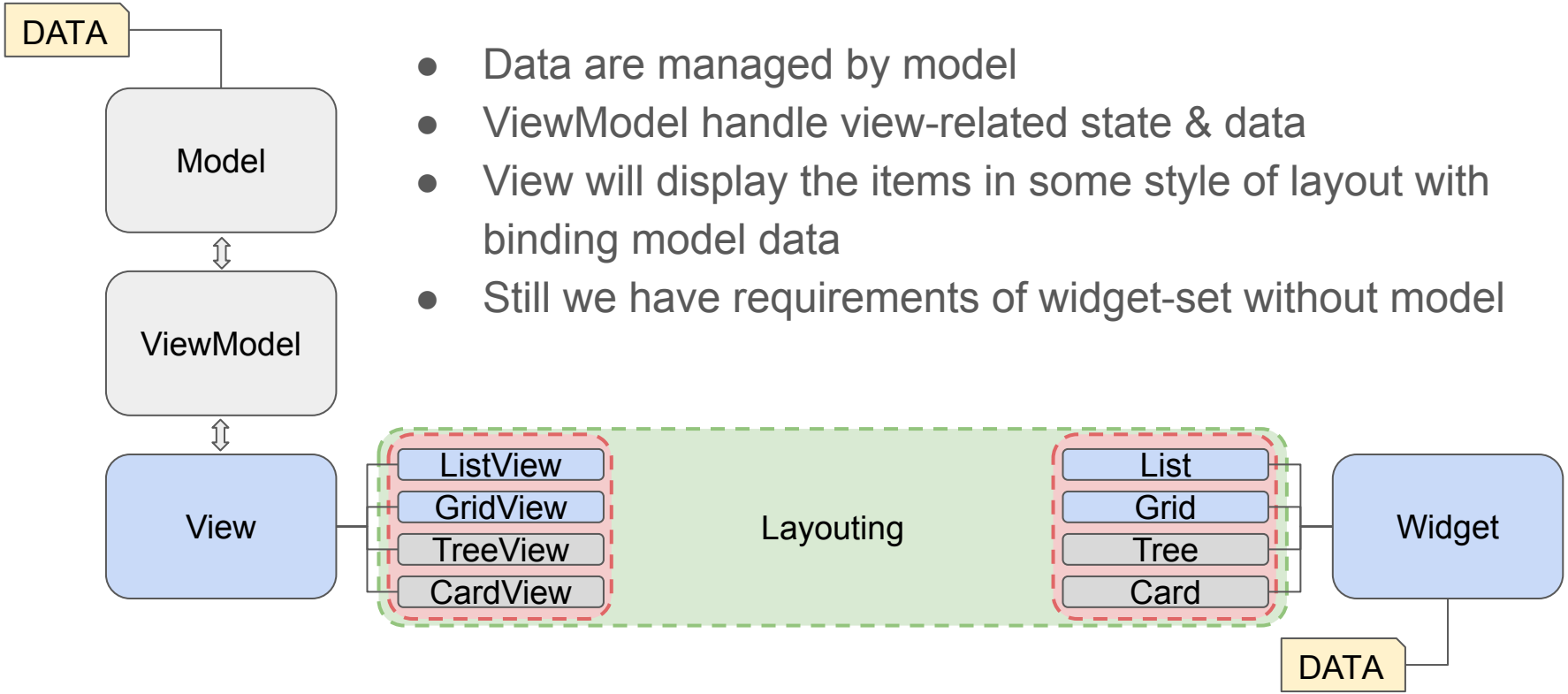
Table of Contents

Gen*



- Data were managed by User
- Dependency Inversion in Data Binding
- Many same codes were duplicated
- Item is not a widget class
- Hard to customize class
- Walking through the inlist for item search
- Hard to test widget behavior
- List was not powerful as genlist

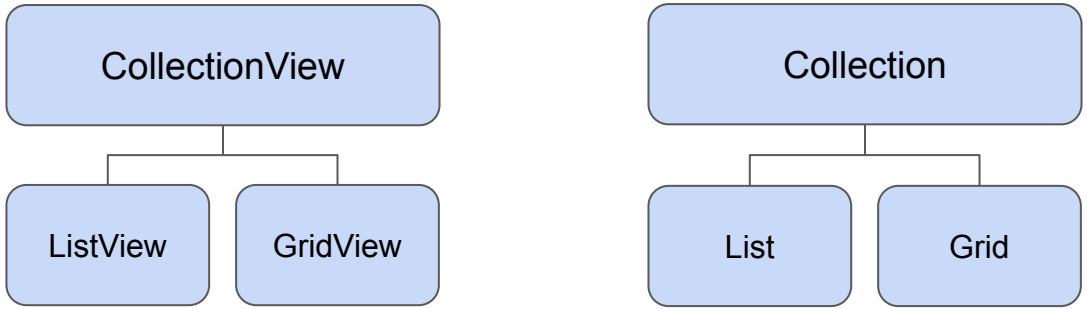
Overview



- Data are managed by model
- ViewModel handle view-related state & data
- View will display the items in some style of layout with binding model data
- Still we have requirements of widget-set without model

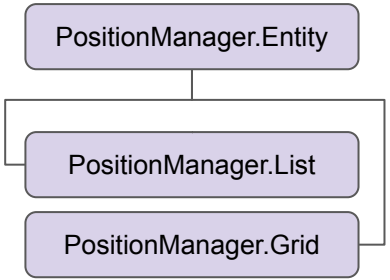
Collection

- Get rid of Duplicated Codes
- Classify Model-binding part and Layouting part
- Easy to add new view and widget class

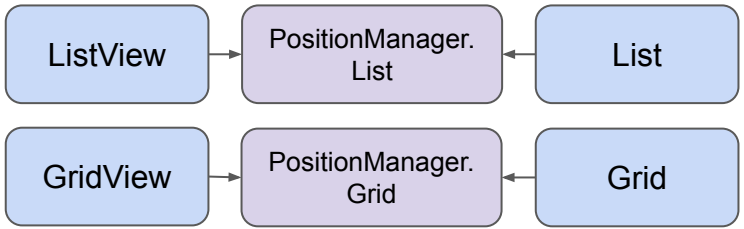


Position Manager

- Positioning(Layouting) can be shared both Widget and View
- Model - Binding Logic can be commonized
- Easy to customize class by inheriting PositionManager
- Request batch of data depending on viewport geometry (Lazy Loading)




```
var colView = new Efl.Ui.CollectionView(win);
var posCircle = new Efl.Ui.PositionManger.CircleList(win);
colView.PositionManager = posCircle;
```



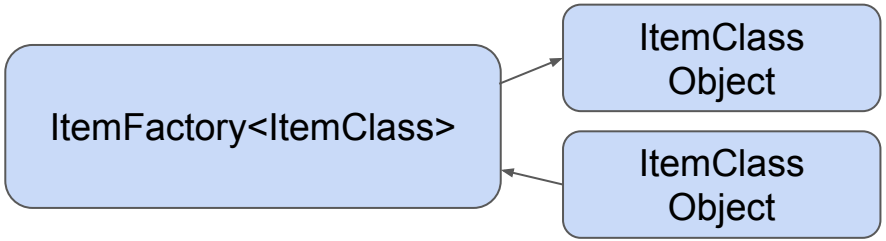
e.g

Factory & Item

- Legacy ItemClass delegated Data binding and item edje style creation
- Legacy Caching and Recycling Item was internally hidden
- Legacy Part of Item was hidden in edje group definition.
- Using Factory Pattern, binding and creation is more intuitive
- Class itself inhere the style of edje, Part need to be defined in it’s eo class.
- Item inherited layout, text, content parts are exposed in the class.

```
Elm_Genlist_Item_Class *itc =
elm_genlist_item_class_new();

itc->item_style = “default”;
itc->func.text_get = _gl_text_get;
itc->func.content_get = _gl_content_get;
itc->func.del_cb = _gl_del_cb;
…
elm_genlist_item_class_free(itc);
```

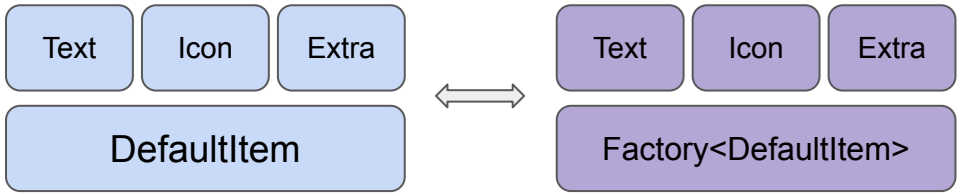


Data Binding

- User can easily bind the data
- Property to Edje Part (e.g. text, content)
- Property to Factory (e.g. image, check, radio)
- Property to Signal Events (e.g. odd/even)
- Factory binding properties to Efl.Part by proxy(Introspectdion)

Efl.Ui.Property_Bind

Efl.Ui.Factory_Bind




```
var itemFact = new Efl.Ui.ItemFactory<Efl.Ui.ListDefaultItem>(listView);
itemFact.TextPart().Markup().Bind("Region");
itemFact.IconPart().BindFactory(imgFact);
```

Example in C#(Mono)


```
class WeatherCast {
        public String Region;
        public String Weather;
        public static Efl.UserModel<WeatherCast> CreateModel(Efl.Loop loop)
        {
                Efl.UserModel<WeatherCast> casts = new Efl.UserModel<WeatherCast>(loop);
                casts.Add(new WeatherCast { Region = "Seoul", Weather = Predefined.Resource + "windy.png"});
                casts.Add(new WeatherCast { Region = "Tokyo", Weather = Predefined.Resource + "cloudy.png"});
                casts.Add(new WeatherCast { Region = "Hong Kong", Weather = Predefined.Resource + "lightning.png"});
                casts.Add(new WeatherCast { Region = "Delhi", Weather = Predefined.Resource + "hot.png"});
                casts.Add(new WeatherCast { Region = "Istanbul", Weather = Predefined.Resource + "hot.png"});
                casts.Add(new WeatherCast { Region = "Moscow", Weather = Predefined.Resource + "snow.png"});
                casts.Add(new WeatherCast { Region = "London", Weather = Predefined.Resource + "rainy.png"});
                casts.Add(new WeatherCast { Region = "Paris", Weather = Predefined.Resource + "cloud_sunny.png"});
                casts.Add(new WeatherCast { Region = "Barcelona", Weather = Predefined.Resource + "sunny.png"});
                casts.Add(new WeatherCast { Region = "New York", Weather = Predefined.Resource + "cloudy.png"});
                casts.Add(new WeatherCast { Region = "California", Weather = Predefined.Resource + "sunny.png"});
                casts.Add(new WeatherCast { Region = "Montreal", Weather = Predefined.Resource + "cloudy.png"});
                return casts;
        }
        private static Efl.Ui.ListView CreateListView(Efl.Object parent)
        {
                var weatherView = new Efl.Ui.ListView(parent);
                var weatherModel = WeatherCast.CreateModel(Efl.App.AppMain);
                var itemFact = new Efl.Ui.ItemFactory<Efl.Ui.ListDefaultItem>(weatherView);
                var imgFact = new Efl.Ui.ImageFactory(weatherView);
                itemFact.TextPart().Markup().Bind("Region");
                imgFact.BindProperty("filename", "Weather");
                itemFact.IconPart().BindFactory(imgFact);

                weatherView.Model = weatherModel;
                weatherView.Factory = itemFact;
                return weatherView;
        }
}
```

```
class WeatherCast {
	public String Region;
	public String Weather;
	public static Efl.UserModel<WeatherCast> CreateModel(Efl.Loop loop)
	{
		Efl.UserModel<WeatherCast> casts = new Efl.UserModel<WeatherCast>(loop);
		casts.Add(new WeatherCast { Region = "Seoul", Weather = Predefined.Resource + "windy.png"});
		casts.Add(new WeatherCast { Region = "Tokyo", Weather = Predefined.Resource + "cloudy.png"});
		casts.Add(new WeatherCast { Region = "Hong Kong", Weather = Predefined.Resource + "lightning.png"});
		casts.Add(new WeatherCast { Region = "Delhi", Weather = Predefined.Resource + "hot.png"});
		casts.Add(new WeatherCast { Region = "Istanbul", Weather = Predefined.Resource + "hot.png"});
		casts.Add(new WeatherCast { Region = "Moscow", Weather = Predefined.Resource + "snow.png"});
		casts.Add(new WeatherCast { Region = "London", Weather = Predefined.Resource + "rainy.png"});
		casts.Add(new WeatherCast { Region = "Paris", Weather = Predefined.Resource + "cloud_sunny.png"});
		casts.Add(new WeatherCast { Region = "Barcelona", Weather = Predefined.Resource + "sunny.png"});
		casts.Add(new WeatherCast { Region = "New York", Weather = Predefined.Resource + "cloudy.png"});
		casts.Add(new WeatherCast { Region = "California", Weather = Predefined.Resource + "sunny.png"});
		casts.Add(new WeatherCast { Region = "Montreal", Weather = Predefined.Resource + "cloudy.png"});
		return casts;
	}
	private static Efl.Ui.ListView CreateListView(Efl.Object parent)
	{
		var weatherView = new Efl.Ui.ListView(parent);
		var weatherModel = WeatherCast.CreateModel(Efl.App.AppMain);
		var itemFact = new Efl.Ui.ItemFactory<Efl.Ui.ListDefaultItem>(weatherView);
		var imgFact = new Efl.Ui.ImageFactory(weatherView);
		itemFact.TextPart().Markup().Bind("Region");
		imgFact.BindProperty("filename", "Weather");
		itemFact.IconPart().BindFactory(imgFact);

		weatherView.Model = weatherModel;
		weatherView.Factory = itemFact;
		return weatherView;
	}
}
```

Example in C#(Mono)

DISCUSSIONS

Stabilized BETA

- Multi_Selectable / Select_Model
- Items (placeholder / context / title / double_text)
- Boolean Model
- Size Model
- Position Manager
- Mono Classes

Model & ViewModel Features

- Model Insertion
- SQL Model
- Sorting Model
- itemClass Model
- Support more Widget States
- Boolean Property support in c#(Mono)

Factory

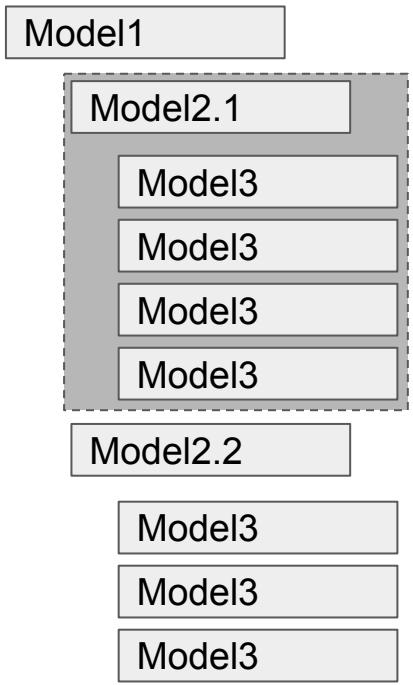
Factory for All

- Check / Radio / Button / Label
- Automated by Introspection

Multiple ItemClass in Factory

- Binding ItemClass with Model?
- Factory in Factory?
- Group / Tree

Group & Tree



- Group and Tree can be described Model in Model
- One Single View handle multiple models?
- ItemFactory?
- What is size of item on Model2.1?
- View in View vs View for Multiple models?

Animations

- Tree Expand / Collapse (Unfold / Fold)
- Scroll End
- Item New / Delete / Update
- Reordering
- Can it be replaced by User?

```
listView.Animations.Add = new Efl.Animation();
listView.Animations.Add = (sender, ev) => {
      Efl.Ui.Item item = (Efl.Ui.Item) sender;
      Efl.Ui.CollectionViewAnimationAddEventArgs addEvt = ev.args;
      // addEvt.pos : current animation position [0.0 - 1.0]
      // addEvt.interpolation : linking Efl.Interpolation
      // addEvt.userData : userData
      item.Geometry = (Eina.Rect) addEvt.UserData;
}
```

Header & Footer

- Header and Footer is Unique position Item
- Different Type of Data can be came.
- Append / Prepend should not effect.
- Item can be out of Model
- Item can be out of Factory

```
listView.Header = new Efl.Ui.TitleItem(listView);
listView.Header.TextPart.SetText("Fruits");
listView.Footer = new Efl.Ui.ListDefaultItem(listView);
listView.Footer.TextPart.SetText("Load More ...");
```

THANK YOU
FOR WATCHING!