

Flex Data Visualization

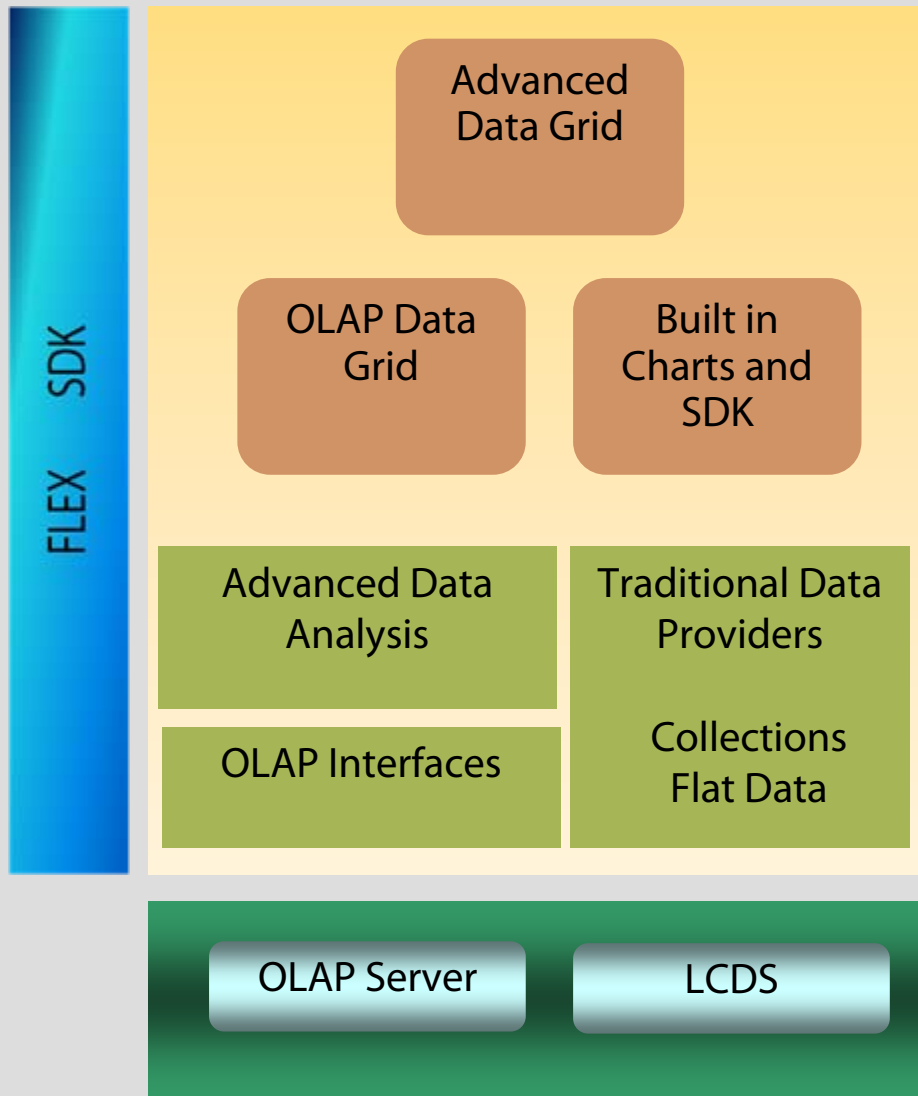
Sreenivas Ramaswamy, Sr. Computer Scientist

Sunil Bannur, Computer Scientist

Adobe Systems



FLEX Data Visualization Landscape



- Visualization
 - Charts
 - Advanced Data Grid
 - OLAP Data Grid
- Advanced data analysis
 - CRUD, sorting/filtering of Hierarchical data
 - Grouping and summary on flat data
 - Multi dimensional aggregations and trend analysis

Flex Chart Types

- Comparative data:
 - Plot charts (a.k.a Scatter Plot) for positioning two values along the axes
 - Pie charts for proportional relevance of multiple values of a total sum
 - Bar (horizontal) and Column (vertical) charts for data by category
- Date / Time based- data:
 - Line / Area/Bubble Charts can be used to analyze value changes over a time period
- Multi-Value relations display:
 - Bubble charts can be used to represent data with at least 3 values, containing information for position and relational volume / radius display
- Financial data:
 - HLOC (High Low Open Close) and Candlestick charts can be used for comparative multi-value data such as stock information
- Stacked representation:
 - When using multiple series in a chart, the bar, column and area series could be stacked.
 - The type of stacking could be overlaid, stacked or 100%

Flex Chart Types contd ..

- Axis Types
 - Category, Linear, Logarithmic, DateTime axis
- All Flex charts use similar attributes (Chart, Series, Axes, Elements) to define the data, how it is displayed, and interactions with the chart

```
<mx:BarChart id="bar" dataProvider="{medalsAC}" >
  <mx:verticalAxis>
    <mx:CategoryAxis categoryField="Country"/>
  </mx:verticalAxis>
  <mx:series>
    <mx:BarSeries id = "barseries1" yField="Country" xField="Gold" displayName="Gold" />
  </mx:series>
</mx:BarChart>
<mx:Legend dataProvider="{bar}"/>
</mx:BarChart>
```

Formatting Charts

- Applying Styles
 - Numerous styles like font, color, ticks, which could be applied through CSS or inline when defining the components or through MXML tags.
- Adding background/foreground objects
 - Add ChartElement objects (such as images, grid lines, and strokes) to the charts by using the backgroundElements and annotationElements properties of the chart classes.
- Showing Datatips
 - DataTips show a small pop-up window that shows the data value for the data point under the mouse pointer.
 - showDataTips enables datatips and dataTipFunction allows customization of the datatip
- Padding and Gutters
 - Padding is the area between the outside bounds of the chart control and its content .
 - Gutter is the area between the padding area and the actual axis line. Chart controls adjust the gutters to draw label, titles and ticks to the axis, but one can specify explicit gutter values.

Formatting Charts contd ...

- Item Renderers for Chart Items
 - Each of the ChartItem represented in a particular chart could be represented by an itemRenderer
 - Default renderers like BoxItemRenderer, CircleItemRenderer can be extended
- Legend Controls
 - provide visual key to associate groups of data to their visual counterpart
 - match the fill patterns on the chart to labels that describe the data series shown with those fill patterns
- Formatting Axes
 - The axis titles, labels, tick marks and the axis line can be formatted
 - stagger or rotate the axis labels for better visualizations

Flex Charts – Events and Effects

- Chart Events
 - Mouse Events – itemClick, itemDoubleClick, itemMouseDown, itemMouseMove, itemRollOut, itemRollOver, itemMouseUp
 - Extensible APIs to control what items could be interactive (example: findDataPoints)
 - Control interactivity with the interactive property
- Chart Effects
 - Standard Flex effects such as Zoom and Fade
 - Charts trigger effects with each with 2 triggers: hideDataEffect and showDataEffect
 - Additional effects supported by series
 - SeriesInterpolate, SeriesSlide and SeriesZoom

Flex 3 Charting Enhancements

- Selection, Region Selection
- Drag and Drop
- Keyboard accessibility

Charting Interaction

Multiple Axes and Renderers

- Ability to associate each series with an axis of its own
- Rendering of multiple axes

Per Item Label

- Labels for Bar and column chart

- Fill function, fills Array for most of the charts

Custom fills

Work Week Axis

Filtering of dates from a DateTime axis

Flex 3 Charting Enhancements

- Canvas like functionality to draw on charts as annotations
- Cartesian and Polar canvases
- Ability to have pixel based offset



Data Graphics



Miscellaneous enhancements

- Data functions to get item data for series, category axis
- Non interactive data tips, Negative Stacked Charts
- Additional styles –
 - vertical alignment, label alignment in axis renderer
 - minRadius in bubble series
- Title renderer

Flex 3 Charts – Labels, Selection

- Selection in charts is achieved by setting the selection mode property which could be single, multiple or none
- The Labels on the Series could be specified as inside and outside

```
<mx:BarChart id="chart" height="50%" width="50%" paddingRight="5" paddingLeft="5"  
    showDataTips="true" dataProvider="{medalsAC}" color="blue" selectionMode = "multiple">
```

```
<mx:verticalAxis>
```

```
    <mx:CategoryAxis categoryField="Country"/>
```

```
</mx:verticalAxis>
```

```
<mx:series>
```

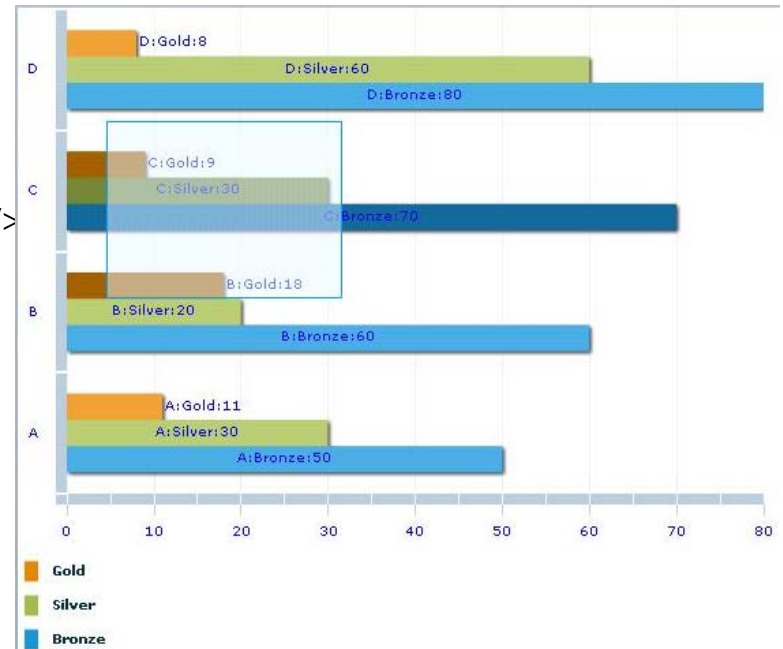
```
    <mx:BarSeries ... labelPosition="outside" labelFunction="myLabel"/>>
```

```
    <mx:BarSeries ... labelPosition="inside" labelFunction="myLabel"/>
```

```
    <mx:BarSeries ... labelPosition="inside" labelFunction="myLabel"/>
```

```
</mx:series>
```

```
</mx:BarChart>
```



Flex 3 Charts – Data function, All data tips

- Complex data providers could be handled now where xValue, yValue and categoryField could be dynamically calculated by the datafunction.
- Datatips can be shown now at all times, without having to hover the mouse around the item

```
dp:Array = [ {month: "Aug", close: {High:45.87,Low:12.2}, open:25.19 }, {month: "Sep", close: {High:45.74,Low:10.23}, open:35.29} ];
```

```
<mx:ColumnChart id="chart" dataProvider="{dp}" showAllDataTips="true" >  
  <mx:horizontalAxis> <mx:CategoryAxis id="h1" dataFunction="catFunc"/> </mx:horizontalAxis>  
  <mx:series>  
    <mx:ColumnSeries displayName="Close (High)"  
      dataFunction="dataFunc" />  
  </mx:series>
```

```
private function catFunc(axis:AxisBase, item:Object):Object
```

```
{  
  return item.month;  
}
```

```
private function dataFunc(series:Series, item:Object,  
  fieldName:String):Object
```

```
{  
  if ( fieldName == "yValue") return(item.close.High);  
  else if (fieldName == "xValue") return (item.month); return null;  
}
```



Flex 3 Charts – Data Graphics, Multiple Axes

- Datagraphics allows to draw on charts as annotation elements or background elements based on data values

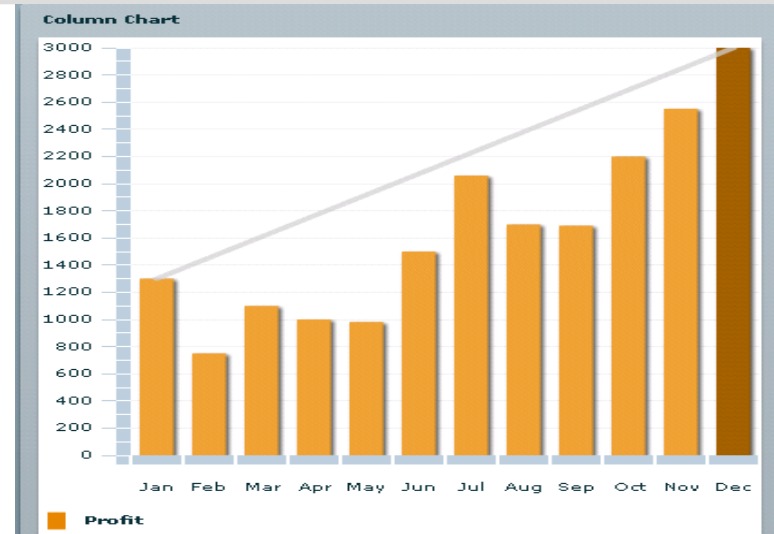
```
canvas.lineStyle(4, 0xCCCCCC, .75, true, LineScaleMode.NORMAL, .., ..);
```

```
canvas.moveTo("Jan", 1250); canvas.lineTo("Dec", 3000);
```

```
<mx:annotationElements>
```

```
  <mx:CartesianDataCanvas id="canvas" includeInRanges="true"/>
```

```
</mx:annotationElements>
```



- Multiple Axes and Renderers:

```
<mx:verticalAxisRenderers>
```

```
  <mx:AxisRenderer axis="{v2}"/>
```

```
</mx:verticalAxisRenderers>
```

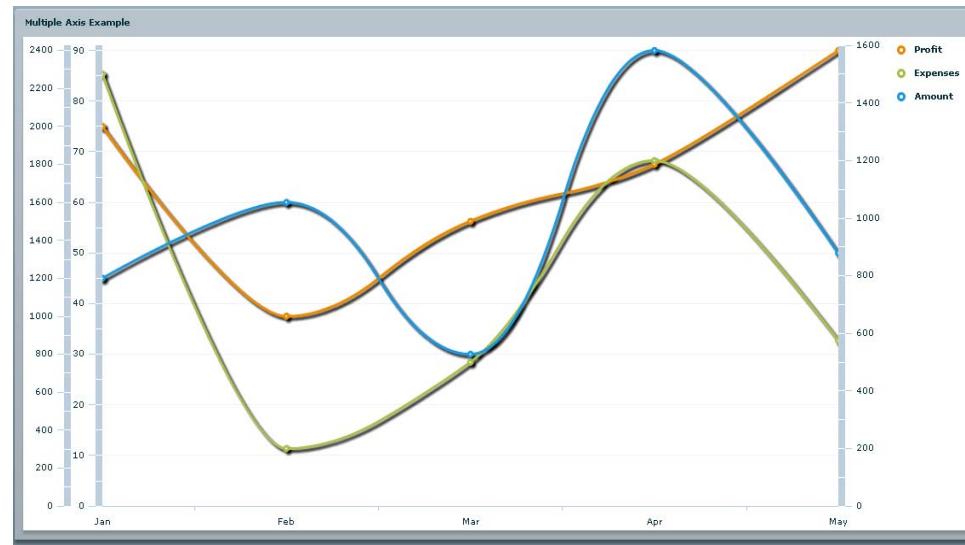
```
<mx:LineSeries yField="Expenses" form="curve">
```

```
  <mx:verticalAxis>
```

```
    <mx:LinearAxis id="v2" />
```

```
  </mx:verticalAxis>
```

```
</mx:LineSeries>
```



AdvancedDataGrid features

- Multi column sorting
- TreeView for hierarchical/grouped Data
- Column grouping
- Column/Cell spanning of itemRenderers
- Conditional formatting and support for formatters
- Summary using Grouping Collection
- Cell selection

Multi column sorting

Region	Territory	1 ▲	Territory ...	2 ▲	Estimate	3 ▲	Actual
Southwest	Arizona		Barbara Jennings		40000		38865
Southwest	Arizona		Dana Binn		30000		29885
Southwest	Central Californ		Joe Schmoe		30000		29134
Southwest	Nevada		Bethany Pittmar		45000		52888
Southwest	Northern Califor		Lauren Ipsum		40000		38805
Southwest	Northern Califor		T.R. Smith		40000		55498

TreeView for Hierarchical data

- Hierarchical data is in a structure of parent and child data items
 - HierarchicalData can have the input as ArrayCollection or XMLList
 - Supports sorting and filtering

```
<AdvancedDataGrid dataProvider={new HierarchicalData(input)} />
```

```
var input:Array = [ { Region: "Southwest",  
    children: [ { Territory: "Arizona", Estimate:35000, Actual :33000 } ... ] },  
    { Region: "Southwest",  
    children: [ { Territory: "Nevada", Estimate:45000, Actual : 40000 } ... ] } .... ];
```

Region	Territory	Territory Rep	Est...	Actual
▼ Southwest				
▼	Arizona			
Southwest	Arizona	Barbara Jennings	40000	38865
Southwest	Arizona	Dana Binn	30000	29885
▼	Central California			
Southwest	Central California	Joe Schmoe	30000	29134
▼	Nevada			
Southwest	Nevada	Bethany Pittman	45000	52888

Grouping

- GroupingCollection
- Grouped data is generated from flat data by specifying a set of group fields
 - GroupingCollection allows asynchronous grouping

```
<AdvancedDataGrid creationComplete="gc.refresh()">
```

```
<GroupingCollection id="gc" source="{dpFlat}">
```

```
<Grouping>
```

```
<GroupingField name="Region"/>
```

```
<GroupingField name="Territory"/>
```

```
</Grouping>
```

```
</GroupingCollection>
```

```
</AdvancedDataGrid>
```

Region	Territory	Territory Rep	Actual	Estimate
▼ Southwest				
▼ Arizona				
Southwest	Arizona	Barbara Jennings	38865	40000
Southwest	Arizona	Dana Binn	29885	30000
▶ Central California				
▶ Nevada				
▶ Northern California				
▶ Southern California				

Summary support in grouping

- Aggregate information such as Sum, Avg, Max, Min, count and custom summaries from Grouping Collection

```
<mx:GroupingCollection id="gc" source="{dpFlat}">
```

```
<mx:Grouping>
```

```
<mx:GroupingField name="Region">
```

```
<mx:SummaryRow summaryPlacement="group">
```

```
<mx:SummaryField dataField="Actual" label="Min" operation="MIN"/>
```

```
</mx:SummaryRow>
```

```
</mx:GroupingField>
```

```
</mx:Grouping>
```

```
</mx:GroupingCollection>
```

Region	Territory Rep	Actual	Estimate	Min Actual	Max Actual
▼ Southwest				29134	55498
▼ Arizona				29885	38865
Southwest	Barbara Jennings	38865	40000		
Southwest	Dana Binn	29885	30000		
▶ Central California				29134	29134
▶ Nevada				52888	52888
▶ Northern California				38805	55498
▶ Southern California				44913	44985

Column Grouping

- Allows any level of nesting

MajorGroup-1				MajorGroup-2			
SubGroup1-1		SubGroup1-2		SubGroup2-1		SubGroup2-2	
SubCol1-1	SubCol1-2	SubCol2-1	SubCol2-2	SubCol1-1	SubCol1-2	SubCol2-1	SubCol2-2

Column grouping in MXML

- Group the Actual and Estimate columns under a group named Revenues

```
<mx:AdvancedDataGrid >
```

```
<mx:groupedColumns>
```

```
  <mx:AdvancedDataGridColumn dataField="Region" headerText="Region"/>
```

```
  <mx:AdvancedDataGridColumn dataField="Territory" headerText="Territory"/>
```

```
  <mx:AdvancedDataGridColumn dataField="Territory_Rep"  
    headerText="Territory Rep"/>
```

```
  <mx:AdvancedDataGridColumnGroup headerText="Revenues">
```

```
    <mx:AdvancedDataGridColumn dataField="Actual"/>
```

```
    <mx:AdvancedDataGridColumn dataField="Estimate"/>
```

```
  </mx:AdvancedDataGridColumnGroup>
```

```
</mx:groupedColumns>
```

```
</mx:AdvancedDataGrid>
```

Column grouping ...

Region	Territory	Territory Rep	Revenues	
			Actual	Estimate
Southwest	Arizona	Barbara Jennings	38865	40000
Southwest	Arizona	Dana Binn	29885	30000
Southwest	Central California	Joe Smith	29134	30000
Southwest	Nevada	Bethany Pittman	52888	45000
Southwest	Northern California	Lauren Ipsum	38805	40000
Southwest	Northern California	T.R. Smith	55498	40000
Southwest	Southern California	Alice Treu	44985	45000
Southwest	Southern California	Jane Grove	44913	45000

Column/Cell spanning

- Span multiple columns with a renderer

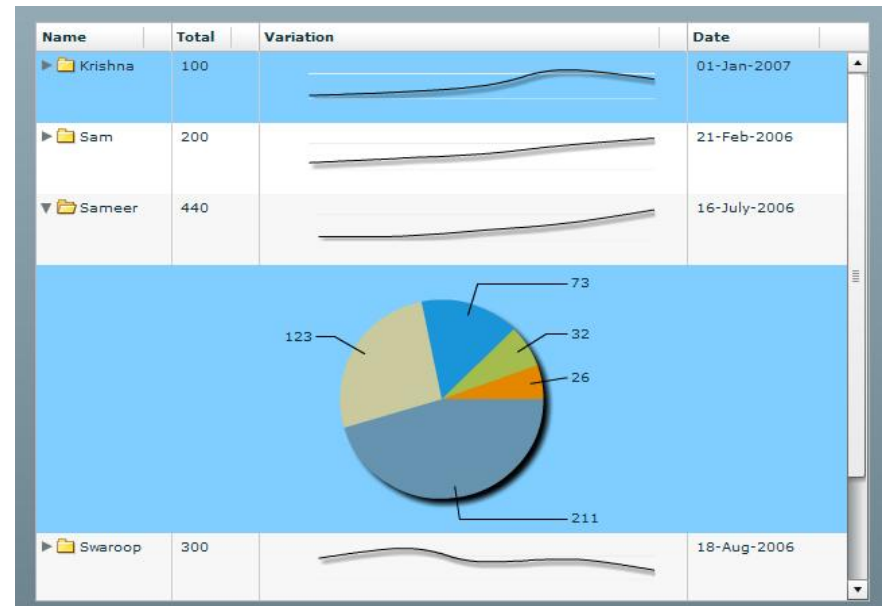
```
<mx:AdvancedDataGrid>
```

```
<mx:rendererProviders>
```

```
<mx:AdvancedDataGridRendererProvider columnIndex="0" columnSpan="0" depth="2"  
    renderer="mx.charts.PieChart"/>
```

```
</mx:rendererProviders>
```

```
</mx:AdvancedDataGrid>
```



Custom styling and formatters

```
<CurrencyFormatter currencySymbol="$" id="curFormat" />

<AdvancedDataGrid id="adg" >
  <columns>
    <AdvancedDataGridColumn headerText="Estimate" textAlign="center"
dataField="Estimate" formatter="{curFormat}" styleFunction="formatSummary" />
    .....
  </columns>
</AdvancedDataGrid>
```

```
private function formatSummary(data:Object, col:AdvancedDataGridColumn):Object
{
    if (adg.dataProvider.getRootModel().canHaveChildren(data))
    {
        //change text color to red
        return { color:0xFF0000 };
    }
    //no style changes
    return {};
}
```

Custom styling and formatters ...

Territory Rep	Sales Figures	
	Estimate	Actual
▼  Southwest	\$315,000	\$334,973
▼  Arizona	\$70,000	\$68,750
 Barbara Jennings	\$40,000	\$38,865
 Dana Binn	\$30,000	\$29,885
▶  Central California	\$30,000	\$29,134
▼  Nevada	\$45,000	\$52,888
 Bethany Pittman	\$45,000	\$52,888
▶  Northern California	\$80,000	\$94,303
▼  Southern California	\$90,000	\$89,898
 Alice Treu	\$45,000	\$44,985
 Jane Grove	\$45,000	\$44,913

Cell selection

- selectionMode = singleCell | multipleCells, singleRow, multipleRows

Region	Territory	Territory ...	Estimate	Actual
Southwest	Arizona	Barbara Jennings	40000	38865
Southwest	Arizona	Dana Binn	30000	29885
Southwest	Central Californ	Joe Schmoe	30000	29134
Southwest	Northern Califor	Lauren Ipsum	40000	38805
Southwest	Northern Califor	T.R. Smith	40000	55498
Southwest	Southern Califor	Jane Grove	45000	44913

OLAP : Online Analytical Processing

- OLAP Interfaces
 - IOLAPCube
 - IOLAPDimension, IOLAPAttribute
 - IOLAPHierarchy, IOLAPLevel
 - IOLAPMember
 - IOLAPQuery, IOLAPQueryAxis
 - IOLAPResult, IOLAPResultAxis
 - IOLAPTuple, IOLAPSet,
- OlapDataGrid for query result visualization
- Default implementation provides a client side in memory cube.

OLAP: Steps

1. Start with a flat data in a local collection
2. Configure a local cube by specifying
 - Dimensions
 - Attributes
 - Hierarchies
 - Levels
 - Measures
3. Build the cube by calling `refresh()`
4. Query the cube by specifying
 - column and row axes
 - measure
 - aggregation operation
 - Optionally specify slicer axis
5. Visualize the query result in an `OLAPDataGrid`

Defining OLAPCube

```
<mx:OLAPCube id="salesCube" dataProvider="{companySalesData}" >
```

```
  <mx:OLAPDimension name="SalesData" >
```

```
    <mx:OLAPAttribute name="Company" dataField="company" />
```

```
    <mx:OLAPAttribute name="Region" dataField="region" />
```

```
    <mx:OLAPAttribute name="Market" dataField="market" />
```

```
    <mx:OLAPAttribute name="Product" dataField="product" />
```

```
  <mx:OLAPHierarchy name="Region-Market-Store" >
```

```
    <mx:OLAPLevel attributeName="Company" />
```

```
    <mx:OLAPLevel attributeName="Region" />
```

```
    <mx:OLAPLevel attributeName="Market" />
```

```
    <mx:OLAPLevel attributeName="Product" />
```

```
  </mx:OLAPHierarchy>
```

```
</mx:OLAPDimension>
```

```
<mx:OLAPMeasure name="Revenue" dataField="revenue" />
```

```
</mx:OLAPCube>
```

OLAP Demo

- Analyzing company revenue data



Revolutionizing
how the world engages
with ideas and information