BIRT Chart Interactivity

Functional Specifications

Draft 5: August 24, 2005

Abstract

This document describes the functional specifications of the Chart interactivity features for BIRT 2.0

Document Revisions

Version	Date	Primary Author(s)	Description of Changes
Draft 1	July 21, 2005	David Michonneau	Initial Draft
Draft 2	July 26, 2005	David Michonneau	Revised TriggerConditions. Added a new refresh entry in the interactivity menu and actions. Added a new paragraph about the event flow
Draft 3	July 28, 2005	David Michonneau	Added mockups for default interactivity
Draft 4	August 5, 2005	David Michonneau	Marked Rotation as POST 2.0 feature. Added chart dimension in menu. More details on JavaScript API. Removed Builder section.
Draft 5	August 24, 2005	David Michonneau	Added: - Use Cases - Drilldown - Filtering - Interactivity Configuration Removed: - 3D rotation - Subtype/Dimension change Improved: - Zoom usability - Toolbar

Contents

1. Introduction

The chart interactivity allows the users to perform different actions on the chart they are viewing.

Each action is mapped to a user gesture, this mapping is called a trigger. These UI gestures can be standard mouse or keyboard events. The Actions can modify the visuals of the chart in different ways, but they can also perform some change at the viewer level, such as redirect to a new page for instance.

Interactivity features depend on the output type, but also on the environment (report/html/birt viewer). The goal is to achieve the best interactivity using SVG inside the HTML BIRT viewer. Other outputs will benefit some interactivity features as well.

2. Use Cases

These use cases are extracted from the Chart interactivity design document of IBM and feedback received on the chart-dev mailing list.

2.1 Highlight

A pie chart is displayed in the eclipse workbench. Therefore the chart is rendered in an SWT device. The user has the ability to select a pie slice. When the user select a pie slice the pie slice changes color indicating that it has been selected. Furthermore, property information associated with the pie slice is displayed in the property view.

2.2 Visibility Toggle

A line chart consisting of two datasets is displayed in an html page using the SVG renderer. The user has the ability to select a legend item to toggle the visibility of the associated dataset in the plot area.

2.3 URL redirection

A pie chart is displayed in an html page using the SVG renderer. When the user selects a pie section the browser is redirected to a URL.

2.4 ToolTip

A pie chart is displayed in the eclipse workbench. When the user hovers over a pie section tooltip text is displayed.

2.5 DrillDown

2.5.1 Example scenario #1

I am viewing a bar chart of world-wide sales by year -- with a bar for each year showing me total sales. I think click on bat for year "2005" and the chart changes to show me the sales for each quarter in 2005. Effectively, this "drill down" has applied a filter to only look at "2005", and has changed the series from year to quarter. If I wanted, could also drill back out to the year level.

2.5.2 Example scenario #2

I am looking at a pie chart that shows total sales for each product line in each slice of the pie. I click on the pie slice and we drill into a pie chart that shows the total sales numbers for all the individual products in that category.

2.6 Filter

If "color" is in the original dataset (but not actually charted) then allow the end user to specify that the chart should only show products where "color=red". Simple filtering would be all that is required here (in order to keep the UI simple and easy to use) - along the lines of Excel's Automatic Filter capability.

3. Triggers

3.1 Overview

Triggers define a mapping between a trigger condition (a UI gesture) and an action. They can be associated with any Chart Block or Serie (see Block.getTriggers() and Series.getTriggers()).

3.2 Trigger Conditions

This table lists all the possible trigger conditions. There are similar to most of the HTML40 intrisic events except that they apply to chart elements and not html elements.

Note that some events such as doubleclick will not work in all renderers/browsers/os. In such case they will simply be ignored.

Name	Description
onclick	Occurs when the pointing device button is clicked over an element
ondblclick	Occurs when the pointing device button is double clicked over an element.
onmousedown	Occurs when the pointing device button is pressed over an element.
onmouseup	Occurs when the pointing device button is released over an element.
onmouseover	Occurs when the pointing device is moved onto an element.
onmousemove	Occurs when the pointing device is moved while it is over an element.

Name	Description
onmouseout	Occurs when the pointing device is moved away from an element.
onfocus	Occurs when an element receives focus either by the pointing device or by tabbing navigation
onblur	Occurs when an element loses focus either by the pointing device or by tabbing navigation
onkeydown	Occurs when a key is pressed down on an element
onkeypress	Occurs when a key is pressed on an element
onkeyup	Occurs when a key is up on an element
onload	Occurs when the chart is loaded in the viewer

Each trigger is associated with any visual component of the chart, called hotspot: that can be a legend item, the chart title, a chart serie, etc… Each hotspot will react to some predefined triggers with one or several actions. Note that this action is not limited to the hotspot it originated.

3.3 Trigger Flow

The trigger flow will follow the W3C DOM specifications event model: http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/events.html#Events-flow

That is, trigger conditions will be first captured down the target element and then will bubble back up. Any block having registered the same trigger during those two sequences will have the correspond action executed. The TriggerFlow class will define whether it reacts to the capturing or bubbling sequence, the default being the bubbling sequence. It will also allow the propagation sequence to stop in the bubbling phase.

3.3.1 API

A new TriggerFlow class will define three constants:

Capture: the trigger's action will be triggered in the capturing phase

Bubble (default): the trigger's action will be triggered in the bubbling phase

BubbleAndStop: the trigger's action will be triggered in the bubbling phase, and will stop the bubbling flow.

4. Actions

Actions are executed in answer to some trigger events.

4.1 Description

Here are the possible actions:

4.1.1 Url Redirect

Only works in an html environment. This redirects the browser to a new url. A possible extension is to redirect the browser to a different page of the report or another report.

4.1.2 Show ToolTip

This shows a tooltip on a chart element. It is normally associated with a mouse hover event.

4.1.3 Toggle Visibility

This allows to toggle the visibility of one or several series

4.1.4 Highlight (new)

This highlights a chart element. It is normally associated with a mouse click event.

4.1.5 Refresh (new)

This requests a new chart to be rendered based on the current data. It is a one time refresh, but there is an option to set it as automatic with any time period defined in seconds.

4.1.6 Invoke Script

This invokes a script inside the viewer. The scripts are written in a language specific to the rendering environment and output type.

4.2 Support matrix

Not all actions are supported for all outputs and environments. Here is a table summarizing the actions on an output basis.

Output Type/Action	SVG	SWT	Swing	Static image in PDF	Static image in HTML
Url Redirect	✓	✓	✓	✓	✓
Show Tool tip	✓	✓	✓	✓	✓
Toggle Visibility	✓	✓	✓		
Highlight	✓	✓	✓		
Refresh	✓	✓	✓		✓
Invoke Script	✓				✓

5. Scripting

When the built-in interactivity features defined by the API are not enough, the user can write his own interactivity script, which is dependent on the output type. This is not to be confused with Chart Scripting which is done at the rendering time.

This release will provide scripting support for SVG (in any svg viewer) and static images (inside a browser)

5.1 SVG Scripting

SVG output supports the ECMAScript language, also known as JavaScript.

An API will be provided to allow the user to manipulate the SVG DOM and access the chart engine on the server.

The capabilities of SVG Scripts allow it to communicate with the chart engine at a specific url through web services. It can also modify the SVG output

5.1.1 SVG JavaScript API

To be defined in the design document.

5.2 Static Images Scripting

It's possible to write scripts for static images, using JavaScript. An API will be provided to communicate with the chart engine or access other report elements. These scripts can only be used in a javascript-enabled environment.

5.2.1 Static images Script API

function refresh(Chart) : this will cause the chart element to be regenerated by the server and refreshed inside the browser, using the provided chart model as a parameter.

function Chart getModel(): returns the current chart model

function refresh() : refresh the chart using the current chart model. This is useful for getting live data charts.

6. Chart Interactivity API

Several classes in the Chart API define the interactivity properties. These properties are then used by each device renderer. Here are the key classes and interfaces to define interactivity at the API level:

6.1 InteractionEvent

The InteractionEvent defines the triggers and action mapping on given hotspots:

public final class **InteractionEvent**
extends PrimitiveRenderEvent

Author:
Actuate Corporation

See Also:
Serialized Form

Field Summary

Fields inherited from class org.eclipse.birt.chart.event.PrimitiveRenderEvent
`DRAW, FILL, iObjIndex`

Constructor Summary

`InteractionEvent``(java.lang.Object source)`

Method Summary

`void`	**`addTrigger`**`(Trigger t)`
`Action`	**`getAction`**`(TriggerCondition tc)`
`PrimitiveRenderEvent`	**`getHotSpot`**`()`
`Trigger[]`	**`getTriggers`**`()`
`void`	**`reuse`**`(java.lang.Object oNewSource)`
`void`	**`setHotSpot`**`(PrimitiveRenderEvent pre)`

6.2 Trigger

public interface **Trigger**

extends org.eclipse.emf.ecore.EObject

A representation of the model object '***Trigger***'. This type defines a Trigger. A trigger defines interactivity for a chart component.

The following features are supported:

- *Condition*
- *Action*

See Also:
`DataPackage.getTrigger()`

Method Summary

`Action`	**`getAction`**`()` Returns the value of the '***Action***' containment reference.
`TriggerCondition`	**`getCondition`**`()` Returns the value of the '***Condition***' attribute.
`boolean`	**`isSetCondition`**`()` Returns whether the value of the ' *`Condition`*' attribute is set.
`void`	**`setAction`**`(Action value)` Sets the value of the '*`Action`*' containment reference.
`void`	**`setCondition`**`(TriggerCondition value)` Sets the value of the '*`Condition`*' attribute.
`Void`	**`unsetCondition`**`()` Unsets the value of the '*`Condition`*' attribute.
`void`	**`setFlowReaction(`**TriggerFlow flow**`)`** Set the trigger flow reaction
TriggerFlow	**`getFlowReaction()`** Returns the trigger flow reaction

6.3 Action

public interface **Action**
extends org.eclipse.emf.ecore.EObject

A representation of the model object '***Action***'. This type defines an Action. An action is a property defining interactivity for an element. It is associated in a trigger with a trigger condition that defines when the action is to be processed.

The following features are supported:

- *`Type`*
- *`Value`*

See Also:
`DataPackage.getAction()`

Method Summary

`ActionType`	**`getType`**`()` Returns the value of the '***Type***' attribute.
`ActionValue`	**`getValue`**`()` Returns the value of the '***Value***' containment reference.
`boolean`	**`isSetType`**`()`

	Returns whether the value of the '*Type*' attribute is set.
`void`	**`setType`**`(ActionType value)` Sets the value of the '*Type*' attribute.
`void`	**`setValue`**`(ActionValue value)` Sets the value of the '*Value*' containment reference.
`void`	**`unsetType`**`()` Unsets the value of the '*Type*' attribute.

7. BIRT HTML Viewer Interactivity

7.1 Browser Requirements

Full interactivity will only be available when the user is using a SVG and JavaScript enabled browser. The Chart full interactivity will be available on IE6+ with Adobe SVG Plugin or Firefox 1.1.

7.2 Automatic SVG support detection

The report will automatically show SVG content if the browser supports it, otherwise it will use a static image of the chart, with limited interactivity.

8. Built-in Interactivity

BIRT will provide default built-in interactivity for the Chart, in addition to what the user defines. This section describes what is this default interactivity. Note that this interactivity behavior is not available in the report designer's editor in design mode.

8.1 Drill Down / Drill Up

8.1.1 Feature Description

This feature allows the user to click on any orthogonal value of the chart to drill down through the data, and see an updated chart showing more granular data for this value. What it does is:

- Apply a data filter on the chart dataset on the field represented by the clicked value.
- Change the X serie data definition and grouping for that value
- Change the chart type/subtype (optional)
- Change the chart title (optional)

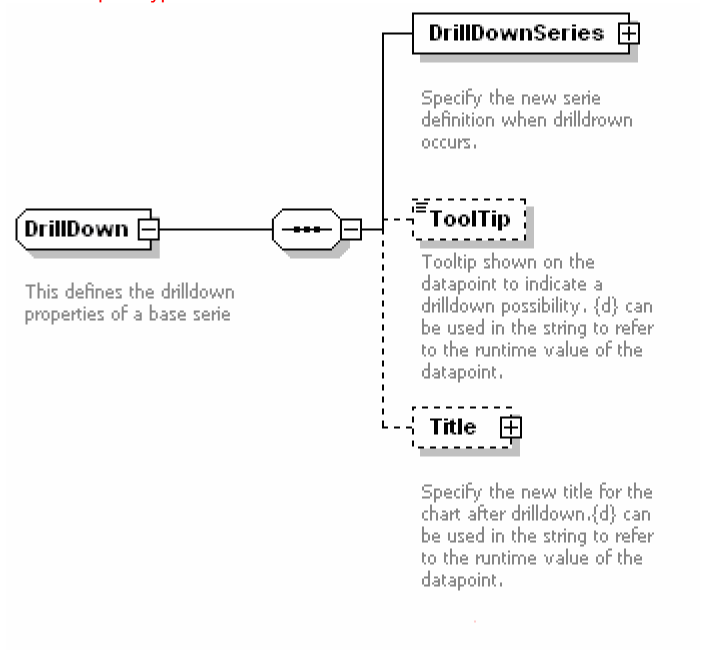
It also allows drilling up to the previous chart, by clicking on a left arrow, at the bottom right of the chart:

8.1.2 Levels

Any number of levels is supported by the engine and model, although the chart builder might allow only one or two levels to keep the UI simple.

8.1.3 Model API Change

```
<xsd:complexType name=”SeriesGrouping”>
        …
        <xsd:element name=”DrillDown” type=”DrillDown” minOccurs=”0” maxOccurs=”1”>
                <xsd:annotation>
                        <xsd:documentation xml:lang="en">
                                Specify the DrillDown properties related to this grouping
                        </xsd:documentation>
                </xsd:annotation>
        </xsd:element>
</xsdcomplexType>
```


DrillDownSeries
Specify the new serie definition when drilldrown occurs.
DrillDown
This defines the drilldown properties of a base serie
ToolTip
Tooltip shown on the datapoint to indicate a drilldown possibility. {d} can be used in the string to refer to the runtime value of the datapoint.
Title
Specify the new title for the chart after drilldown.{d} can be used in the string to refer to the runtime value of the datapoint.

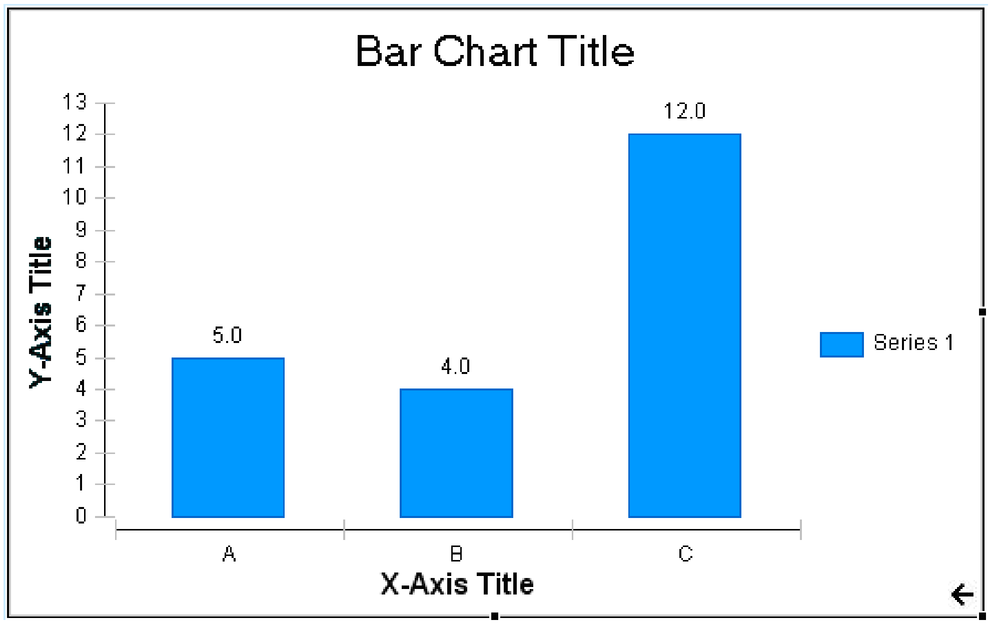
```
<xsd:complexType name="DrillDown">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            This defines the drilldown properties of a base serie
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="DrillDownSeries" type="data:SeriesDefinition">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                    Specify the new serie definition when drilldrown occurs.
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="ToolTip" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Tooltip shown on the datapoint to indicate a drilldown possibility. {d} can be used in
the string to refer to the runtime value of the datapoint.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="Title" type="component:Label" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation xml:lang="en">
                    Specify the new title for the chart after drilldown.{d} can be used in the string to refer to the runtime value
of the datapoint.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
```

8.1.4 Runtime UI

Drilldown/Drillup will be available in SVG, Swing, and SWT. If drilldown is available on a datapoint, a tooltip will appear when hovering the mouse. The user must click on the datapoint he is interested in to drilldown. If the mouseclick event has been overridden with another action, no drilldown will occur.

When the user clicks to drilldown, the chart is regenerated using a new SerieDefinition for the base serie, in order to refine the X data. This could be more granular grouping or use of a different data field. The original grouping and data definition of the base serie will be overridden by this new one. A filter will also be applied to the new chart with the value of the X value of the datapoint. The title of the chart can possibly change during the drilldown.

A drill-up arrow will appear at the right bottom of the chart, so that the user can drill back up to the previous chart. This icon has a "Drill Up" tooltip.



8.1.5 Examples

8.2 Zoom

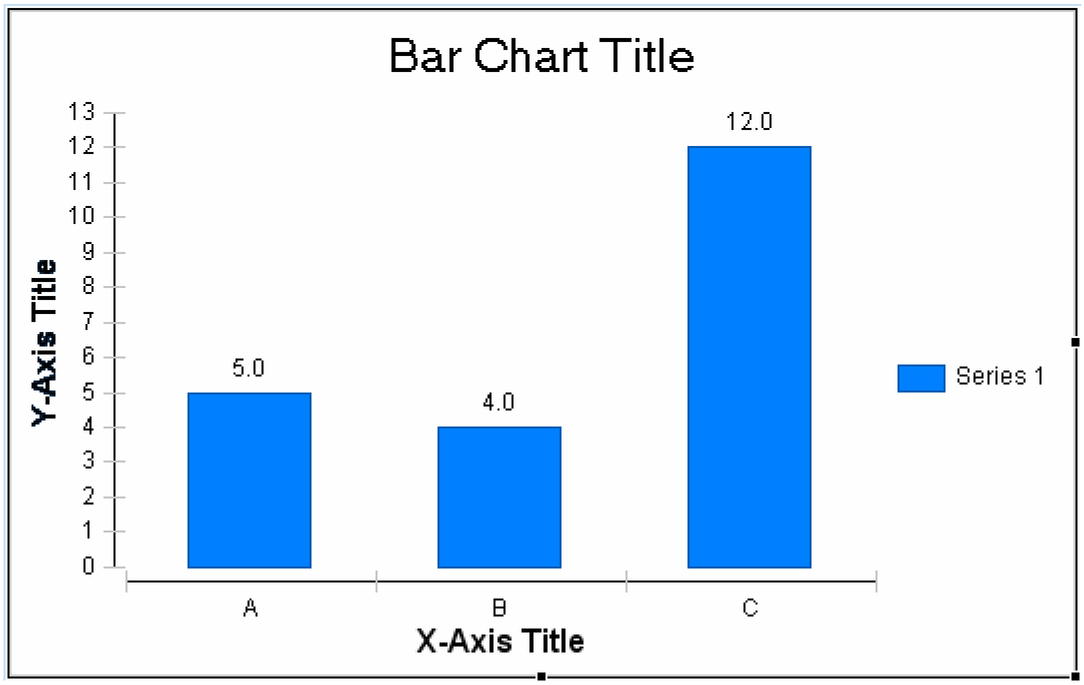
Zoom allows the user to select ranges he is interested in on the X and Y Axes. To do this, he simply clicks and drags the mouse anywhere on the chart: that will dynamically shade a square, while changing the mouse cursor to a magnifier icon. When the mouse is released, the chart will be regenerated using a new scale for X and Y axes defined by the square. It is possible to do several zooms sequentially.

To revert to the original chart, the user can click on the zoomout icon that is shown on the bottom right corner of the chart. This icon has a "Undo Zoom" tooltip.

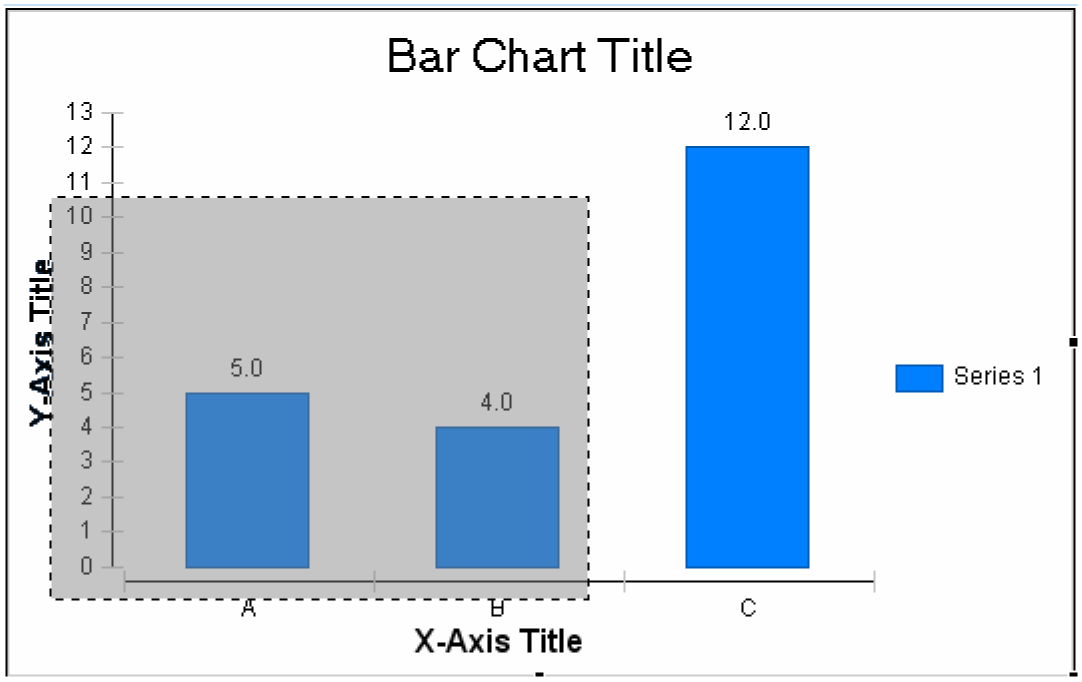
8.2.1 Zoom Mockups

Here are some mockups showing how the zoom works:

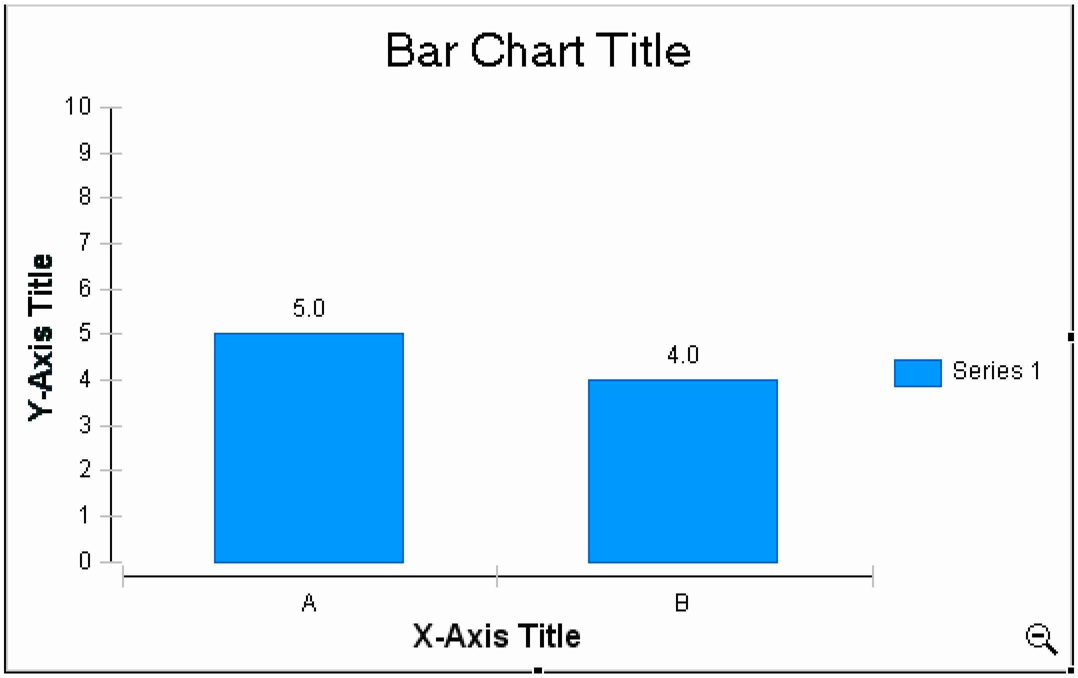
1 - Original Chart:



2 - User selects a square, by click and drag:



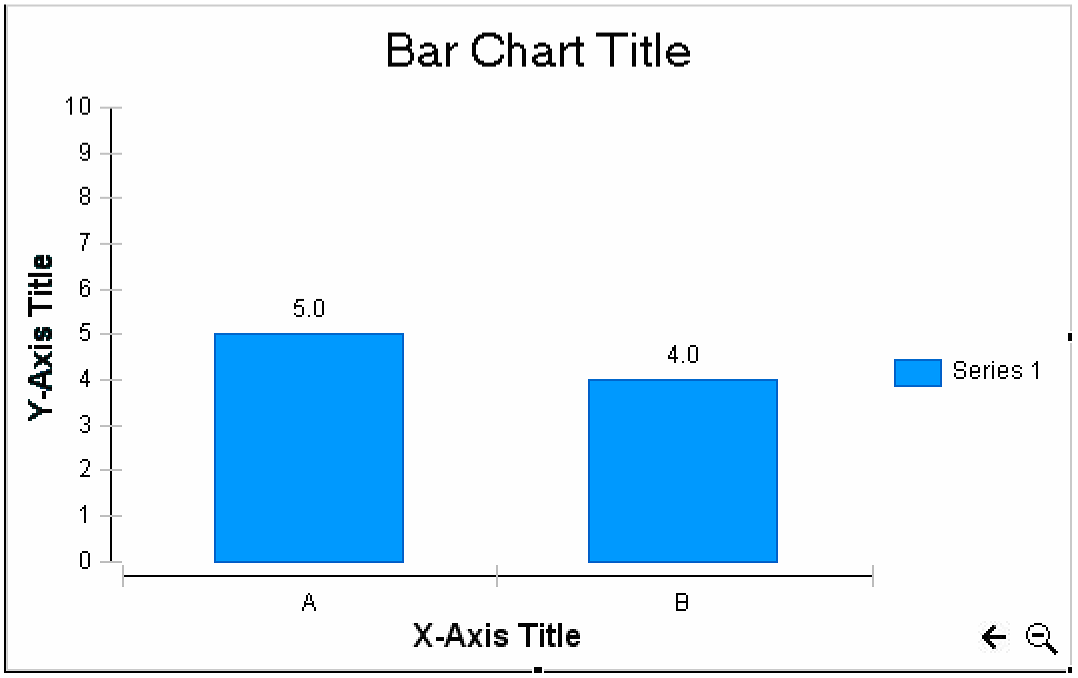
3 - User releases the mouse, a new chart is generated:



8.2.2 Zoom/Drilldown combination

It is possible to drilldown and then zoom. However zoom followed by a drilldown will remove the zoom in the drilled down chart. The drill up will restore the zoom as it was when the drilldown occurred.

When both a drilldown and zoom are active at the same time, two icons are shown at the right bottom of the chart, one to remove the zoom, one to drillup. Here is how it looks like:



8.3 Toolbar

A toolbar will be displayed over the chart to allow several actions on the chart. Note that limitations can apply in some environments, in such case, some menu items might be hidden.

In the Browser environment, no toolbar action will reload the page. If the action requires the chart engine on the server to regenerate the chart, only the chart element should be updated on the page by an asynchronous JavaScript call.

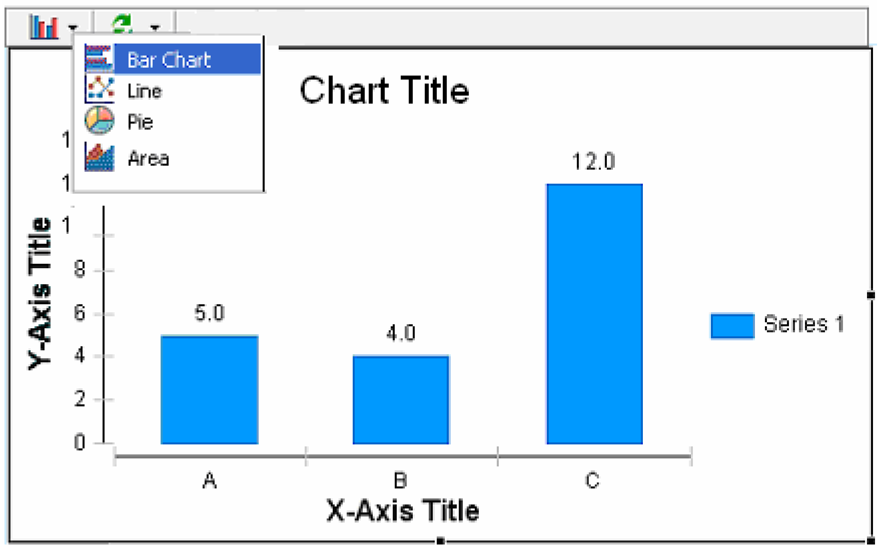
8.3.1 Contents Overview

Here is a short tree describing the Toolbar structure:

- Chart Type <dropdown button>
  - ✓ Bar
  - Line
  - Area
  - Pie
- Refresh <dropdown button>
  - Refresh Chart Now
  - Automatic Refresh
    - ✓ Off
    - 5s
    - 10s
    - 30s
    - 1min
    - 5min
- CustomLabel: <combobox>
  - <All> (default)
  - FieldValue1
  - FieldValue2
  - …

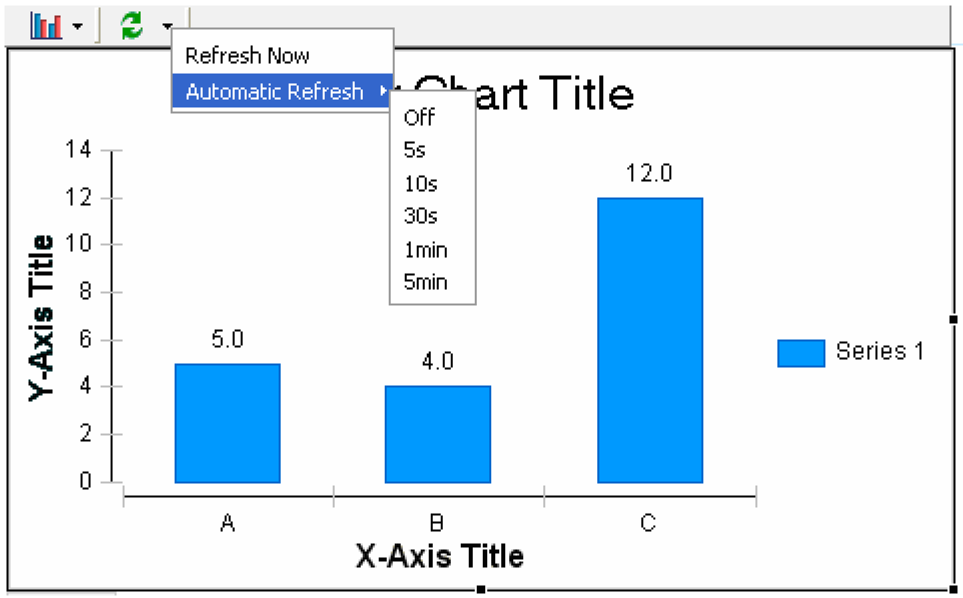
8.3.2 Chart Types

The top menu will allow changing the type of the chart. It will ask the chart engine to regenerate it. Only bar/line/pie/area types will be available. Custom types, stock and scatter won't be available through this menu, as they have a different databinding model. This entry will only appear if the chart is of any of those types.



8.3.3 Refresh

The refresh menu will request an updated chart based on the current data. This can be set as automatic for predefined amounts of time.

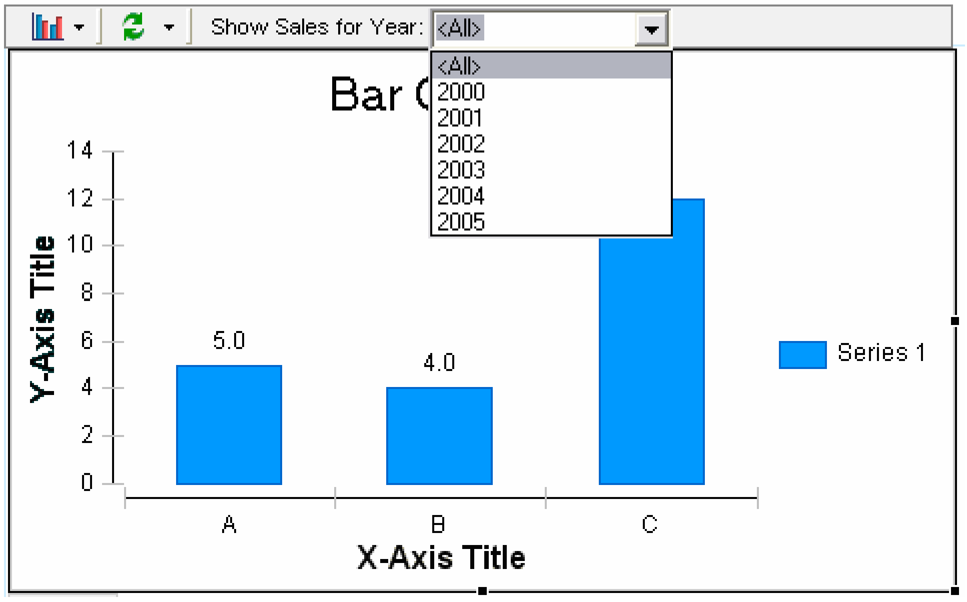


8.3.4 Filter by

This shows a label and a combobox, where the user can select a value he would like to use as a filter for the chart. Here is an example:

Show Sales For Year: <combobox>

When selecting a value, the chart is regenerated with the filter. To cancel the filter, the user simply chooses <All>, which is the first and default value.

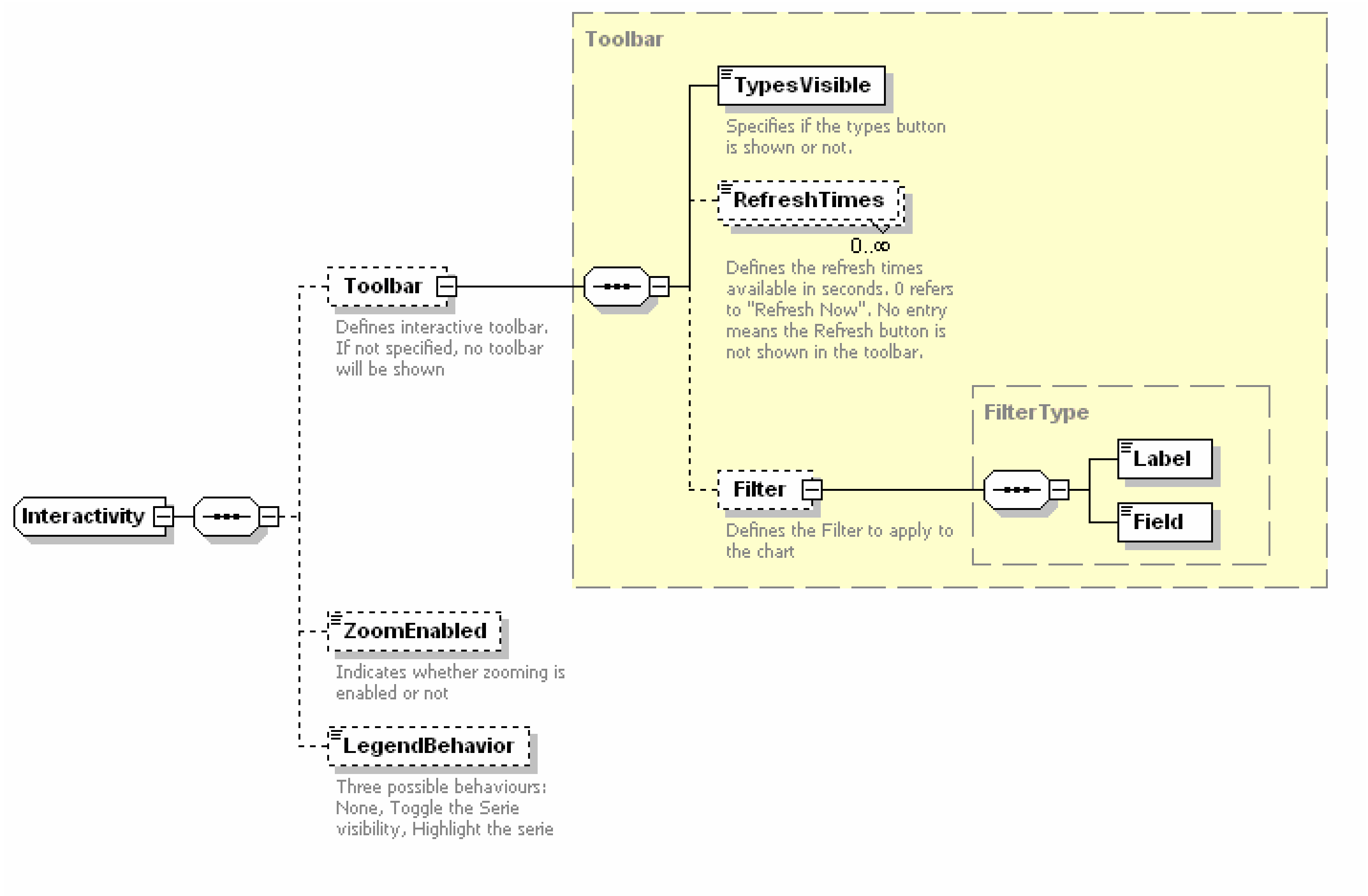


8.4 Legend Interactivity

Legend interactivity responds to mouse left clicks. It can either toggle the visibility of the serie it maps to, or highlight it. This is set at design time by the user (see Configuration API)

8.5 Configuration API

The Chart class will have a new optional Interactivity attribute. This Interactivity holds various configuration properties of the interactivity features.


Toolbar
TypesVisible
Specifies if the types button is shown or not.
RefreshTimes
0..∞
Defines the refresh times available in seconds. 0 refers to "Refresh Now". No entry means the Refresh button is not shown in the toolbar.
Toolbar
Defines interactive toolbar. If not specified, no toolbar will be shown
FilterType
Label
Field
Filter
Defines the Filter to apply to the chart
Interactivity
ZoomEnabled
Indicates whether zooming is enabled or not
LegendBehavior
Three possible behaviours: None, Toggle the Serie visibility, Highlight the serie

```
<xs:complexType name="Interactivity">
          <xs:sequence>
               <xs:element name="Toolbar" type="Toolbar" minOccurs="0">
                    <xs:annotation>
                         <xs:documentation>Defines interactive toolbar. If not specified, no toolbar will be
shown</xs:documentation>
                    </xs:annotation>
               </xs:element>
               <xs:element name="ZoomEnabled" type="xs:boolean" minOccurs="0">
                    <xs:annotation>
                         <xs:documentation>Indicates whether zooming is enabled or not</xs:documentation>
                    </xs:annotation>
               </xs:element>
               <xs:element name="LegendBehavior" type="LegendType" minOccurs="0">
                    <xs:annotation>
                         <xs:documentation>Three possible behaviours: None, Toggle the Serie visibility, Highlight the
serie</xs:documentation>
                    </xs:annotation>
               </xs:element>
          </xs:sequence>
     </xs:complexType>
     <xs:complexType name="Toolbar">
          <xs:sequence>
               <xs:element name="TypesVisible" type="xs:boolean">
                    <xs:annotation>
                         <xs:documentation>Specifies if the types button is shown or not.</xs:documentation>
                    </xs:annotation>
               </xs:element>
               <xs:element name="RefreshTimes" type="xs:unsignedInt" minOccurs="0" maxOccurs="unbounded">
                    <xs:annotation>
                         <xs:documentation>Defines the refresh times available in seconds. 0 refers to "Refresh Now". No entry
means the Refresh button is not shown in the toolbar.</xs:documentation>
```

```
                </xs:annotation>
            </xs:element>
            <xs:element name="Filter" type="FilterType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Defines the Filter to apply to the chart</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="FilterType">
        <xs:sequence>
            <xs:element name="Label" type="xs:string"/>
            <xs:element name="Field" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="TypesType"/>
    <xs:complexType name="RefreshType"/>
    <xs:complexType name="LegendType">
        <xs:simpleContent>
            <xs:restriction base="xs:string">
                <xs:enumeration value="ToggleSerieVisibility"/>
                <xs:enumeration value="HighlightSerie"/>
                <xs:enumeration value="None"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
```