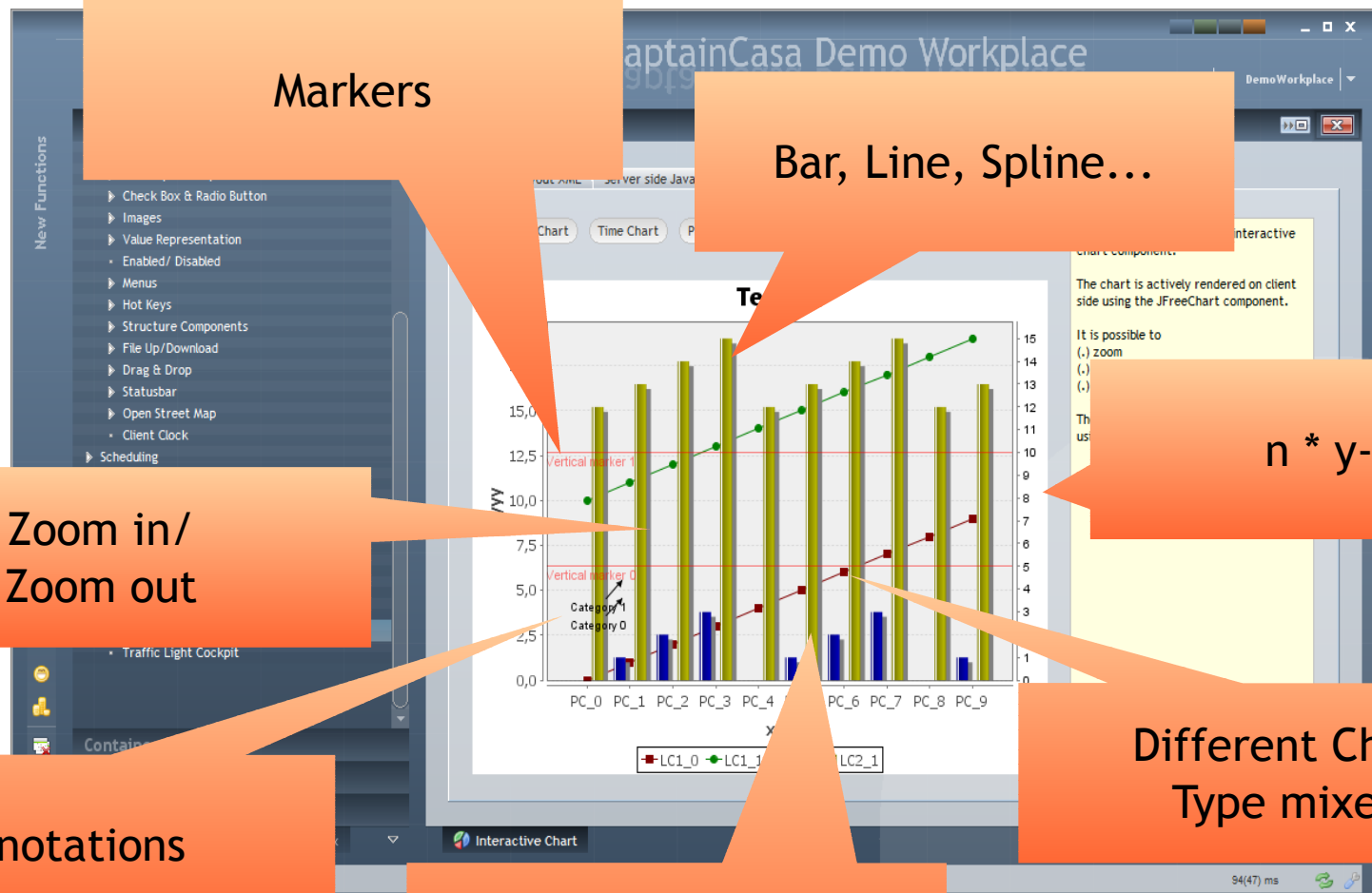


Interactive Charting



Category based Chart I



Category based Chart II

The screenshot displays the CaptainCasa Demo Workplace interface. On the left, a 'Components' panel lists various UI elements, with 'Interactive Chart' selected. The main workspace shows an 'Interactive Chart' component with tabs for 'Demo', 'Layout XML', and 'Server side Java Code'. Under the 'Demo' tab, there are buttons for 'Category Chart', 'Time Chart', 'Pie Chart', and 'XY Chart'. The 'Pie Chart' is active, showing two pie charts labeled 'LC1_0' and 'LC1_1'. Each pie chart is divided into ten segments, labeled PC_0 through PC_9. A legend at the bottom of the chart area maps colors to these labels. On the right, a help panel explains that the CHART component is interactive and rendered on the client side using JFreeChart. It lists capabilities like zooming, positioning a crosshair, and selecting parts of the chart, and notes that chart definitions are simple and use a server-side API.

Interactive Chart

Demo | Layout XML | Server side Java Code

Category Chart | Time Chart | Pie Chart | XY Chart

Test Pie

LC1_0 | LC1_1

Legend: PC_0, PC_1, PC_2, PC_3, PC_4, PC_5, PC_6, PC_7, PC_8, PC_9

The CHART component is an interactive chart component.
The chart is actively rendered on client side using the JFreeChart component.

It is possible to
(.) zoom
(.) position a crosshair
(.) select a part of a chart

The definition of charts is very simple using a server side API.

78(124) ms

Time based Charts

The screenshot shows the CaptainCasa Demo Workplace interface. On the left is a 'Components' sidebar with a tree view including 'Interactive Chart'. The main window displays an 'Interactive Chart' titled 'Test'. The chart has a time-based x-axis labeled 'xxx' with dates from 19-Jul to 22-Jul. The y-axis is labeled 'VVV' and ranges from 0 to 110. There are multiple data series represented by different colored lines and markers. A legend at the bottom identifies series L10 through L20. Annotations include 'This is a marker' pointing to a purple arrow and 'An annotation (24)' pointing to a red line. To the right of the chart is a text box explaining the CHART component and its interactive capabilities.

Interactive Chart

Demos: Layout XML, Server side Java Code

Category Chart | Time Chart | Pie Chart | XY Chart

Test

This is a marker

An annotation (24)

Legend: L10, L11, L20, L21, L22, L20

Y-axis: VVV (0 to 110)

X-axis: xxx (19-Jul to 22-Jul)

Second Chart: 0 to 25.0

Third Chart: 0.0 to 25.0

The CHART component is an interactive chart component.
The chart is actively rendered on client side using the JFreeChart component.

It is possible to
(.) zoom
(.) position a crosshair
(.) select a part of a chart

The definition of charts is very simple using a server side API.

x-Axis is time based

x,y Charts

The screenshot shows the CaptainCasa Demo Workplace interface. On the left is a 'Components' sidebar with a tree view. The main area displays an 'Interactive Chart' component. The chart is titled 'Test XY' and features four data series: L10 (red squares), L11 (green circles), L20 (blue triangles), and L21 (yellow diamonds). The x-axis is numeric, ranging from 0 to 40. The y-axis is labeled 'VWV' and ranges from 0 to 17. Annotations include 'This is a marker' pointing to a red vertical line at x=25, 'This is a marker on the y-Axis (50)' pointing to a red horizontal line at y=50, and 'Annotation 0', 'Annotation 1', and 'Annotation 2' pointing to specific data points. A legend at the bottom identifies the series. The interface also includes a top bar with 'Effects' and 'DemoWorkplace' menus, and a bottom bar with 'Functions', 'Search', and 'Inbox' options.

x-Axis is numeric

Internals...

- Charts are rendered on Client side using the JFreeChart component
- Very simple server side API for feeding chart info
 - Chart Info contains...
 - Charts, chart contains...
 - Lines, line contains...
 - Data Points representing values
- Chart selections („Click“) are passed to the server side together with their corresponding association (chart, line, point)