

TWaver Java Training

Serva Software LLC

TWaver Java Introduction

TWaver™

- **What is TWaver?**
- **Java Swing Basics**
- **TWaver Basics**
 - Hello TWaver!
 - MVC Pattern(data-view/event-driven)
 - Data items / Data models / Views

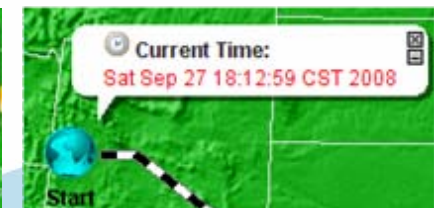
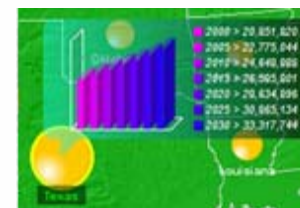
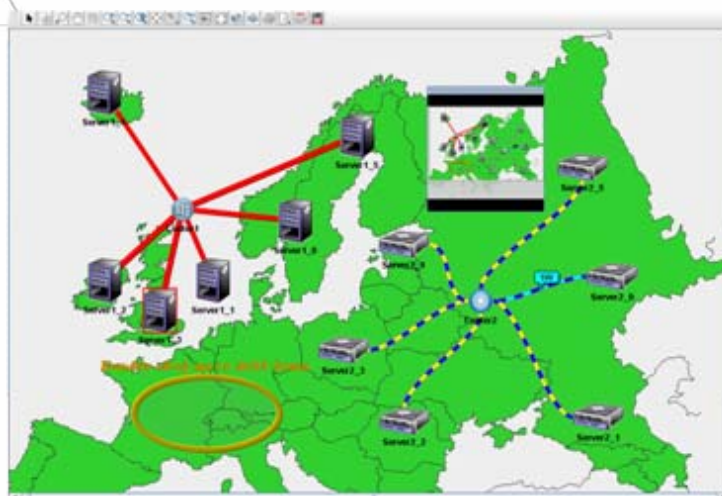
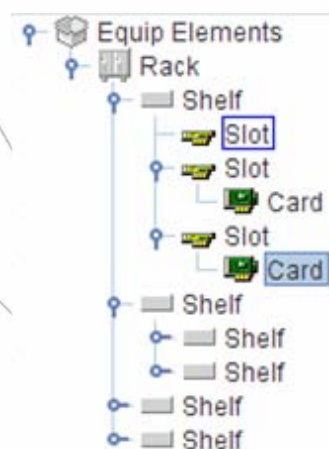
What is TWaver?

- **High efficiency and lightweight graphical component library**
- **Multi-platform solutions**
- **TWaver Java package contents**
- **TWaver license**

TWaver™

A Graphical Component Library

- TWaver focus on graphical display of data
- TWaver is for developers, needs secondary development
- TWaver provides documentation, license, training and technical supports

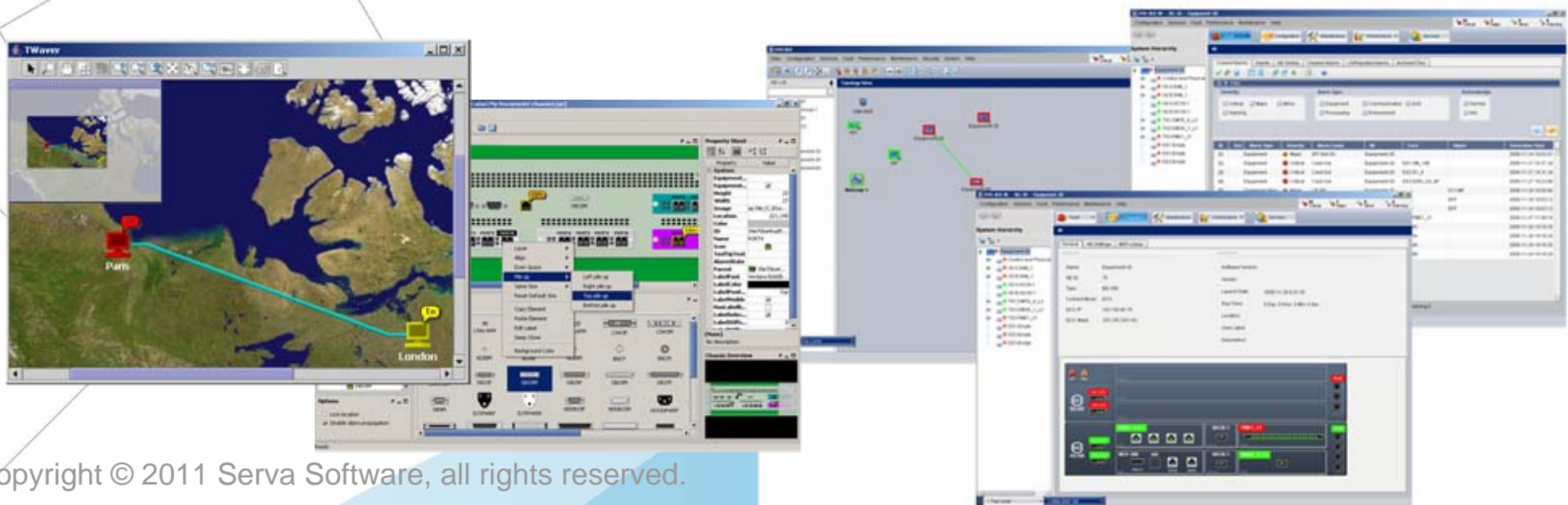


/3Page 1-20/58Row

AlarmID	AlarmSeverity	Acked	ElementID	P
57	Major		Computer9	Loss of multi frame
56	Minor		Computer2	Invalid MSU receive
55	Critical		Computer5	Link failure
54	Warning			High wind
53	Indeterminate			Data set or modem
52	Critical			Battery breakdown

Focus on Telecom Network Management Components

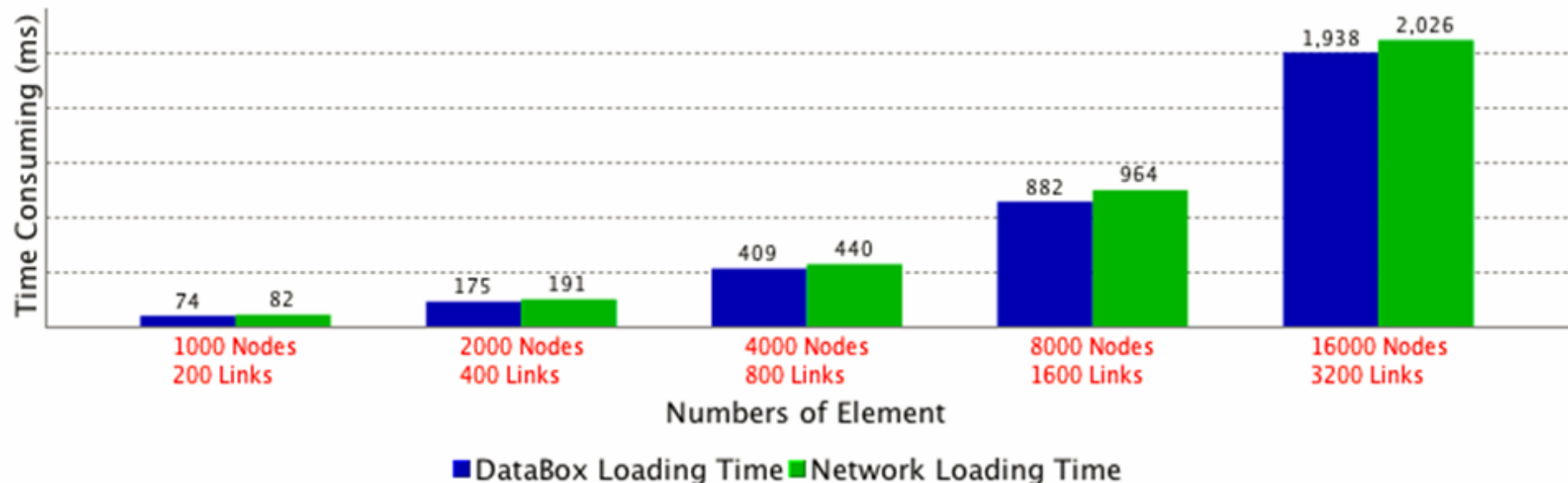
- Provides telecommunications business models
- Have a large number of telecommunication application cases
- But not limited to the telecommunications



High Efficiency and Lightweight

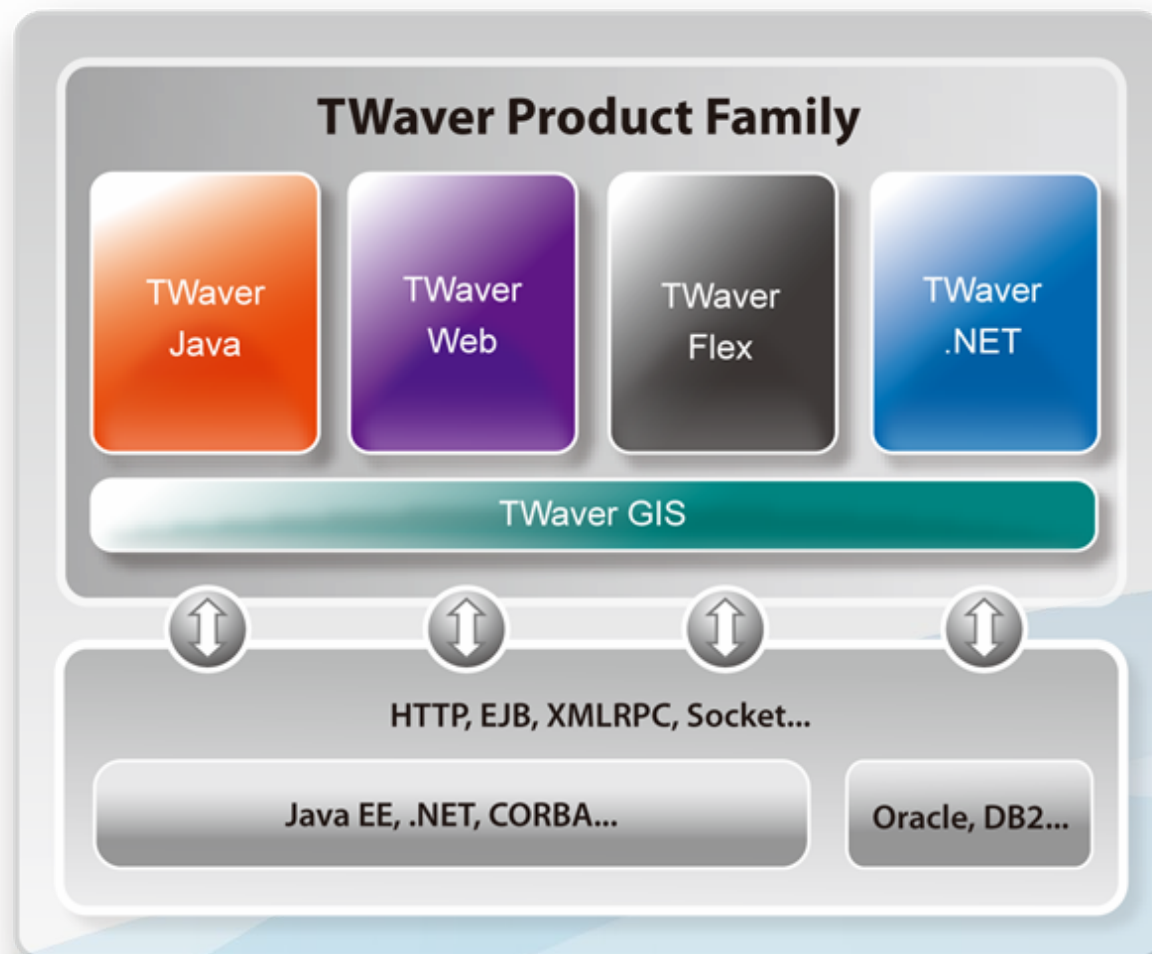
- **twaver.jar 1.8MB**
- **The network component supports ten thousands of nodes and links**
- **The table supports one hundred thousand data**

TWaver Performance Report



Multi-Platform Solutions

- **Java**
- **Web**
- **Flex**
- **.NET**
- **HTML5...**



TWaver™

Customers

TM
waver



TWaver Java Package Contents

TWaver™

twaver-java-3.7	--
demo.bat	4 KB
demo.jar	1.8 MB
demo.sh	4 KB
documents	--
TWaver Java 3.7 Developer Guide.pdf	5.6 MB
twaver-3.7.dtd	4 KB
twaver-beaninfo-3.7.dtd	4 KB
TWaver.template.xml	57 KB
javadoc	--
lib	--
twaver.jar	1.8 MB
License.txt	16 KB
README.txt	4 KB
src	--

demo run files

developer guider

API docs

twaver.jar

demo source

TWaver License

TWaver™

TWaver has three types of license: Evaluation License, Development License, and Runtime License. Click Ctrl+Shift+T to check license information on Swing interface.

- **Evaluation** - used for prophecy on the initial stage or a period of technical selection, with watermark of "TWaver Evaluation Version" on component interface
- **Development** - Grant you the right to use TWaver to develop your software product or project. No watermark shown but a license information dialog will popup for every 2 hours.
- **Runtime** - Used for your final project/product deployment. No watermark, no dialog popup.

How to Use License File

Twaver™

- **“license.dat” is a plain text format file, it contains license information**
- **Used in three ways:**
 - Replace `"/resource/license.dat"` file in `"twaver.jar"` directly
 - Put it under your project source folder `"/resource/license.dat"`
 - Use the following code:
`TwaverUtil.validateLicense("/resource/license.dat");`

Java Swing Basics

TWaver™

- **TWaver and Swing**
- **Event Despatch Thread**
- **LookAndFeel**
- **Java2D Basics**

TWaver and Swing

TNetwork TTree TElementTable ...



TWaver

TWaver™

JTree JPanel JTable ...



JComponent

JFrame

Swing



Container → Window → Frame

Component

AWT

TWaver and Swing

TWaver Components	Inherited from Swing Components
TNetwork	JComponent
TElementTable	JTable
TPropertySheet	JTable
TTree	JTree
***Chart	JPanel
TList	JList
...	...

TWaver™

Event Dispatch Thread

TWaver™

- **Event Dispatch Thread(EDT)**
- **EDT is a queue, the last event has been processed, the next one will begin**
- **Any code that make GUI changes must be invoked in EDT**
- **Don't do heavy works in EDT**
- **How to use: SwingUtilities.invokeLater/invokeAndWait, queuing and queue-jumping**

```
SwingUtilities.invokeLater(new Runnable() {  
    public void run() {  
        sheet.updateTViewUI();  
    }  
});
```

EDT and TWaver

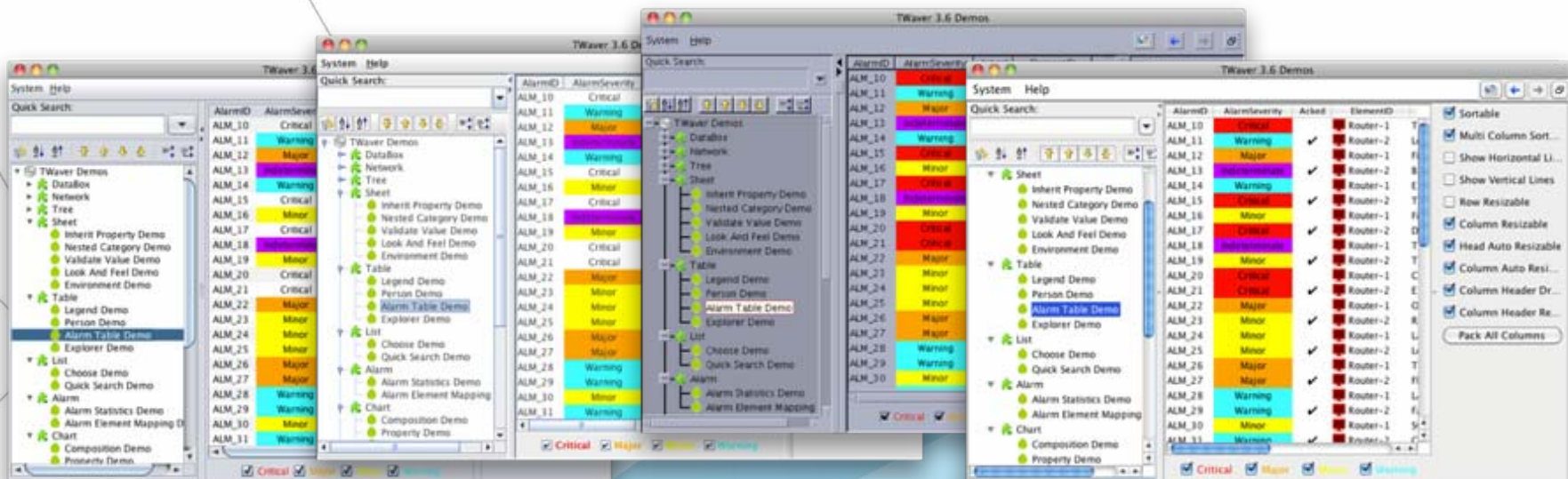
TWaver™

- **For TDataBox, most operations are not thread-safe**
- **If a dataBox associate with a component, any operation on this dataBox must be invoked in EDT**
- **If an element in dataBox, set the properties of the element must invoked in EDT**
- **An element can only be placed in one dataBox**

LookAndFeel

- UIManager manages all default properties of current look and feel
- Use get *** (key) or put (key, value) methods to get or set these properties
- Use setLookAndFeel (...) to set your own look and feel

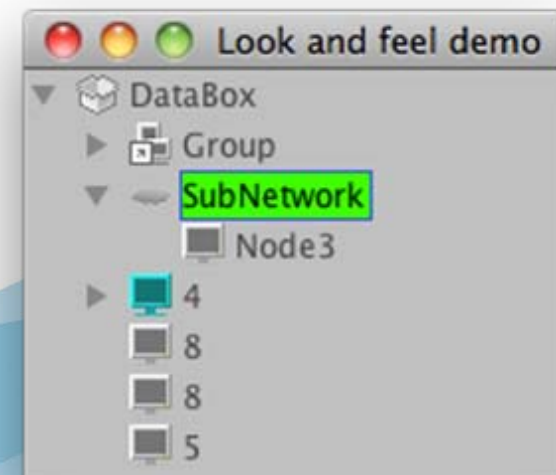
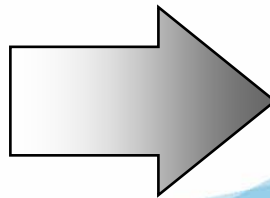
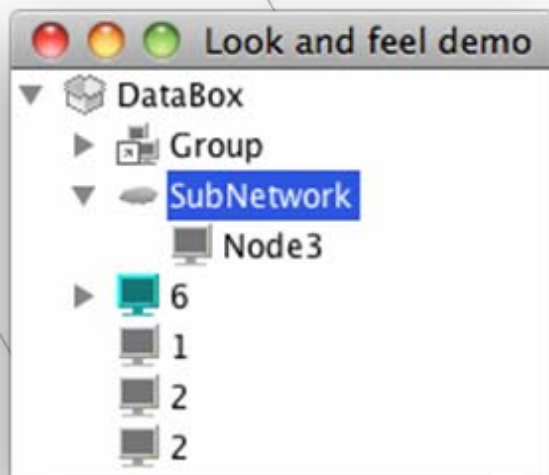
UIManager.setLookAndFeel([com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel](http://www.java.com/javase6/javase6api/ui/lookandfeel/1.0.0/overview.html));



LookAndFeel

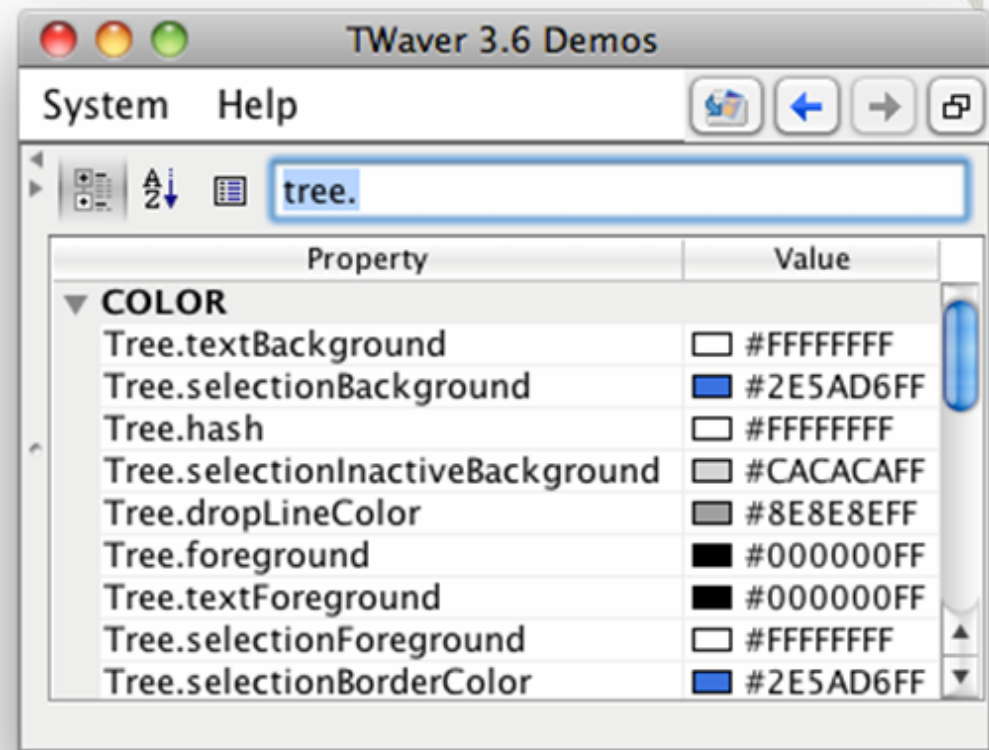
```
UIManager.put("Tree.textForeground", Color.DARK_GRAY);  
UIManager.put("Tree.textBackground", Color.LIGHT_GRAY);  
UIManager.put("Tree.background", Color.LIGHT_GRAY);  
UIManager.put("Tree.selectionForeground", Color.BLACK);  
UIManager.put("Tree.selectionBackground", Color.GREEN);
```

```
TestUtil.showFrame("Look and feel demo", TestUtil.getTree());
```



LookAndFeel Tool in TWaver Demo

- **LookAndFeelDemo.java**
- **Use sheet component lists all default properties of the look and feel, it is easy for search**
- **See: TWaver Java Demo**



Java2D

Twaver™

- **If you want to custom components, you need some Java2D knowledges**
- **Graphics2D**
- **Shape, Area, Paint, Composite**
- **Stroke, AffineTransform**
- **...**

Java2D Example

```
public class Java2DDemo extends JComponent {  
    protected void paintComponent(Graphics g) {  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);  
        Rectangle bounds = this.getBounds();  
        g2d.translate(bounds.getCenterX(), bounds.getCenterY());  
        g2d.setColor(Color.DARK_GRAY);  
        g2d.setFont(new Font(Font.SANS_SERIF, Font.ITALIC, 18));  
        g2d.drawString("Hello TWaver", -50, 5);  
        g2d.setColor(Color.GRAY);  
        g2d.setStroke(new BasicStroke(5, BasicStroke.CAP_ROUND));  
        g2d.drawRect(-60, -25, 130, 50);  
    }  
    public static void main(String[] args) {  
        TestUtil.showFrame("Java2D Demo", new Java2DDemo());  
    }  
}
```



TWaver™

TWaver Basics

TWaver™

- **Hello TWaver**
- **MVC design pattern (data-view/event-driven)**
- **Data items and data models**
- **Views**

Hello TWaver

TWaver Java development environment

- **twaver.jar**
- **JDK 1.4+**
- **Eclipse, NetBeans, JBuilder ...**

TWaver™

Hello TWaver

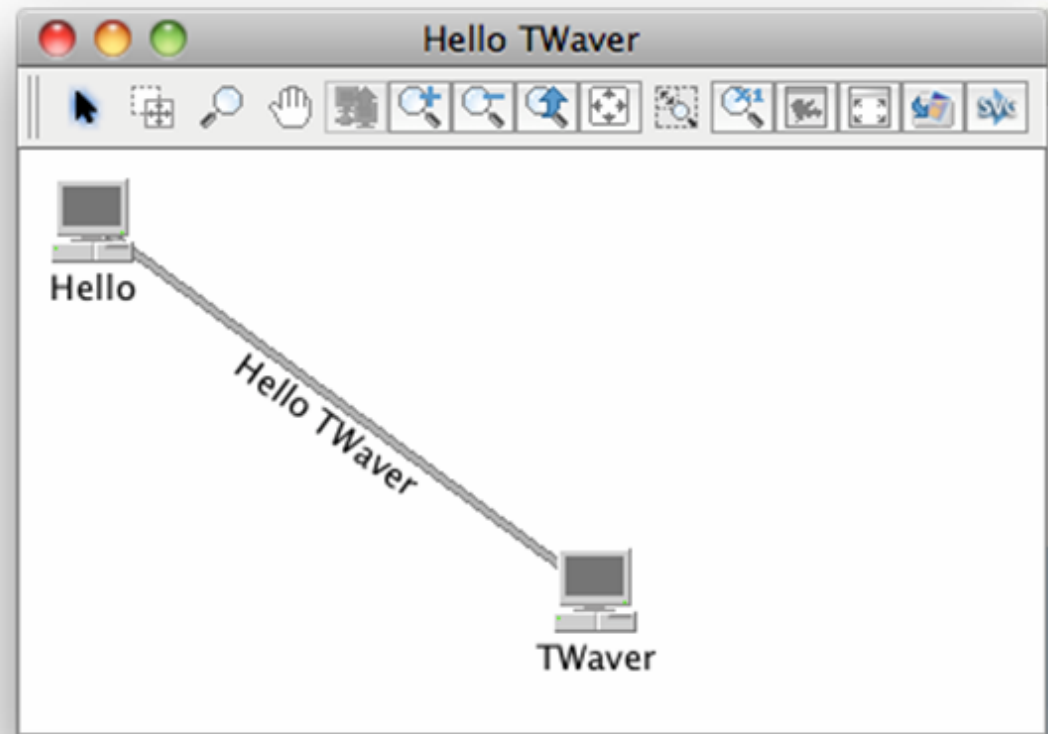
```
TDataBox box = new TDataBox();  
Node node = new Node();  
node.setName("Hello");  
node.setLocation(10, 10);  
box.addElement(node);
```

```
Node node2 = new Node();  
node2.setName("TWaver");  
node2.setLocation(200, 150);  
box.addElement(node2);
```

```
Link link = new Link(node, node2);  
link.setName("Hello TWaver");  
link.putLinkLabelRotatable(true);  
box.addElement(link);
```

```
TNetwork network = new TNetwork(box);
```

```
showFrame("Hello TWaver", network);
```



TWaverTM

Adding Tree, Sheet, Table

TWaver™

```
TNetwork network = new TNetwork(box);
```

```
TTree tree = new TTree(box);
```

```
TPropertySheet sheet = new TPropertySheet(box);
```

```
TElementTable table = new TElementTable(box);
```

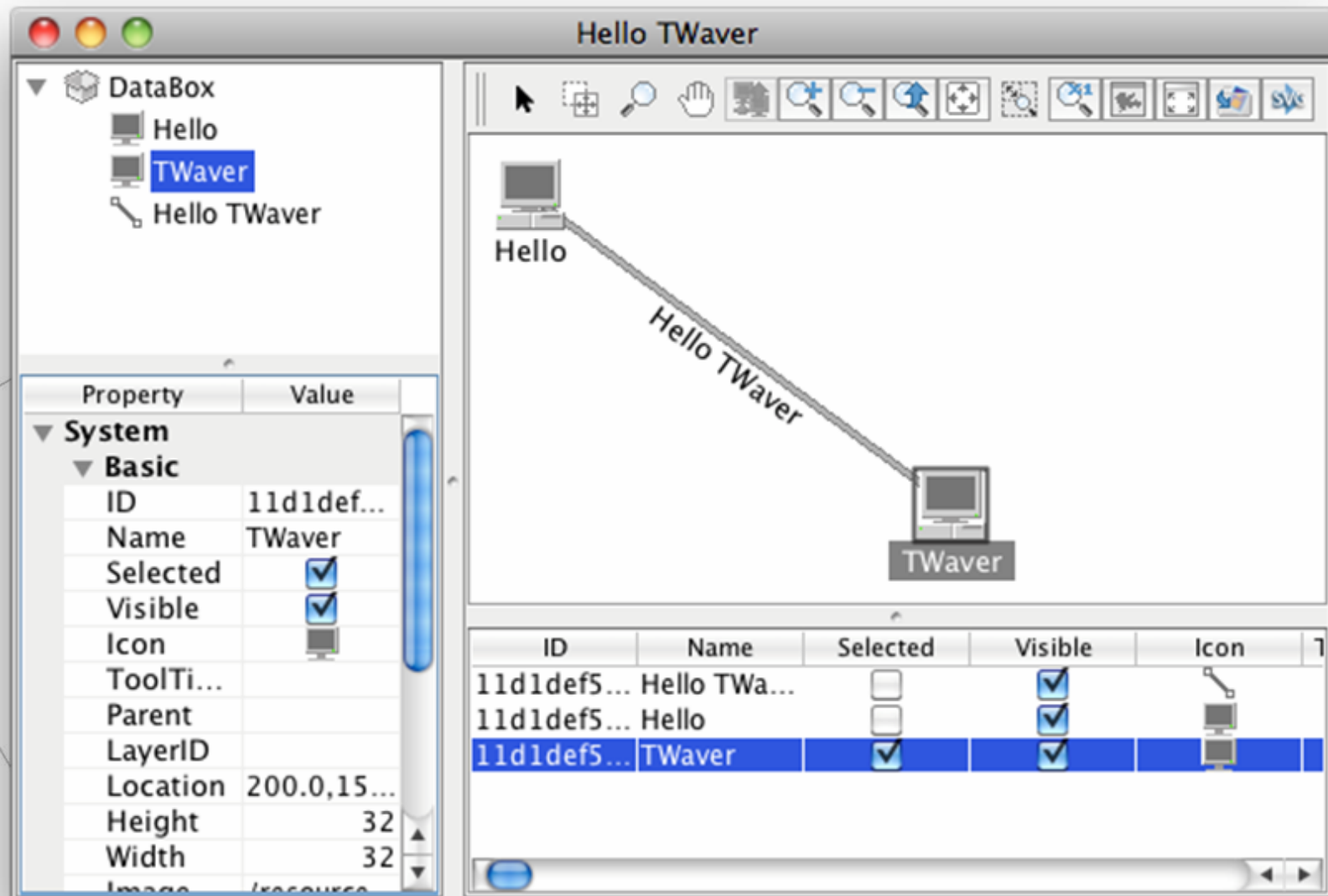
```
table.setElementClass(Element.class);
```

```
JScrollPane tablePanel = new JScrollPane(table);
```

```
JScrollPane sheetPanel = new JScrollPane(sheet);
```

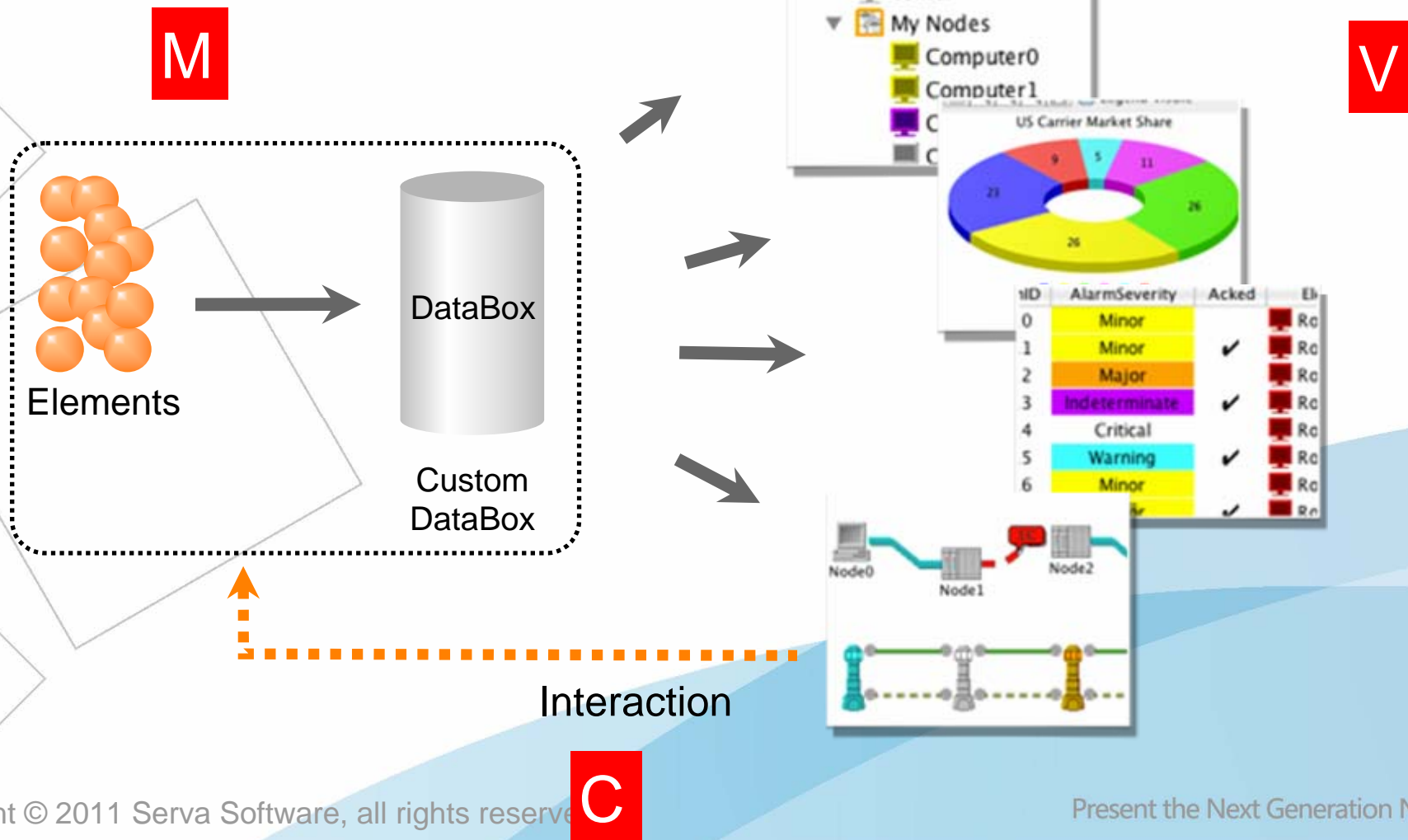
Hello TWaver

TWaver™



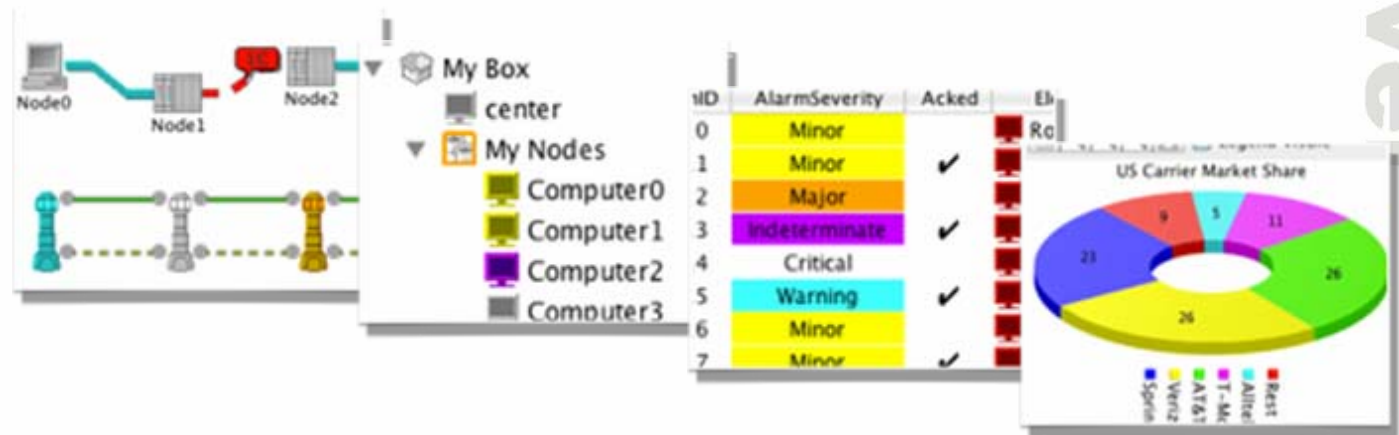
MVC Design Pattern

TWaver™



Model & Views

Views(V)



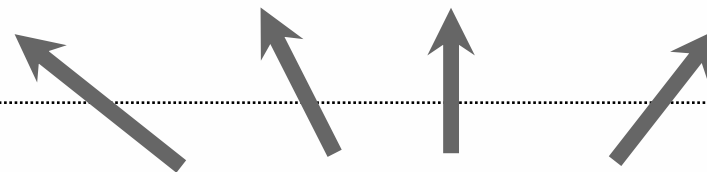
Model(M)

Element

Node
Link
Group
...



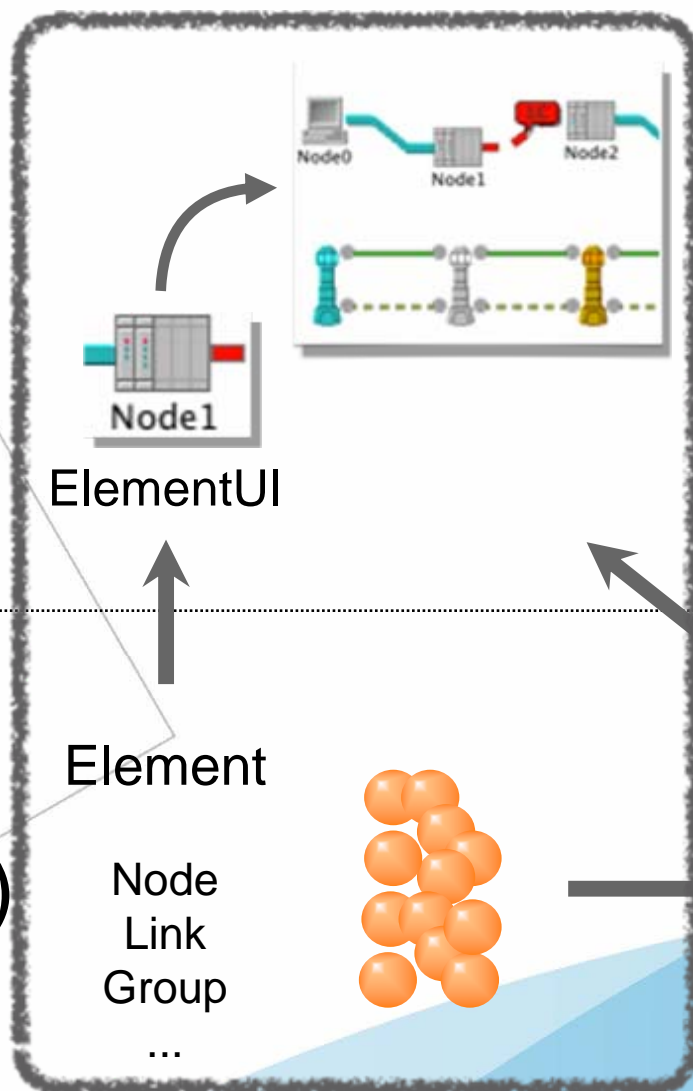
TDataBox



TMwaveTM

Model & Views

Views(V)

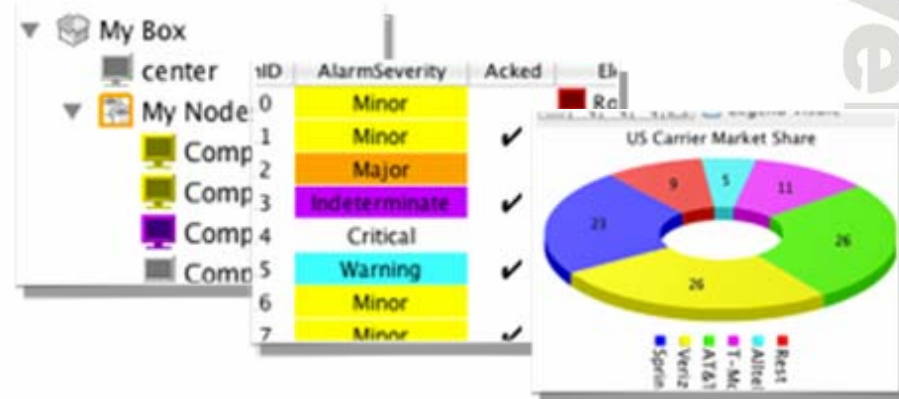


ElementUI

Element

Model(M)

Node
Link
Group
...



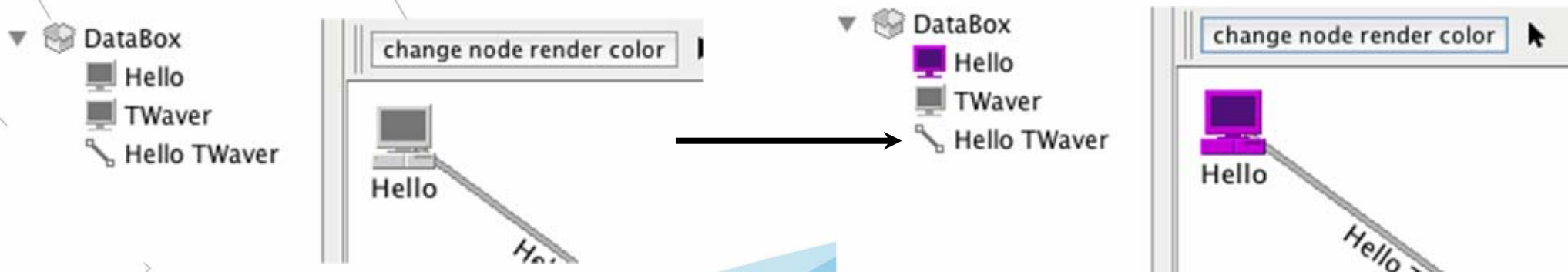
TDatBox

Model & Views

TWaver™

```
JBButton changeRenderColorButton = new JBButton("change node render color");  
changeRenderColorButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        node.putRenderColor(TWaverUtil.getRandomColor());  
    }  
});  
network.getToolBar().add(changeRenderColorButton, 0);
```

all views will update automatically
when data is modified



Event-Driven

Twaver™

Element

`node.putRenderColor(Color.RED)`

`firePropertyChange`

DataBox

`elementPropertyChangeListeners`

`fireElementPropertyChange`

View

network, tree, chart ...

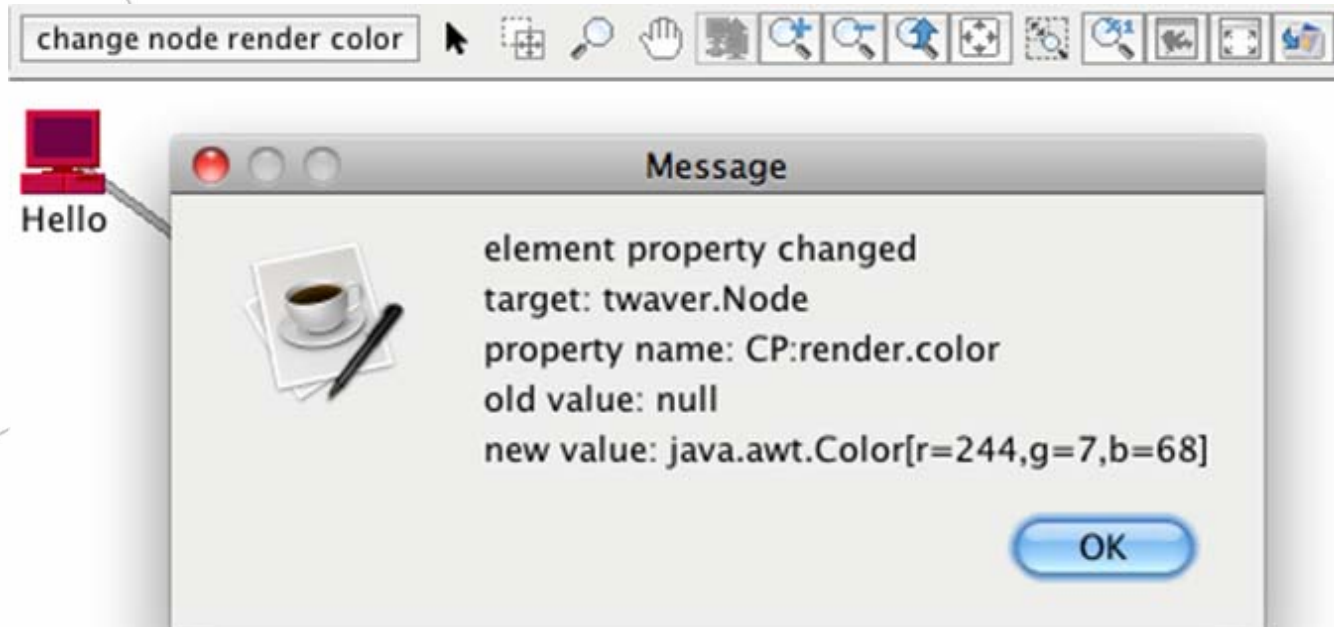
`dataBoxPropertyChangeListener`

`repaint(element)`



Event Listeners

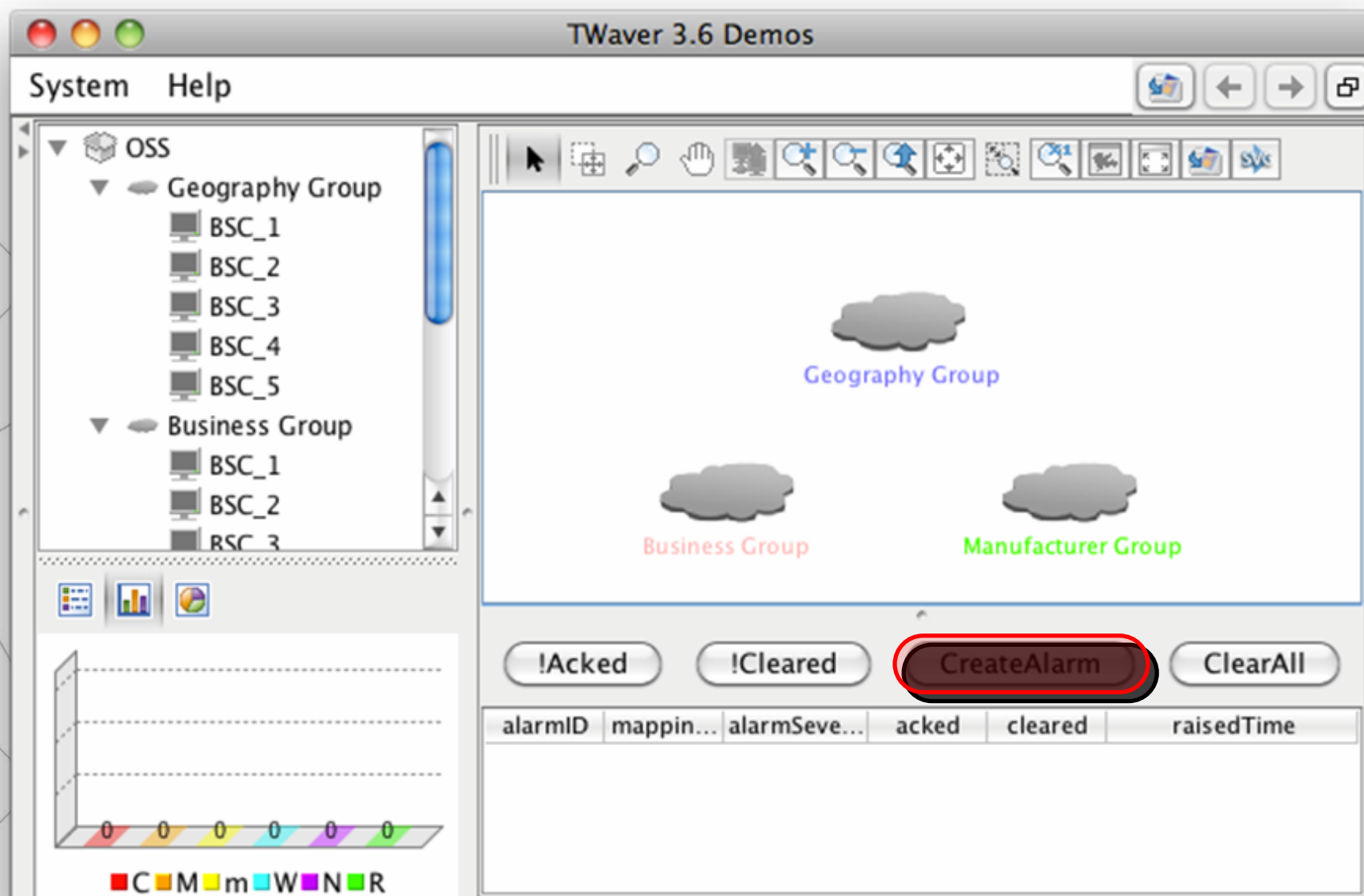
```
box.addElementChangeListener(new PropertyChangeListener() {  
    public void propertyChange(PropertyChangeEvent evt) {  
        JOptionPane.showMessageDialog(network,  
            "element property changed\ntarget: " + evt.getSource() +  
            "\nproperty name: " + evt.getPropertyName() +  
            "\nold value: " + evt.getOldValue() +  
            "\nnew value: " + evt.getNewValue());  
    }  
});
```



Event Listeners

Model	Listener
TDataBox	DataBoxListener DataBoxSequenceListener BatchListener
DataBoxSelectionModel	DataBoxSelectionListener
UndoRedoManager	UndoRedoListener
AlarmModel	AlarmModelListener
AlarmSeverity	AlarmSeverityChangeListener
Network	ZoomListener InteractionListener CanvasPaintListener
LayerManager	LayerModelListener
TTable	PageListener

Views Synchronize Update



TWaver™

Views Synchronize Update

TWaver 3.6 Demos

System Help

OSS

- Geography Group**
 - BSC_1
 - BSC_2
 - BSC_3
 - BSC_4
 - BSC_5
- Business Group**
 - BSC_1
 - BSC_2
 - BSC_3

Business Group

Manufacturer Group

Geography Group

!Aked **!Cleared** **CreateAlarm** **ClearAll**

alarmID	mappin...	alarmSeve...	acked	cleared	raisedTime
Alarml...	2	Critical	✓		2011-03-11 12:...

1 0 0 0 0 0

■ C ■ M ■ m ■ W ■ N ■ R

TWaver™

Data Item and Data Models

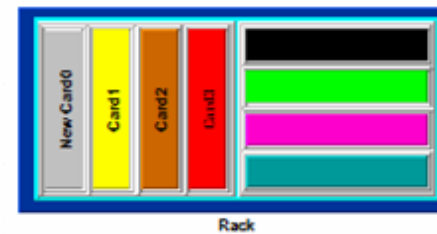
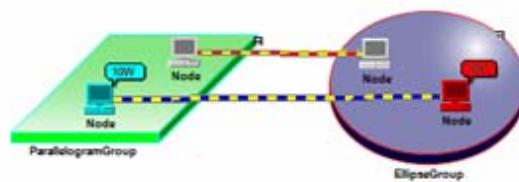
Data Item : The basic unit of data

Data Model : the data management container, provides add/remove/clear operations, can monitor property change of data

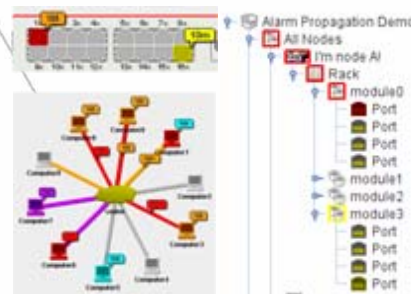
Data Items	Data Models
Element	TDataBox
Alarm	AlarmModel
Layer	LayerModel

Data Types

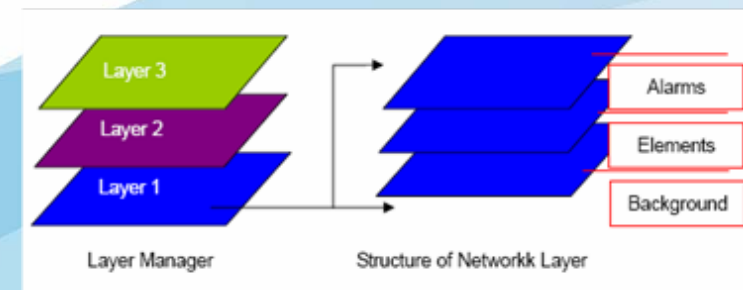
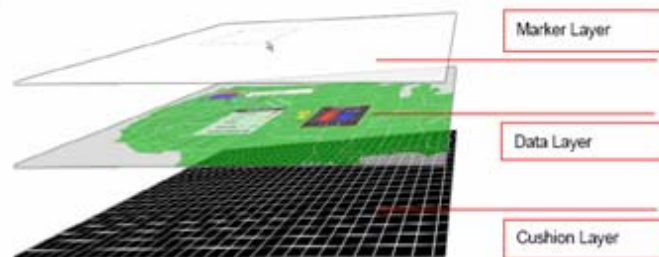
Element (Node / Link / Group / SubNetwork / ...)



Alarm

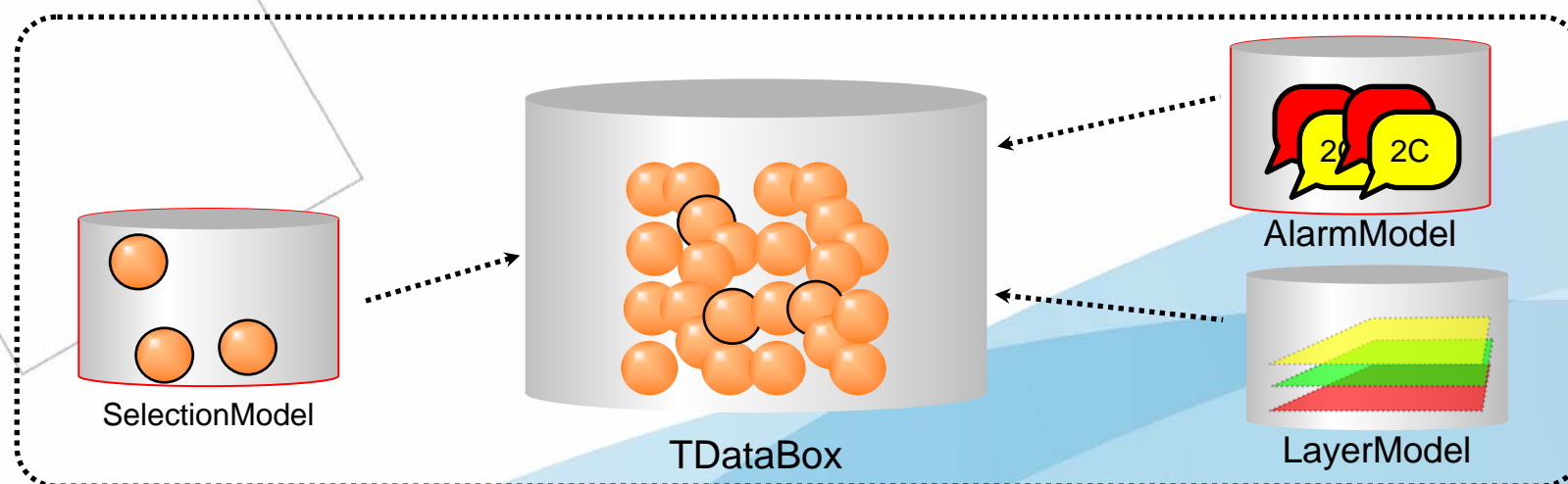


Layer



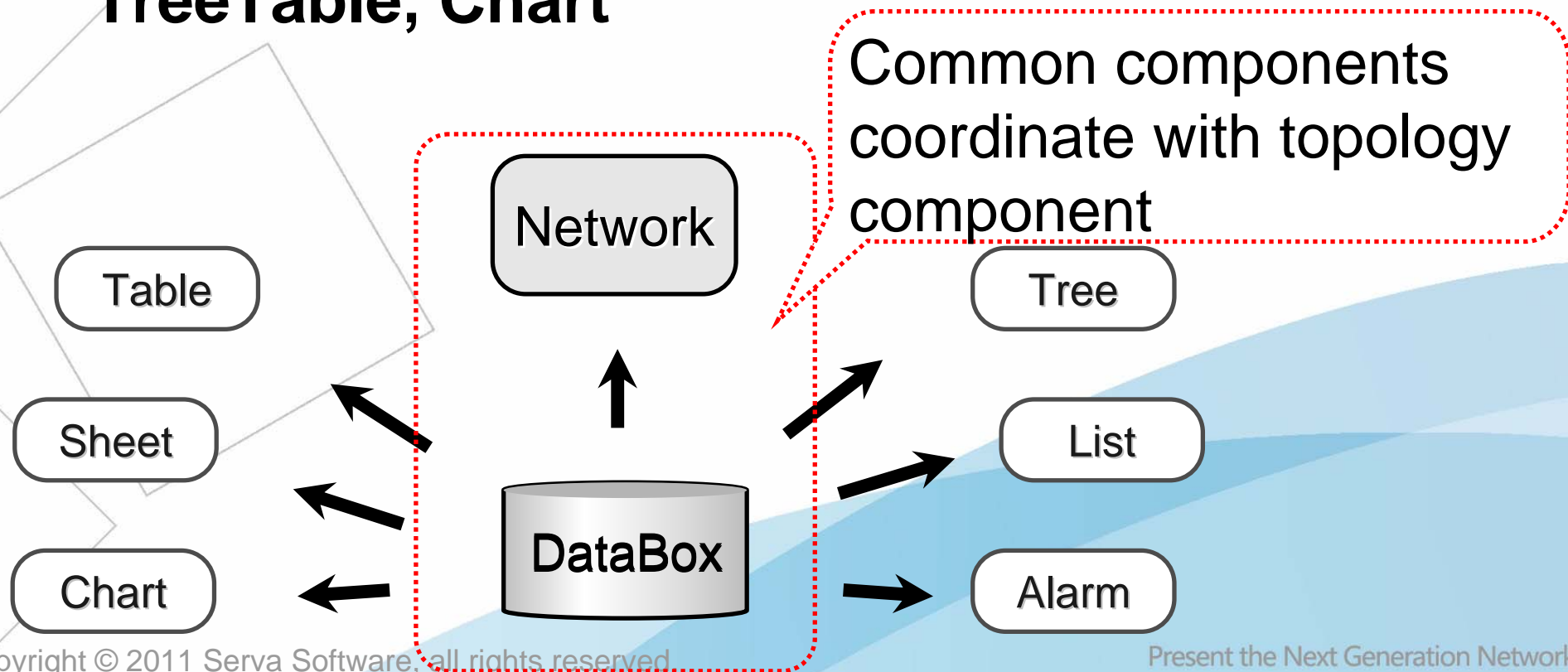
Data Models

- **TDataBox is a elements container, it includes a alarm model, a layer model and a selection model**



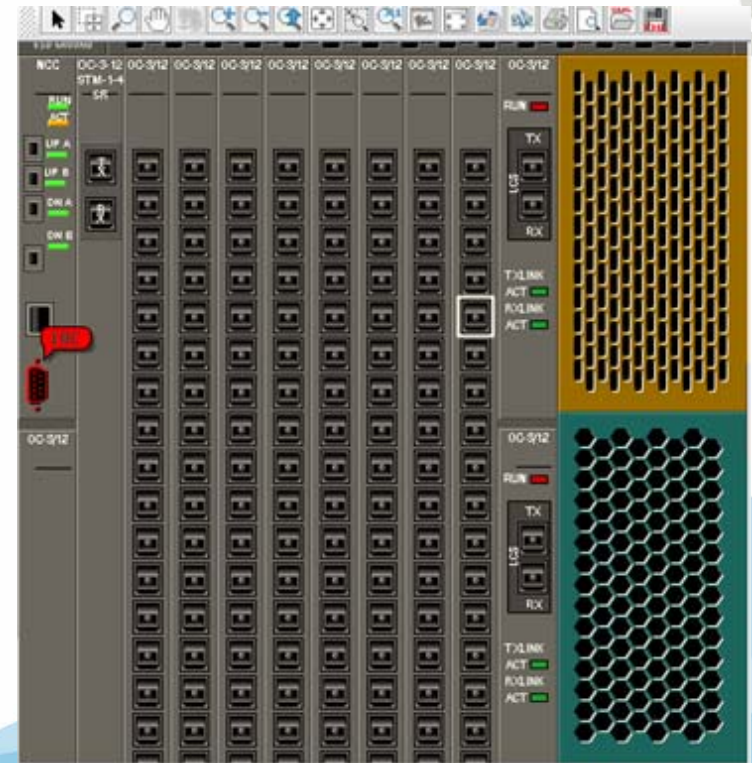
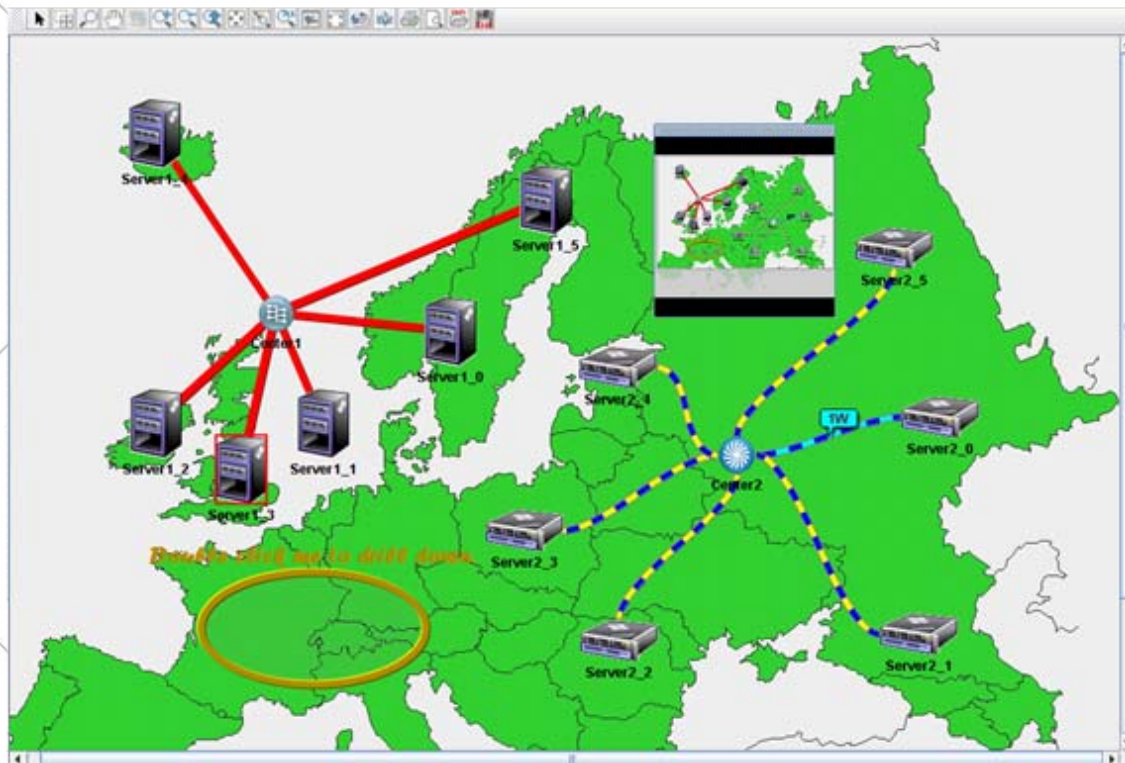
View Types

- **Topology component: Network**
- **Common components: Tree, Table, Sheet, TreeTable, Chart**



Network

Presenting Topology, equipment panels, etc.

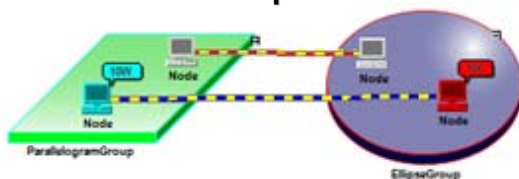


TM
Tmaver

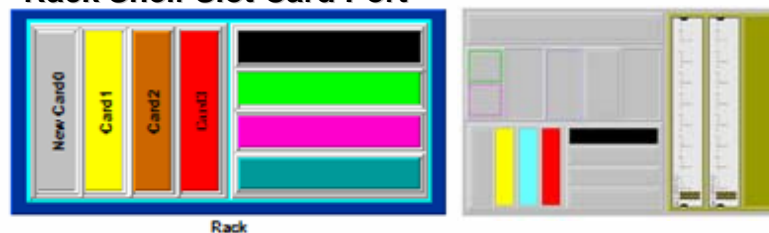
Topology Elements

TWaver™

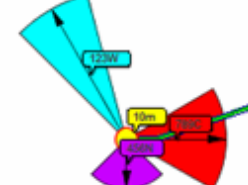
Node Link Group



Rack Shelf Slot Card Port



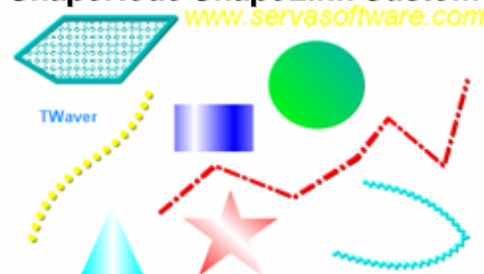
BTS Antenna



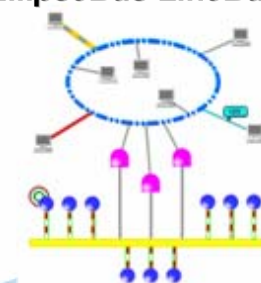
SubNetwork



ShapeNode ShapeLink CustomVector Text



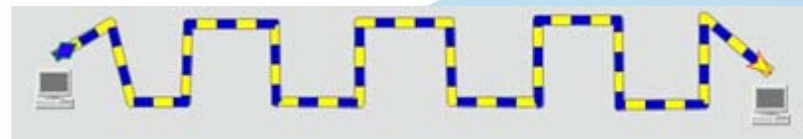
EllipseBus LineBus



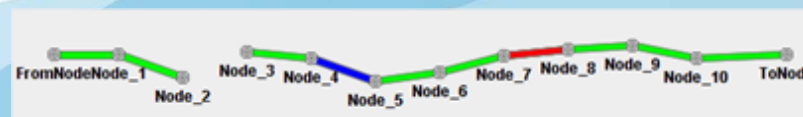
ResizableNode Follower



ShapeLink



Polyline

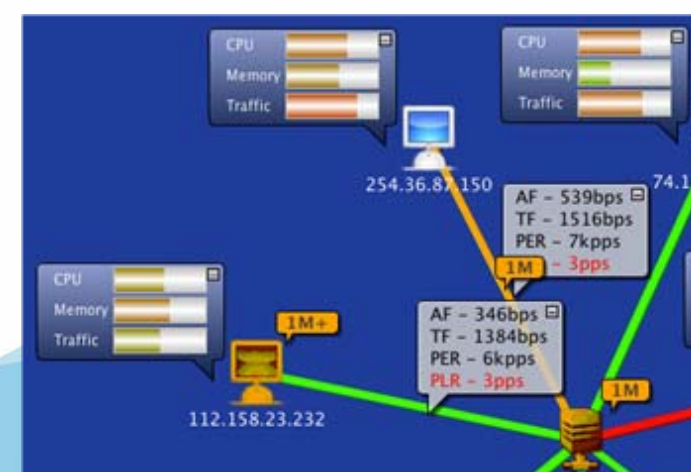
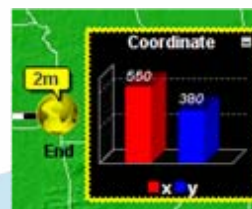
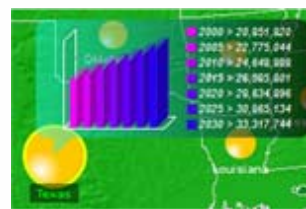
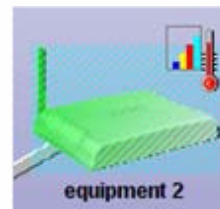
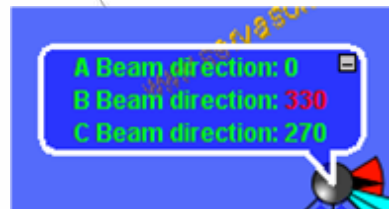
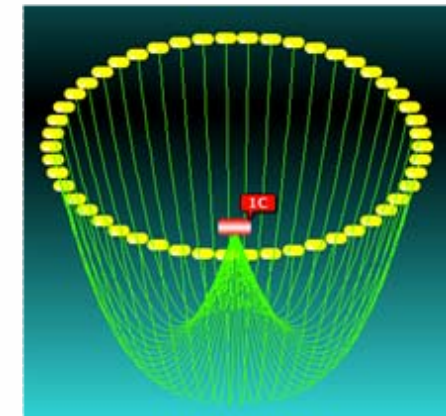


Custom UI

Twaver™

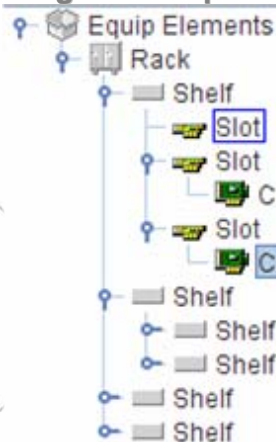


	0	1	2	3	4	5	6	7	8
0	0-0	0-1	0-2	0-3	0-4	0-5	0-6	0-7	0-8
1	1-0	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8
2	2-0	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8
3	3-0	3-1	3-2	3-3	3-4	3-5	3-6	3-7	3-8

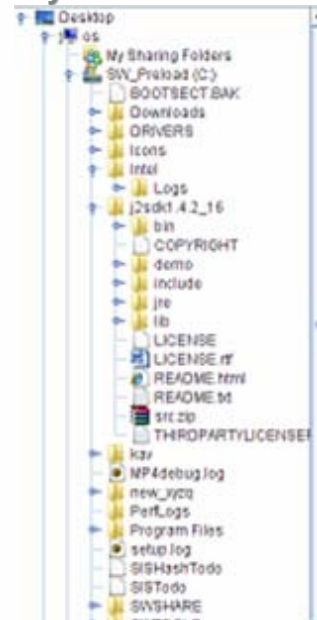


Tree

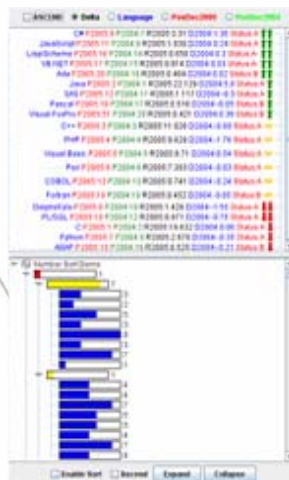
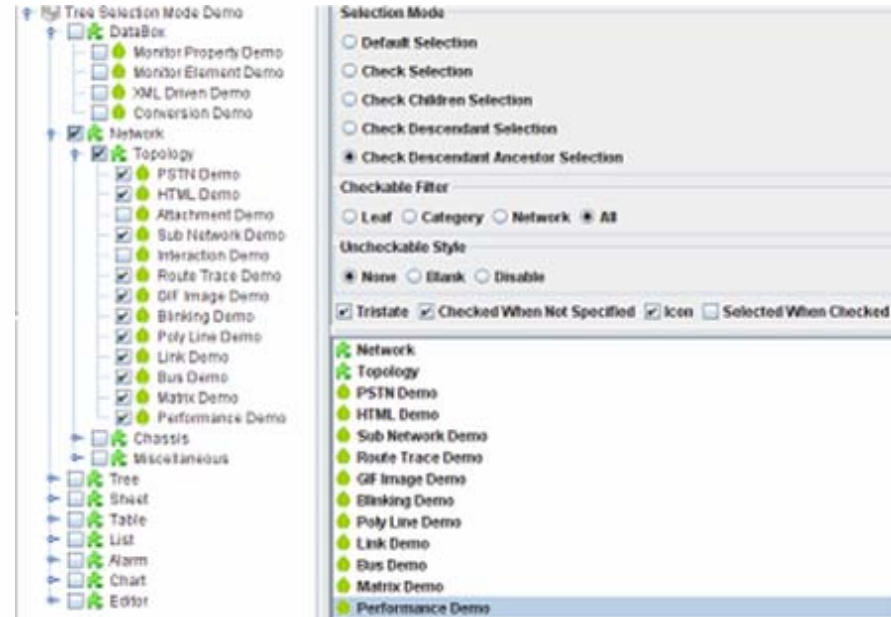
Drag and Drop



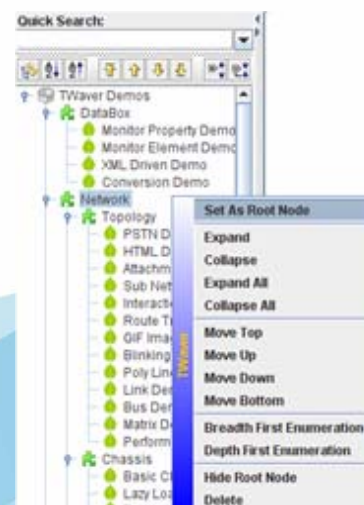
Lazy Load



Check Selection Mode



Compare Sort Custom Icon



Switch Root Node

Move node up/down/top/bottom
Breadth/Depth First Enumeration
Visible Filter

Table

VisibleFilter Multi-Column Sort

First Name	Last Name	Full Name	Age	Birth	First Name	Last Name	Full Name	Salary	Married	Sex	Severity	Description
Aluko	Bin	Aluko Bin	91	1912	Aluko	Bin	Aluko Bin	\$1000.0	<input checked="" type="checkbox"/>	M	Warning	Aluko Bin with monthly salary of \$1000.0
John	Brine	John Brine	91	1914	John	Brine	John Brine	\$1000.0	<input checked="" type="checkbox"/>	M	Warning	John Brine with monthly salary of \$1000.0
Travis	Bunker	Travis Bunker	60	1917	Travis	Bunker	Travis Bunker	\$2000.0	<input checked="" type="checkbox"/>	M	Critical	Travis Bunker with monthly salary of \$2000.0
Celso	Aguiar	Celso Aguiar	84	1923	Celso	Aguiar	Celso Aguiar	\$1000.0	<input checked="" type="checkbox"/>	M	Major	Celso Aguiar with monthly salary of \$1000.0
Cylan	Asha	Cylan Asha	81	1924	Cylan	Asha	Cylan Asha	\$1000.0	<input checked="" type="checkbox"/>	M	Critical	Cylan Asha with monthly salary of \$1000.0
Anil	Batra	Anil Batra	84	1943	Anil	Batra	Anil Batra	\$11000.0	<input checked="" type="checkbox"/>	M	Minor	Anil Batra with monthly salary of \$11000.0
Jason	Boyer	Jason Boyer	42	1947	Jason	Boyer	Jason Boyer	\$1000.0	<input checked="" type="checkbox"/>	M	Warning	Jason Boyer with monthly salary of \$1000.0
Christopher	Brickford	Christopher Brickford	58	1948	Christopher	Brickford	Christopher Brickford	\$1000.0	<input checked="" type="checkbox"/>	M	Critical	Christopher Brickford with monthly salary of \$1000.0
Puneet	Akha	Puneet Akha	48	1959	Puneet	Akha	Puneet Akha	\$1000.0	<input checked="" type="checkbox"/>	M	Critical	Puneet Akha with monthly salary of \$1000.0
Gangadhar	Bathula	Gangadhar Bathula	43	1964	Gangadhar	Bathula	Gangadhar Bathula	\$1000.0	<input checked="" type="checkbox"/>	M	Warning	Gangadhar Bathula with monthly salary of \$1000.0
Art	Americus	Art Americus	42	1967	Art	Americus	Art Americus	\$1000.0	<input checked="" type="checkbox"/>	M	Minor	Art Americus with monthly salary of \$1000.0

TreeTable PackColumn

File	File Size	File Type	Last Modified
Desktop			2008-09-27 18:58:58
My Sharing Folders		System Folder	1970-01-01 08:00:00
SW_Preload (C:)		Local Disk	2008-09-27 21:26:19
BOOTSECT.BAK	8 KB	BAK File	2006-11-10 09:23:37
Downloads		File Folder	2008-09-27 20:56:37
DRIVERS		File Folder	2007-11-27 17:13:35
Icons		File Folder	2007-11-27 16:46:09
Intel		File Folder	2007-11-27 17:29:37
J2sdk1.4.2_16		File Folder	2008-01-28 21:44:28
bin		File Folder	2008-01-28 21:44:12
COPYRIGHT	4 KB	File	2007-09-17 02:21:50
demo		File Folder	2008-01-28 21:43:31
include		File Folder	2008-01-28 21:43:17
jre		File Folder	2008-01-28 21:44:21
lib		File Folder	2008-01-28 21:44:12
LICENSE	19 KB	File	2007-09-17 01:33:46
LICENSE.rtf	20 KB	RTF 格式	2007-09-17 01:33:46
README.html	18 KB	HTML Document	2007-09-17 01:33:46
README.txt	11 KB	Text Document	2007-09-17 01:33:46
src.zip	11,550 KB	WinRAR ZIP archive	2007-09-17 01:33:19
THIRDPARTYLICENSEREADME.txt	10 KB	Text Document	2008-01-28 21:43:16
jav		File Folder	2008-01-28 15:02:07
MP4dubup.log	1 KB	Text Document	2008-02-18 11:44:59
Realtek		File Folder	1970-01-01 08:00:00







Binding AlarmModel Paging Lock

1/3Page 1-20/58Row

Size20

Page1

☒ Lock

AlarmID	AlarmSeverity	Acked	ElementID	ProbableCause	key2	
57	Major		 Computer9	Loss of multi frame	2008-09-27 22:16:28	Th
56	Minor		 Computer2	Invalid MSU received	2008-09-27 22:16:27	Th
55	Critical		 Computer5	Link failure	2008-09-27 22:16:26	Th
54	Warning			High wind	2008-09-27 22:16:25	Th
53	Indeterminate			Data set or modem error	2008-09-27 21:13:37	Th
52	Critical			Battery breakdown	2008-09-27 21:13:36	Th

Sheet

Filter & Sort Properties

Property Sheet

Property Value

System

Equipment... ☐

Equipment... ☒

Height 22

Width 27


Image jar:file:/C:/Doc...

Location 221,190

Color


ID 39e706a46ad5...

Name PORT4

Icon 

ToolTipText

AlarmState

Parent  39e706a4...

LabelFont Verdana Bold/B...

LabelColor

LabelPosit... Top

LabelVisible ☒

HasLabelB... ☐

LabelSelec... ☒

LabelXOffs... 0

(None)

No description.

Nested Category & Property Description

Property Value

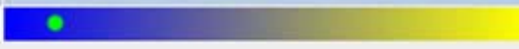
System

GROUP1

PropertyA ABC


PropertyB 123456789@#\$%^&


1980-03-06~2080-04-23

Scale 

GROUP2

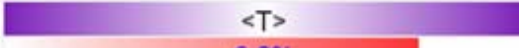
Married ☒


Color  #0A5A33FF

State  STOPPED

City New York

Language 53

Character  <T>

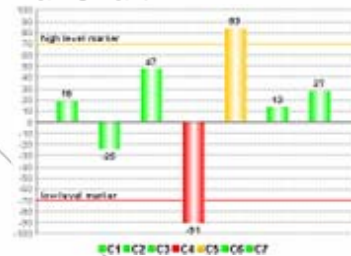
Proportion  0.8%

1980-03-06~2080-04-23:

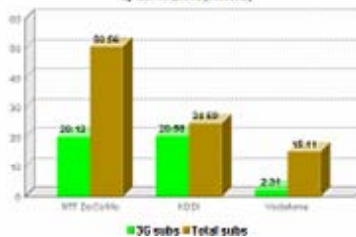
TWaver™

Chart

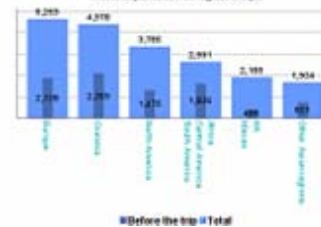
Bar Chart



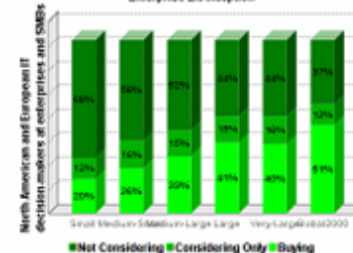
Japan's cellular subs & 3G penetration by carrier, 2005 (millions)



Travel expenditure on leisure trips



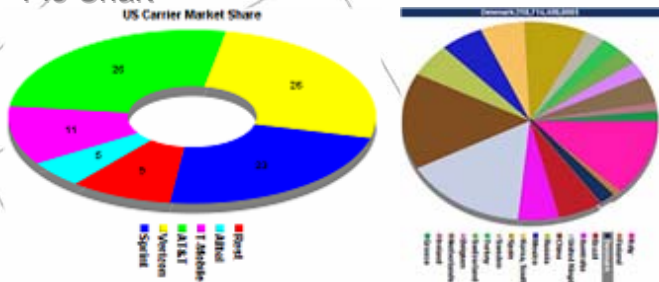
Enterprise 2.0 Adoption



Dial Chart



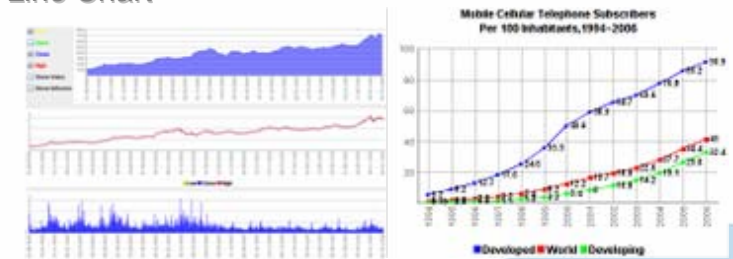
Pie Chart



Radar Chart



Line Chart



Percent Chart



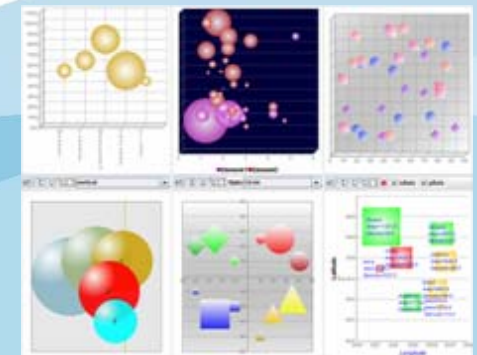
Using Ajax frameworks, toolkits, or libraries:



Using Ajax in conjunction with:



Bubble Chart





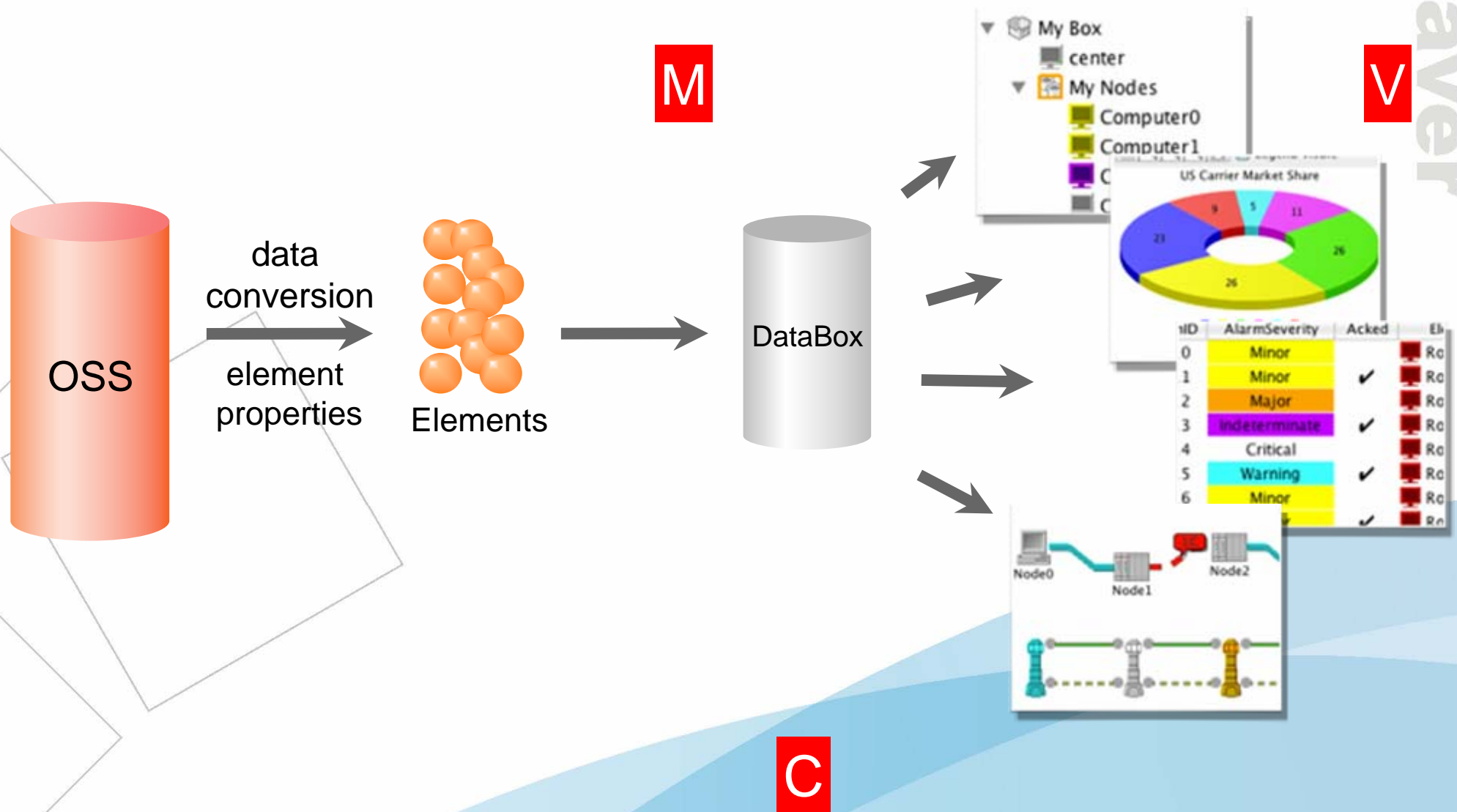
- **Home - ServaSoftware.com**
- **Email - tw-service@servasoft.com**

TWaver Core Components

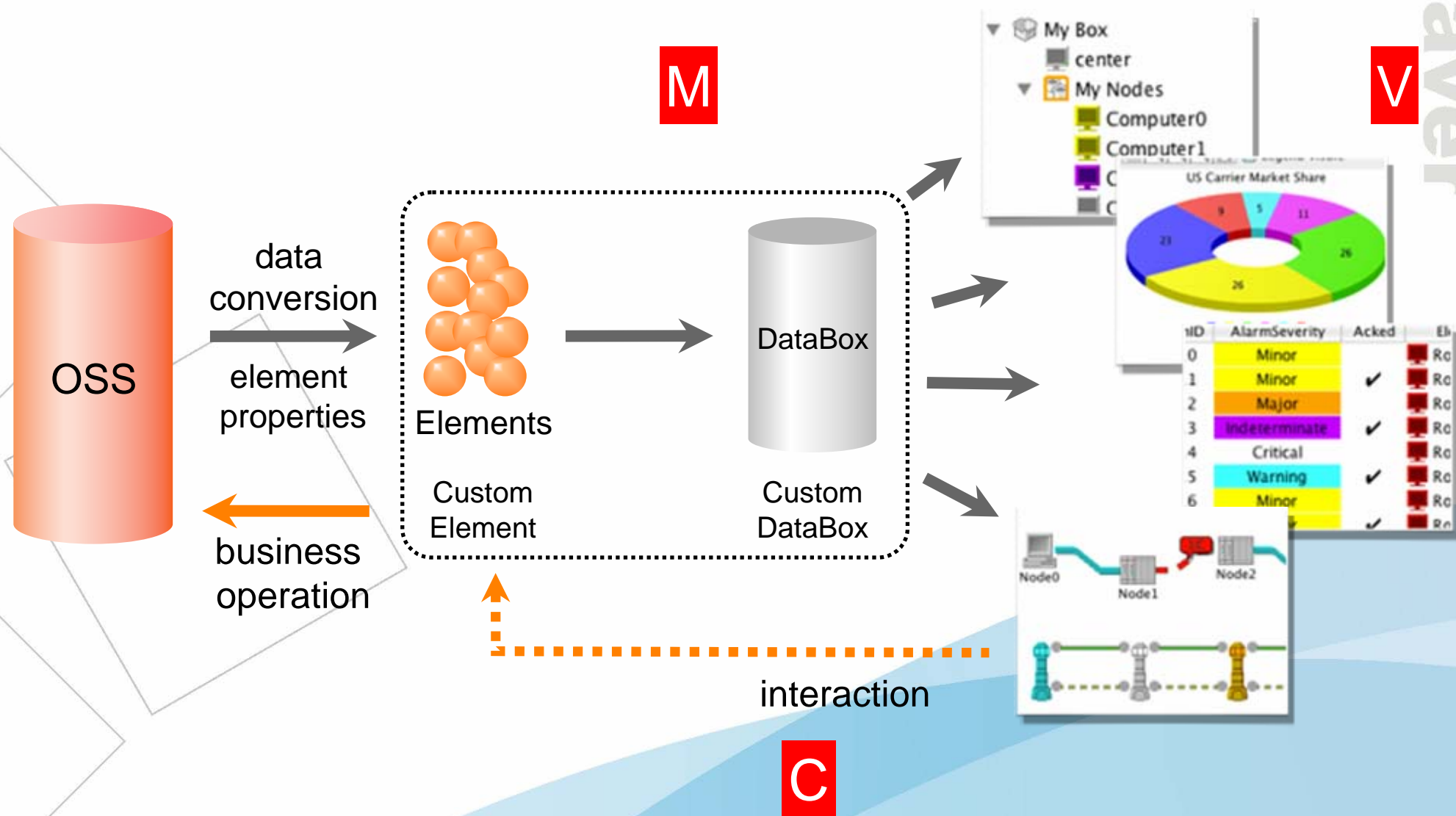
TWaver™

- **Development process**
- **Using TDataBox**
- **Predefined elements**
- **Using TNetwork**

TWaver Development Process



TWaver Development Process



TWaver Development Process

TWaver™

```
//data acquisition
```

```
List<Device> devices = getDevicesFromOSS();  
List<Relationship> relationships = getDevicesRelationshipFromOSS();
```

```
//data conversion
```

```
TDataBox box = new TDataBox();  
translateToTWaverNode(box, devices, relationships);
```

```
//show the interface
```

```
final TNetwork network = new TNetwork(box);  
network.doLayout(TWaverConst.LAYOUT_SYMMETRIC);  
showFrame("Develop Flow", network);
```

```
//add interaction
```

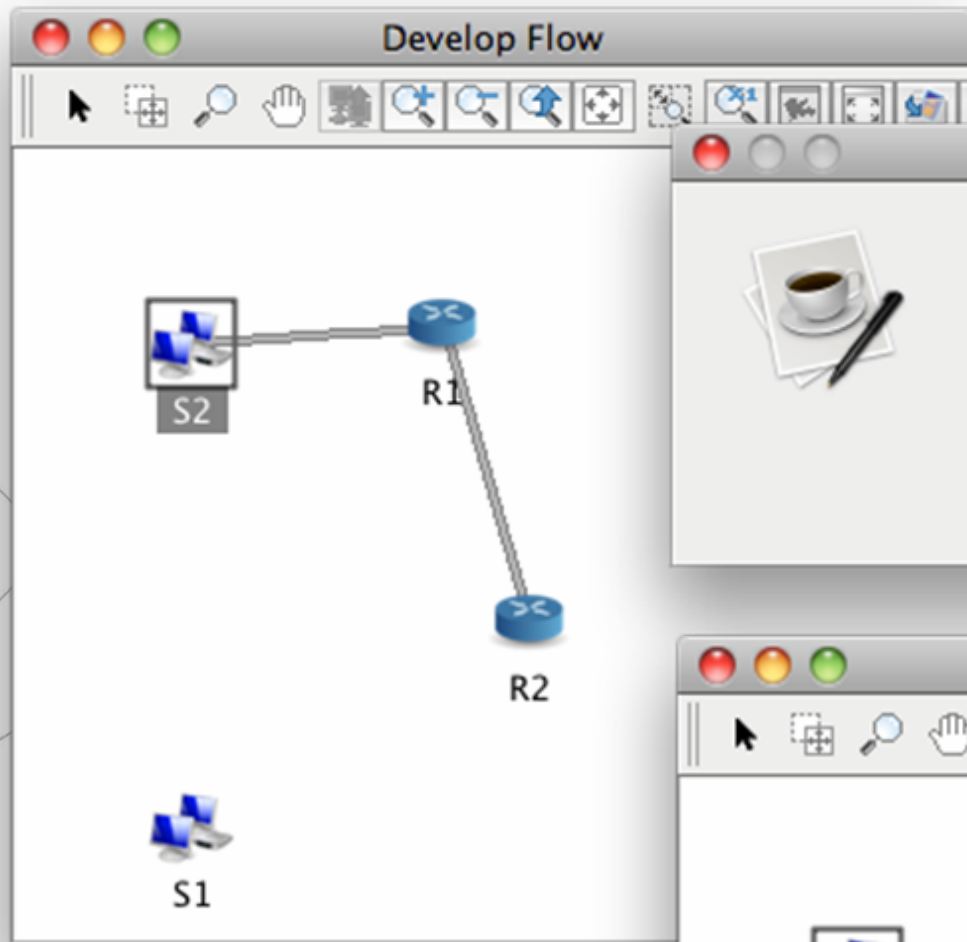
```
network.addElementDoubleClickedActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        Element element = (Element)e.getSource();  
        String newName = JOptionPane.showInputDialog(network, "change device name from : " + element.getName() + " to:");  
        if(newName != null){  
            element.setName(newName);  
            //to storage  
        }  
    }  
});
```

```
private static List<Device> getDevicesFromOSS() {  
    List<Device> devices = new ArrayList<Device>();  
    devices.add(new Device("R1", "router"));  
    devices.add(new Device("R2", "router"));  
    devices.add(new Device("S1", "device"));  
    devices.add(new Device("S2", "device"));  
    return devices ;  
}
```

```
private static List<Relationship> getDevicesRelationshipFromOSS(){  
    List<Relationship> relationships = new ArrayList<Relationship>();  
    relationships.add(new Relationship("2I", "R1", "R2"));  
    relationships.add(new Relationship("3I", "R1", "S2"));  
    return relationships;  
}
```

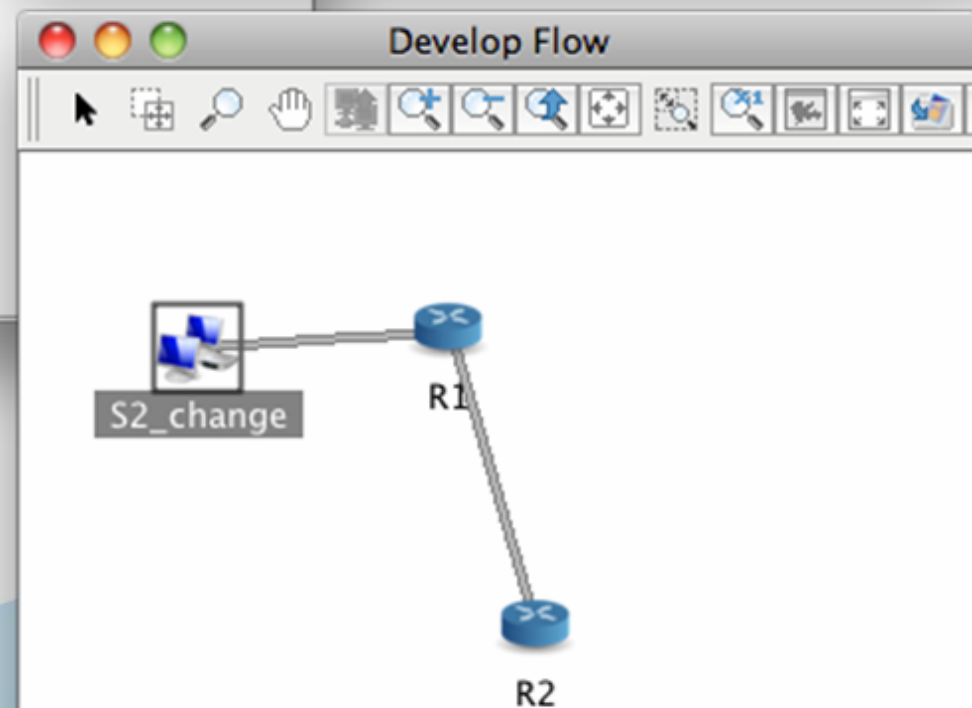
```
private static void translateToTWaverNode(TDataBox box, List<Device> devices, List<Relationship> relationships) {  
    for(Device device : devices){  
        Node node = new Node();  
        node.setName(device.name);  
        node.setImage("/ppt/images/" + device.type + ".png");  
        box.addElement(node);  
    }  
    for(Relationship relationship : relationships){  
        Link link = new Link((Node)box.getElementByName(relationship.fromName),  
        (Node)box.getElementByName(relationship.toName));  
        box.addElement(link);  
    }  
}
```

```
static class Device{  
    String name;  
    String type;  
  
    Device(String name, String type){  
        this.name = name;  
        this.type = type;  
    }  
}  
  
static class Relationship{  
    String name;  
    String fromName;  
    String toName;  
  
    Relationship(String name, String from, String to){  
        this.name = name;  
        this.fromName = from;  
        this.toName = to;  
    }  
}
```



Input

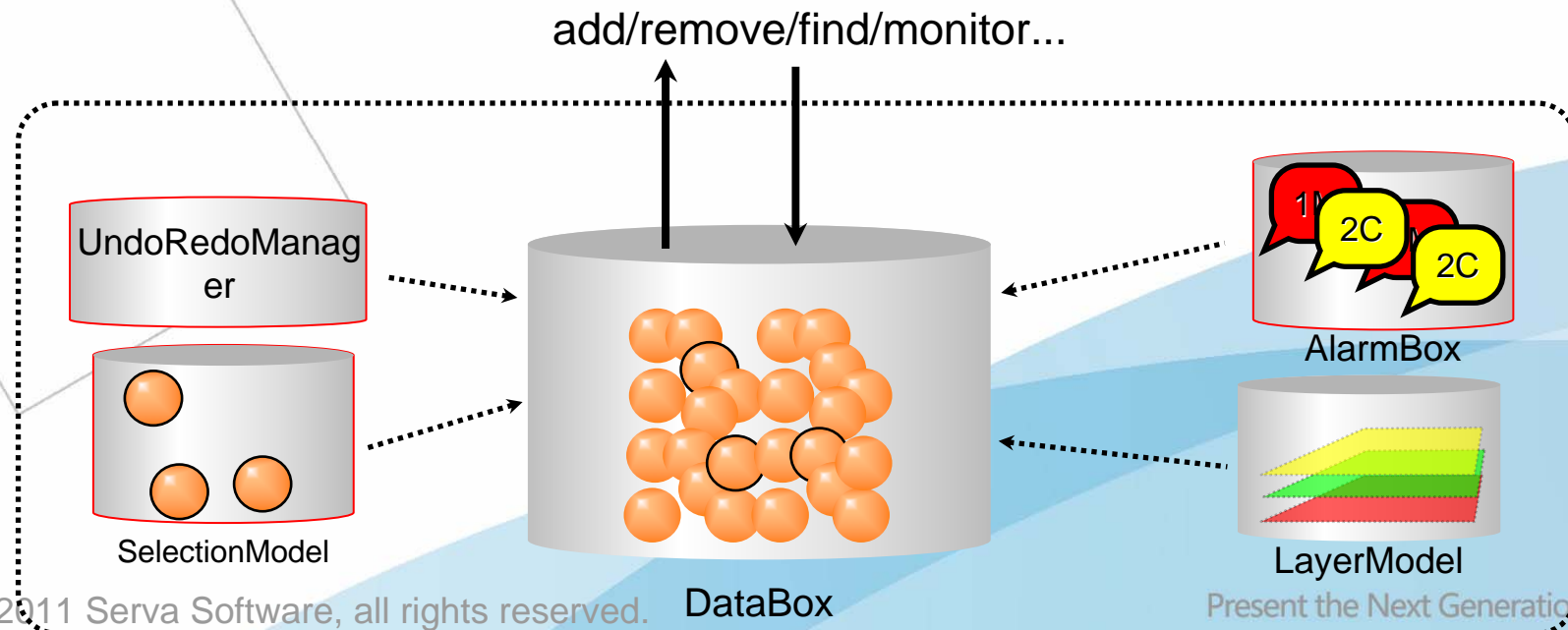
 change device name from : 'S2' to:



Using TDataBox

TDataBox - Data container for managing network elements.

Provides basic operations like add, delete, edit, also can monitor elements changes, level changes and other events, it's includes alarm model, layer model and selection model



Basic Operations

TM
waver

- **Add**

addElement(final Element element)/addElement(int index, final Element element)/addElement(Element element, Element parentOfRootElement)

addElementWithDescendant(Element element)/addElementWithDescendant(Element element, VisibleFilter filter)

addElements(Collection collection)/addElements(Collection collection, Element parentOfRootElement)

- **delete**

removeElement/ removeDescendant/ removeElementById/ removeSelectedElements

- **clear**

dataBox.clear();

Get Elements

get elements - get***Element***

getAllElements()/getAllElementsReverse()/getRootElements()/getRootElementsReverse()getElementByID(Object id)
getElementByName(String name)

Traverse elements - iterator***

iterator()/iterator(Class clazz)/iteratorReverse()/iteratorSelection(ElementCallbackHandler handler)/iteratorReverse(ElementCallbackHandler handler)
iteratorReverseByLayer(ElementCallbackHandler handler)
/iteratorByLayer(ElementCallbackHandler handler)

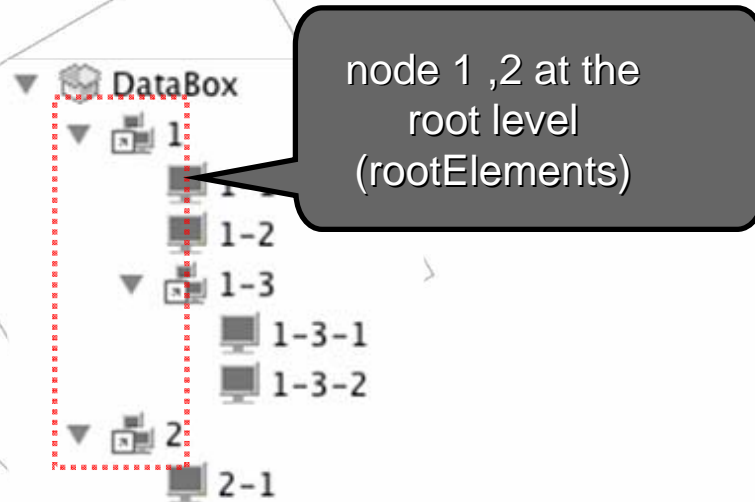
Depth traversal and breadth traversal

depthFirstEnumeration()/depthFirstEnumeration(Element root)/breadthFirstEnumeration()/breadthFirstEnumeration(Element root)

Twaver™

Hierarchy of Elements

- **Element hierarchy is built by parent-child relationships**
- **move***methods of TDataBox are used to adjust element index**



```
TDataBox
  ● moveTo(int, Element) : void
  ● moveToUp(Element) : void
  ● moveToDown(Element) : void
  ● moveToTop(Element) : void
  ● moveToBottom(Element) : void
  ● moveSelectionToUp() : void
  ● moveSelectionToDown() : void
  ● moveSelectionToTop() : void
  ● moveSelectionToBottom() : void
  ● moveElements(Iterator, MovableFilter, double, double)
```

Quick Finder

Quick Finder - DataBoxQuickFinder:

To find elements by a name or other property of element,
for example: List result =
`box.createJavaBeanFinder("name").find("PC");`

Create a quick finder:

`TDataBox#create***Finder(String name)`

- `createJavaBeanFinder` — by javaBean
- `propertycreateClientPropertyFinder` —by client
- `propertycreateUserPropertyFinder` — by user property

Quick Finder Example

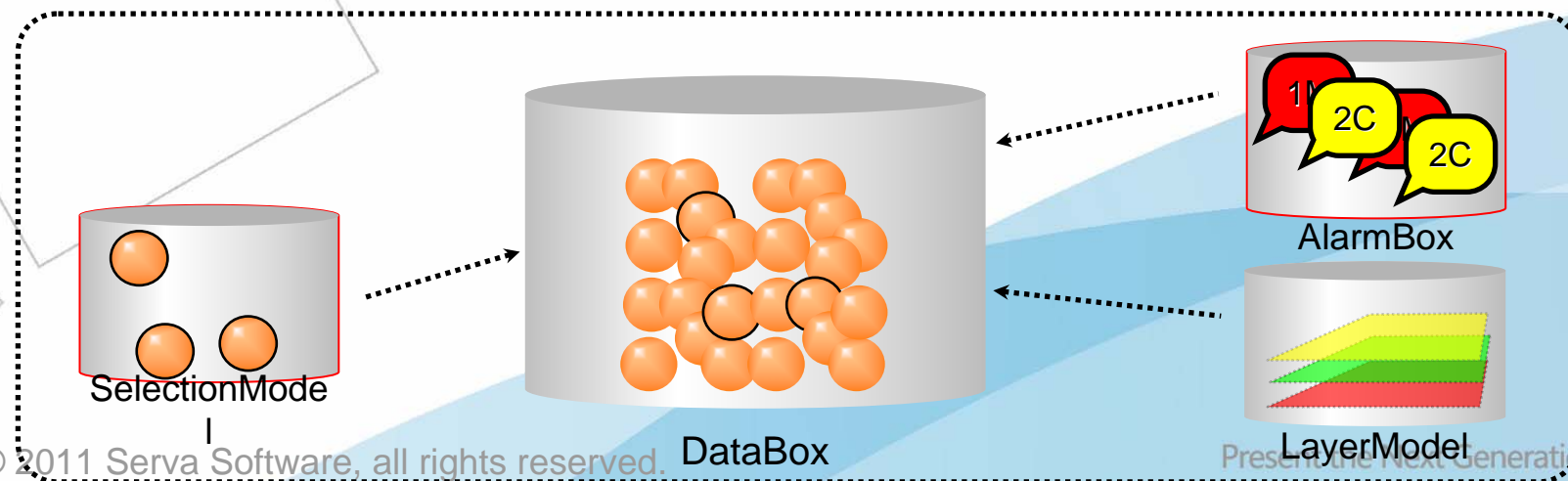
TWaver™

```
TDataBox box = new TDataBox();  
for (int i = 0; i < 30; i++) {  
    Node node = new Node();  
    node.setName("node_" + i);  
    node.putClientProperty("age", new Integer(i%10));  
    box.addElement(node);  
}  
DataBoxQuickFinder finder = box.createClientPropertyFinder("age");  
List elements = finder.find(new Integer(8));  
for(Object node : elements){  
    System.out.println(((Element)node).getName());  
}
```

```
output:  
node_28  
node_18  
node_8
```

Models in TDataBox

- A dataBox includes a alarm model, a layer model and a selection model
- **dataBox.getLayerModel()**
LayerModel#addLayer/removeLayer/moveTo
- **dataBox.getAlarmModel()**
AlarmModel#addAlarm/removeAlarm/clear
- **dataBox.getSelectionModel()**
DataBoxSelectionModel#appendSelection/removeSelection/clearSelection



Listeners in TDataBox

Listeners	Functions
dataBoxListener	detect element's add, remove, clear
propertyChangeListener	detect dataBox's property change
elementPropertyChangeListener	detect element's property change
dataBoxSequenceListener	detect dataBox's sequence change
batchListener	dataBox's batch event listener

add listener - TDataBox#add***Listener

remove listener - TDataBox#remove***Listener

```
box.addElementPropertyChangeListener(new PropertyChangeListener() {  
    public void propertyChange(PropertyChangeEvent evt) {  
        System.out.println(((Element)evt.getSource()).getName() + "s property changed");  
    }  
});
```

Listener Example

```
TDataBox box = new TDataBox();
//add element property change listener
box.addElementPropertyChangeListener(new PropertyChangeListener() {
    public void propertyChange(PropertyChangeEvent evt) {
        System.out.println(((Element)evt.getSource()).getName() + "'s property changed");
    }
});
//add dataBox change listener
box.addDataBoxListener(new DataBoxAdapter() {
    public void elementRemoved(DataBoxEvent e) {
        System.out.println("element removed: " + e.getElement().getName());
    }
    public void elementAdded(DataBoxEvent e) {
        System.out.println("element added: " + e.getElement().getName());
    }
});
Node node = new Node();
node.setName("001");
//add element
box.addElement(node);
//modified element's property
node.setDisplayName("server-001");
//remove the element
box.removeElement(node);
```

results:

element added: 001
001's property changed
element removed: 001

Import & Export TDataBox

Twaver™

- **TDataBox can both import and export xml**
- **The export xml contains all elements and their properties**
- **The images of elements also can be exported to xml with base64 encode**

Import & Export TDataBox

TM
Twaiver

Export - **dataBox.output***(...)** / **dataBox.toXML(...)**

```
public void output(String fileName,  
    boolean withElementId,  
    boolean withAlarmState,  
    ElementPersistentFilter elementFilter,  
    ClientPropertyPersistentFilter clientPropertyFilter)
```

Import - **dataBox.parse***(...)**

```
public void parse(String url, Element parentOfRootElement)
```

Import & Export TDataBox

```
TDataBox box = new TDataBox();  
box.putClientProperty("databoxPropertyA", "a box  
property");
```

```
Node node = new Node();  
node.setName("a node");  
node.setLocation(100, 100);  
box.addElement(node);
```

//output xml

```
String xml=box.toXML();  
System.out.println(xml);
```

//parse xml

```
TDataBox box2 = new TDataBox();  
box2.parseXML(xml);  
System.out.println(box2.getClientProperty("databoxPropertyA"));  
System.out.println(box2.getElementByName("a  
node").getLocation());
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<java version="1.6.0_24" class="java.beans.XMLDecoder">  
  <string>databoxPropertyA</string>  
  <string>a box property</string>  
  <object class="twaver.Node">  
    <void property="location">  
      <object class="java.awt.Point">  
        <int>100</int>  
        <int>100</int>  
      </object>  
    </void>  
    <void property="name">  
      <string>a node</string>  
    </void>  
  </object>  
</java>
```

a box property

java.awt.Point[x=100,y=100]

Twaver™

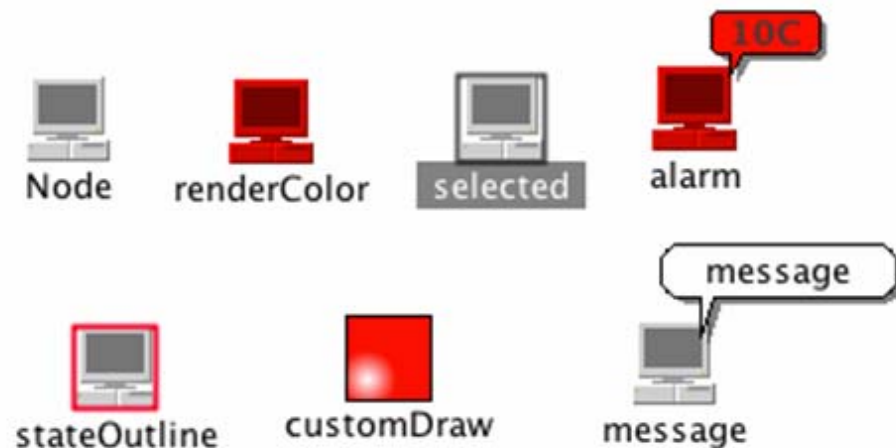
Elements

Element - topology data

- |-- Node — basic node
 - |-- Dummy — dummy node
 - |-- Group — a group contains other elements
 - |-- Text — node without image
 - |-- ShapeNode — polygon
 - |-- ShapeImage/ShapeSubNetwork
 - |-- ResizableNode — resizable node
 - |-- Follower
 - |-- BaseEquipment
 - |-- BTS/BTSAntenna/Card/Port/Rack/Shelf/Slot/Grid
 - |-- SubNetwork
- |-- Link
 - |-- LinkSubNetwork

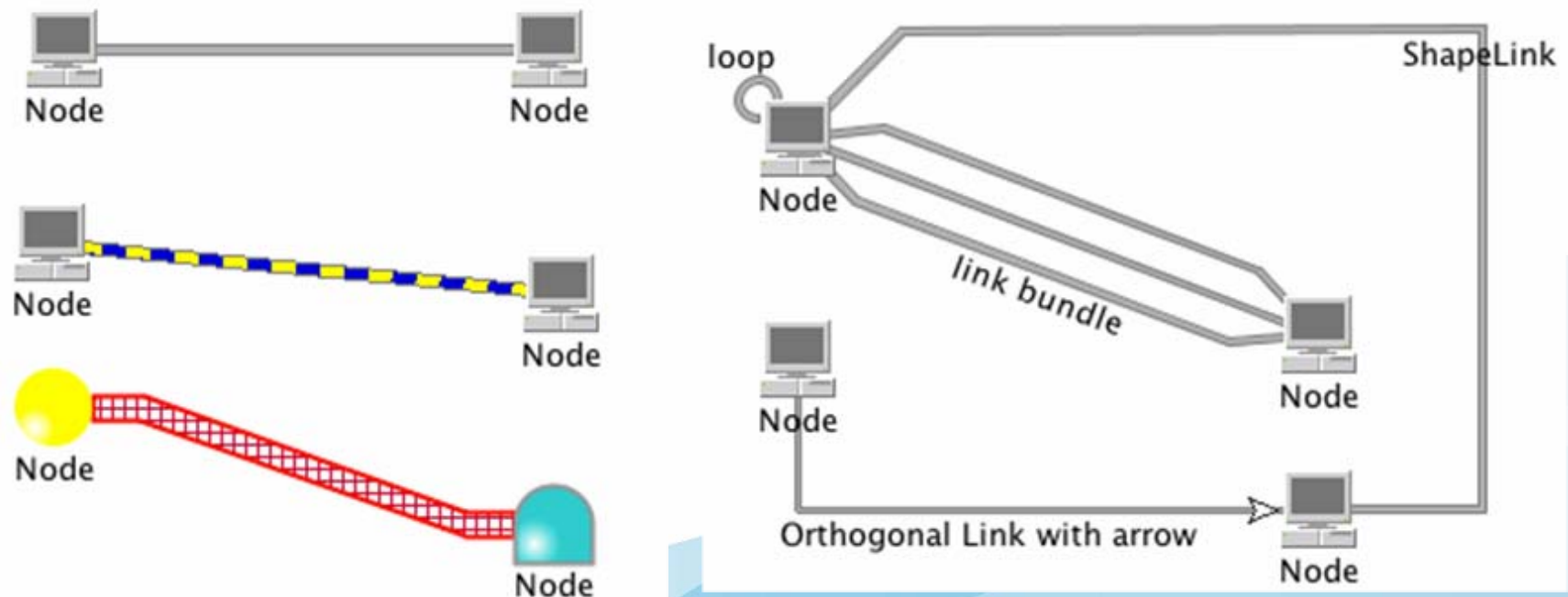
Node

- Basic node type, can set image, outline, render color ...
- `twaver.Node` is the base class for other elements



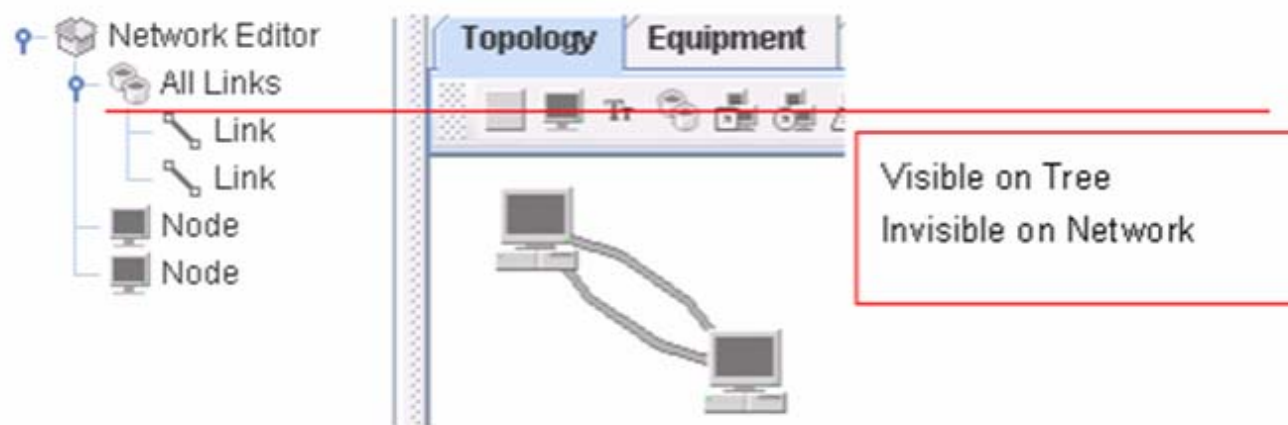
Link

- Link, linking two nodes, it has many types and styles, it provides arrows, loop, flowing.
- `twaver.link` is the base class for other link types.



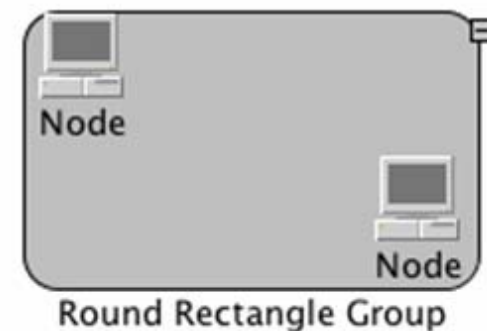
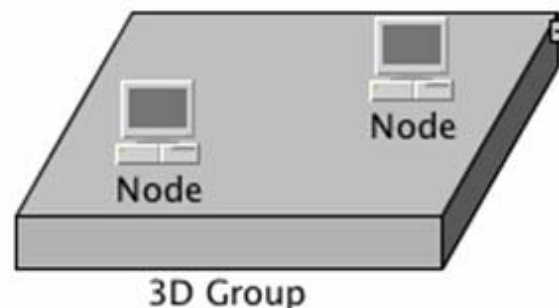
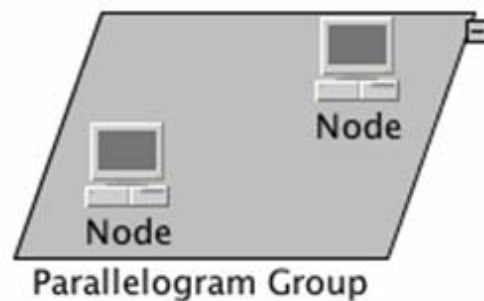
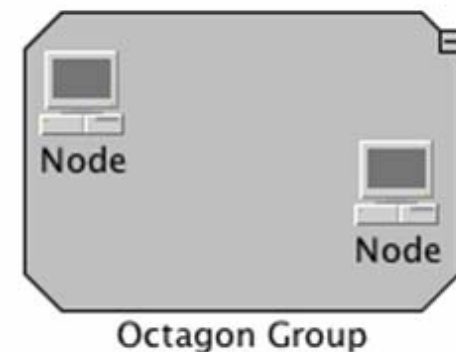
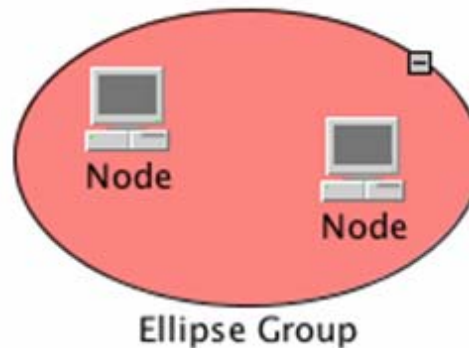
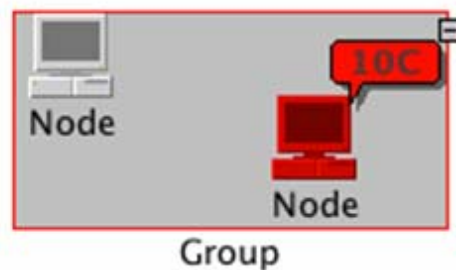
Dummy

- **Dummy is a type of node, it is invisible on network, while it is visible on Tree, Table and Property Table**
- **Dummy can be used to reorganize tree structure and avoid to affect network**



Group

- **Group**, contains other nodes, it can close or open, it has many types of shapes, such as ellipse, parallelogram, octagon and 3D, etc.



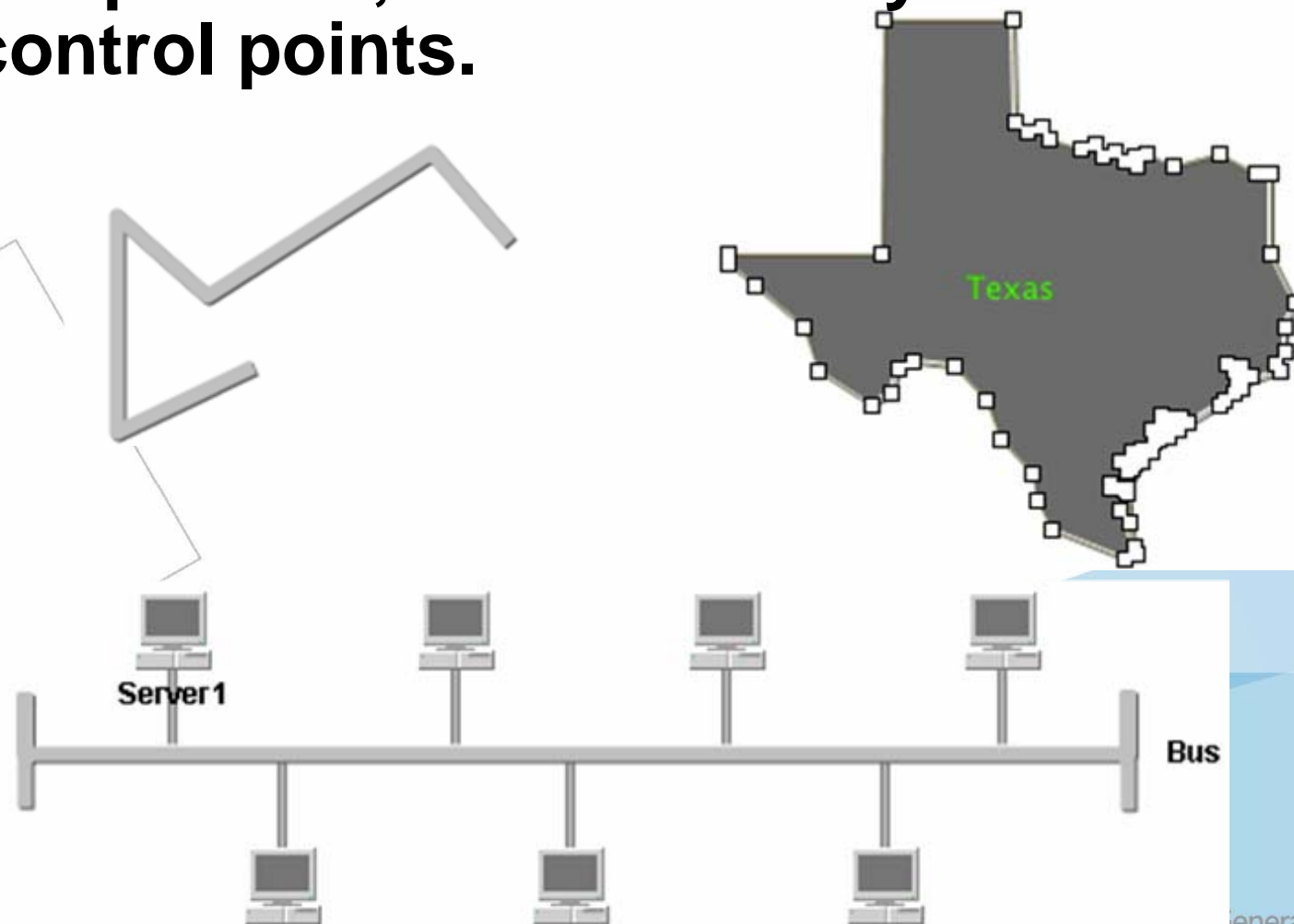
SubNetwork

- Subnetwork, you can drop in or out by double click. When you drop in a subnetwork, the children of the subnetwork will be displayed; when it drop out, it will come back to the upper subnetwork. When previous subnetwork is null, it shows top
- Subnetwork has its own background



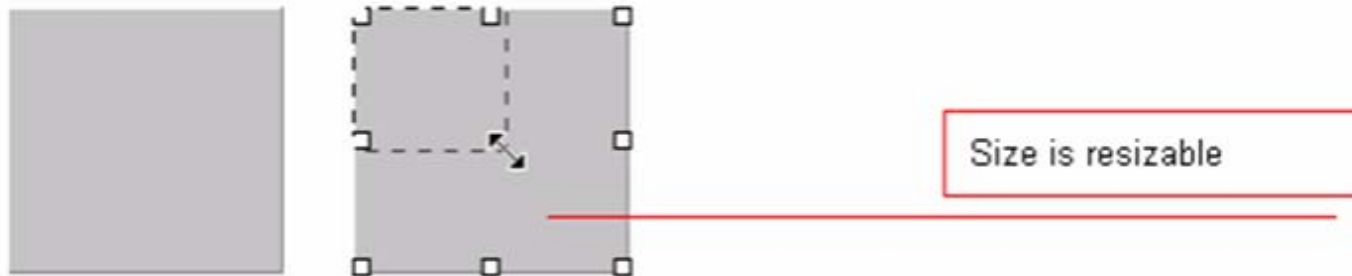
ShapeNode

- **ShapeNode, surrounded by a number of control points.**



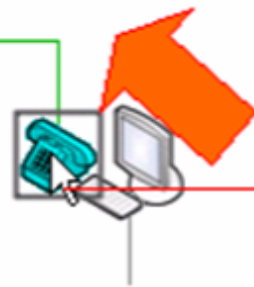
ResizableNode

- **ResizableNode, can resize width and height**



Follower

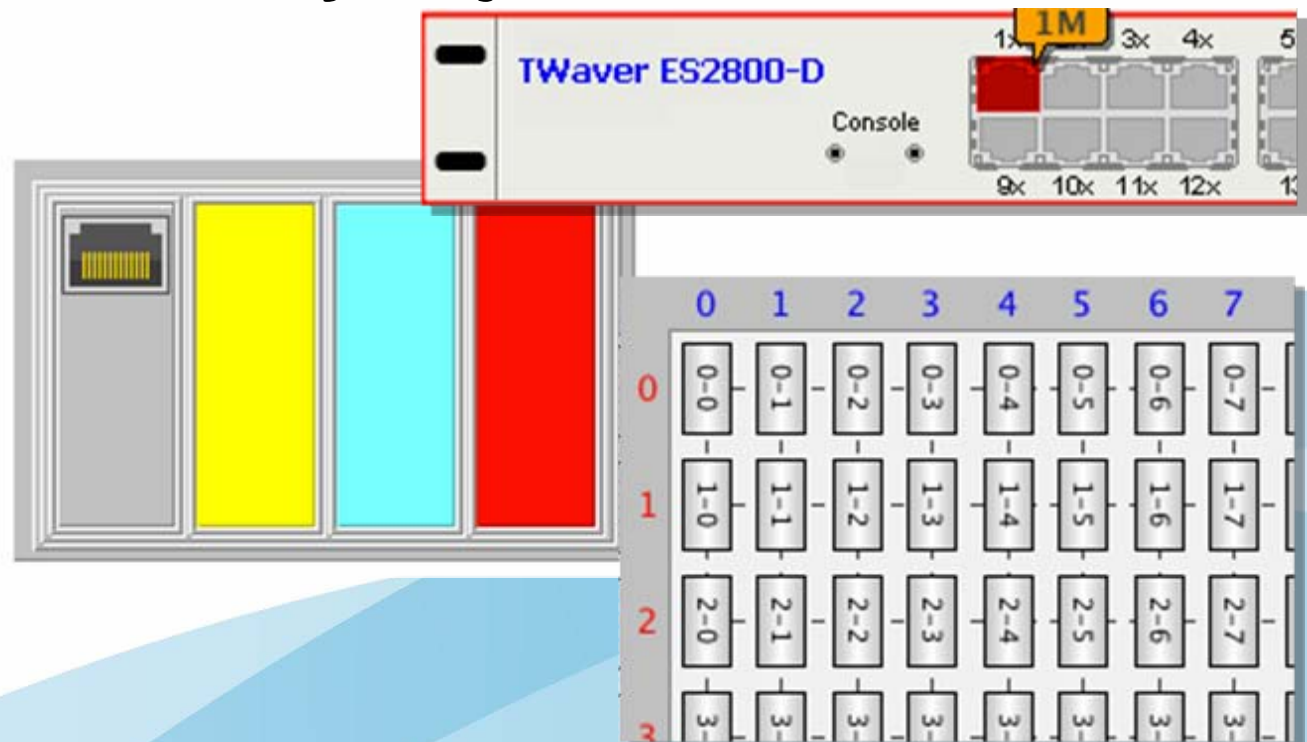
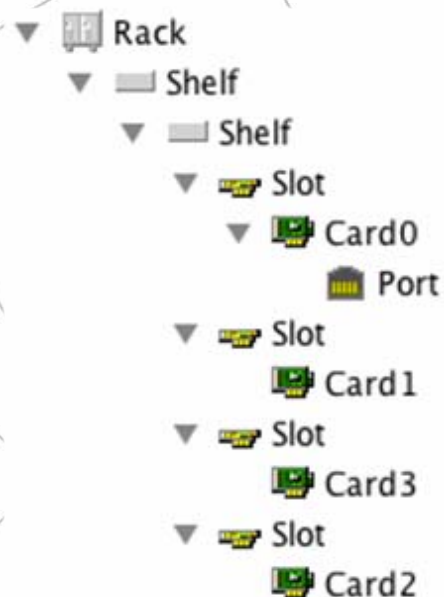
- **Follower, it can set a host node, follower will follow it when host is moving.**



I move, and you move

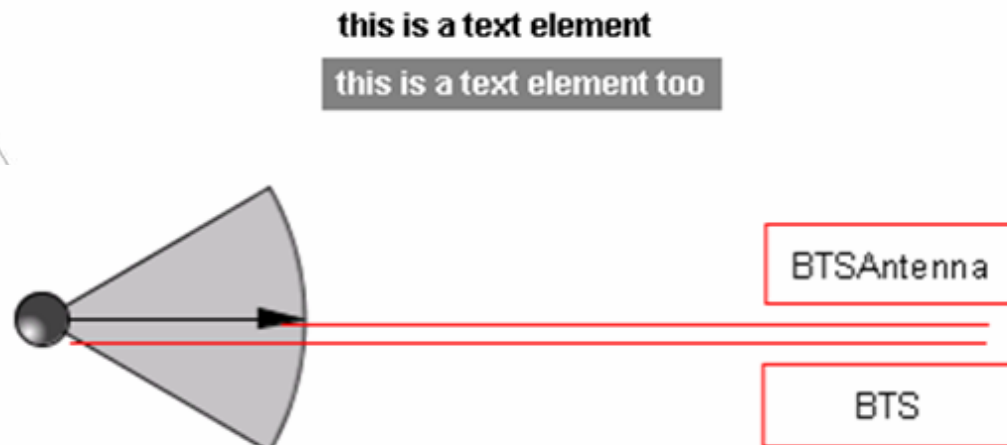
Equipment

- **Equipment, it is the base class for Rack, Shelf, Slot, Card and Port.**
- **It can automatically adjust size and position.**



Other Types

- **twaver.Text**
- **BTS, BTSAntenna**
- **ShapelImage**
- ...



Element Properties

Element

JavaBeans - name, id ...

Client Properties - Graphics-related properties, like:
render color, border ...

User Properties - user properties

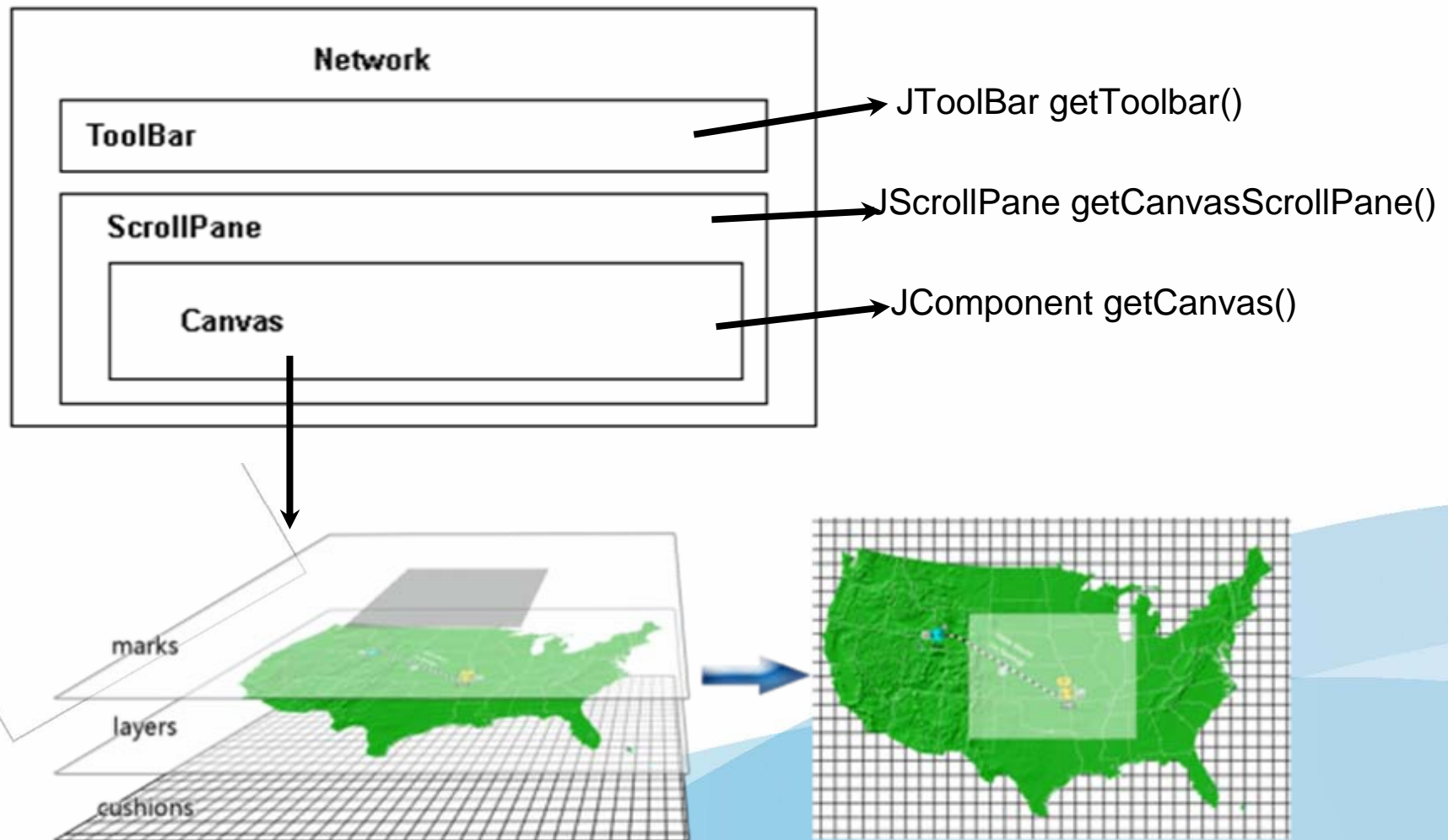
Business Object - user business object attached to
the element

Using TNetwork

- **TNetwork hierarchy**
- **TNetwork interaction**
- **TNetwork filters**
- **TNetwork generators**
- **Custom ElementUI**

Twaver™

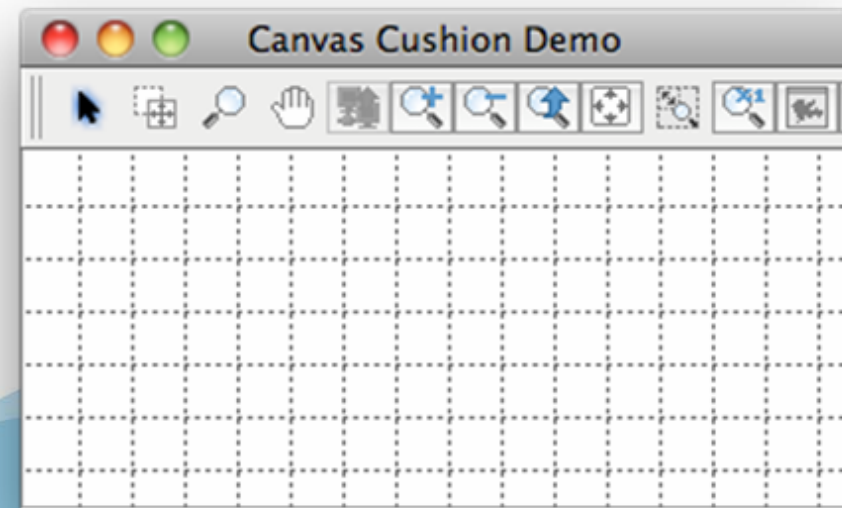
TNetwork Hierarchy



Marker & Cushion

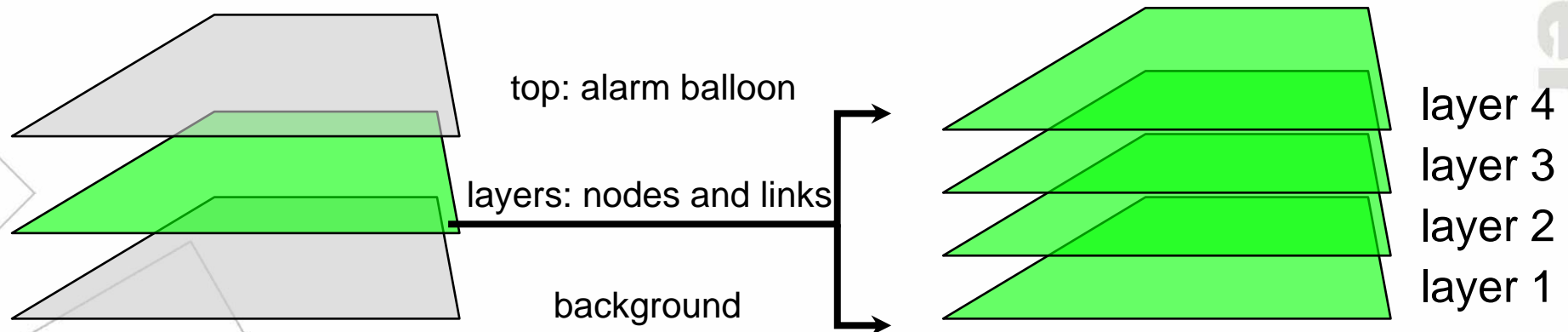
add/removeCanvasMarker(CanvasMarker m)
add/removeCanvasCushion(CanvasCushion canvasCushion)

```
final TNetwork network = new TNetwork();  
CanvasCushion canvasCushion = new CanvasCushion() {  
    Color color = new Color(100, 100, 100);  
    Stroke stroke = new BasicStroke(0, BasicStroke.CAP_BUTT, BasicStroke.JOIN_ROUND, 10.0f, new float[]{2}, 0.0f);  
    int gridSize = 20;  
    public void paint(Graphics2D g2d) {  
        g2d.setColor(color);  
        g2d.setStroke(stroke);  
        int width = network.getCanvas().getWidth();  
        int height = network.getCanvas().getHeight();  
        for (int x = gridSize; x < width; x += gridSize) {  
            g2d.drawLine(x, 1, x, height);  
        }  
        for (int y = gridSize; y < height; y += gridSize) {  
            g2d.drawLine(0, y, width, y);  
        }  
    }  
};  
network.addCanvasCushion(canvasCushion);
```



Layer Management

TWaver™



```
LayerModel layerModel = box.getLayerModel();  
Layer bottomLayer = new Layer("bottom");  
layerModel.addLayer(0, bottomLayer);
```

```
ResizableNode bottomNode = new ResizableNode();  
bottomNode.setName("node at bottom");  
bottomNode.setLayerID(bottomLayer.getID());  
box.addElement(bottomNode);
```

TNetwork Interaction

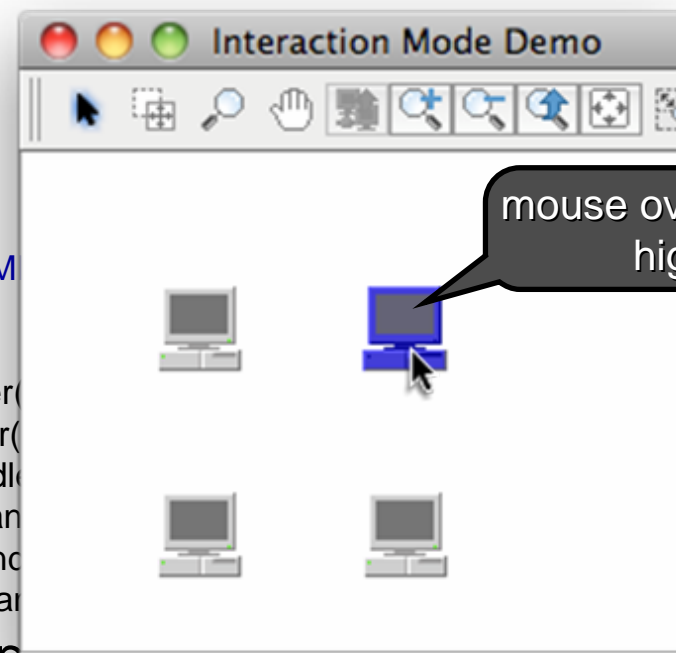
- **InputHandler** - receives all mouse and keyboard events
- **InteractionMode** - a set of input handlers form an interaction mode
- **Predefined Modes** - Network provides several predefined interaction modes, such as the default mode, edit mode, creating link mode ...

```
InputHandler[] listeners = new InputHandler[] {  
    //add selection input handler, make network can select elements  
    new SelectionInputHandler(network),  
    //add move input handler, then network can move elements  
    new EagerMoveInputHandler(network),  
    new InteractionInputHandler(network),  
    new PopupMenuInputHandler(network),  
};  
InteractionMode defaultMode = new InteractionMode(listeners);  
network.setInteractionMode(defaultMode);
```

Custom Input Handler

TWaver™

```
public static void main(String[] args) {  
    TNetwork network=new TNetwork();  
    network.getDataBox().addElement(new Node());  
    network.getDataBox().addElement(new Node());  
    network.getDataBox().addElement(new Node());  
    network.getDataBox().addElement(new Node());  
    network.doLayout(TWaverConst.LAYOUT_SYMM  
  
    InputHandler[] handlers = new InputHandler[] {  
        new DefaultInputHandler(  
        new ResizeInputHandler(  
        new SelectionInputHandl  
        new EagerMoveInputHan  
        new InteractionInputHanc  
        new PopupMenuInputHar  
  
        new HighlightInputHandler(network)  
    };  
    network.setInteractionMode(new InteractionMode(handlers));  
    showFrame("Interaction Mode Demo", network);  
}
```



mouse over, and node highlight


```
public static class HighlightInputHandler extends InputAdapter {
    private TNetwork network;
    private Element highlightElement;
    private Color highLightColor = new Color(100, 100, 255);
    private Color oldColor;

    public HighlightInputHandler(TNetwork network) {
        this.network = network;
    }

    public void mouseMoved(MouseEvent e) {
        Element element = network.getElementLogicalAt(e.getPoint());
        if(element != null){
            highlight(element);
        }else{
            reset();
        }
    }

    private void highlight(Element element) {
        if(element == null || element.equals(highlightElement)){
            return;
        }
        reset();
        highlightElement = element;
        oldColor = element.getRenderColor();
        element.putRenderColor(highLightColor);
    }

    private void reset() {
        if (highlightElement != null) {
            highlightElement.putRenderColor(oldColor);
            highlightElement = null;
        }
    }
}
```

TNetwork Filters

Filters are used for global control, such as control elements to show or hide, move or not move

add***Filter/remove***Filter

```
network.addVisibleFilter(new VisibleFilter() {  
    public boolean isVisible(Element element) {  
        return !(element instanceof Link);  
    }  
});
```

set***Filter

```
network.setMovableFilter(new MovableFilter(){  
    public boolean isMovable(Element element) {  
        if(editMode.isSelected()){  
            return true;  
        }  
        return !(element instanceof Equipment);  
    }  
});
```

TNetwork Filters

Filter Types	Description
visibleFilter	be concealed and displayed
movableFilter	movable or unmovable
selectableFilter	selectable or unselectable
resizableFilter	resizable
sendToTopFilter	showing on top
paintSelectionStateFilter	whether paint element's selection borders
elementPropertyChangeRepaintFilter	whether repaint when element's property changed
elementBoundsInvalidatableFilter	whether invalidate element's bounds when its property changed
doubleClickableFilter	whether double clickable
elementLabelEditableFilter	whether label editable

add
remove

set
get

TNetwork Generators

set***Generator

Generator can have the general control of element properties, not need to modify property of each element

```
network.setElementLabelGenerator(new Generator() {  
    public Object generate(Object object) {  
        Element element = (Element) object;  
        return element.getClientProperty("STATE_NAME");  
    }  
});
```



TNetwork Generators

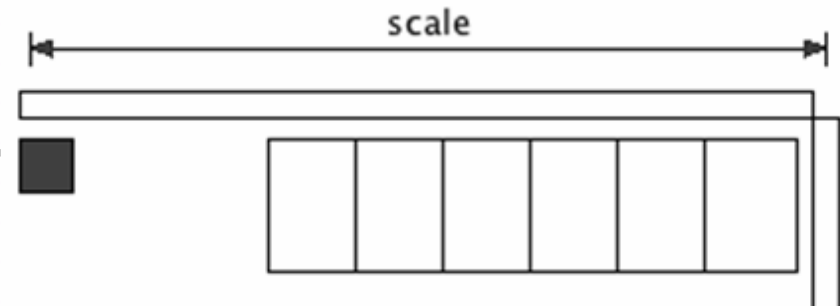
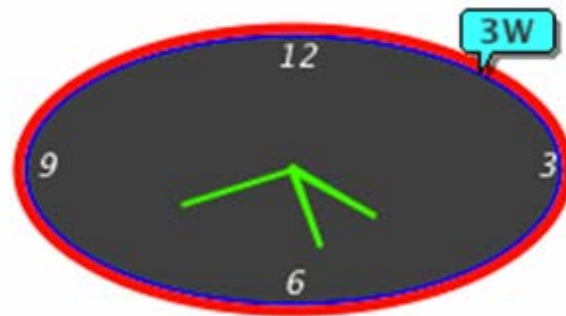
Generator types	Description
ElementLabelGenerator	element label
AlarmLabelGenerator	alarm ball label
ElementToolTipTextGenerator	element tooltip text
MessageContentGenerator	element message text
ElementSelectColorGenerator	element selection border color
PopupMenuGenerator	context menu
AlarmColorGenerator	alarm ball color
ElementOutlineGenerator	element outline color
ElementBodyColorGenerator	node body color

TWaver™

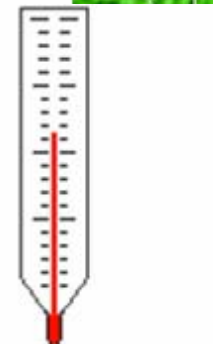
Custom ElementUI

There are two ways to custom element's UI:

- **Override ElementUI**
- **Use Attachments**



X-001



TM
T
W
a
v
e
r

Override ElementUI

TWaver™

- **Define a class: MyElementUI**

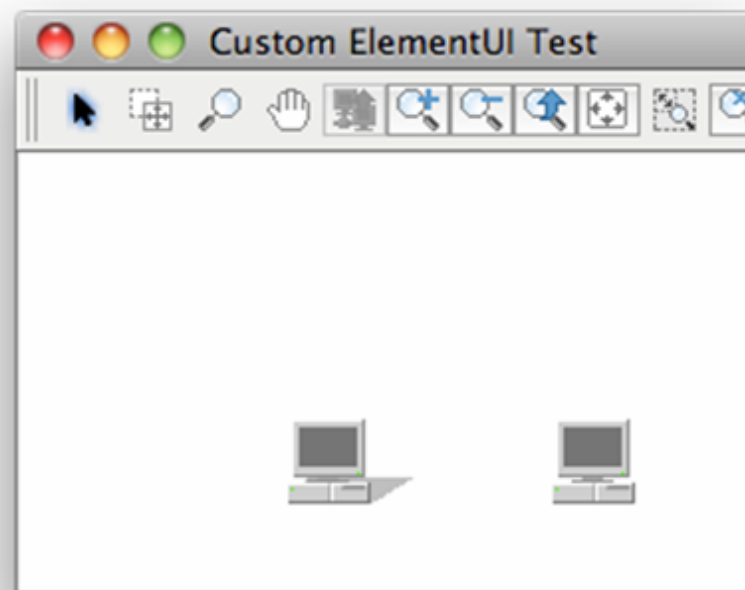
```
public class MyElementUI extends NodeUI {  
    ...  
    public void paintBody(Graphics2D g2d) {  
        //custom draw  
        super.paintBody(g2d);  
    }  
}
```

- **Associated with the Element class by override the function Element#public String getUIClassID()**

```
public class MyElement extends Node {  
    ...  
    public String getUIClassID() {  
        return MyElementUI.class.getName();  
    }  
}
```

ElementUI Example

- **The following example shows how to create a shadow for a node by custom element's UI**



```
public static class MyElement extends Node {
    //add "showShadow"property
    public final static String PROPERTYNAME_SHOW_SHADOW = "propertyname.show.shadow";
    public void putShowShadow(boolean show){
        this.putClientProperty(PROPERTYNAME_SHOW_SHADOW, show);
    }
    public boolean isShowShadow(){
        return Boolean.TRUE.equals(this.getClientProperty(PROPERTYNAME_SHOW_SHADOW));
    }
    //
    public MyElement() {
        super();
    }
    public MyElement(Object id){
        super(id);
    }
    // associated with MyElementUI class
    public String getUIClassID() {
        return MyElementUI.class.getName();
    }
}
```

```
public static class MyElementUI extends NodeUI {  
    public MyElementUI(TNetwork network, Node element) {  
        super(network, element);  
        resetShadowShape();  
    }  
  
    public void paintBody(Graphics2D g2d) {  
        if (shadow != null) {  
            Color shadowColor = new Color(128, 128, 128, 128);  
            g2d.setColor(shadowColor);  
            g2d.fill(shadow);  
        }  
        super.paintBody(g2d);  
    }  
    private Polygon shadow = null;  
    private Polygon resetShadowShape() {  
        //...  
    }  
  
    public void elementPropertyChange(PropertyChangeEvent evt) {  
        super.elementPropertyChange(evt);  
        resetShadowShape();  
    }  
  
    public Rectangle getUIBounds() {  
        Rectangle result = super.getUIBounds();  
        if (shadow != null) {  
            result.add(shadow.getBounds());  
        }  
        return result;  
    }  
}
```



```

public static void main(String[] args) {
    TDataBox box = new TDataBox();
    MyElement node = new MyElement();
    node.putShowShadow(true);
    node.setLocation(100, 100);
    box.addElement(node);

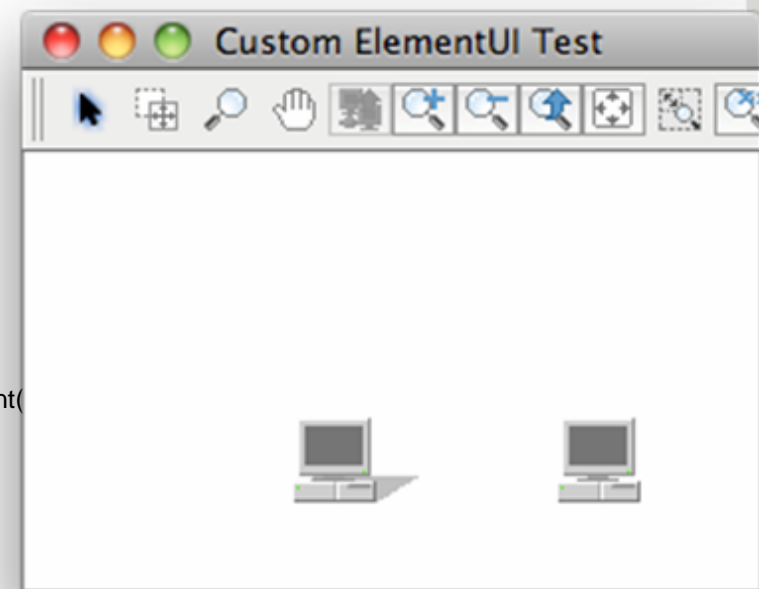
    node = new MyElement();
    node.setLocation(200, 100);
    box.addElement(node);

    PopupMenuGenerator popupMenuGenerator = new PopupMenuGenerator(){
        public JPopupMenu generate(TView tview, MouseEvent mouseEvent) {
            TNetwork network=(TNetwork)tview;

            final Element element=network.getElementLogicalAt(mouseEvent.getPoint(
            if(!(element instanceof MyElement)){
                return null;
            }

            final boolean showShadow = ((MyElement)element).isShowShadow();
            JPopupMenu menu=new JPopupMenu();
            JMenuItem menuItem=new JMenuItem(showShadow ? "Hide Shadow" : "Show Shadow");
            menuItem.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    ((MyElement)element).putShowShadow(!showShadow);
                }
            });
            menu.add(menuItem);
            return menu;
        }
    };
    TNetwork network = new TNetwork(box);
    network.setPopupMenuGenerator(popupMenuGenerator);
    TestUtil.showFrame("Custom ElementUI Test", network);
}

```



TWaver™

Custom Attachment

twaver.network.ui.Attachment interface.

Implementation classes:

- **LabelAttachment/AlarmAttachment/MessageAttachment/IconAttachment...**
- **ComponentAttachment**



Custom Attachment

TWaver™

- **Define MyAttachment:**
public class **MyAttachment** extends **ComponentAttachment**
- **Register MyAttachment:**
TUIManager.registerAttachment("myAttachment", MyAttachment.class)
- **Add to a node:**
node.addAttachment("myAttachment")

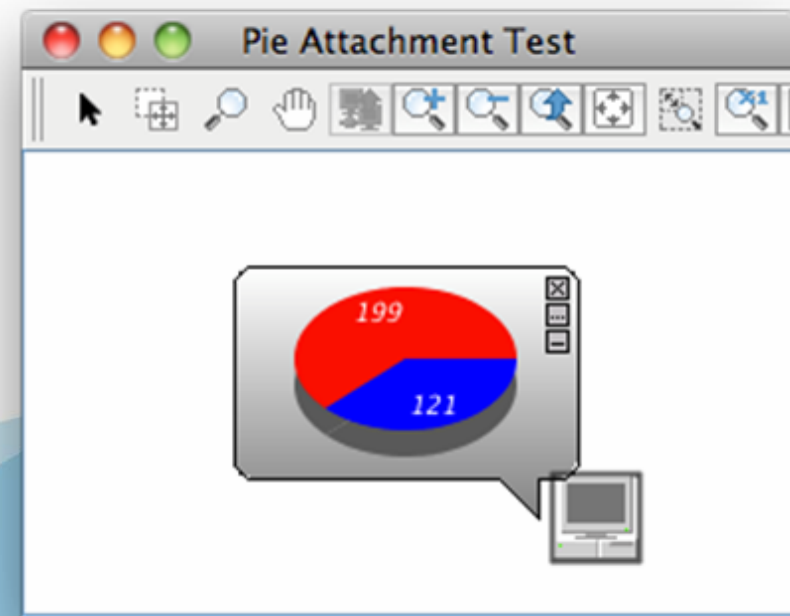
Use Attachment

TWaver™

```
static {  
    TUIManager.registerAttachment("PieAttachment", PieAttachment.class);  
}  
  
public static void main(String[] args) {  
    TDataBox box = new TDataBox();  
    Node node = new Node();  
    node.setLocation(100, 100);  
    node.addAttachment("PieAttachment");  
    box.addElement(node);  
  
    TestUtil.showFrame("Pie Attachment Test", new TNetwork(box));  
}
```

Register
attachment

Add attachment



Define a Attachment

Define
PieAttachment

```
public class PieAttachment extends ComponentAttachment {
```

```
private PieChart pie;
private Item xItem;
private Item yItem;
```

```
public PieAttachment(String name, ElementUI ui) {
```

```
super(name, ui);
```

```
xItem = new Item("x", element.getX(), Color.RED);
yItem = new Item("y", element.getY(), Color.BLUE);
Vector items = new Vector();
items.addElement(xItem);
items.addElement(yItem);
```

```
pie = new PieChart(items, null, Color.WHITE);
```

```
pie.setValueTextColor(Color.WHITE);
pie.setValueTextFont(TWaverUtil.getFont(Font.ITALIC, 10));
pie.getLegendPanel().setVisible(false);
pie.setBackground(Color.GRAY.darker());
pie.setOpaque(true);

this.setClosable(true);
this.setMinimizable(true);
this.setResizable(true);
this.setStyle(TWaverConst.ATTACHMENT_STYLE_BUBBLE);
this.setPosition(TWaverConst.POSITION_LEFT);
this.setDirection(TWaverConst.ATTACHMENT_DIRECTION_TOP_LEFT);
this.setBorderVisible(true);
this.setBorderColor(Color.BLACK);
this.setBorderVisible(true);
this.setBodyVisible(true);
this.setBodyGradient(true);
this.setBackgroundColor(Color(0, 0, 0, 125));
this.setBodyGradientColor(new Color(255, 255, 255, 125));
this.setWidth(100);
this.setHeight(80);
```

```
this.setComponent(pie);
```

```
public void elementPropertyChange(PropertyChangeEvent evt) {
super.elementPropertyChange(evt);
if (evt.getPropertyName().equals(TWaverConst.PROPERTYNAME_LOCATION)) {
xItem.setValue(element.getX());
yItem.setValue(element.getY());
}
}
```

Define a constructor

Create a pie chart

Set the main component

Listen node property change event, and refresh the chart



- Home - ServaSoftware.com
- Email - tw-service@servasoft.com

Common Components & Alarms

Twaver™

- **Common components**

- TTree
- TElementTable
 - TTreeTable
- TPropertySheet
- Chart

- **Alarm**

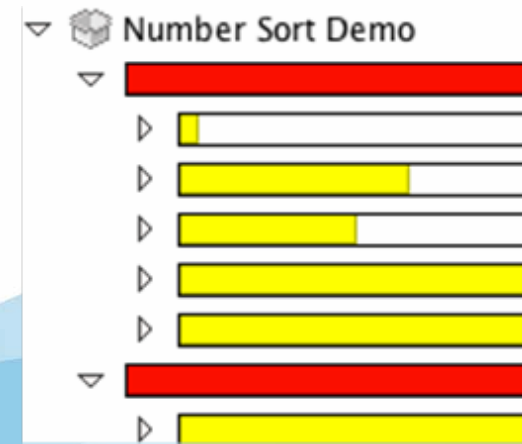
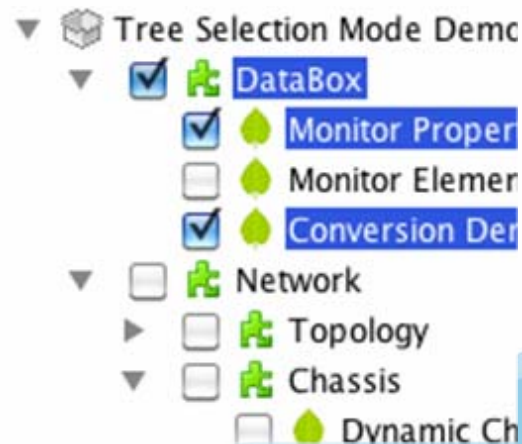
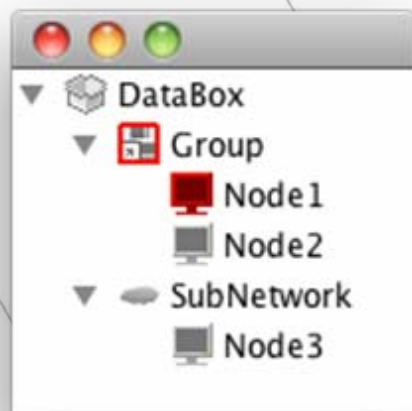
- Alarm/AlarmSeverity/AlarmModel/Alarm
StateStatistics/Alarm views

TTree

TWaver™

twaver.tree.TTree provides a hierarchical view of elements contained in a DataBox

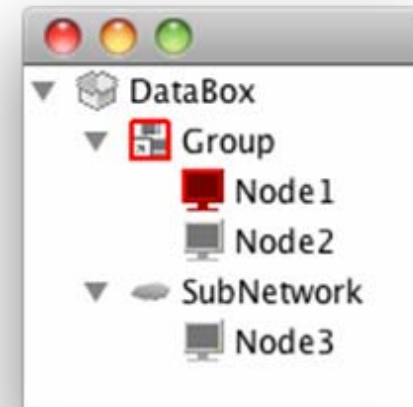
```
TTree tree = new TTree(box);
```



Create a Tree

```
TDataBox box = new TDataBox();  
TTree tree = new TTree(box);  
Group group = new Group();  
box.addElement(group);  
SubNetwork subnetwork = new SubNetwork();  
box.addElement(subnetwork);
```

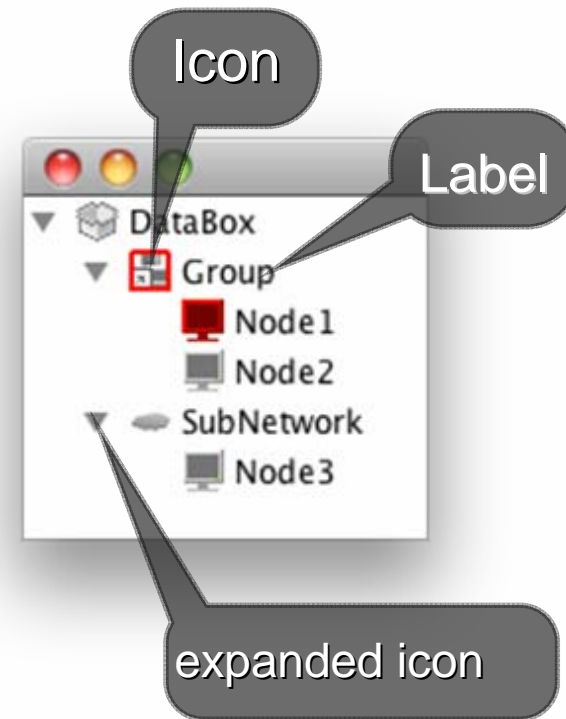
```
Node node1 = new Node();  
node1.setName("Node1");  
node1.setParent(group);  
node1.getAlarmState().addNewAlarm(AlarmSeverity.CRITICAL);  
box.addElement(node1);  
Node node2 = new Node();  
node2.setName("Node2");  
node2.setParent(group);  
box.addElement(node2);  
Node node3 = new Node();  
node3.setName("Node3");  
node3.setParent(subnetwork);  
box.addElement(node3);  
  
showFrame("Tree Demo", tree);
```



element parent-child
relationship

Custom Tree Node

- **Node icon**
`element.putElementTreeIcon(icon)`
- **Node label**
`node.setName(name);`
- **Collapse and expand icons**
`tree.setCollapsedIcon(icon)`
`tree.setExpandedIcon(icon)`



```
tree.setCollapsedIcon(TWaverUtil.getIcon("/demo/sheet/nested/collapse.png")  
);tree.setExpandedIcon(TWaverUtil.getIcon("/demo/sheet/nested/expand.png")  
);
```


Node Hierarchy and Order

Twaver™

- **Parent-child hierarchy**

```
node.setParent(parent);  
node.addChild(child);
```

- **Element order**

```
box.moveTo***(node);
```

- **Sorting**

```
tree.setSortComparator(comparator);
```

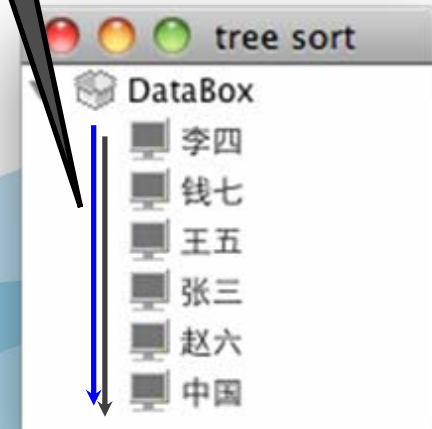
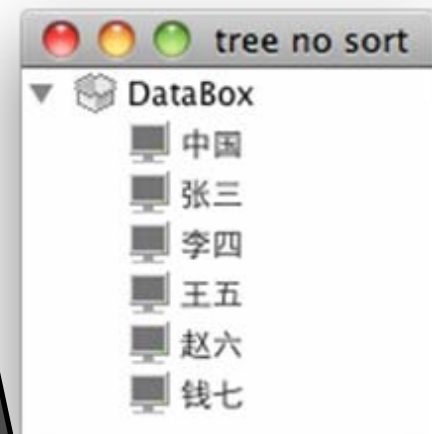
Tree Sort Example

```
TDataBox box = new TDataBox();  
createNode(box, "中国");  
createNode(box, "张三");  
createNode(box, "李四");  
createNode(box, "王五");  
createNode(box, "赵六");  
createNode(box, "钱七");
```

```
final Comparator comparator = new Comparator() {  
    Comparator cmp = Collator.getInstance(java.util.Locale.CHINA);  
    public int compare(Object obj1, Object obj2) {  
        Element node1 = (Element) obj1;  
        Element node2 = (Element) obj2;  
        return cmp.compare(node1.getName(), node2.getName());  
    }  
};
```

```
TTree tree = new TTree(box);  
tree.setSortComparator(comparator);  
  
showFrame("tree sort", tree);  
showFrame("tree no sort", new TTree(box));
```

Sort by Chinese Pinyin



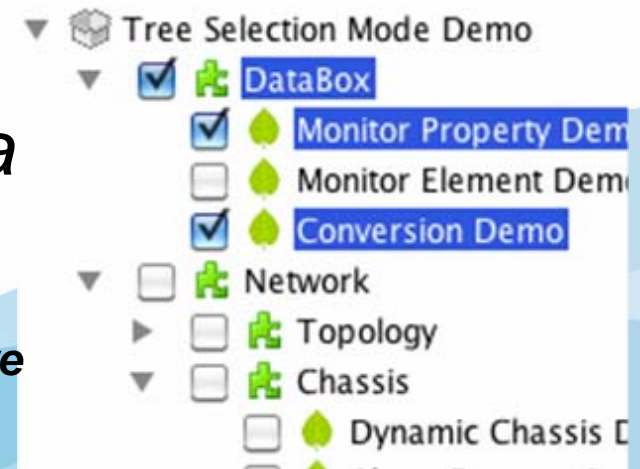
TTree Check Mode

TTree supports check selection mode, setting as follow:

TTree.setTTreeSelectionMode(int mode)

TTree has five selection modes, see demo: *TreeSelectionModeDemo.java*

```
tree.setTTreeSelectionMode(TTree.CHECK_SELECTION);tree.setPaintSelectedStateWhenChecked(true);
```



TElementTable

Twaver™

- **List all elements that in the dataBox, each row refers one element**
- **Configuration table by both API and XML**
- **Can sort the column, supports multi-column sorting**
- **Custom table by renderer and editor**

Configuration Table

TWaver™

- **Configuration table by two steps:**
- **Set element class**

```
public void setElementClass(Class elementClass)
```

- **Register element attributes**

```
public void registerElementClassXML(Class elementClass, String url)
```

```
public void registerElementClassXML(Class elementClass, InputStream inputStream)
```

```
public void registerElementClassAttributes(Class elementClass, List attributes)
```

```
table.registerElementClassXML(Person.class, "/resource/bean/table.xml");
```

```
table.setElementClass(Person.class);
```


TElementTable Example

```
TDataBox box = new TDataBox();
TElementTable table = new TElementTable(box);
table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);

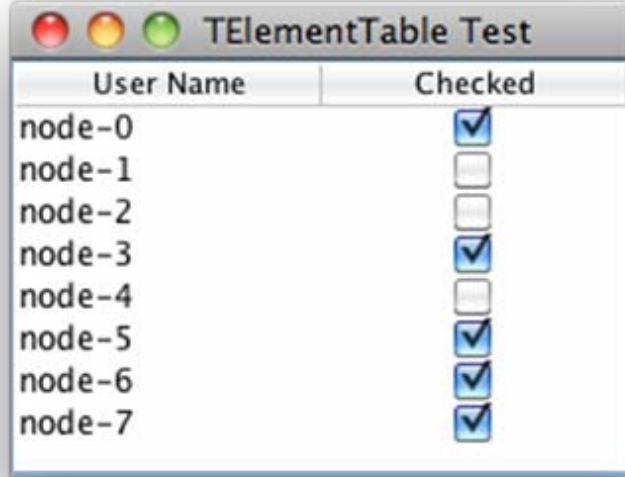
List attributes = new ArrayList();
ElementAttribute attribute = new ElementAttribute();
attribute.setUserPropertyKey("userName");
attribute.setDisplayName("User Name");
attributes.add(attribute);

attribute = new ElementAttribute();
attribute.setUserPropertyKey("checked");
attribute.setDisplayName("Checked");
attributes.add(attribute);

table.registerElementClassAttributes(Node.class, attributes);
table.setElementClass(Node.class);

for (int i = 0; i < 8; i++) {
    Node element = new Node();
    element.putUserProperty("userName", "node-" + i);
    element.putUserProperty("checked", TWaverUtil.getRandomBool());
    box.addElement(element);
}

TestUtil.showFrame("TElementTable Test", new JScrollPane(table));
```



User Name	Checked
node-0	<input checked="" type="checkbox"/>
node-1	<input type="checkbox"/>
node-2	<input type="checkbox"/>
node-3	<input checked="" type="checkbox"/>
node-4	<input type="checkbox"/>
node-5	<input checked="" type="checkbox"/>
node-6	<input checked="" type="checkbox"/>
node-7	<input checked="" type="checkbox"/>

Configuration Table by XML

TWaver™

Using XML configuration files can achieve the same functionality

```
<beaninfo>  
  <attribute  
    userPropertyKey="userName"  
    displayName="User Name"/>  
  <attribute  
    userPropertyKey="checked"  
    displayName="Checked"/>  
</beaninfo>
```

```
table.registerElementClassXML(Node.class, "/ppt/table/table.xml");  
table.setElementClass(Node.class);
```

TElementTable Order

- **Default order**

- By default, the later added elements listed to the bottom.

- Set converse order

- table.setConverseIncreaseOrder(true)*

- **Table has another order, hierarchy order**

- table.setIteratorByHierarchy(true)*

- call function `dataBox.move***(element)` can adjust the row order

Table Sort

Get table column from table model:

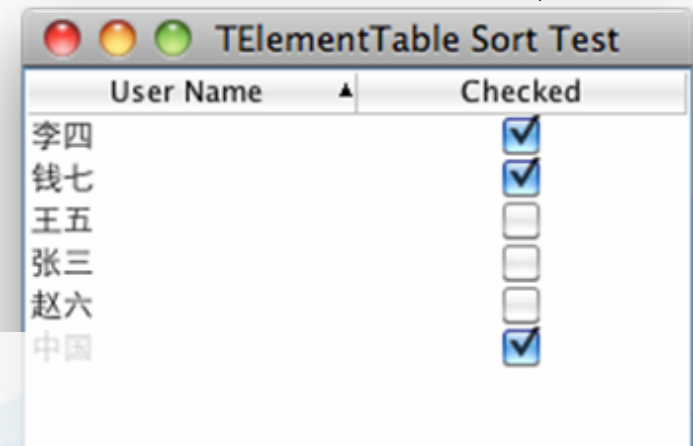
```
TTableModel#public TTableColumn getColumnByName(String name)
```

Sort column:

```
TTableModel#public void sortColumn(TTableColumn sortedColumn, int sortMode, boolean multiColumnSort)
```

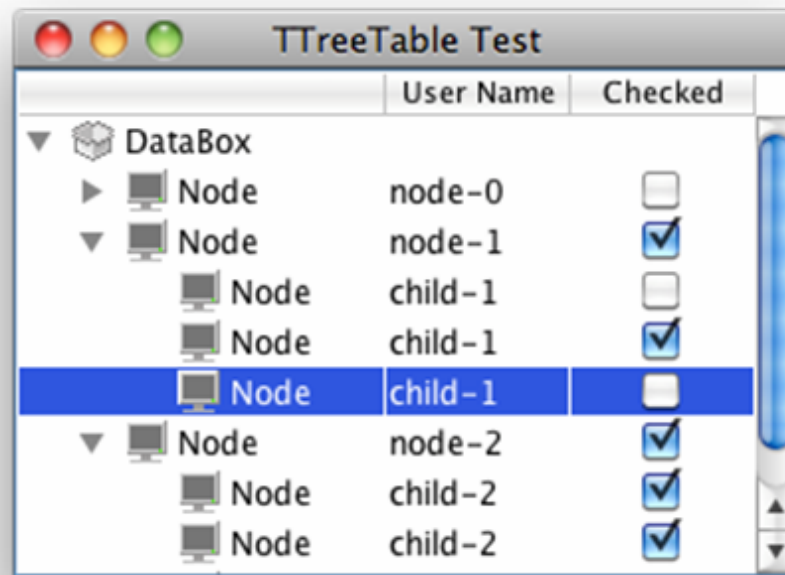
Example, sort the userName column:

```
TTableModel tableModel = table.getTableModel();  
TTableColumn column = tableModel.getColumnByName("userName");  
Collator comparator = Collator.getInstance(java.util.Locale.CHINA);  
column.setSortComparator(comparator);  
tableModel.sortColumn("userName", TTableModel.SORT_ASCENDING);
```



TTreeTable

- **TTreeTable extends from TElementTable, it has a tree column, it provides expand and merge tree node operation**



The Tree in TTreeTable

TWaver™

- **Get the tree:**

TTree tree = treeTable.getTree();

- **Expand or merge row by the tree:**

table.getTree().expandAll();

- **Get the tree column:**

TTableColumn column = treeTable.getTreeColumn();

TTreeTable Order

TWaver™

- Table row order is as same as the tree's.
- Table sorting is as same as the tree's

```
treeTable.setTreeColumnComparator(new Comparator() {  
    Comparator cmp = Collator.getInstance();  
    public int compare(Object obj1, Object obj2) {  
        Element node1 = (Element) obj1;  
        Element node2 = (Element) obj2;  
        return cmp.compare(node1.getName(), node2.getName());  
    }  
});
```

TPropertySheet

TMaver™

- **Sheet used for displaying and editing element properties**
- **Sheet configuration is as same as the table's**
- **Supports property group**

Sheet Example

Sheet configuration is as same as the table's

```
TDataBox box = new TDataBox();
TPropertySheet sheet = new TPropertySheet(box);

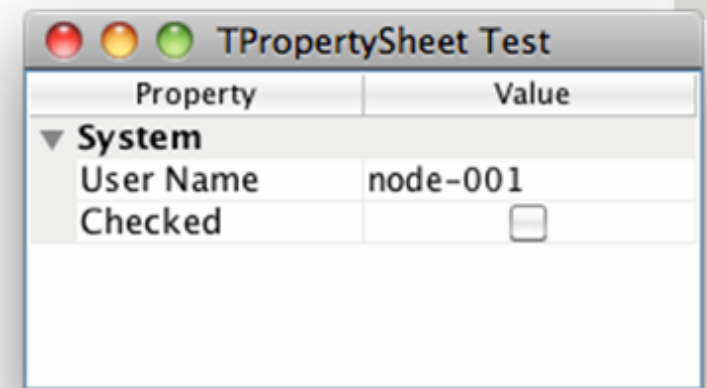
List attributes = new ArrayList();
ElementAttribute attribute = new ElementAttribute();
attribute.setUserPropertyKey("userName");
attribute.setDisplayName("User Name");
attributes.add(attribute);

attribute = new ElementAttribute();
attribute.setUserPropertyKey("checked");
attribute.setDisplayName("Checked");
attributes.add(attribute);

sheet.registerElementClassAttributes(Node.class, attributes);

Node element = new Node();
element.putUserProperty("userName", "node-001");
element.putUserProperty("checked", TWaverUtil.getRandomBool());
box.addElement(element);

box.getSelectionModel().setSelection(element);
showFrame("TPropertySheet Test", new JScrollPane(sheet));
```



Sheet Category

- By default, all properties are in “system” category
- Custom category by two steps: register category, then configuration element attribute.

```
Category.registerCategory("user", "User Info", true);
```

```
Category.registerCategory("basic", "Basic", true);
```

```
List categoryNames = new ArrayList();
```

```
categoryNames.add(0, "user");
```

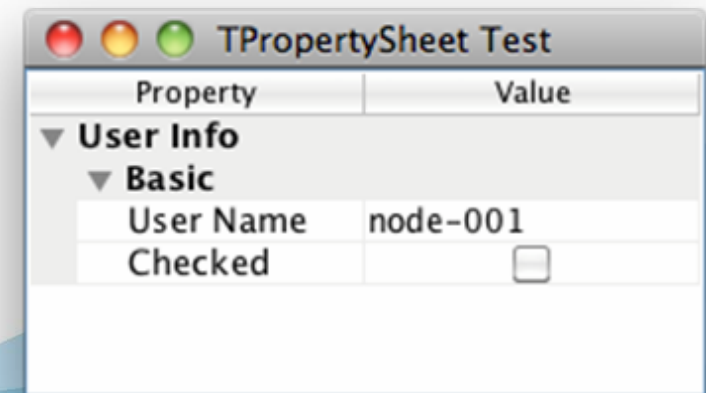
```
categoryNames.add(1, "basic");
```

```
ElementAttribute attribute = new ElementAttribute();
```

```
attribute.setCategoryNames(categoryNames);
```

```
attribute.setUserPropertyKey("userName");
```

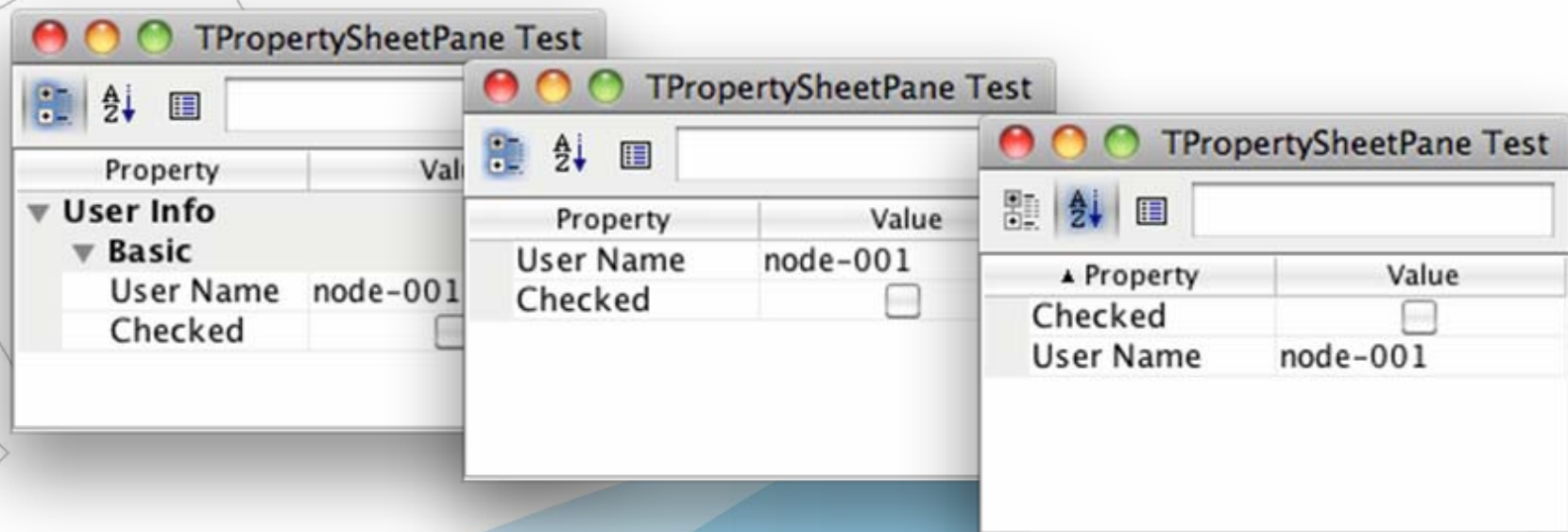
```
attribute.setDisplayName("User Name");
```



Sheet Panel

- **TPropertySheetPane** - it has a toolbar which with some shortcut buttons and other components, like: sort button, description button, category button, search box ...

```
TPropertySheetPane sheetPane = new TPropertySheetPane(sheet);  
showFrame("TPropertySheetPane Test", sheetPane);
```



Filter and Sort

TWaver™

- **Filter**

public void setElementAttributeVisibleFilter(ElementAttributeVisibleFilter
elementAttributeVisibleFilter)

- **Sort**

- Sort by property

public void setPropertySortingComparator(Comparator
propertySortingComparator)

- Sort by category

public void setCategorySortingComparator(Comparator
categorySortingComparator)

- **BarChart**
- **BubbleChart**
- **DialChart**
- **LineChart**
- **PercentChart**
- **PieChart**
- **RadarChart**

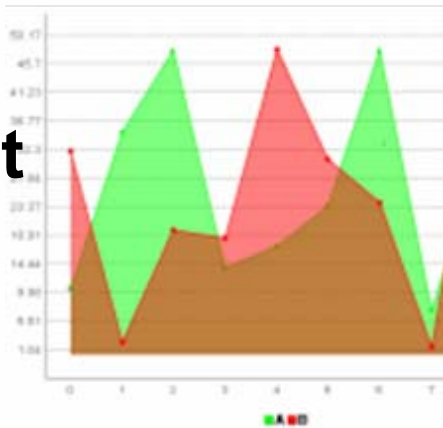
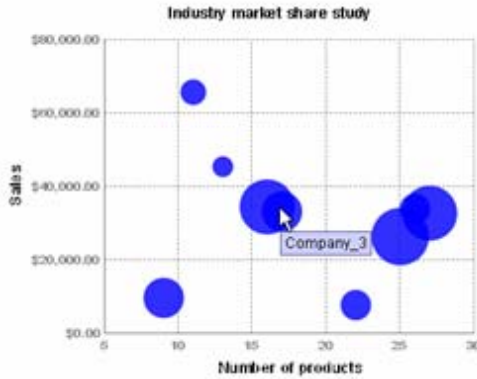
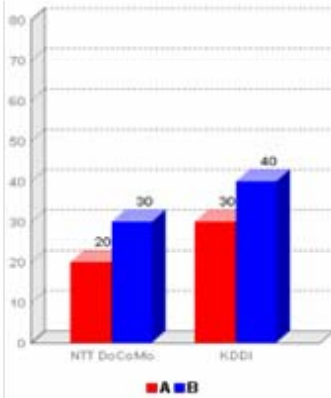


Chart Value & List Values

TWaver™

A element can have a single chart value or a set of chart values, setting these values like this:

twaver.Element#

```
//set single chart value
```

```
public void putChartValue(double value);
```

```
//only for bubble chart
```

```
public void addChartBubble(Bubble bubble);
```

```
//set a set of chart values
```

```
public void addChartValue(double value);
```

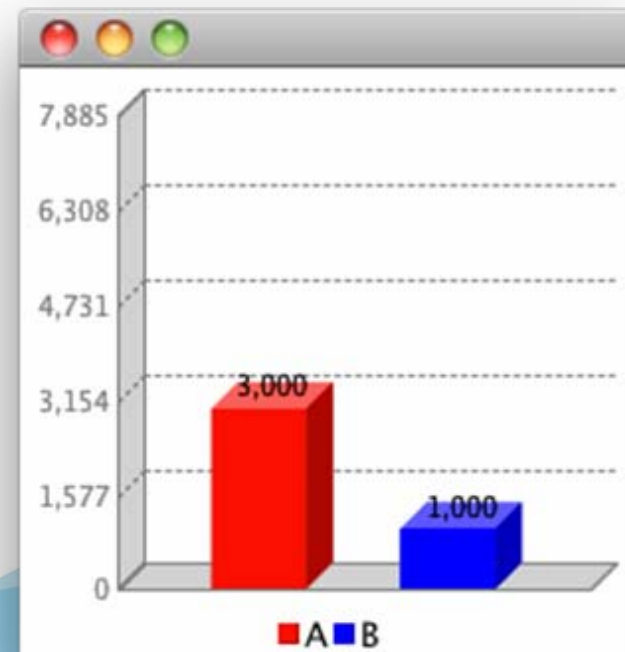
BarChart Example

```
TDataBox box = new TDataBox();
```

```
BarChart barChart = new BarChart(box);  
barChart.setYScaleTextVisible(true);  
barChart.setYScaleMinTextVisible(true);  
barChart.setUpperLimit(7885);  
barChart.setLowerLimit(0);  
barChart.setYScaleValueGap(1577);
```

```
Element A = new Node("A");  
A.setName("A");  
A.putChartColor(Color.RED);  
A.putChartValue(3000);  
box.addElement(A);
```

```
Element B = new Node("B");  
B.setName("B");  
B.putChartColor(Color.BLUE);  
B.putChartValue(1000);  
box.addElement(B);
```



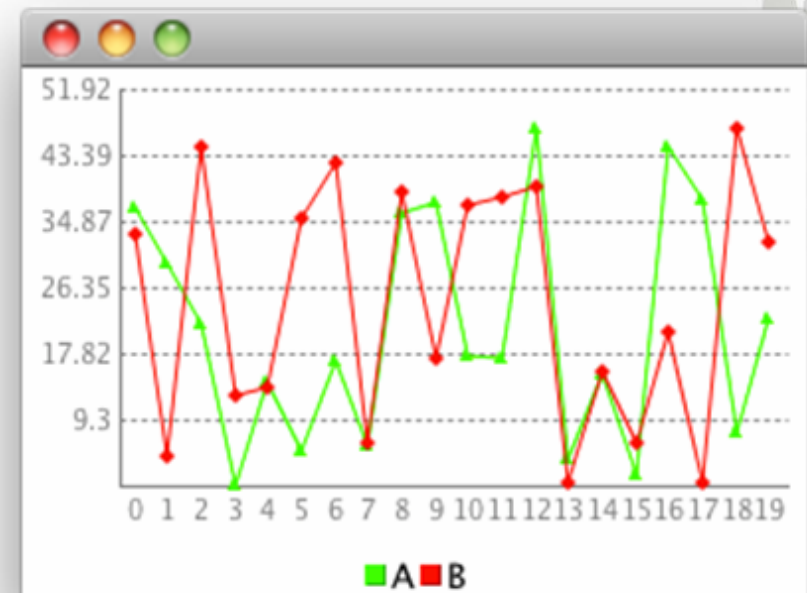
LineChart Example

```

TDataBox box = new TDataBox();
LineChart lineChart = new LineChart(box);
lineChart.setYAxisVisible(true);
lineChart.setYScaleTextVisible(true);
lineChart.setXAxisVisible(true);
lineChart.setXScaleTextVisible(true);
// Display markers for each data value.
lineChart.setInflexionVisible(true);

Element a = new Node();
a.setName("A");
a.putChartColor(Color.GREEN);
a.putChartInflexionStyle(TWaverConst.INFLEXION_STYLE_TRIANGLE);
box.addElement(a);
Element b = new Node();
b.setName("B");
b.putChartColor(Color.RED);
b.putChartInflexionStyle(TWaverConst.INFLEXION_STYLE_DIAMOND);
box.addElement(b);
for (int i = 0; i < 20; i++) {
    lineChart.addXScaleText("" + i);
    a.addChartValue(Math.random() * 50);
    b.addChartValue(Math.random() * 50);
}

```



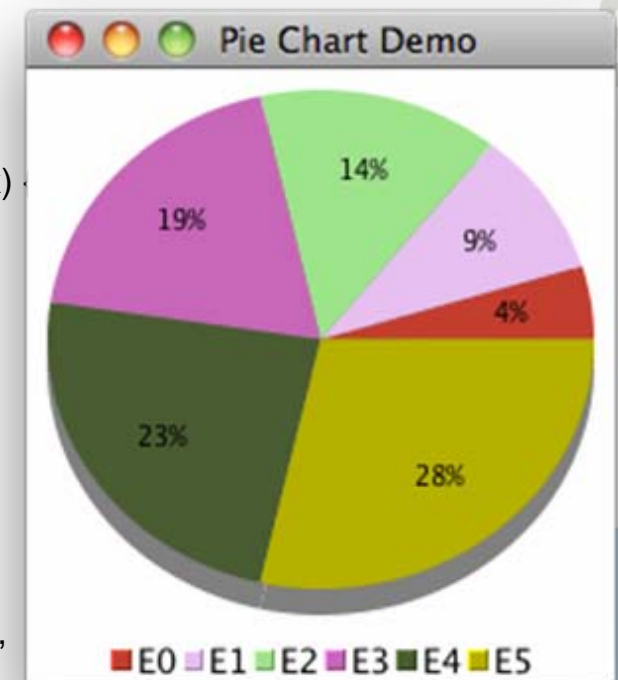
PieChart Example

```

TDataBox box = new TDataBox();
PieChart pieChart = new PieChart(box){
    protected String getFormattedText(Element element, double value, int index) {
        double proportion=element.getChartValue()/sum;
        return (int) (proportion * 100) + "%";
    }
};
Random random = new Random();
for (int i = 0; i < 6; i++) {
    Element element = new Node();
    element.setName("E" + i);
    element.putChartValue(10 * (i + 1));
    element.putChartColor(new Color(random.nextInt(255), random.nextInt(255),
    random.nextInt(255)));
    box.addElement(element);
}

TestUtil.showFrame("Pie Chart Demo", pieChart);

```



Alarm

- **Alarm - describes the element or device operation status**
- **Alarm/AlarmSeverity**
- **AlarmModel**
- **AlarmState/AlarmStateStatistics**
- **Alarm display**

TM
waver

AlarmSeverity

TWaver™

- **AlarmSeverity** - Reflect the urgency of alarms, it has this properties: name, nickName, value,color.
- **There are six alarm severities predefined.**

**CRITICAL, MAJOR, MINOR, WARNING,
INDETERMINATE, CLEARED**

- **Custom alarm severities as follows:**

```
AlarmSeverity.clearAlarmSeverity();  
AlarmSeverity.registerAlarmSeverity("a", "a", 1, Color.RED,  
"AAA");
```

Custom Alarm Severities Example

```
public static void main(String[] args) throws Exception {
```

```
    AlarmSeverity.clearAlarmSeverity();
```

```
    AlarmSeverity a = AlarmSeverity.registerAlarmSeverity("a", "a", 1, Color.RED, "AAA");
```

```
    AlarmSeverity b = AlarmSeverity.registerAlarmSeverity("b", "b", 2, Color.BLUE, "BBB");
```

```
    AlarmSeverity c = AlarmSeverity.registerAlarmSeverity("c", "c", 3, Color.GREEN, "CCC");
```

```
    TDataBox box = new TDataBox();
```

```
    Group node1 = new Group();
```

```
    node1.setExpand(true);
```

```
    Node node2 = new Node();
```

```
    node2.setLocation(100, 100);
```

```
    Node node3 = new Node();
```

```
    node3.setLocation(200, 150);
```

```
    node3.setParent(node1);
```

```
    node2.setParent(node1);
```

```
    box.addElementWithDescendant(node1);
```

```
    addAlarm(box.getAlarmModel(), node1, a);
```

```
    addAlarm(box.getAlarmModel(), node2, b);
```

```
    addAlarm(box.getAlarmModel(), node3, c);
```

```
    TestUtil.showFrame("Custom AlarmSeverity Demo", new TNetwork(box));
```

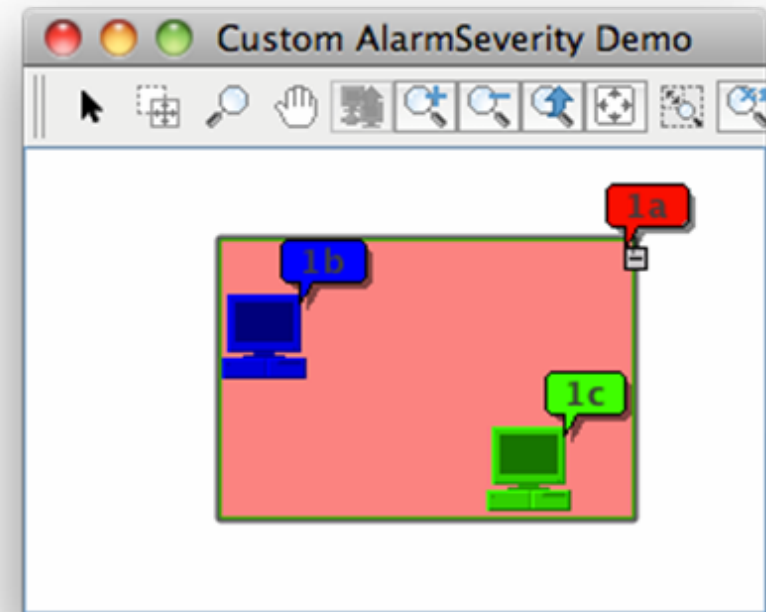
```
}
```

```
private static void addAlarm(AlarmModel alarmModel, Element element, AlarmSeverity severity){
```

```
    Alarm alarm = new Alarm(element.getID(), severity);
```

```
    alarmModel.addAlarm(alarm);
```

```
}
```



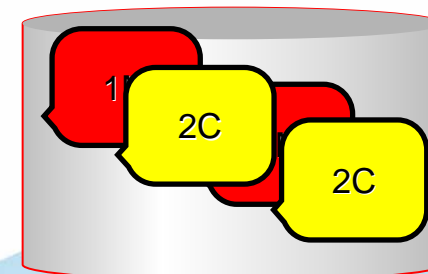
Alarm Object

Twaver™

- **twaver.Alarm** - the basic data type of alarm.
- **Alarm properties:**
elementID、 **alarmSeverity**、 **alarmID**、 **acked**、
cleared、 **alarmState**
#putClientProperty(key, value)

AlarmModel

- **AlarmModel** - Data container for managing alarm objects.
- **Basic Operations**
 - addAlarm/addAlarms
 - removeAlarm***
- **Quick finder and listeners**



AlarmBox

AlarmState

- **AlarmState** - describes current alarm status of element or devices
- Includes how many new alarms, how many acked alarms and so on.
- Any specific alarm object(`twaver.Alarm`) is not included.

Twaver™

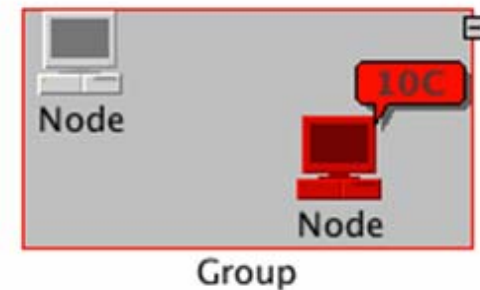
AlarmStateStatistics

TM
waver

- **AlarmStateStatistics - AlarmStateStatistics - Counts the alarm states of all elements in the ElementBox.**
- **Includes all quantities of alarms for each alarm severity**
- **Includes the numbers of all new alarms or acknowledged alarms.**
- **Supports filter, can statistics for specifying element's.**

Alarm Display

- Alarm balloon
- Alarm color - fill or outline
- Alarm table
- Alarm statistics charts



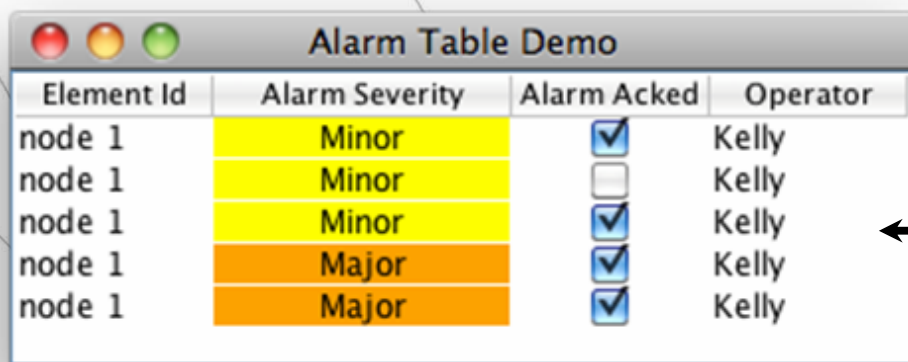
Element Id	Alarm Severity	Alarm Acked	Operator
node 1	Minor	<input checked="" type="checkbox"/>	Kelly
node 1	Minor	<input type="checkbox"/>	Kelly
node 1	Minor	<input checked="" type="checkbox"/>	Kelly
node 1	Major	<input checked="" type="checkbox"/>	Kelly
node 1	Major	<input checked="" type="checkbox"/>	Kelly

Severity	New	Acked	Total
Critical	0	0	0
Major	0	1	1
Minor	1	1	2
Warning	2	0	2
Indetermin...	0	1	1
Cleared	0	0	0
Total	3	3	6

TAlarmTable

TWaver™

- TAlarmTable displays alarm information in AlarmModel, constitutes a MV relationship with AlarmModel
- Similar use as TElementTable, have features of sorting, filtration, etc.



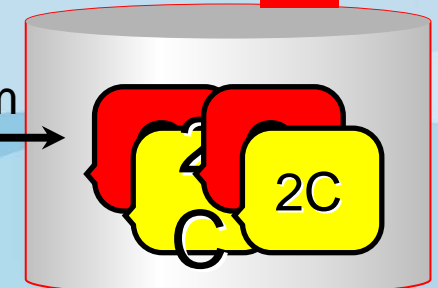
Alarm Table Demo

Element Id	Alarm Severity	Alarm Acked	Operator
node 1	Minor	<input checked="" type="checkbox"/>	Kelly
node 1	Minor	<input type="checkbox"/>	Kelly
node 1	Minor	<input checked="" type="checkbox"/>	Kelly
node 1	Major	<input checked="" type="checkbox"/>	Kelly
node 1	Major	<input checked="" type="checkbox"/>	Kelly

V

each row means a alarm

M



AlarmModel

TAlarmTable Example

TWaver™

```
TDataBox box=new TDataBox();  
Node node = new Node("node 1");  
box.addElement(node);
```

```
TAlarmTable alarmTable = new TAlarmTable(box, new TTableColumn[]{  
    new TTableColumn(Alarm.PROPERTY_ELEMENTID, "Element Id"),  
    new TTableColumn(Alarm.PROPERTY_ALARMSEVERITY, "Alarm Severity"),  
    new TTableColumn(Alarm.PROPERTY_ACKED, "Alarm Acked"),  
    new TTableColumn("operator", "Operator"),  
    new TTableColumn(Alarm.PROPERTY_RAISEDTIME, "Raised Time")  
});
```

alarmTable.setConverseIncreaseOrder(true):

```
AlarmModel alarmModel = box.getAlarmModel();  
for (int i = 0; i < 5; i++) {  
    AlarmSeverity severity = TWaverUtil.getRandomNonClearedSeverity();  
    Alarm alarm = new Alarm(node.getID(), severity);  
    alarm.setAcked(TWaverUtil.getRandomBool());  
    alarm.setRaisedTime(new Date(System.currentTimeMillis() + i*3600));  
    alarm.putClientProperty("operator", TWaverUtil.getRandomBool() ? "Sam" : "Kelly");  
    box.getAlarmModel().addAlarm(alarm);  
}  
TestUtil.showFrame("Alarm Table Demo", new JScrollPane(alarmTable));
```

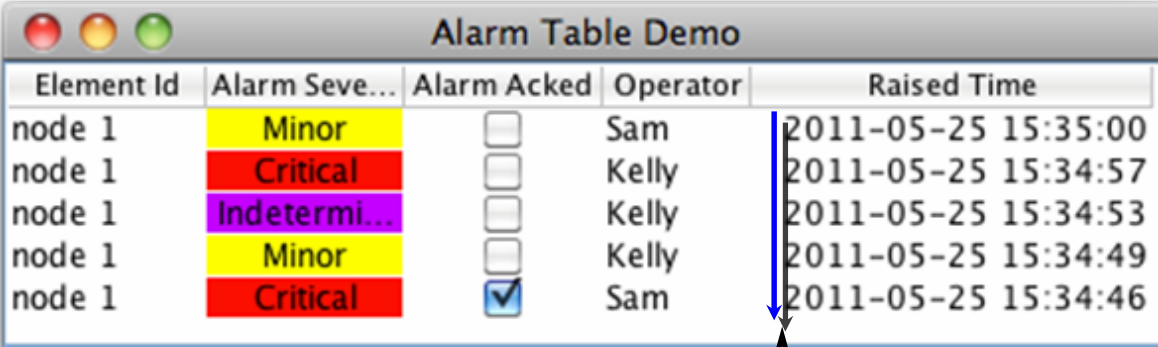
set display columns

set converse order

add alarm to
AlarmModel

TAlarmTable Example

TWaver™



A screenshot of a window titled "Alarm Table Demo". It contains a table with five columns: "Element Id", "Alarm Seve...", "Alarm Acked", "Operator", and "Raised Time". The table has five rows of data. The "Alarm Seve..." column contains colored text: yellow for "Minor", red for "Critical", and purple for "Indetermi...". The "Alarm Acked" column contains checkboxes, with the last one checked. The "Operator" column contains names: Sam, Kelly, Kelly, Kelly, and Sam. The "Raised Time" column contains timestamps from 2011-05-25 15:35:00 down to 2011-05-25 15:34:46. A blue arrow points from the "converse order" callout to the "Raised Time" column.

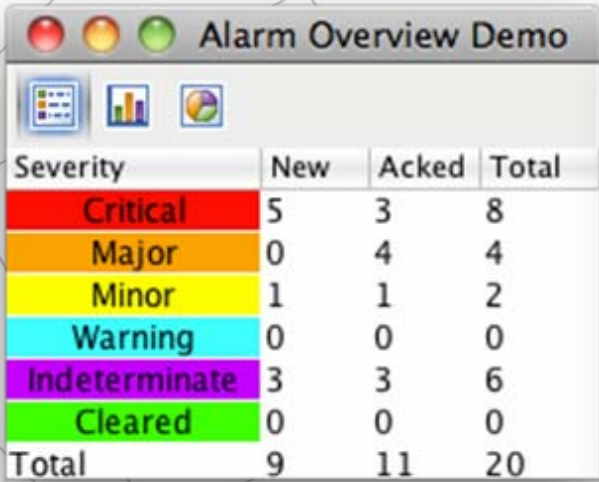
Element Id	Alarm Seve...	Alarm Acked	Operator	Raised Time
node 1	Minor	<input type="checkbox"/>	Sam	2011-05-25 15:35:00
node 1	Critical	<input type="checkbox"/>	Kelly	2011-05-25 15:34:57
node 1	Indetermi...	<input type="checkbox"/>	Kelly	2011-05-25 15:34:53
node 1	Minor	<input type="checkbox"/>	Kelly	2011-05-25 15:34:49
node 1	Critical	<input checked="" type="checkbox"/>	Sam	2011-05-25 15:34:46

converse order

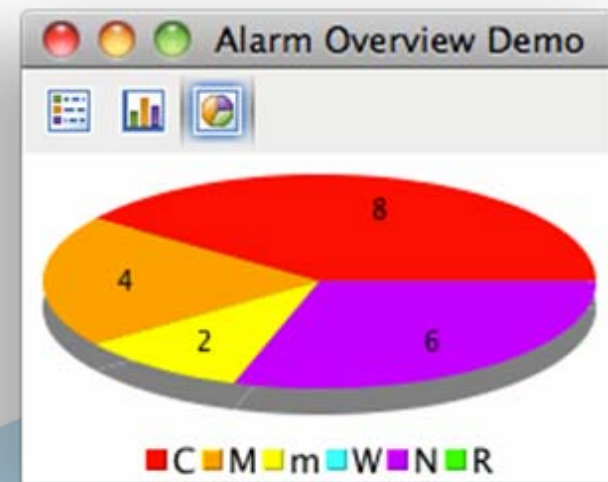
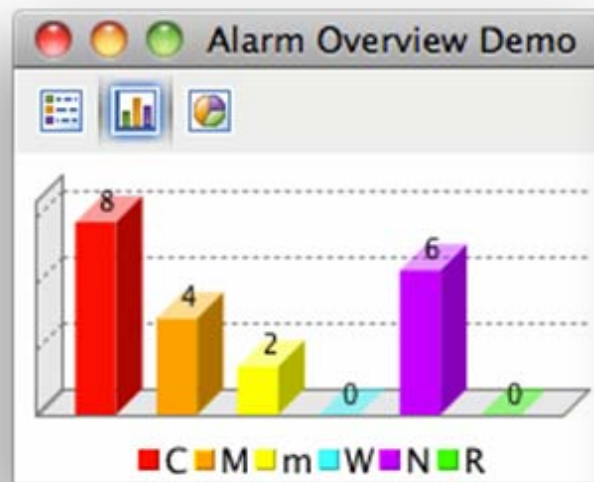
```
alarmTable.setConverseIncreaseOrder(true);
```

TAlarmOverview

- **TAlarmOverview** - presents alarm state statistics informations by table and charts



Severity	New	Acked	Total
Critical	5	3	8
Major	0	4	4
Minor	1	1	2
Warning	0	0	0
Indeterminate	3	3	6
Cleared	0	0	0
Total	9	11	20



TAlarmOverview Example

```

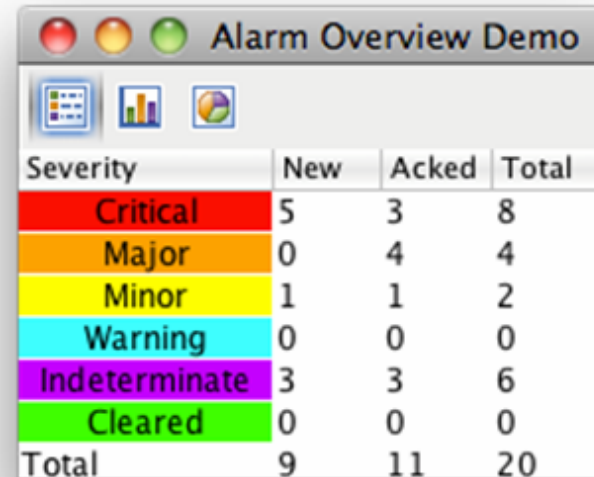
TDataBox box=new TDataBox();
Node node = new Node("node 1");
box.addElement(node);
Node node2 = new Node("node 2");
box.addElement(node2);

for (int i = 0; i < 20; i++) {
    AlarmSeverity severity = TWaverUtil.getRandomNonClearedSeverity();
    Alarm alarm = new Alarm((TWaverUtil.getRandomBool() ? node : node2).getID(), severity);
    alarm.setAcked(TWaverUtil.getRandomBool());
    alarm.putClientProperty("operator", TWaverUtil.getRandomBool() ? "Sam" : "Kelly");
    box.getAlarmModel().addAlarm(alarm);
}

TAlarmOverview overview = new TAlarmOverview(box);

TestUtil.showFrame("Alarm Overview Demo", overview);

```



Severity	New	Acked	Total
Critical	5	3	8
Major	0	4	4
Minor	1	1	2
Warning	0	0	0
Indeterminate	3	3	6
Cleared	0	0	0
Total	9	11	20

Thank you

- Home - ServaSoftware.com
- Email - tw-service@servasoft.com