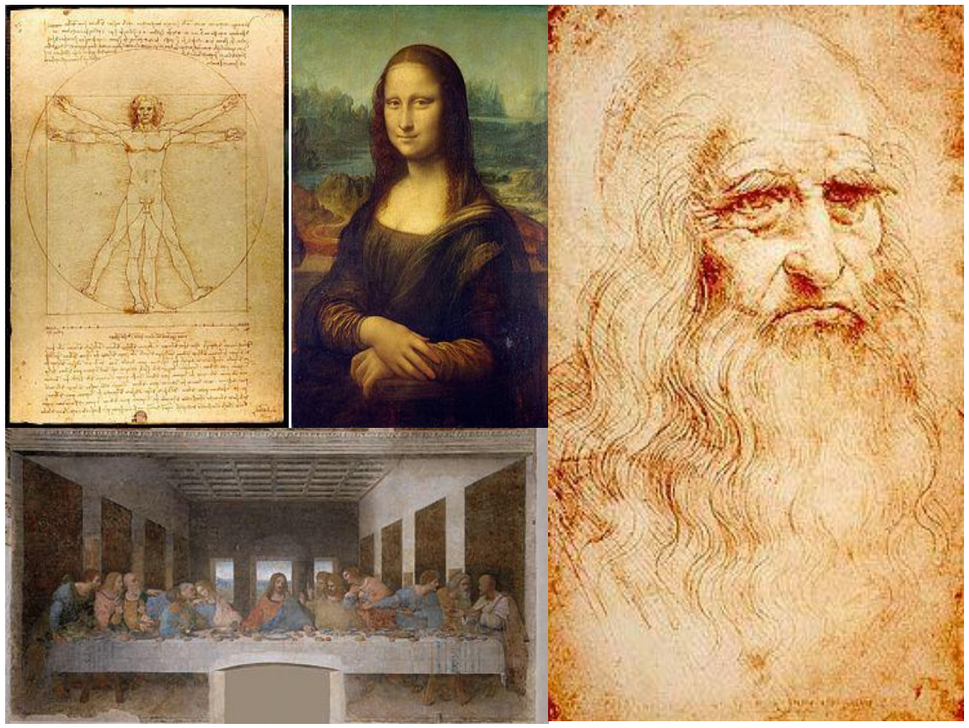


DaVinci: A lightweighted Information Visualization Framework V1.0

Nan Cao(nan.cao@gmail.com)

Aug. 2011

Related Work

- Graphic Toolkits
  - Java 2D, Java 3D, OpenGL, Piccolo, Jazz

- InfoVis Toolkits
  - XML Toolkit, xAnVi, InfoVis Toolkit, Prefuse
  - Peony : developed with Martin Wattenberg, Michelle X Zhou's teams for 2 years. Wide deployed in IBM CRL and Watson, all the CRL visualization solutions are using Peony, ManyEyes use its data interfaces.
  - DaVinci : Open source project for my Ph.D research only

- GraphVis Toolkits
  - See "Graph Drawing Algorithms In Visualization Designs"

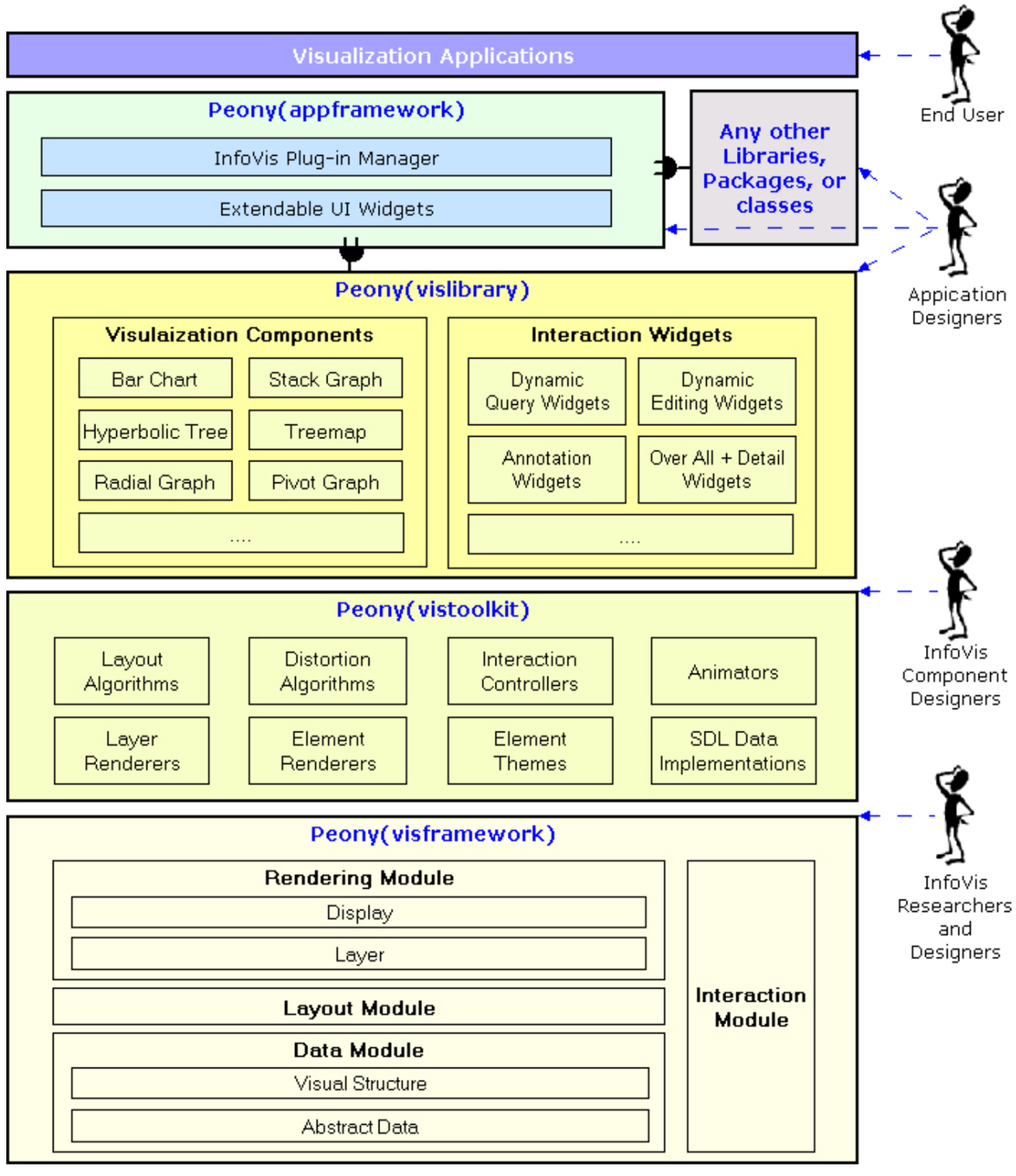
- Why design DaVinci?
  - Need a framework with the ***simplest interfaces*** that facilitate ***fast learning***

Setup DaVinci (Java Version)

- Download
  - JDK: http://www.oracle.com/technetwork/java/javase/downloads/index.html
  - Eclipse: http://www.eclipse.org/downloads/
  - DaVinci: http://www.cse.ust.hk/~nancao/architecture.html
- Install
  - Setup Java
  - Open Eclipse
  - Unzip DaVinci package
  - File → Import → Existing Project into Workspace → Browse → Finish

InfoVis Architecture

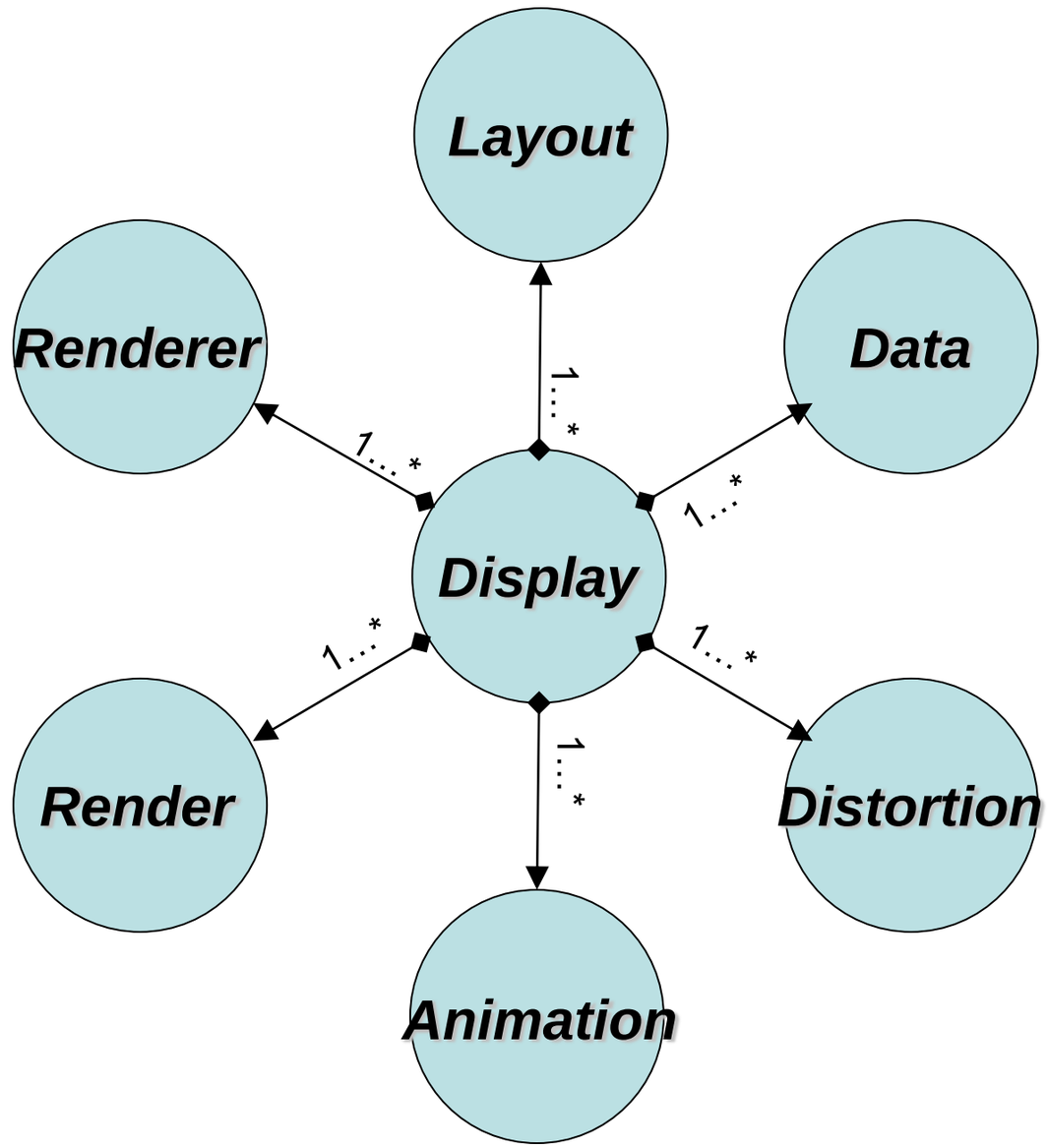
- Concepts:
  - Tool and Toolkit
  - Vjit and Vjit lib
  - Application
- Architecture
  - Application framework
  - Component library
  - Vis toolkit
  - Vis framework



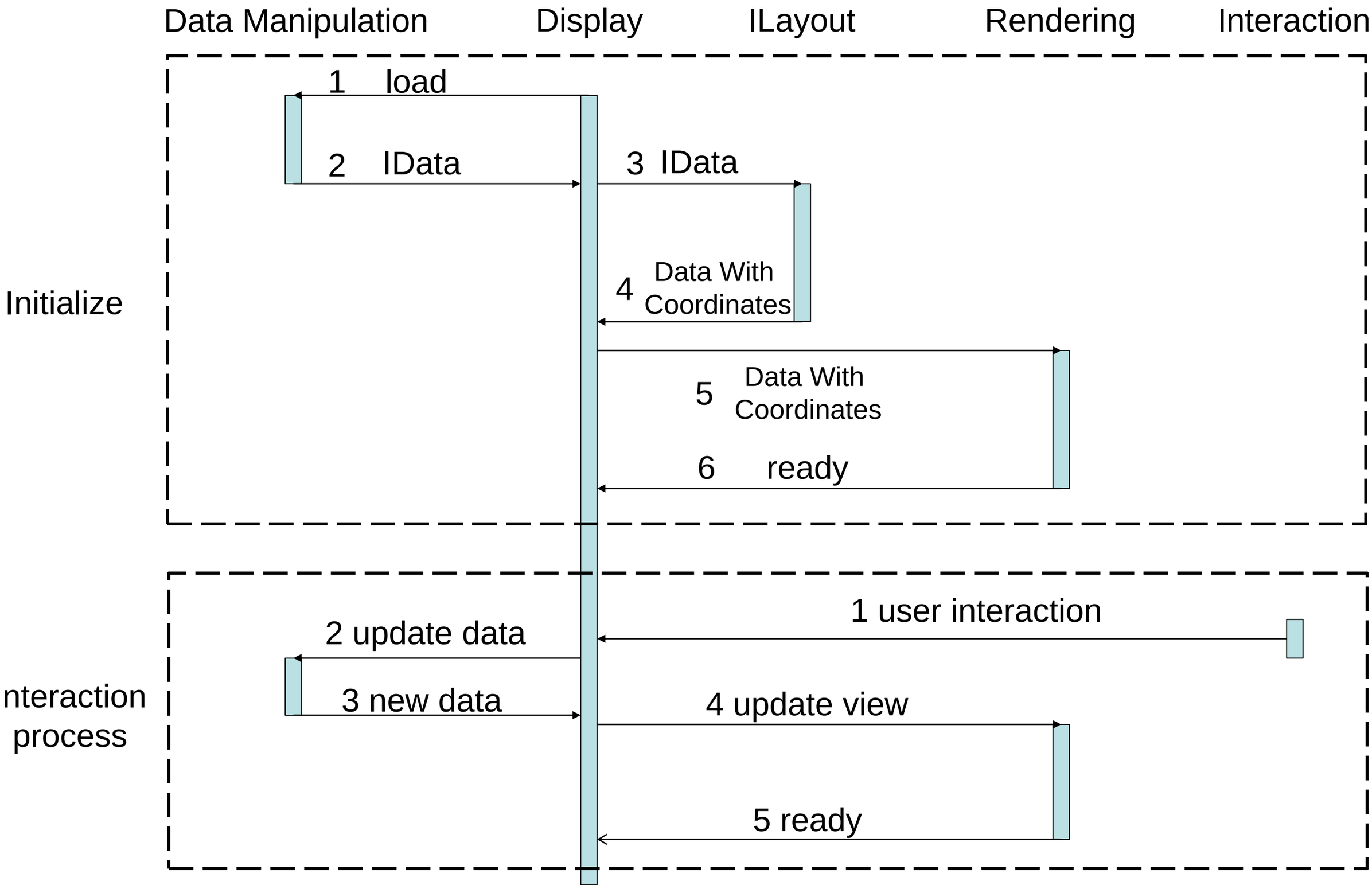
Reference: Peony

DaVinci Framework : Module View

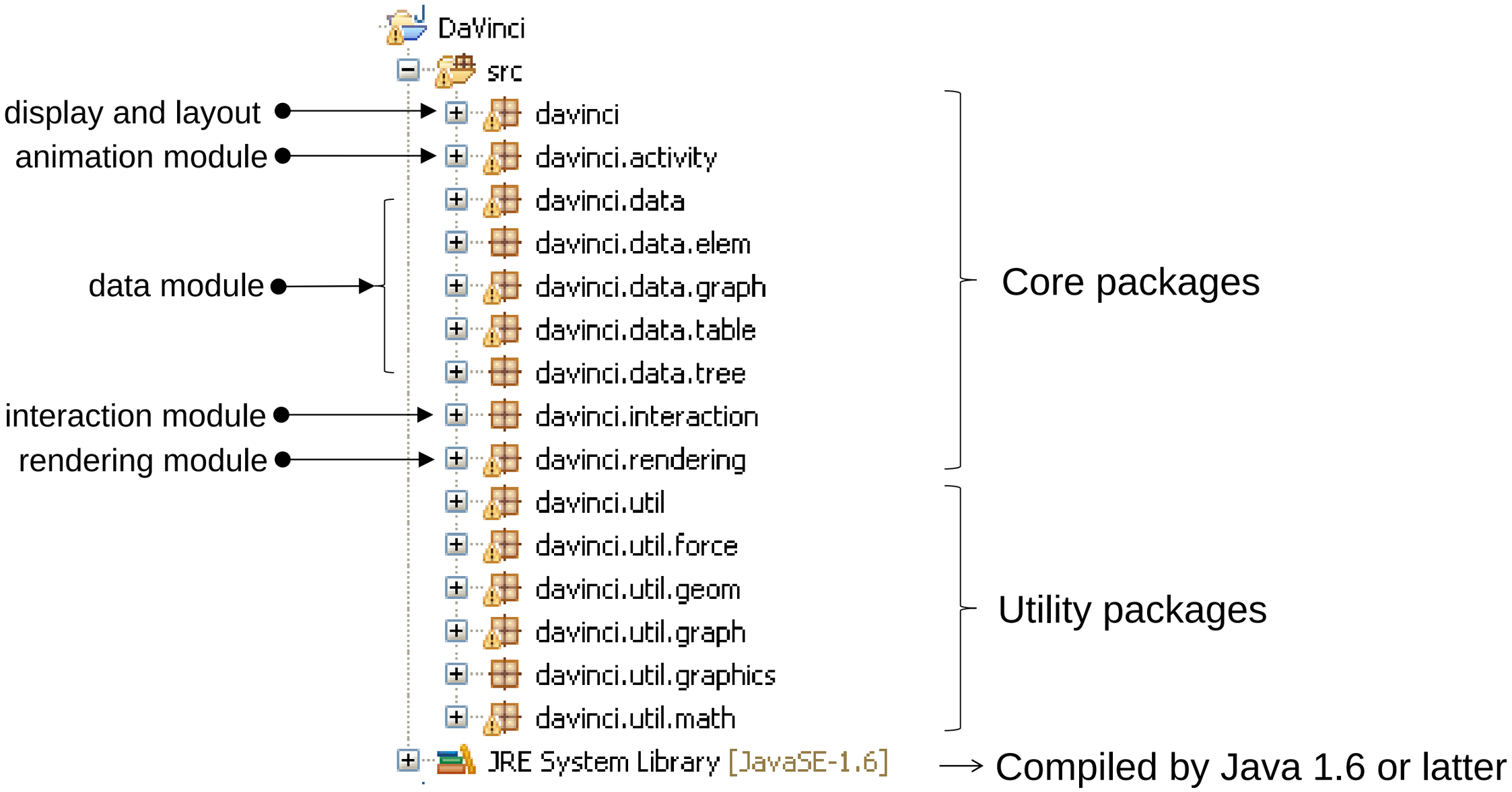
- V1.0 Download:
- Aggregation oriented design patterns:
  - Data structures
  - Layout
  - Render
  - Interaction
  - Animation
  - Distortion



DaVinci Framework : Sequence View



Java Packages



202 kb when ziped

Code Example: TestVjit.java

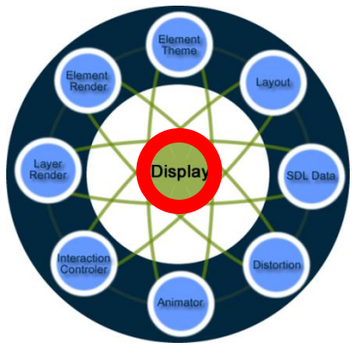
```java
package vis.vjit.test;

import vis.vjit.test.action.TestAction;

public class TestVjit extends Display {

    private static final long serialVersionUID = 484591750534155O575L;

    private Activity m_animator = null;

    public TestVjit() {

        m_animator = new TestAnimator();

        // register layout algorithm
        this.addLayout(new TestLayout());

        // add interactions
        this.addAction(new TestAction());

        // add elemnt finder
        this.setElemFinder(new TestElemFinder());

        // register renders
        this.setDisplayRender(new TestVjitRender());
        this.addElemRender("node", new TestNodeRender());
        this.addElemTheme("node", new TestNodeTheme());
        this.addElemRender("edge", new TestEdgeRender());
        this.addElemTheme("edge", new TestEdgeTheme());
    }

    public void doLayout() {

        super.doLayout();

        m_animator.setDisplay(this);
        m_animator.setStartTime(System.currentTimeMillis());
        this.getActivityManager().addActivity(m_animator);
    }
}
```

Assemble a visualization vjit (< 15 line codes)

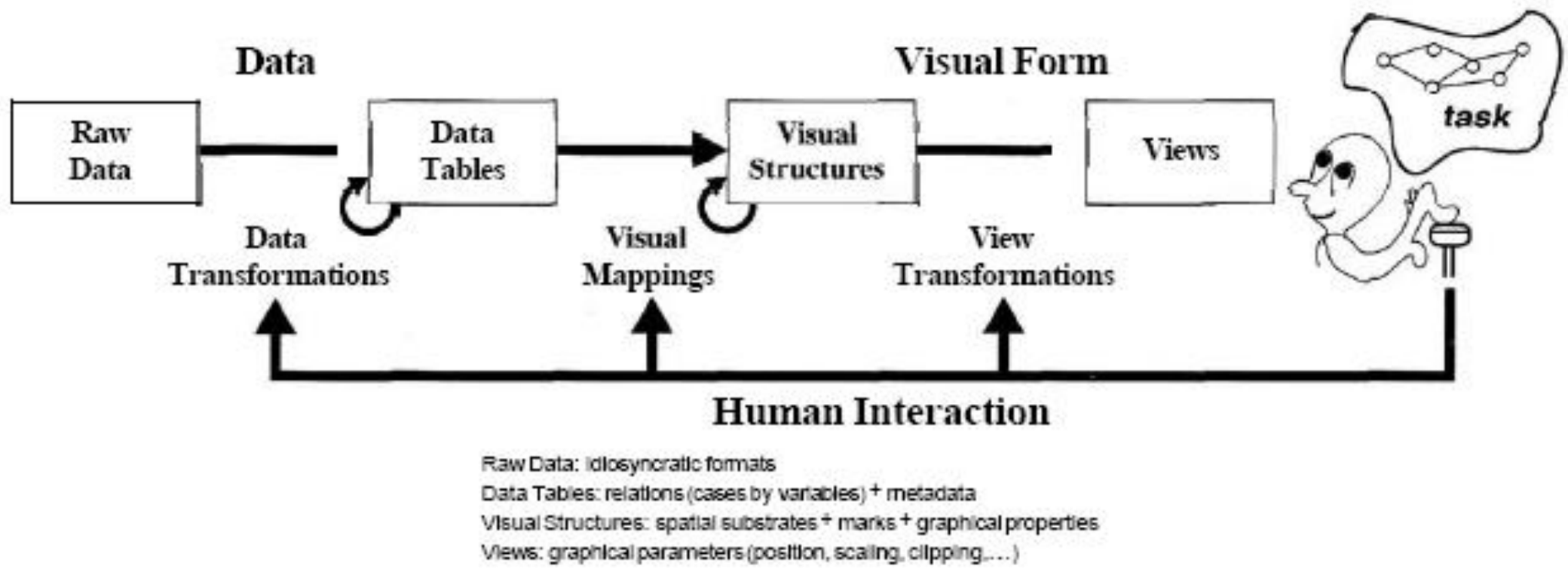
Code Example: TestDemo.java

```
package vis.vjit.test;

import java.awt.BorderLayout;

public class TestDemo extends JFrame {

    private static final long serialVersionUID = 797644095066587826 9L;

    public TestDemo() {

        // initiate your visualization
        TestVjit vjit = new TestVjit();
        vjit.setBackground(Color.white);

        // prepare the data
        Graph<IVisualNode> graph = GraphLib.getGrid(5, 5);
        graph.setID("mygraph");
        vjit.addData(graph);

        this.getContentPane().setLayout(new BorderLayout());
        this.getContentPane().add(vjit, BorderLayout.CENTER);
        this.setSize(1024, 768);
        this.setTitle("This is test using DaVinci");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        TestDemo demo = new TestDemo();
        demo.setVisible(true);
    }

}
```

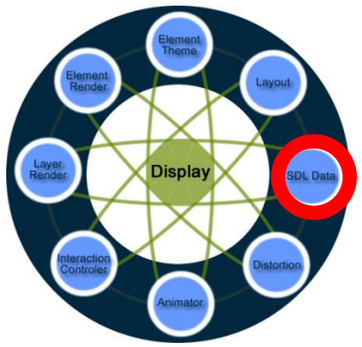
Use a vjit in an application (< 10 line codes)



Display

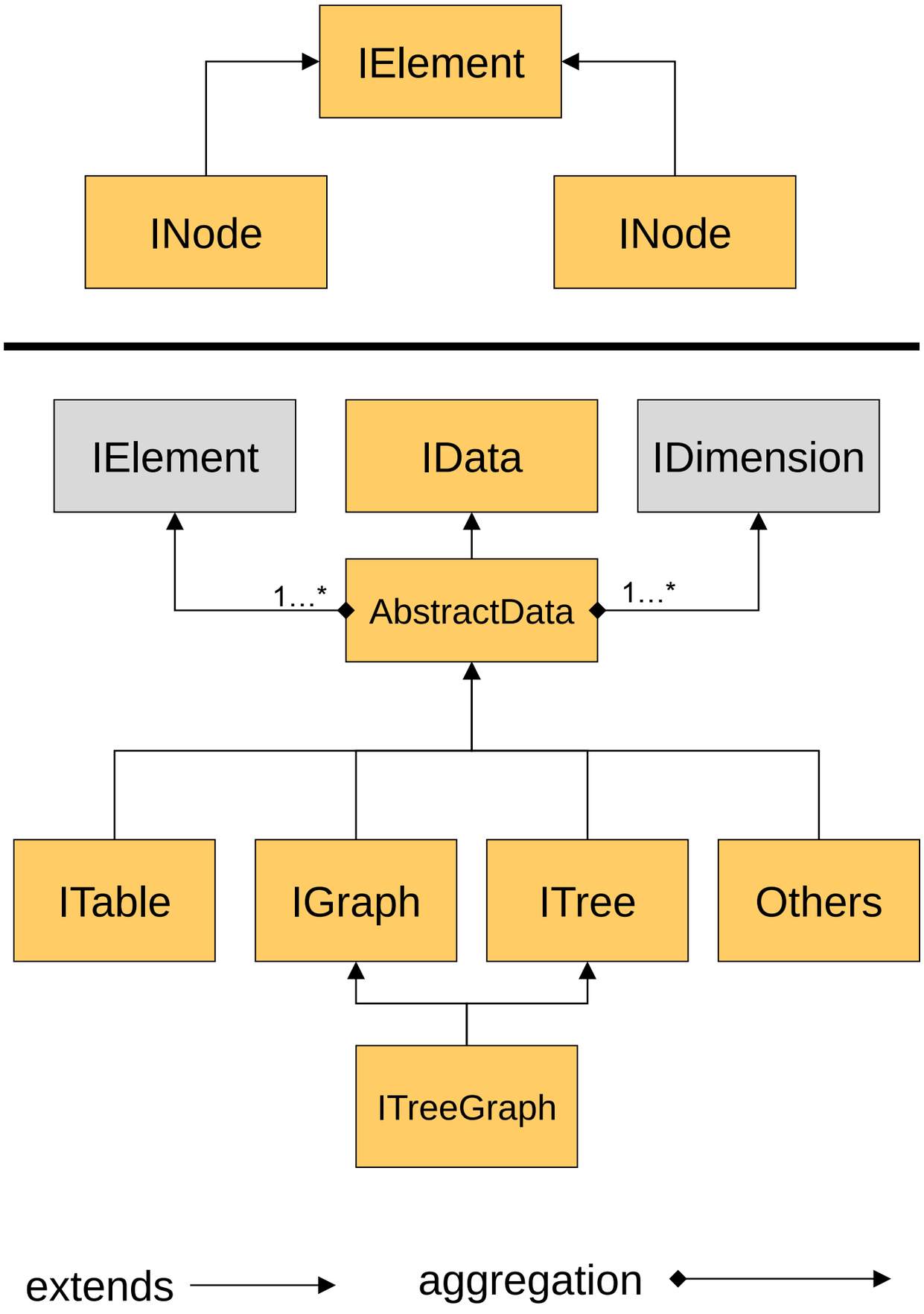


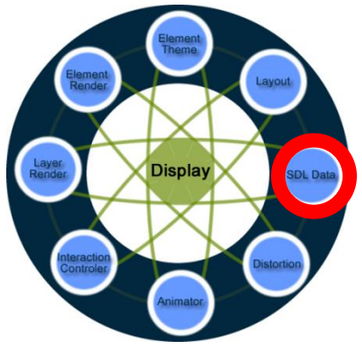
- Handles all the logic transitions of the visualization pipeline
- Displays all the visualization layout and rendering results
- Captures all the mouse and keyboard inputs
- The interface with Java-Swing (extends from JPanel)



Data Module

- Element
  - Node
  - Edge
  - Others (defined by users)
- Dimension
- Standard data structures
  - table
  - graph
  - tree
- Others
  - designed by users when necessary

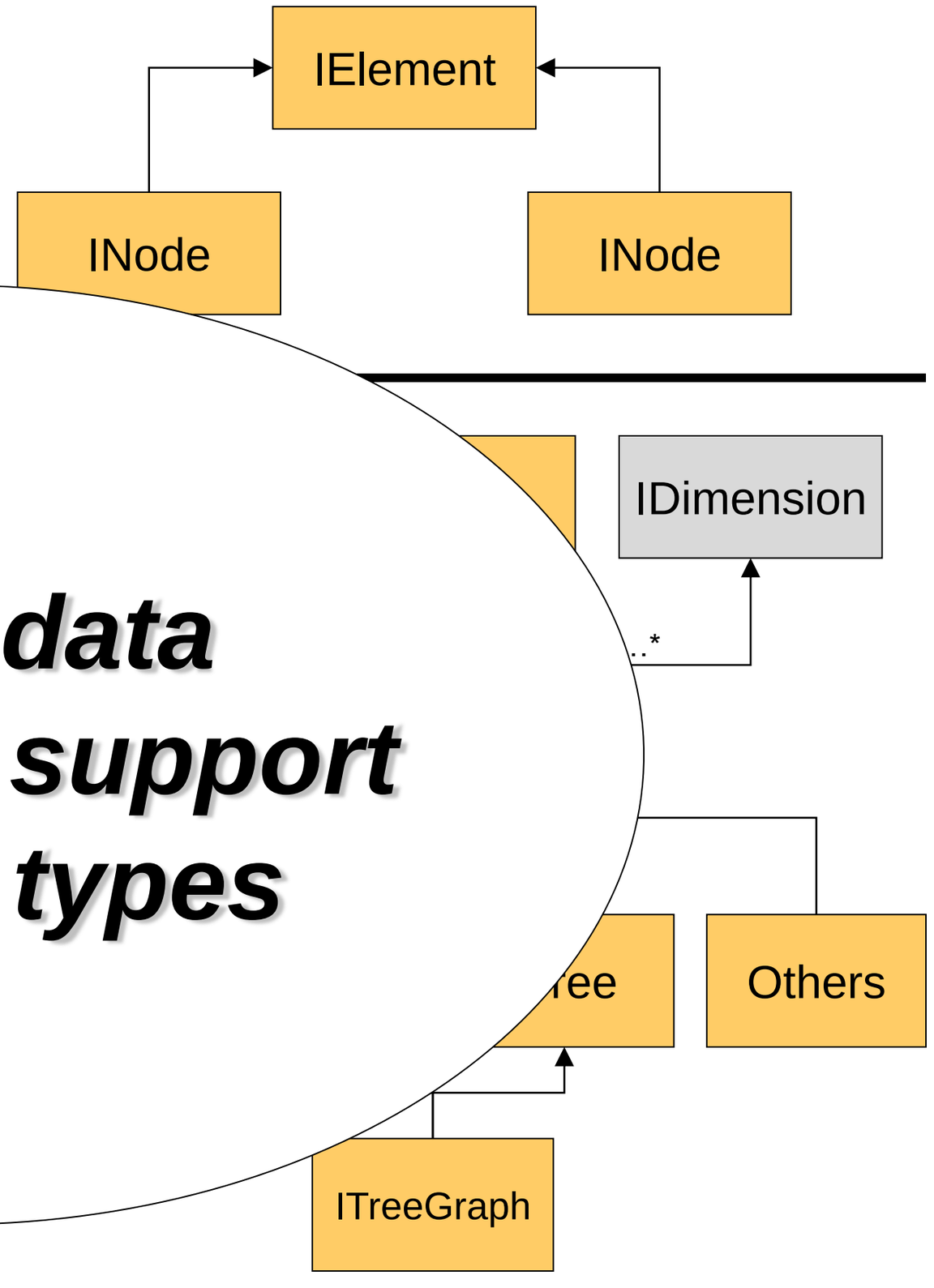




Data Module

- Element
  - Node
  - Edge
  - Others
- Dimens
- Standa
  - table
  - graph
  - tree
- Others
  - designed by users wh… necessary

All the data structures support generic types

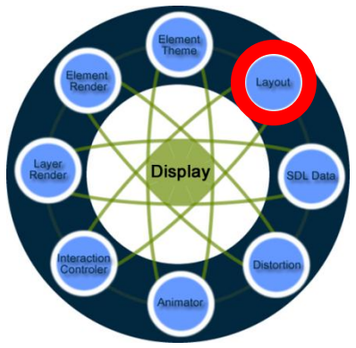


Data Module: Code Example

add data dimensions

add data elements

```
1   Graph<INode> graph = new Graph<INode>()
2   // add node dimensions
3   graph.addNodeDimension(new Dimension( “dim0”, “dimension 0”, “TEXT”));
4   ……
5   graph.addNodeDimension(new Dimension( “dimn”, “dimension n”, “NUMERICAL”));
6   // add edge dimensions
7   graph.addEdgeDimension(new Dimension( “edim1”, “dimension 0”, “NUMERICAL”));
8   ……
9   graph.addEdgeDimension(new Dimension( “edimn”, “dimension n”, “NUMERICAL”));
10
11  // creat a new node
12  for(int i = 0; i < 100; ++i){
13          INode node = new AnyUserDefinedNode();
14
15          // add attributres for the new node
16          node.add(dim0, value-0);
17          ……
18          node.add(dimn. value-n);
19
20          // add the node into the graph
21          graph.addNode(node);
22  }
23  ……
24  // add edge between node1 and node2
25  IEdge edge = graph.addEdge(node1, node2);
26  //add edge attributes
27  edge.add(edim1, value-0)
```



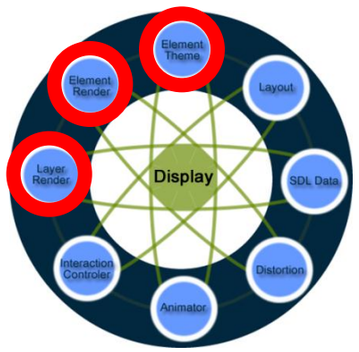
Layout Module

- ILayout interface
  - All the layout algorithms should implement this interface
  - When the window is initialized or size is changed, layout() method will be automatically called
  - Reset() method should be called by users when necessary. It resets the layout to the initial status

ILayout
• layout(Display disp) • reset()

Code Example: TestLayout.java

```
public class TestLayout implements ILayout {

    public TestLayout() {
    }

    public String getName() {
        return "TestLayout";
    }

    public void layout(Display disp) {

        Graph<IVisualNode> graph = (Graph<IVisualNode>)disp.getData("mygraph");
        if(null == graph) {
            return;
        }

        VisualNode node = null;
        int ncnt = graph.getNodeCount();
        for(int i = 0; i < ncnt; ++i) {
            node = (VisualNode)graph.getNode(i);
            node.setX(Math.random() * disp.getWidth());
            node.setY(Math.random() * disp.getHeight());
            node.setWidth(20);
            node.setHeight(10);
        }
    }

    public void reset() {
    }

}
```



Rendering Module

- IElemRender
  - Defines the painting logic of one single element and paint the geometry properties of visual elements. (Shape, Lines, Curves, Text)

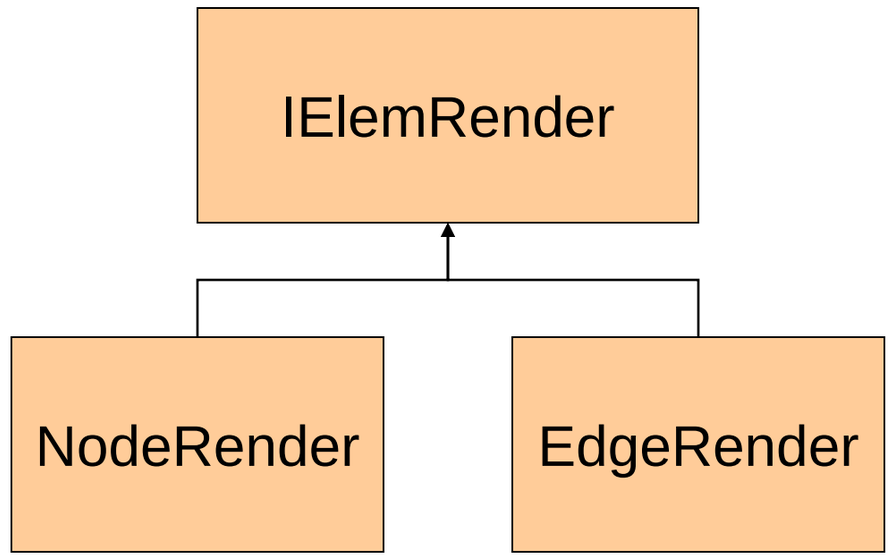
- IElemTheme
  - Defines the painting style of the elements. (The thickness of the borders, the finll color, border color, text font)

- IDisplayRender
  - Defines the painting pipeline of all the elements
  - FIFO, stack, tree, user defined orders.

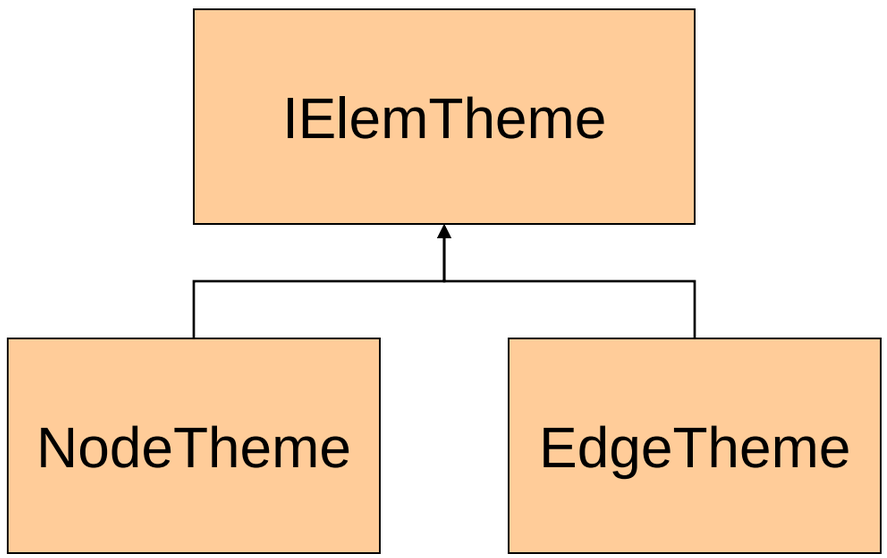
- All the renders are aggregated in the Display

Rendering Module: Elem Render and Theme

- One on one mapping
  - Each element has one element render and one element theme, if the users are not specified, a default render and theme will be returned



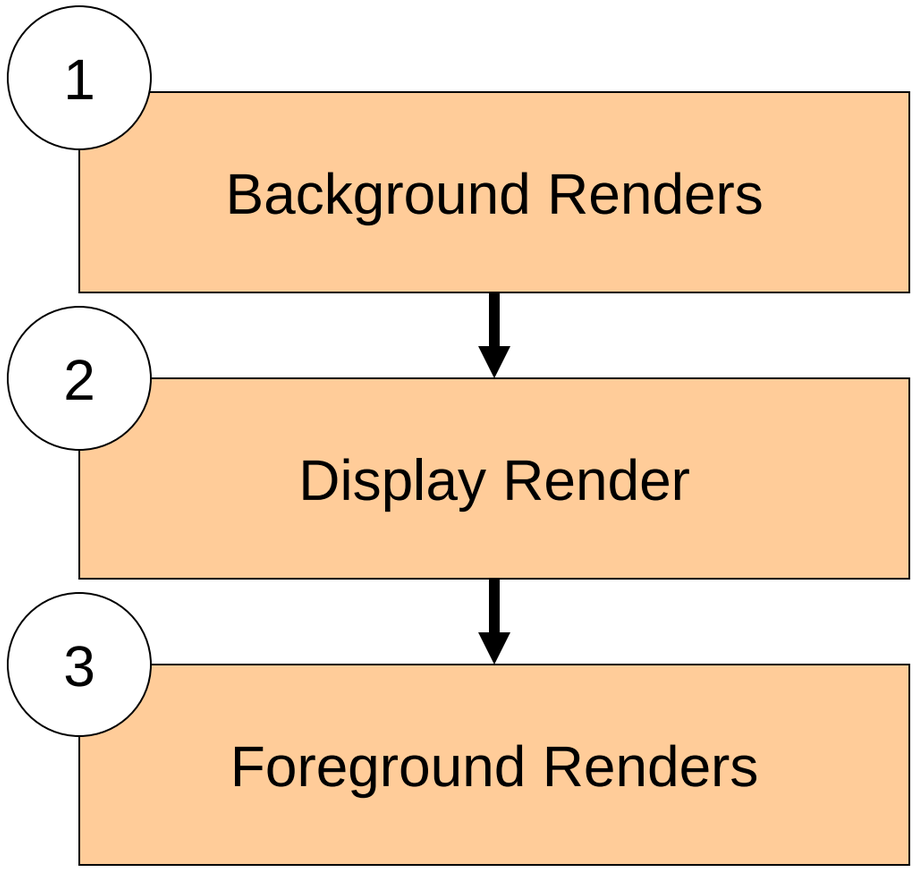
- Element renders and themes are stored and managed in the Display
  - display.addElemTheme(key, theme);
  - display.addElemRender(key, render);



Rendering Module: Display Render

- Store in the Display in a layer by layer manner
  - display.addBackgroundRender(r);
  - display.setDisplayRender(r);
  - display.addForegroundRender(r);

- Call individual element renders in order
  - FIFO, Tree, Stack



Code Example: TestVjitRender.java

Call all the edge renders in order

Call all the node renders in order

```
public void render(Graphics2D g) {

    Graph<IVisualNode> graph = (Graph<IVisualNode>)m_owner.getData("mygraph");
    if(null == graph) {
        return;
    }

    IElemRender r = null;
    IElemTheme t = null;
    IVisualNode node = null;
    IEdge<IVisualNode> edge = null;
    int ecnt = graph.getEdgeCount();
    for(int i = 0; i < ecnt; ++i) {
        edge = graph.getEdge(i);
        if(!edge.isVisible()) {
            continue;
        }

        r = m_owner.getElemRender(edge.getID());
        if(r == null) {
            r = m_owner.getElemRender("edge");
        }
        t = m_owner.getElemTheme(edge.getID());
        if(t == null) {
            t = m_owner.getElemTheme("edge");
        }
        r.render(g, edge, t, edge.isHighlight() || edge.isFocused());
    }

    int ncnt = graph.getNodeCount();
    for(int i = 0; i < ncnt; ++i) {
        node = graph.getNode(i);
        if(!node.isVisible()) {
            continue;
        }
        r = m_owner.getElemRender(node.getID());
        if(r == null) {
            r = m_owner.getElemRender("node");
        }
        t = m_owner.getElemTheme(node.getID());
        if(t == null) {
            t = m_owner.getElemTheme("node");
        }

        r.render(g, node, t, node.isHighlight() || node.isFocused());
    }

}
```

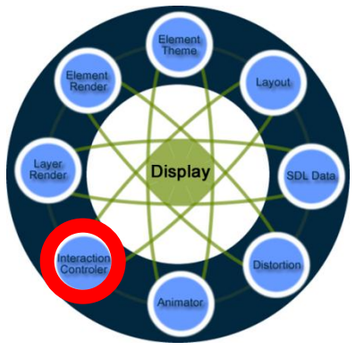
Code Example: TestXXXRender.java

TestEdgeRender.java

```
public class TestEdgeRender extends ElemRender {

    private Line2D m_line = null;

    public TestEdgeRender() {
        m_line = new Line2D.Double();
    }

    public Shape getRawShape(IElement elem) {
        Edge<IVisualNode> edge = (Edge<IVisualNode>)elem;
        IVisualNode n1 = edge.getFirstNode();
        IVisualNode n2 = edge.getSecondNode();

        m_line.setLine(n1.getX(), n1.getY(), n2.getX(), n2.getY());

        return m_line;
    }

    public void render(Graphics2D g, IElement elem, IElemTheme theme,
            boolean highlight) {
        Shape s = getRawShape(elem);
        g.setColor(theme.getBorderColor(elem));
        g.draw(s);
    }

}
```

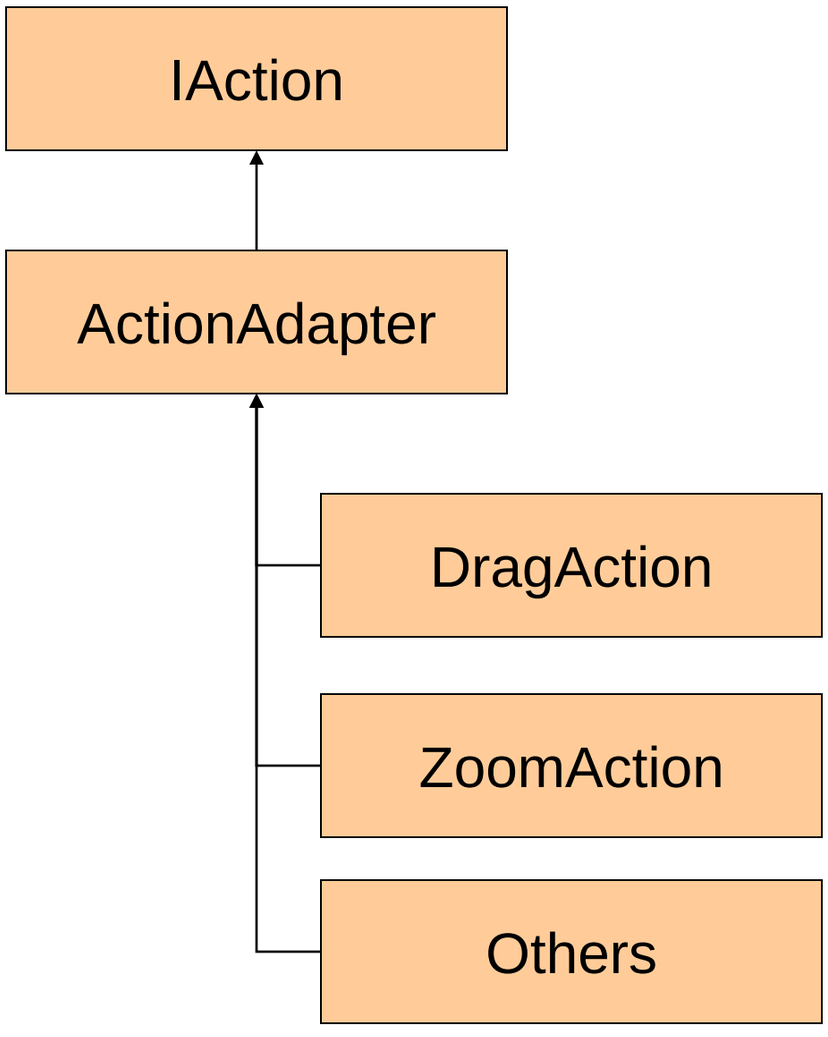
TestNodeRender.java

```
public class TestNodeRender extends ElemRender {

    private Ellipse2D m_shape = null;

    public TestNodeRender() {
        m_shape = new Ellipse2D.Double();
    }

    public Shape getRawShape(IElement elem) {

        IVisualNode node = (IVisualNode)elem;

        m_shape.setFrameFromCenter(
                node.getX(),
                node.getY(),
                node.getX() + node.getWidth() / 2,
                node.getY() + node.getWidth() / 2);

        return m_shape;
    }

    public void render(Graphics2D g, IElement elem, IElemTheme theme,
            boolean highlight) {
        Shape s = getRawShape(elem);
        g.setColor(theme.getFillColor(elem));
        g.fill(s);
        g.setColor(theme.getBorderColor(elem));
        g.draw(s);
    }

}
```



Interaction Module

- IAction interface
- Handled by Display
  - display.addAction(new MyAction())
- Two levels of control
  - Display level: When mouse is focused on a display but no element is focused
    - mouseXXX(MouseEvent e)
  - Element level: when mouse is focused on an element
    - elemXXX(IElement e, MouseEvent evn);

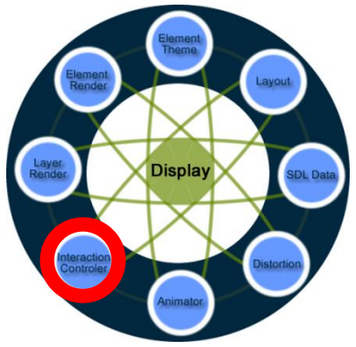


Code Example: TestAction.java

```
public class TestAction extends ActionAdapter {

    public TestAction() {

    }

    public void elemPressed(IElement elem, MouseEvent e) {
        System.out.println("Element Pressed: " + elem.getID());
    }

    public void mousePressed(MouseEvent e) {
        System.out.println("Mouse Pressed");
    }

}
```

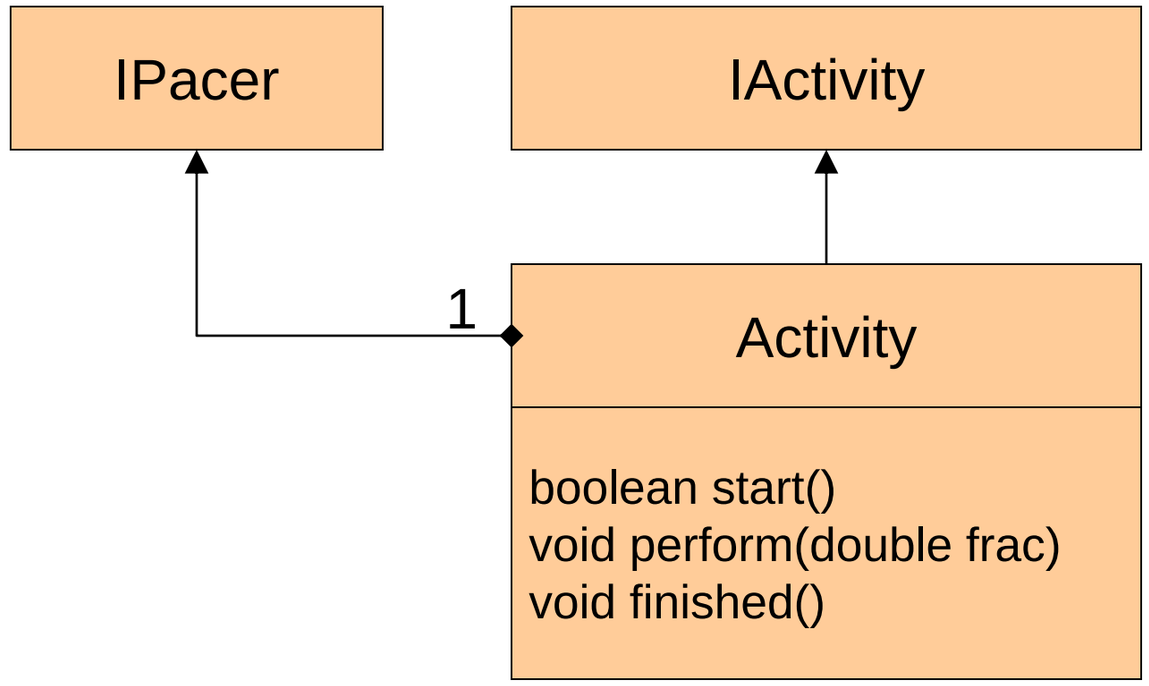
Called when element is pressed

Called when mouse is pressed in an empty space

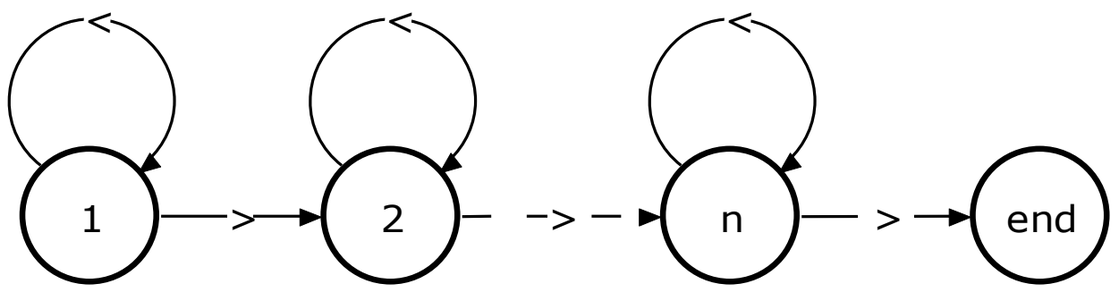


Animation Module

- Each animation is defined as an activity
- Each activity has three states
  - start, perform, finish
- Activities can be scheduled in chain

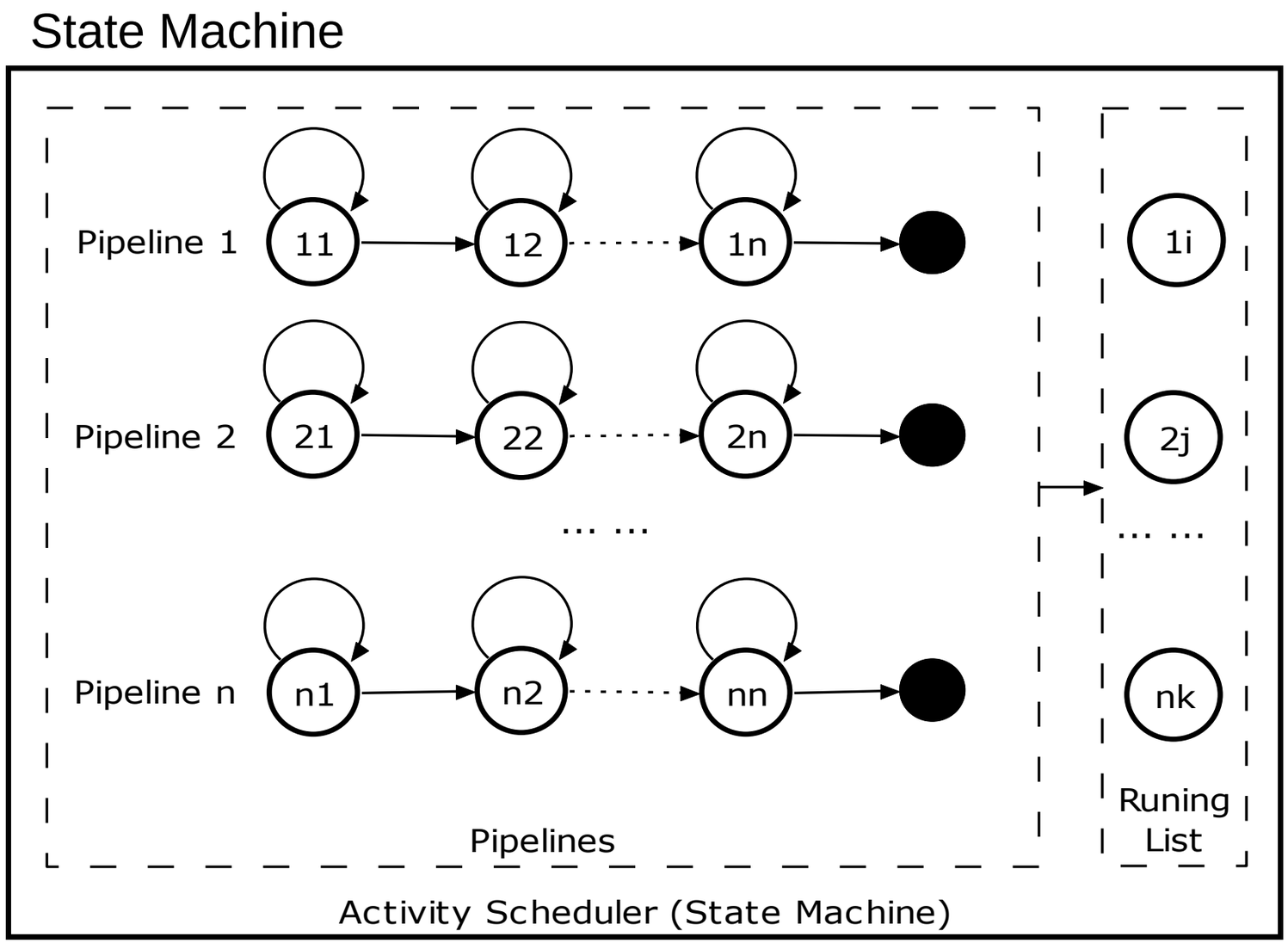


Activity Chain



Animation Module

- The state transitions are automatically handled by ActivityManager as a state machine



Example Code: Animator

TestAnimator.java (Design your own animator)

```
public class TestAnimator extends Activity {

    private static final long serialVersionUID = 2469923962814774228L;

    private Graph<IVisualNode> m_graph = null;

    public TestAnimator() {
        super(5000, 30);
    }

    public boolean start() {
        m_graph = (Graph<IVisualNode>)m_disp.getData("mygraph");
        if(m_graph == null) {
            return false;
        }
        return true;
    }

    public void perform(double frac) {
        IVisualNode node = null;
        int ncnt = m_graph.getNodeCount();
        for(int i = 0; i < ncnt; ++i) {
            node = m_graph.getNode(i);
            node.updateLocation(frac);
            node.updateSize(frac);
        }
        super.perform(frac);
        m_disp.repaint();
    }

    public void finish() {
        super.finish();
    }

}
```

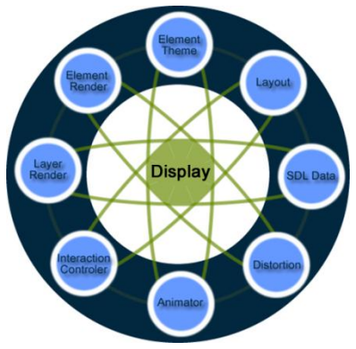
TestVjit.java (User your animator)

```
public void doLayout() {

    super.doLayout();

    m_animator.setDisplay(this);
    m_animator.setStartTime(System.currentTimeMillis());
    this.getActivityManager().addActivity(m_animator);
}
```

Element Finder

- Find the underlying element in the data structure base on a specified position (x, y)

- Various implementations
  - Linear O(n)
  - QuadTree O(log(n))



Code Example: TestElemFinder.java

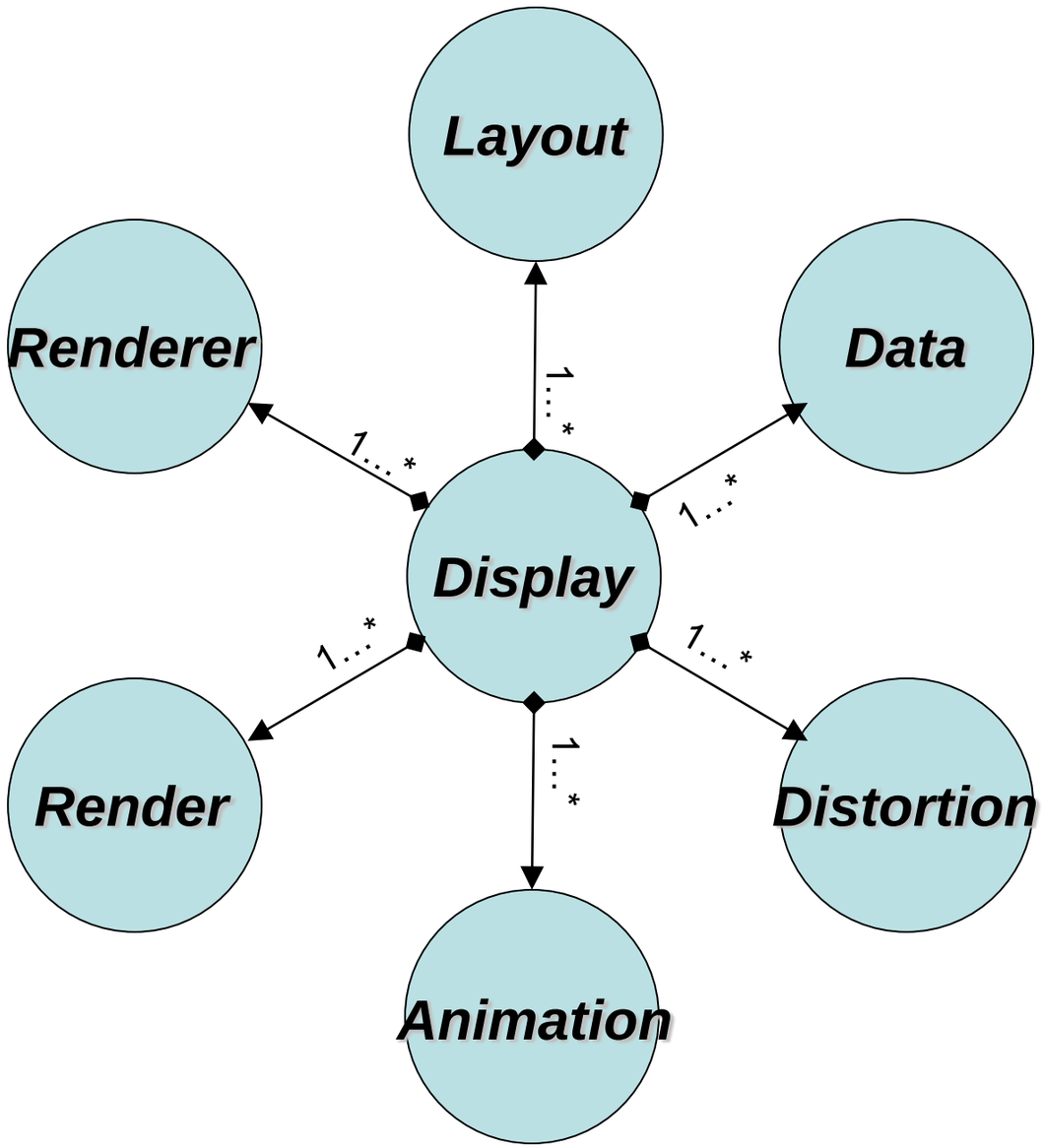
```
public class TestElemFinder extends ElemFinder {

    public TestElemFinder() {
    }

    public IElement find(double x, double y) {

        Graph<IVisualNode> graph = (Graph<IVisualNode>)m_owner.getData("mygraph");
        if(graph == null) {
            return null;
        }

        VisualNode node = null;
        IElemRender r = null;
        int ncnt = graph.getNodeCount();
        for(int i = 0; i < ncnt; ++i) {
            node = (VisualNode)graph.getNode(i);
            r = m_owner.getElemRender(node.getID());
            if(r == null) {
                r = m_owner.getElemRender("node");
            }
            if(r.locatePoint(x, y, node)) {
                return node;
            }
        }
        return null;
    }

}
```

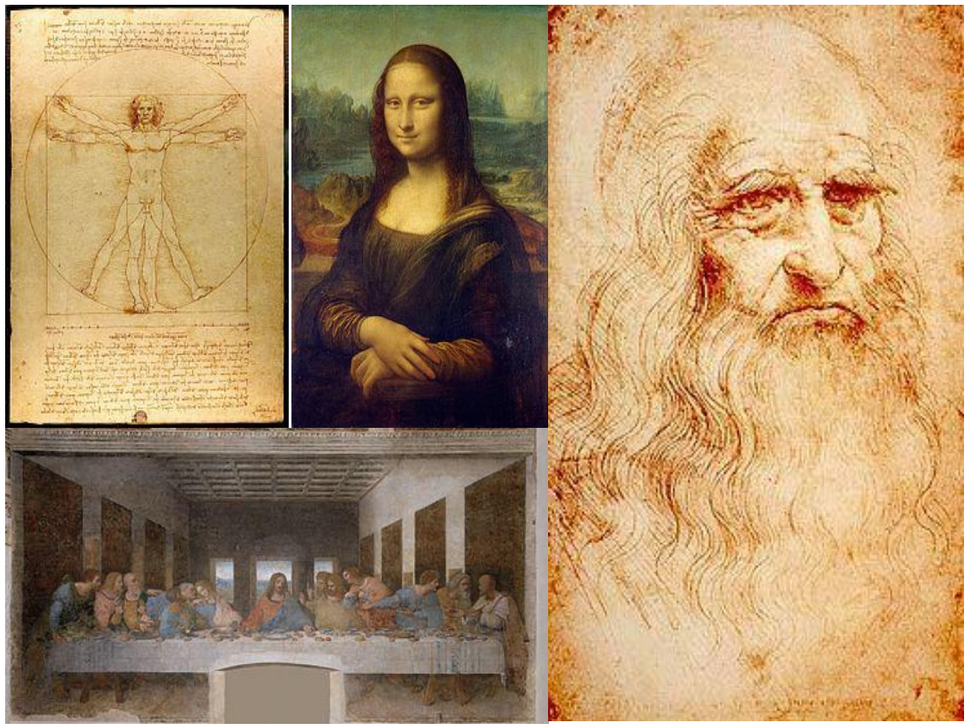
How to develop an InfoVis Vijt ?

- Visualization Design
  - Photoshop : draw your designs

- If I have a coding team:
  - Data structure (everybody)
  - Agile development
    - One person one module
    - Fast iteration

- If I am a single person:
  - Data structure
  - Render
  - Animation
  - Interaction





DaVinci: A lightweighted Information Visualization Framework V1.0

Nan Cao(nan.cao@gmail.com)

Aug. 2011