



VARCHART

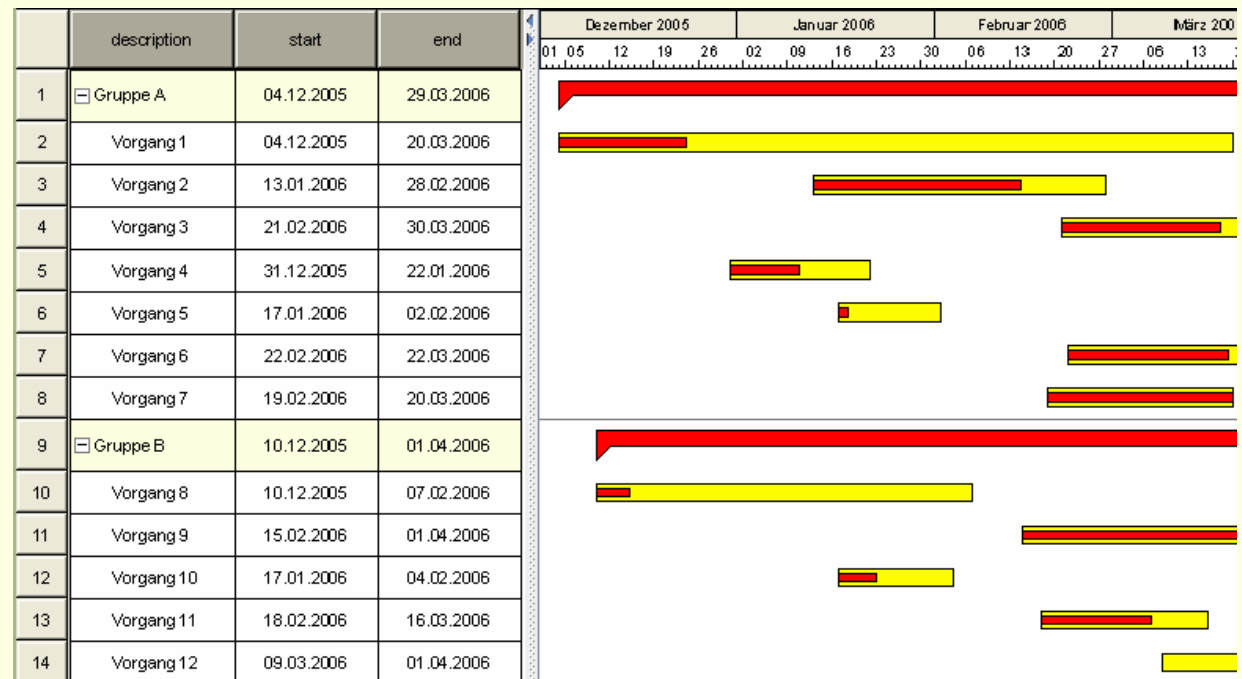
JGantt Workshop 4

- Advanced Graphical Design**
- The New Web Server Edition**

**Udo Corban
Jürgen Theis**

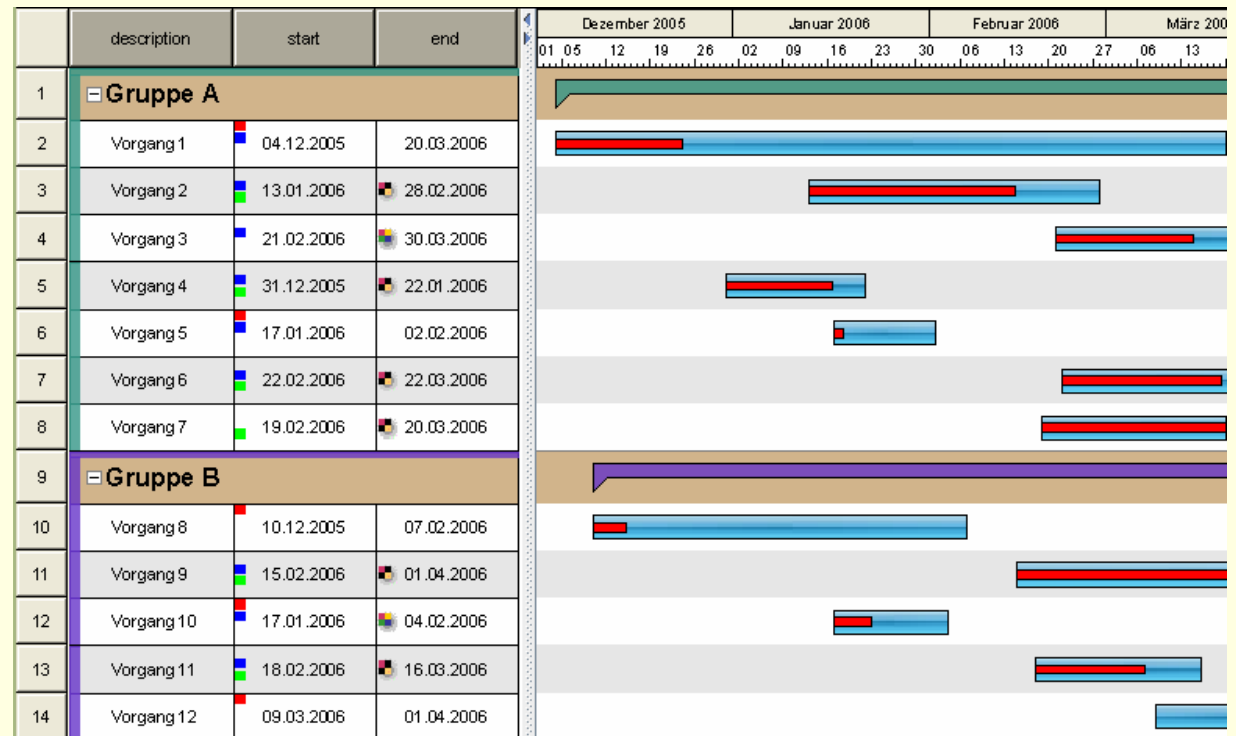
Advanced Graphical Design

- how to get ...
from here



Advanced Graphical Design

- to there:



... in 30 minutes !

What is involved ?

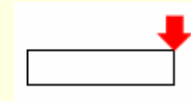
- Layer definitions
- Row definitions
- Pictures and labels
- Dynamic elements
 - Colors
 - Labels
 - Pictures

Pictures and labels

- Interfaces NeIPicture and NeLabel
 - for adding 'decorations' to graphical elements
 - build - in or custom implementations

- NeLabel

- placed at a position
- has its own size



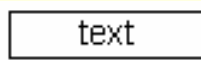
- NeIPicture

- placed inside a rectangle
- adapts to given size
- 'how' depends on implementation

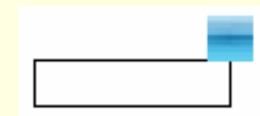
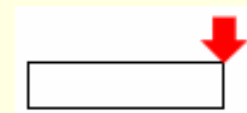
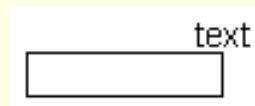


Pictures and labels


- Implementations provided with JGantt
 - NeAnnotation
 - NeSymbol
 - NePicture



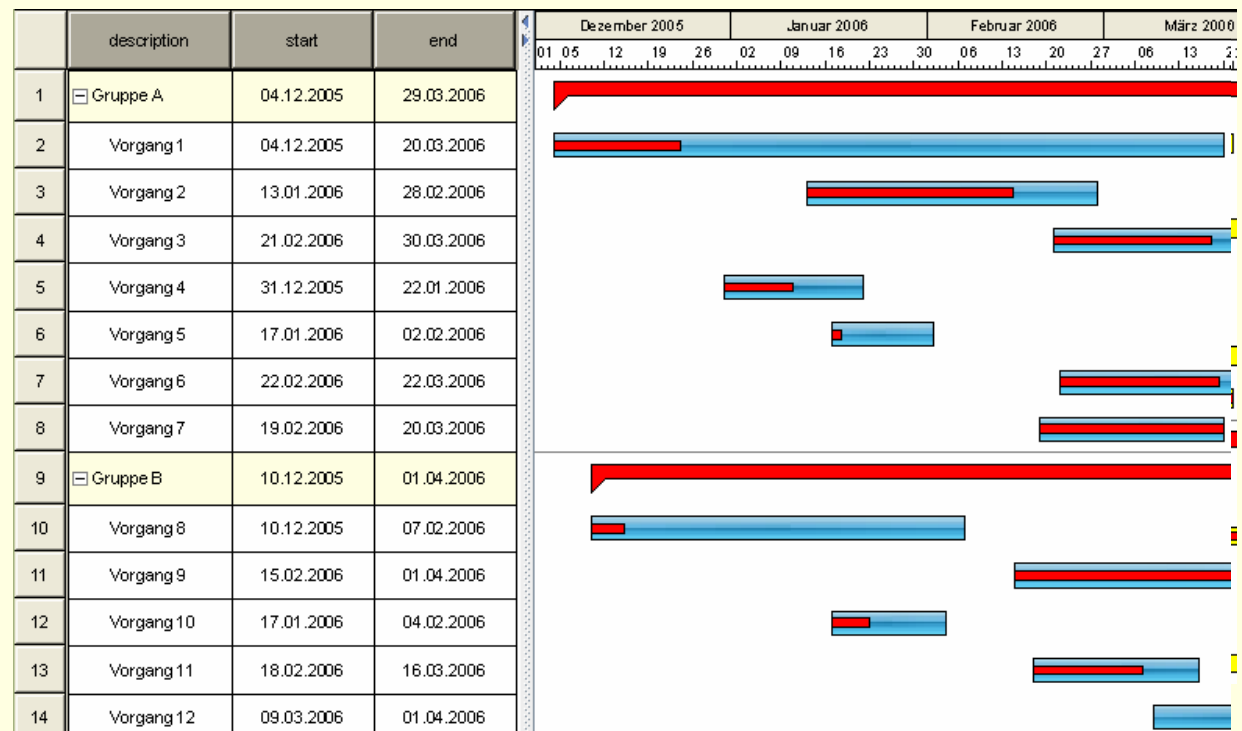
- Use as NeIPicture or NeILabel



Enhancing the bars

- Layer definitions instead of node designs
- Using bitmaps as background
 - stretched horizontally 

	description	start	end
1	Gruppe A	04.12.2005	29.03.2006
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	Gruppe B	10.12.2005	01.04.2006




Accentuating the group header

- RowDefinition for group rows
- Table cells may span multiple columns

	description	start	end
1	Gruppe A	04.12.2005	29.03.2006
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	Gruppe B	10.12.2005	01.04.2006
10	Vorgang 8	10.12.2005	07.02.2006

	description	start	end
1	Gruppe A		
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	Gruppe B		
10	Vorgang 8	10.12.2005	07.02.2006
11	Vorgang 9	15.02.2006	01.04.2006
12	Vorgang 10	17.01.2006	04.02.2006
13	Vorgang 11	18.02.2006	16.03.2006
14	Vorgang 12	09.03.2006	01.04.2006



Advanced Graphical Design



Hands on ...

layer definitions

```
NeLayerDefinition startEndLayer, pcLayer;  
jGantt.setNodeDesign (null);  
  
NeValueReference startRef      = new NeValueReference ("start");  
NeValueReference endRef        = new NeValueReference ("end");  
NeRelativeValueReference pcRef = new NeRelativeValueReference ("pc", 0.01f, startRef, endRef);  
NeIFilter forNodes = new JGEntitySetFilter (jGantt.getNodeSetName ());  
  
startEndLayer = jGantt.getGanttGraph ().createLayerDefinition ("startEndLayer");  
startEndLayer.setFilter (forNodes);  
startEndLayer.setStartValueReference (startRef);  
startEndLayer.setEndValueReference (endRef);  
startEndLayer.setHeight (400);  
  
NePicture layerBack = new NePicture ();  
layerBack.setStretchMode (NePicture.STRETCHED_ANISOTROPIC);  
layerBack.setPicture (new MediaTracker (this), "layerbackBlue.JPG", 0, "file:");  
startEndLayer.setCenterPicture (layerBack);  
  
pcLayer = jGantt.getGanttGraph ().createLayerDefinition ("pcLayer");  
pcLayer.setDrawingPriority (100);  
pcLayer.setFilter (forNodes);  
pcLayer.setStartValueReference (startRef);  
pcLayer.setEndValueReference (pcRef);  
pcLayer.setAreaStyle (Color.red);  
pcLayer.setHeight (150);
```

row definitions

// create filter, only for group nodes

```
NeIFilter groupFilter = new JGEntitySetFilter (jGantt.getGroupNodeSetName ());
```

// a row definition defines a row type in the table

```
groupRowDefinition = new NeRowDefinition (groupFilter, 1);
```

//a field style defines the 'graphical' attributes of a table field

```
NeFieldStyle groupNameStyle = new NeFieldStyle (new Font ("Arial",  
    Font.BOLD, 12), NeIDrawingConstants.ALIGNMENT_CENTER_LEFT,  
    Color.black, groupRowColor);  
groupNameStyle.setMarginLeft (300);
```

//a field definitions defines the (data) content of a field and which style to use

```
NeFieldDefinition groupNameField = new NeFieldDefinition ("description",  
    groupNameStyle);  
groupRowDefinition.addFieldDefinition (groupNameField); // add field to row definition  
groupRowDefinition.setFieldRange (groupNameField, 3); // the field spans 3 columns!
```

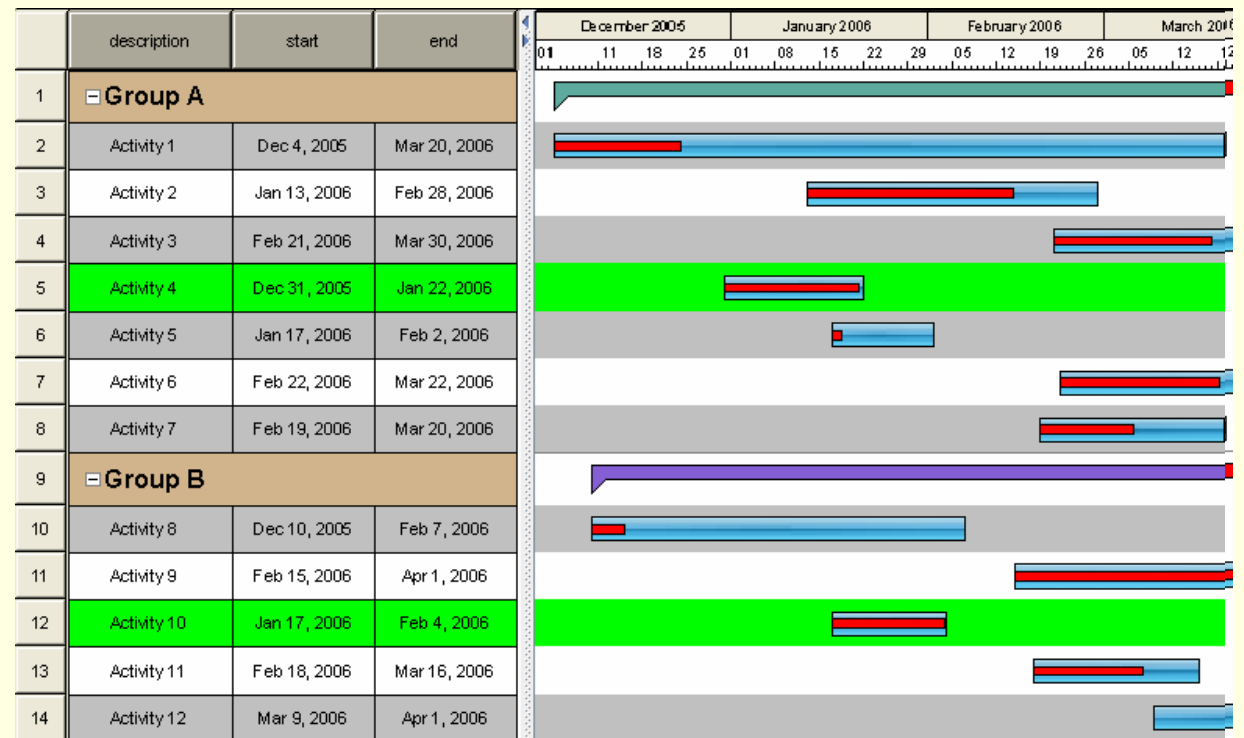
// fetch contents definition from table and add row definition

```
jGantt.getTable ().getContentsDefinition ().addRowDefinition (groupRowDefinition);
```

Coloring the rows

- NeIDynamic color for gantt/table background
- Depends on entity in row
- JGDynamicRowColor ...
- ... or custom implementation for special needs

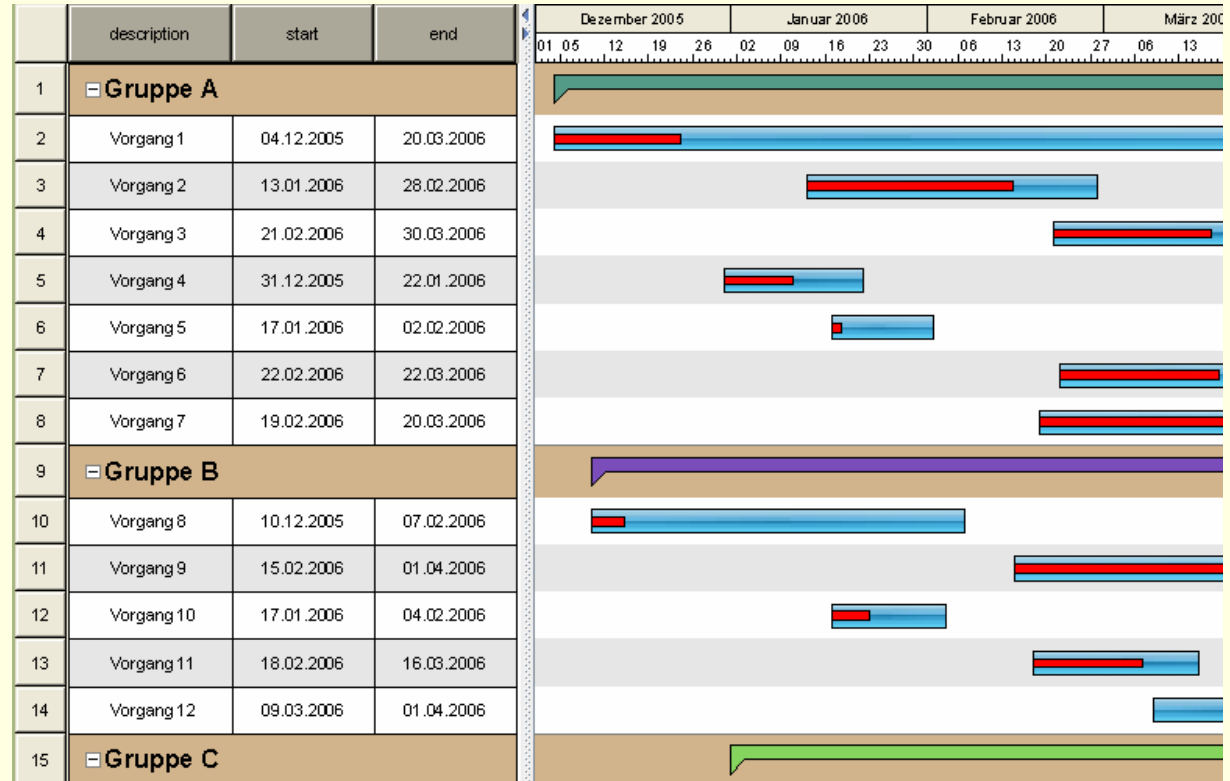
	description	start	end
1	Gruppe A	04.12.2005	29.03.2006
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	Gruppe B	10.12.2005	01.04.2006
10	Vorgang 8	10.12.2005	07.02.2006



Color for the groups

- NeMapped color, depending on group code
- Used as alternate color of group color scheme

	description	start	end
1	-Gruppe A		
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	-Gruppe B		
10	Vorgang 8	10.12.2005	07.02.2006
11	Vorgang 9	15.02.2006	01.04.2006
12	Vorgang 10	17.01.2006	04.02.2006
13	Vorgang 11	18.02.2006	16.03.2006
14	Vorgang 12	09.03.2006	01.04.2006
15	-Gruppe C		



Advanced Graphical Design



Hands on ...

dynamic colors for rows

```
// two colors for odd and even rows
```

```
Color evenRowColor = Color.white;
```

```
Color oddRowColor = new Color (230, 230, 230);
```

```
// first approach: utilize JGDynamicRowColor, using groupRowColor for group rows
```

```
Color dynamicRowColor = new JGDynamicRowColor (jGantt, evenRowColor, oddRowColor,  
    groupRowColor);
```

```
// get the gantt/table color scheme from jGantt and replace the shaded color,
```

```
// which is responsible for the background.
```

```
// CAUTION: you have to set the color schemes to JGantt again to make changes apply.
```

```
JGColorScheme colorScheme = jGantt.getTableColorScheme ();
```

```
colorScheme.setShadedColor (dynamicRowColor);
```

```
jGantt.setTableColorScheme (colorScheme);
```

```
colorScheme = jGantt.getGanttColorScheme ();
```

```
colorScheme.setShadedColor (dynamicRowColor);
```

```
jGantt.setGanttColorScheme (colorScheme);
```

dynamic colors for rows (second approach)

// define your own class if you need something even more special

```
class DynRowColor extends Color implements NelDynamicColor
```

```
{  
    DynRowColor ()  
        { super (255, 255, 255); };  
  
    public Color getActualColor (Object refObject)  
    {  
        NelEntity node = (NelEntity) refObject;  
  
        if (node.getValueAsInt ("pc") > 90)  
            return Color.green;  
        else  
        {  
            int row = jGantt.getRowIndexForNode (node);  
            if (row % 2 == 0)  
                return Color.white;  
            else  
                return Color.lightGray;  
        }  
    }  
}
```

```
dynamicRowColor = new DynRowColor();
```

dynamic colors for groups

// a color map defines mappings between values or ranges of (comparable) values

// and colors

```
NeColorMap groupColorMap = new NeColorMap ();  
groupColorMap.addEntry ("A", new Color (48, 148, 132, 200));  
groupColorMap.addEntry ("B", new Color (100, 48, 200, 200));  
groupColorMap.addEntry ("C", new Color (133, 212, 93));
```

// a mapped color is an implementation of NeIDynamicColor which utilizes a color map

// to determine the actual color, based on the value of an attribut (here "group")

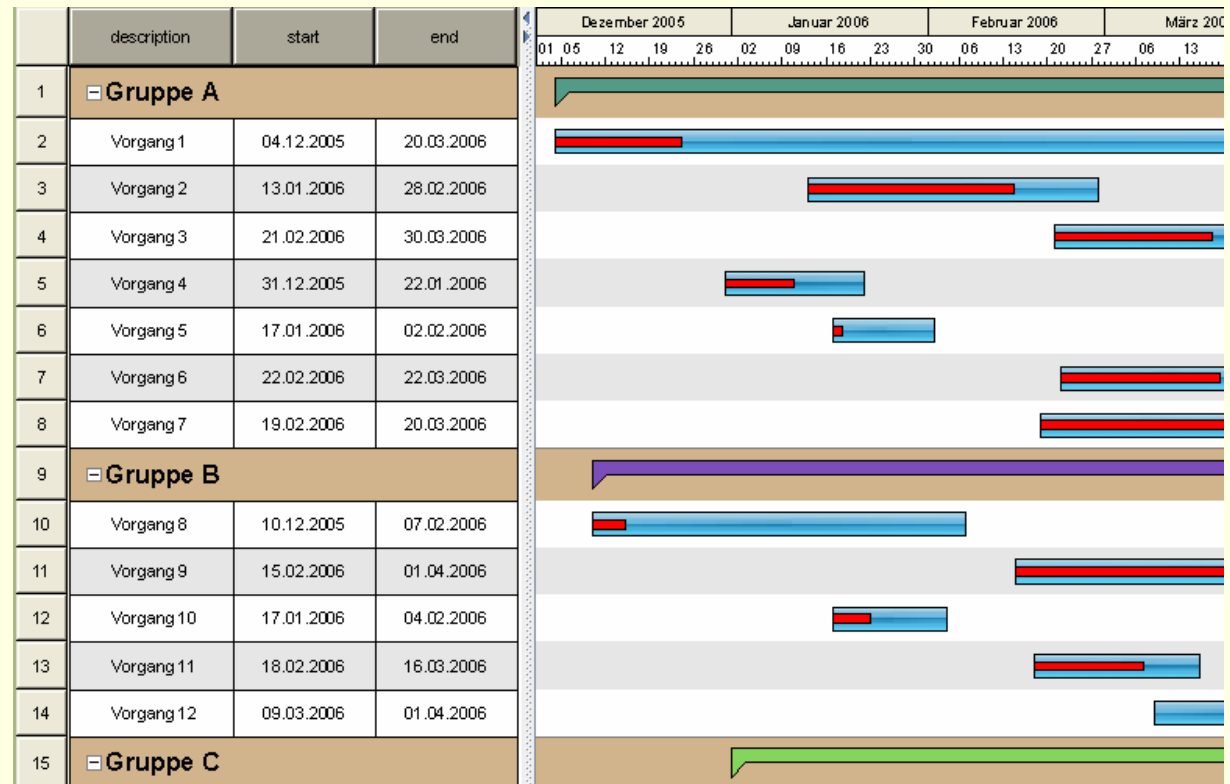
// of the entity in charge. If no value is found in the map, the default is used

```
groupColor = new NeMappedColor (Color.gray, groupColorMap, "group");  
jGantt.setGroupNodeColorScheme (new JGColorScheme (groupColor,  
    groupColor, groupColor, groupColor));
```

Decorating the table

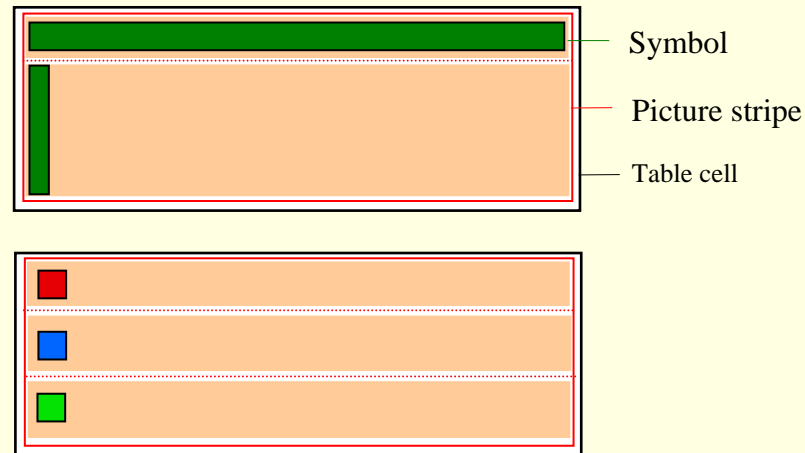
- Symbols as background for cells
- Picture stripe for multiple symbols in one cell
- Dynamic color

	description	start	end
1	-Gruppe A		
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	-Gruppe B		
10	Vorgang 8	10.12.2005	07.02.2006
11	Vorgang 9	15.02.2006	01.04.2006
12	Vorgang 10	17.01.2006	04.02.2006
13	Vorgang 11	18.02.2006	16.03.2006
14	Vorgang 12	09.03.2006	01.04.2006
15	-Gruppe C		



NePictureStripe

- Implementation of NeIPicture
- Container for pictures
 - horizontal or vertical
 - distributes space among sub pictures

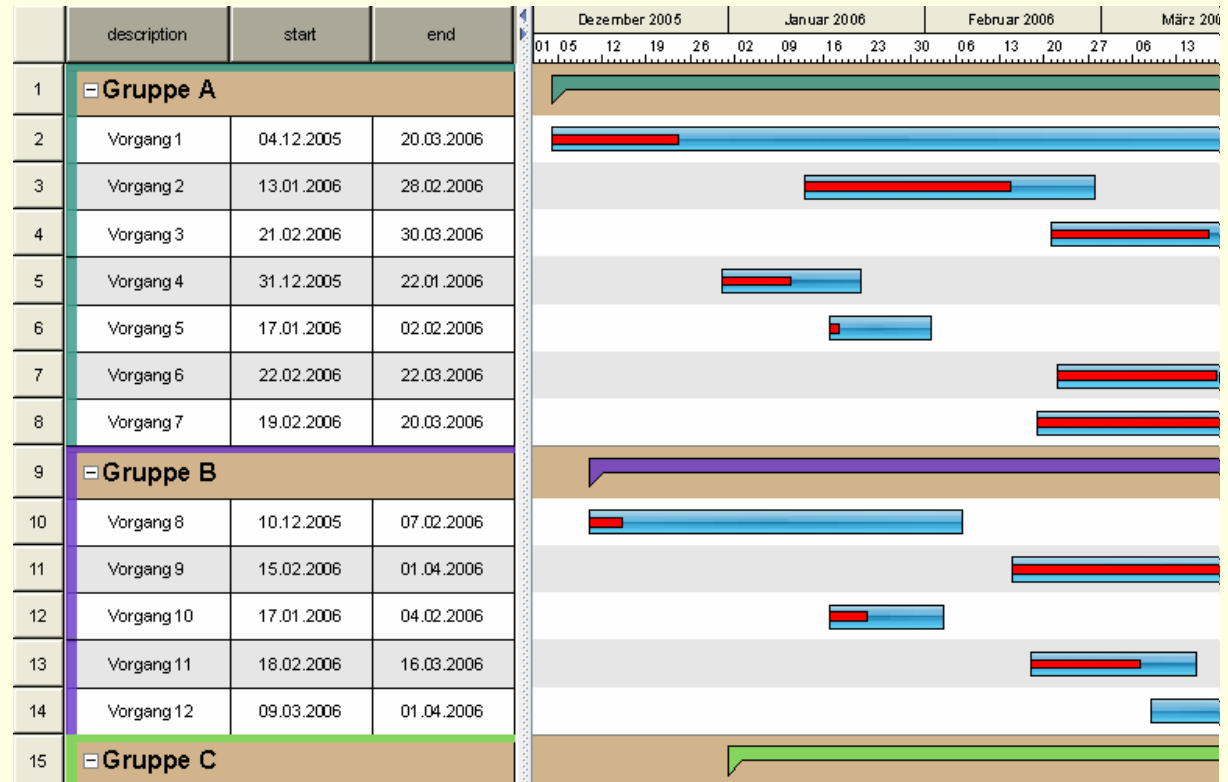


- Can be used recursively

More than text in table cells

- Graphical information depending on data
- Dynamic pictures
 - NeMappedPicture
- Picture stripe and dynamic colors

	description	start	end
1	Gruppe A		
2	Vorgang 1	04.12.2005	20.03.2006
3	Vorgang 2	13.01.2006	28.02.2006
4	Vorgang 3	21.02.2006	30.03.2006
5	Vorgang 4	31.12.2005	22.01.2006
6	Vorgang 5	17.01.2006	02.02.2006
7	Vorgang 6	22.02.2006	22.03.2006
8	Vorgang 7	19.02.2006	20.03.2006
9	Gruppe B		
10	Vorgang 8	10.12.2005	07.02.2006
11	Vorgang 9	15.02.2006	01.04.2006
12	Vorgang 10	17.01.2006	04.02.2006
13	Vorgang 11	18.02.2006	16.03.2006
14	Vorgang 12	09.03.2006	01.04.2006
15	Gruppe C		



Advanced Graphical Design

Hands on ...

picture stripe as cell background

// the upper part (horizontal bar) of the collar

```
NeSymbol upperPart = new NeSymbol (8, NeSymbol.SYMBOL_TABLE_GR_MMG);  
upperPart.setAreaStyle (groupColor);  
upperPart.setLineStyle (groupColor);  
upperPart.setExtent (new Dimension (0, 100)); // fill complete area horizontally, 1 mm high
```

// the lower part (left bar) of the collar

```
NeSymbol lowerPart = new NeSymbol (8, NeSymbol.SYMBOL_TABLE_GR_MMG);  
lowerPart.setHorizontalPictureFillMode (NeSymbol.LEFT);  
lowerPart.setAreaStyle (groupColor);  
lowerPart.setLineStyle (groupColor);  
lowerPart.setExtent (new Dimension (175, 0)); // 1.75mm width, fill complete area vertically
```

// combine both elements in a vertical picture stripe

```
NePictureStripe groupCol = new NePictureStripe (true);  
groupCol.addPicture (upperPart, 0.0f);  
groupCol.addPicture (lowerPart, 1.0f);
```

// use picture stripe as background for group rows / cells

```
NeFieldDefinition fieldDef = (NeFieldDefinition) groupRowDefinition.getFieldDefinition (0);  
fieldDef.setPicture (groupCol);
```

// use left part only as background for default rows

```
fieldDef = (NeFieldDefinition) jGantt.getTable ().getContentsDefinition ().getDefaultRowDefinition ()  
    .getFieldDefinition (0);  
fieldDef.setPicture (lowerPart);
```

custom implementation of NelDynamicColor

class SwitchedColor extends Color implements NelDynamicColor

```
{
    // returns trueColor as actual color if the value of 'pc' is between from and to, otherwise transparent color.
    Color trueColor;
    Color transparent = new Color (0.0f, 0.0f, 0.0f, 0.0f); //invisible
    int from, to;

    SwitchedColor (Color trueColor, int from, int to)
    {
        super (0.0f, 0.0f, 0.0f, 0.0f);
        this.trueColor = trueColor;
        this.from = from;
        this.to = to;
    }

    public Color getActualColor (Object entity)
    {
        int value = ((NelEntity) entity).getValueAsInt ("pc");
        if (value >= from && value <= to)
            return trueColor;
        else
            return transparent;
    }
}
```

multiple symbols in cell

```
// create three squares (2 x 2 mm) , each using a different switched color
NeSymbol red = new NeSymbol (75, NeSymbol.SYMBOL_TABLE_GR_BAR);
red.setLineStyle (new SwitchedColor (Color.red, 0, 33)); //visible (red) if pc <= 33
red.setExtent (new Dimension (200, 200));

NeSymbol yellow = new NeSymbol (75, NeSymbol.SYMBOL_TABLE_GR_BAR);
yellow.setLineStyle (new SwitchedColor (Color.blue, 10, 95)); //visible (blue) if 10 <= pc <= 95
yellow.setExtent (new Dimension (200, 200));

NeSymbol green = new NeSymbol (75, NeSymbol.SYMBOL_TABLE_GR_BAR);
green.setLineStyle (new SwitchedColor (Color.green, 60, 100)); //visible (green) if 60 <= pc <= 100
green.setExtent (new Dimension (200, 200));

// a vertical picturestripe to hold the three squares
NePictureStripe tl = new NePictureStripe (true);
tl.setHorizontalPictureFillMode(NePictureStripe.LEFT);
tl.addPicture (red, 0.0f);
tl.addPicture (yellow, 0.0f);
tl.addPicture (green, 0.0f);

// use picture stripe as background of second (index starts at 0) cell for default rows
NeFieldDefinition fieldDef = (NeFieldDefinition) jGantt.getTable () .getContentsDefinition ()
    .getDefaultRowDefinition ().getFieldDefinition (1);
fieldDef.setPicture (tl);
```

bitmaps as cell background

```
NePicture iconRed = new NePicture ();           // picture from PNG file
iconRed.setExtent (new Dimension (350, 350));
iconRed.setHorizontalPictureFillMode (NePicture.LEFT); // position at center left
iconRed.setVerticalPictureFillMode (NePicture.CENTER); // in cell
iconRed.setPicture (AGFrame.class.getResource("SunChessRed.png"));

NePicture iconBlue = new NePicture ();
iconBlue.setExtent (new Dimension (350, 350));
iconBlue.setHorizontalPictureFillMode (NePicture.LEFT);
iconBlue.setVerticalPictureFillMode (NePicture.CENTER);
iconBlue.setPicture (AGFrame.class.getResource("SunChessYellow.png"));

NePictureMap pm = new NePictureMap (); // defines mapping between values or ranges and NePictures
pm.addRange (new Long (33), iconBlue); // if pc >= 33 use iconBlue
pm.addRange (new Long (66), iconRed); // but if pc >= 66 use iconRed

// implementation of NePicture which utilizes a picture map to determine the actual picture,
// based on the value of an attribut (here "pc") of the entity in charge.
// If no value is found in the map, the default (here null, means 'no picture') is used
NeMappedPicture icon = new NeMappedPicture (null, pm, "pc");

NeFieldDefinition fieldDef = (NeFieldDefinition) jGantt.getTable ()
    .getContentsDefinition ().getDefaultRowDefinition ()
    .getFieldDefinition (2);
fieldDef.setPicture (icon); // assign as background for the third cell in default rows
```



VARCHAR



Questions?



VARCHART



Additional Slides