



Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

Erweiterung von SECONDO UserInterfaces

Thomas Behr

Fakultät für Mathematik und Informatik
Datenbanksysteme für neue Anwendungen



FernUniversität in Hagen

10.Oktober 2009

©2009 FernUniversität in Hagen



1 Javagui

- Dokumentation
- Erstellung eines neuen Viewers
 - Beispiel - Inquiry-Viewer
- Erweiterung des HoeseViewers
 - Objekte in Textdarstellung
 - Graphische Objekte
 - Raum-Zeitliche Objekte
 - Sonstige Objekte
 - Label und Manipulation anderer Objekte

2 Anzeige im TTY

Überblick

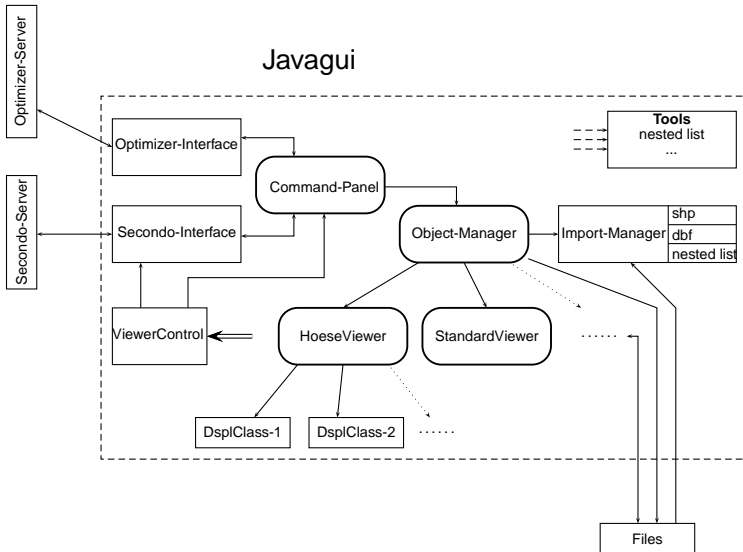
Erweiterung
von
SECONDO

Thomas Behr

Javagui

- Dokumentation
- Erstellung eines neuen Viewers
- Beispiel - Inquiry-Viewer
- Erweiterung des HoeseViewers
- Objekte in Textdarstellung
- Graphische Objekte
- Raum-Zeitliche Objekte
- Sonstige Objekte
- Label und Manipulation anderer Objekte

Anzeige im
TTY





Erweiterung
von
SECONDDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- **im Javagui Verzeichnis**
- **make doc**
- **neues Verzeichnis doc enthält javadoc generierte HTML Seiten**



Nützliche Klassen

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

Mehr als 300 .java Dateien, über 600 Klassen.

- ID
- IDManager
- MenuVector
- ListExpr
- SecondoObject
- SecondoViewer
- Reporter



ID und IDManager

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Paket `gui.idmanager`
- nur für Viewer, die Secondo-Objekte erzeugen interessant
- ID: eindeutige ID für Secondo-Objekte (mit gleichem Wert)
- IDManager: liefert noch nicht verwendete ID



Erweiterung
von
SECONDDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Paket `viewer`
- dient der Erweiterung des Hauptmenüs der Javagui
- bietet Funktionen zum Einfügen und Löschen von Menüeinträgen
- im Normalfall wird eine Instanz im Konstruktor erzeugt und gefüllt
- wird in der `getMenuVector` Methode des Viewers zurückgeliefert



Erweiterung
von
SECONDDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Paket `sj.lang`
- Java-Umsetzung von verschachtelten Listen
- zwei unterschiedliche Implementierungen in Datei `ListExpr.both`
- bietet nahezu identische Funktionalität wie auf C++ Seite
- Interface mehr „objektorientiert“, d.h nur wenige statische Methoden (ähnlich `NList`)



Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Paket `gui`
- enthält ID, Name und Listendarstellung
- muss vom Viewer analysiert und ins eigene Format umgesetzt werden
- interessanteste Methode `toListExpr()`



- Paket `tools`
- bietet Funktionen für
 - Konsolenausgaben
 - Pop-Up Fenster (Meldungen, einfache Fragen)
 - Debug-Ausgaben
- soll stets verwendet werden (statt `System.out` und `System.err`)
- Debug-Infos ignoriert, wenn `DEBUG_MODE=false`
- Ausgaben von Fenstern im Testmodus auf Konsole umgeleitet



Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Paket `viewer`
- abgeleitet von `JComponent`
- Superklasse für *alle* Viewer von Javagui
- abstrakte Klasse
- verschiedene Methoden können / müssen überschrieben werden
- erlaubt teilweise Kontrolle der GUI



Inquiry-Viewer

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- Anzeige des Ergebnisses von `list { databases, objects, ... }`
- Tabelle mittels `html` Code formatiert
- Datei `InquiryViewer.java` im `viewer` Verzeichnis
- Listenformat (Ausschnitt): `(inquiry <value>)`
 - `< value > := (<subtype> (<subvalue>))`
 - `<subtype> := databases, algebras, ...`
 - `<subvalue>` hängt von `subtype` ab
 - `databases, algebras` : einfache Liste von Symbols
 - `constructors`: Liste von Konstruktoren im Format
`(name <property names> <property values>)`
 - ...

- ergibt `true`, wenn der Viewer das übergebene Objekt anzeigen kann
- normalerweise wird der Typ (in der NL) geprüft

```
public boolean canDisplay(SecondoObject o){
    ListExpr LE = o.toListExpr();
    if(LE.listLength()!=2)
        return false;
    if(LE.first().atomType()!=ListExpr.SYMBOL_ATOM ||
        !LE.first().symbolValue().equals("inquiry"))
        return false;
    ListExpr VL = LE.second();
    if(VL.listLength()!=2)
        return false;
    ListExpr SubTypeList = VL.first();
    if(SubTypeList.atomType()!=ListExpr.SYMBOL_ATOM)
        return false;
    String SubType = SubTypeList.symbolValue();
    if(SubType.equals("databases") || SubType.equals("objects") || ...)
        return true;
    return false;
}
```



getDisplayQuality

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- gibt an, wie gut das Objekt durch den Viewer dargestellt wird
- Angabe erfolgt im Bereich von 0 (gar nicht) bis 1 (besonders gut)
- optional
- wird bei der Auswahl des (besten) Viewers berücksichtigt

```
public double getDisplayQuality(SecondoObject SO){  
    if(canDisplay(SO))  
        return 0.9;  
    else  
        return 0;  
}
```



- Initialisiert den Viewer
- fügt Komponenten + Listener hinzu
- darf keine Argumente besitzen
- konstruiert ggf. den MenuVector

```
public InquiryViewer(){  
    /* gewöhnliche Verwendung von Java-Komponenten */  
    /* Aufbau des Menüvektors */  
    ..  
}
```



addObject

- fügt dem Viewer ein Objekt hinzu (oder selektiert es, falls schon vorhanden)
- Umwandlung der Nested List des Objekt in anzeigbares Format
- Objekt wird angezeigt

```
public boolean addObject(SecondoObject o){
    if(!canDisplay(o))
        return false;
    if (isDisplayed(o))
        selectObject(o);
    else{
        ListExpr VL = o.toListExpr().second();
        ObjectTexts.add(getHTMLCode(VL));
        ComboBox.addItem(o.getName());
        SecondoObjects.add(o);
        try{
            ComboBox.setSelectedIndex(ComboBox.getItemCount()-1);
            showObject();
        }
        catch(Exception e){
            Reporter.debug(e);
        }
    }
    return true;
}
```

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY



sonstige Funktionen

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- `removeObject` entfernt ein gegebenes Objekt vom Viewer
- `removeAll` entfernt alle Objekte aus dem Viewer
- `isDisplayed` tested, ob geg. Objekt im Viewer vorhanden ist
- `getMenuVector` gibt die Menüerweiterung zurück
- `selectObject` bringt ein vorhandenes Objekt zur Anzeige
- `getName` gibt den Viewernamen zurück
- `enableTestmode` schaltet automatische Userinteraktion ab



Ergebnis

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation
Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

The screenshot shows the 'Secundo-GUI (InquiryViewer)' window. The top menu bar includes 'Program', 'Server', 'Optimizer', 'Command', 'Help', 'Viewers', and 'Settings'. The main text area contains the following commands and output:

```
Sec>gui addViewer InquiryViewer
Sec>gui connect
you are connected to a secondo server
Sec>list algebra StandardAlgebra
list algebra StandardAlgebra...successful
see result in object list
Sec>
```

Below the text area, a table displays the details for the object 'list algebra StandardAlgebra':

Meaning	Not equal.
Example	query 2.1 # 2.01
starts	
Signature	(string string) -> bool
Syntax	_ starts _
Meaning	Starts.
Example	query "starts" starts "st"
contains	
Signature	(string string) -> bool
Syntax	_ contains _
Meaning	Contains.

At the bottom of the window, there is a search bar with a checked 'Case Sensitive' checkbox and a 'Search' button.



Kontrolle der Javagui

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- möglich durch Klassen-Member `VC (ViewerControl)`
- **mit Vorsicht benutzen**
- notwendig, wenn ein Viewer selbst Objekte erzeugt, z.B. `HoeseViewer` lädt neue Session
- bietet Möglichkeit, mit Secondo-Kern (via TCP/IP) zu kommunizieren, z.B. `UpdateViewer`, der auf bestimmte Queries angewiesen ist
- informiert GUI über Änderung des Hauptmenüs



Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- stellt, textuelle, räumliche und zeitliche veränderbare Objekte dar
- einfache Schnittstellen
- verwendet Displayklassen für die darstellbaren Typen
- Displayklassen werden zur Laufzeit eingebunden
- Manipulation des Aussehens anderer Objekte
- Erzeugung von Objekten möglich



Objekte in Textdarstellung

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- erweitern `DsplGeneric` Klasse
- überschreiben / implementieren `init`-Methode
- Beispiel `Display`-Klasse für rationale Zahlen

- Listenformat:

```
(rational (<sign> <intpart> <numDecimal> / <denomDecimal>))
```

Vorzeichen optional

```
(rational ( - 1 1/2 ))
```

 zur Darstellung von -1.5

- Klasse `Dsplrational` im Paket
`viewer.hoese.algebras`

Dsprational - Hilfsfunktion

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

```
private String getValueString(ListExpr value){
    int len = value.listLength();
    if(len!=4 && len !=5)
        return "ERROR";
    String result="";
    if(len==5){ // with sign
        ListExpr SignList = value.first();
        if(SignList.atomType()!=ListExpr.SYMBOL_ATOM)
            return "ERROR";
        String sign = SignList.symbolValue();
        if(sign.equals("-")) // ignore other values
            result += sign + " ";
        value = value.rest(); // skip the signum
    }
    // check the types
    if( value.first().atomType()!=ListExpr.INT_ATOM ||
        value.second().atomType()!=ListExpr.INT_ATOM ||
        value.fourth().atomType()!=ListExpr.INT_ATOM)
        return "ERROR";
    int intPart = value.first().intValue();
    int numDecimal = value.second().intValue();
    int denomDecimal = value.fourth().intValue();
    result += ""+(denomDecimal*intPart+numDecimal) + " / " + denomDecimal;
    return result;
}
```



Dsprational - Init-Funktionen

Erweiterung
von
SECOND0

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

```
public void init (String name, int nameWidth, int indent,  
                 ListExpr type, ListExpr value, QueryResult qr)  
{  
    String T = name;  
    String V = getValueString(value);  
    T=extendString(T,nameWidth, indent);  
    qr.addEntry(T + " : " + V);  
    return;  
}
```



Graphische Objekte

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- erweitern die `DisplayGraph` Klasse
- erzeugen eine Menge (oft Größe 1) von `java.awt.Shape` Objekten aus der geg. Liste
- verwenden Projektionen zur Darstellung geographischer Objekte
- Textdarstellung durch Überschreiben von `toString()`
- Beispiel: Displayklasse für Rechtecke (`Dsplrect`) auf den nächsten Folien
 - Member `rect` vom Typ `Rectangle.Double` zum Speichern der internen Darstellung

Hilfsfunktion zur Analyse der Liste

```
private void ScanValue (ListExpr v) {
    if (v.listLength() != 4) {
        Reporter.writeError("Error: 4 elements needed");
        err = true;
        return;
    }
    Double X1 = LEUtils.readNumeric(v.first()); ...
    if(X1==null || X2==null || Y1==null | Y2==null){
        Reporter.writeError("Error: not a numeric");
        err =true;
        return;
    }
    try{
        double tx1 = X1.doubleValue();    ...
        if(!ProjectionManager.project(tx1,ty1,aPoint)){
            err = true;
        } else{
            double x1 = aPoint.x; double y1 = aPoint.y;
            if(!ProjectionManager.project(tx2,ty2,aPoint)){
                err=true;
            } else{
                double x2 = aPoint.x; double y2 = aPoint.y;
                double x = Math.min(x1,x2); double w = Math.abs(x2-x1);
                double y = Math.min(y1,y2); double h = Math.abs(y2-y1);
                rect = new Rectangle2D.Double(x,y,w,h);
            }
        }
    }
    }catch(Exception e){
        err = true;
    }
}
```

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY



Init-Funktion

Erweiterung
von
SECOND0

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

```
public void init (String name, int nameWidth, int indent,
                  ListExpr type, ListExpr value, QueryResult qr) {
    AttrName = extendString(name, nameWidth, indent);
    ScanValue(value);
    if (err) {
        Reporter.writeError("Error in ListExpr :parsing aborted");
        qr.addEntry(new String("(" + AttrName + ": GA(rectangle)"));
        return;
    }
    else
        qr.addEntry(this);
}
```



numberOfShapes und getRenderObject

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

```
public int numberOfShapes(){  
    return 1;  
}
```

```
/** Returns the rectangle to display */  
public Shape getRenderObject(int num, AffineTransform at){  
    if(num<1){  
        return rect;  
    } else{  
        return null;  
    }  
}
```



Raum-Zeitliche Objekte

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

**Raum-Zeitliche
Objekte**

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- stellen zeitliche Veränderungen von geometrischen Objekten dar
- erweitern `DisplayTimeGraph` Klasse
- berechnen ihre Darstellung oft neu



DisplayTimeGraph

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

**Raum-Zeitliche
Objekte**

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- erweitert DisplayGraph
 - `Shape getRenderObject(int, AffineTransform)`
 - abhängig von aktueller Zeit
(`RefLayer.getActualTime()`)
- `Rectangle2D getBounds()` liefert Bounding Box des im gesamten Definitionsbereichs besuchten Raums
- `TimeBounds` Zeitintervall, in dem das Objekt definiert ist (ohne Lücken)



Sonstige Objekte

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

Objekte in eigenem Fenster darstellen (z.B. Bilder)

- große (nicht-graphische) Objekte schlecht darstellbar
- Verwendung eines eigenen Fensters (Interface `ExternDisplay`)
 - `displayExtern`
 - `isExternDisplayed`

Zeichnen komplexer Objekte (z.B. gedrehte Strings)

- Graphische Objekte, die nicht durch `Shape` dargestellt werden können
- implementieren `DisplayComplex` Interface
- Überschreiben der `draw` Methode

Label und Manipulation anderer Objekte

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- **Interface** LabelAttribute
 - `String getLabel(double Time)`
- **Interface** RenderAttribute
 - `boolean maybeDefined()`
 - `boolean isDefined(double time)`
 - `double getMaxRenderValue()`
 - `double getMinRenderValue()`
 - `double getRenderValue(double time)`



Anzeige im SecondoTTY(CS)

Erweiterung
von
SECONDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

Allgemeines:

- Anzeige als Liste oft schlecht lesbar
- wünschenswert: formatierte Darstellung
- möglich durch Displayklassen
 - `DisplayTTY.cpp` (Implementierung, Registrierung)
- für einfache und zusammengesetzte Typen möglich



Die Displayklasse (struct)

Erweiterung
von
SECONDDO

Thomas Behr

Javagui

Dokumentation

Erstellung eines
neuen Viewers

Beispiel -
Inquiry-Viewer

Erweiterung des
HoeseViewers

Objekte in
Textdarstellung

Graphische Objekte

Raum-Zeitliche
Objekte

Sonstige Objekte

Label und
Manipulation
anderer Objekte

Anzeige im
TTY

- wird abgeleitet von `DisplayFunction`
- `Display`-funktion wird überschrieben
- `Display(ListExpr type, ListExpr numType, ListExpr value)`
- `type`: gewöhnliche Listendarstellung. z.B. `(int)`
- `numType`: gleicher Inhalt, jedoch Typen kodiert durch Algebranzahl und Nummer des Typkonstruktors, für zusammengesetzte Typen benötigt
- `value`: darzustellendes Objekt

- innerhalb der Initialize Funktion in DisplayTTY.cpp
- `d.insert("typename",
new Displayclass());`