# MedioVis Framework
## Integration of a Hierarchical Data Structure

*Werner A. König*

*Practical Master Course*

*WG Human-Computer Interaction*
*University of Konstanz*

**Universität Konstanz**

# Outline

- History

- Requirements, Complexity & Functionality

- MedioVis Framework

- Start Sequence

- Observer & Singleton Pattern

- Data Handling

- Data Model – Table vs. Tree

- Results & Basket

- Selection, Filtering & Focus

- Future Work

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005
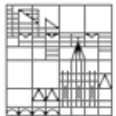
- Insyder

- *2002:* INVISIP (XML, DB)

- *May 2003:* INVISIP, Redesign Data Model

- *January 2004:* MedioVis, Redesign Framework

- *September 2004:* MedioVisX, Refactoring Framework

- *January 2005:* MedioVisX, Multi-Dimensional Data (IMDb)

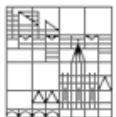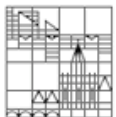- *October 2005:* MedioVisX$^T$, Hierarchical Data Integration

  …

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

**Flexibility in the area of…**

- Data Input

- Information Retrieval

- Data Handling **Flexibility in terms of…**

- Visualizations → Integration

- Window Segmentation → Modification

- Interaction (internal & global) → Extension

- General Appearance

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005
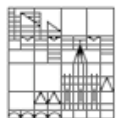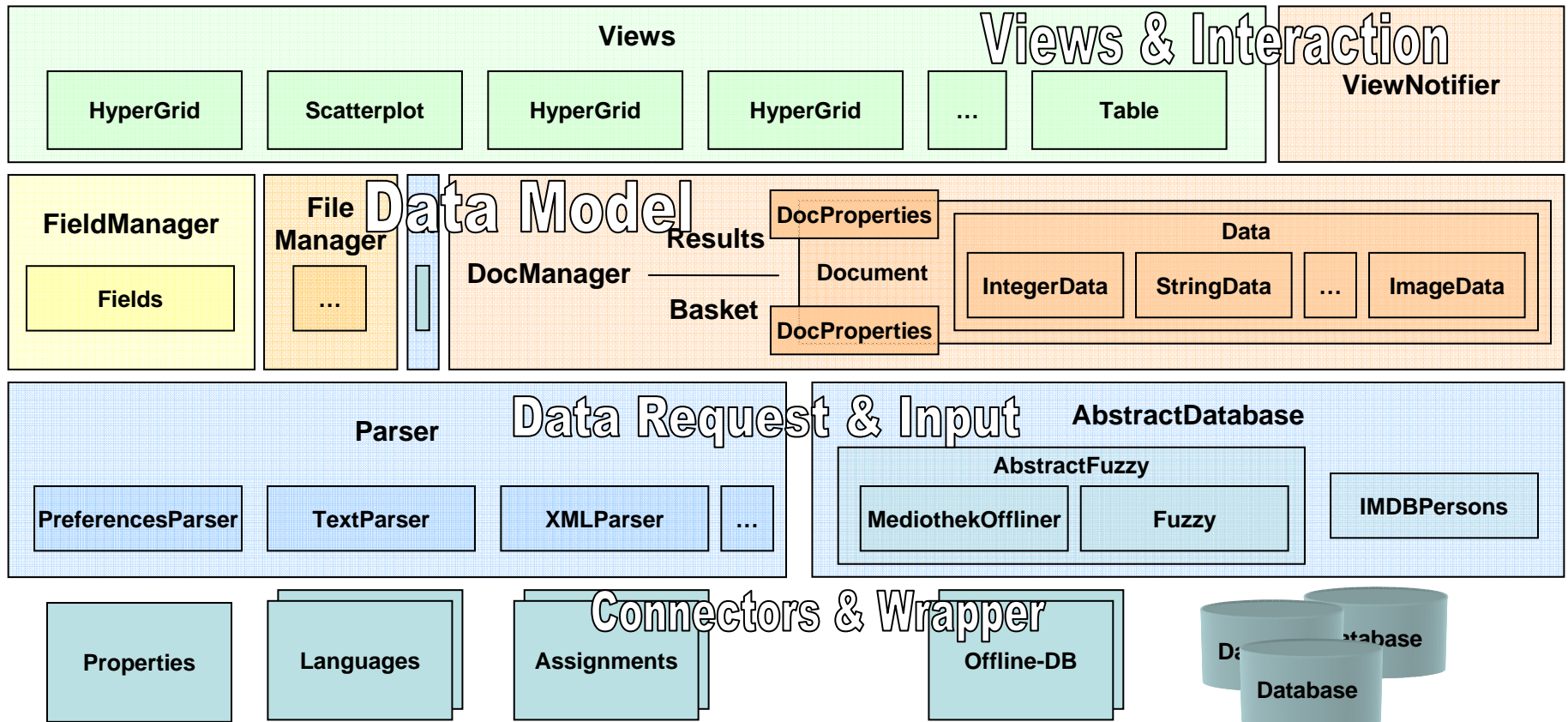
- Data Organization:

    - Heterogeneous Data (string, integer, image, movie,…)

    - Multi-Dimensional Data (meta data, relationships, dependancies)

    - Hierarchical Data (arbitrarily tree structure)

- Basket & Result (independent data, identical interface)

- Temporary and persistent data handling

    - No unique data identifier (hash comparison)

- Different Data Resources and Types

- Performance (dynamic interaction, large information space)

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

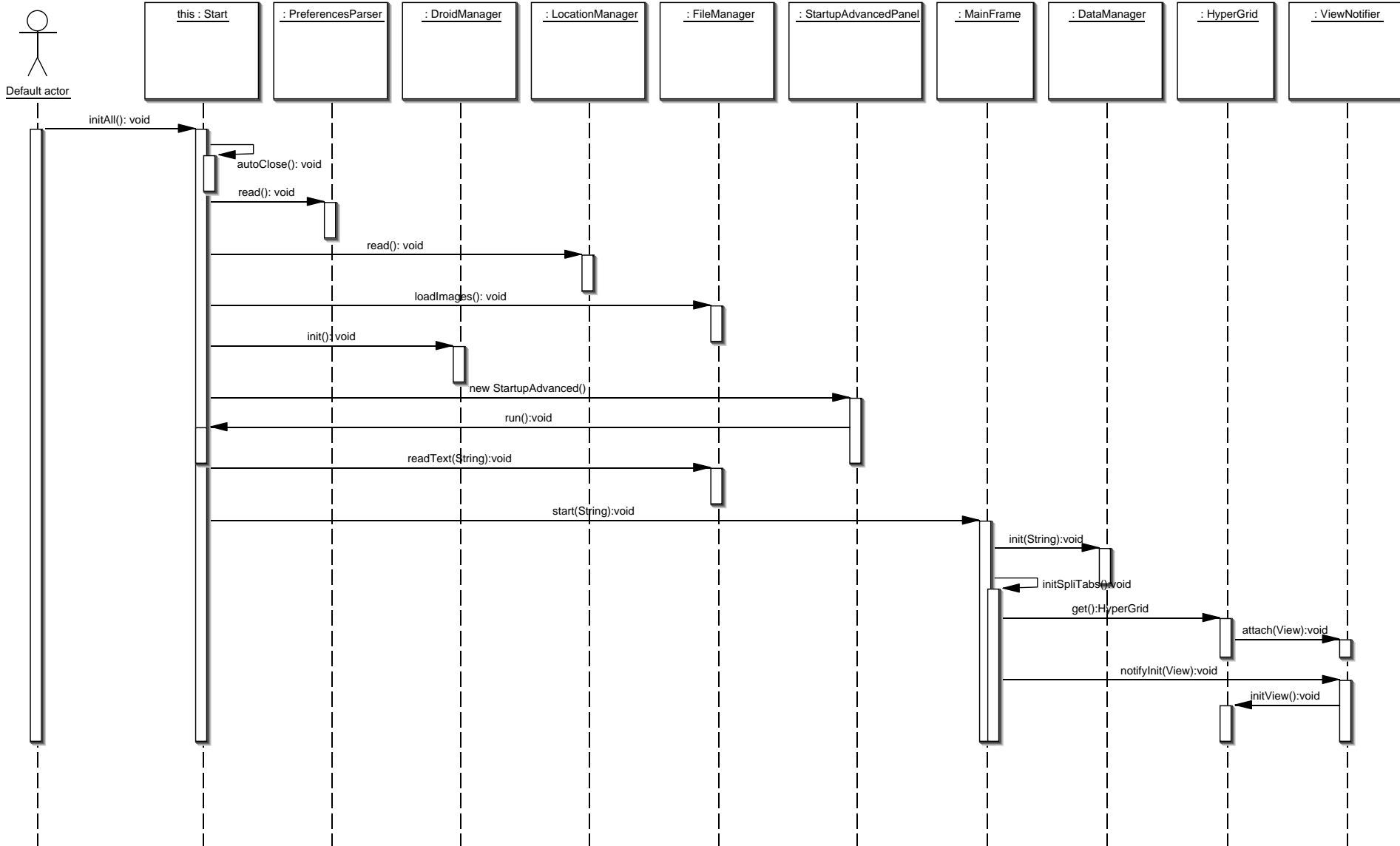Practical Master Course
Project MedioVis
10/24/2005

- Multilevel Abstraction, well defined interfaces

- Intelligent loading and rendering techniques

- Client / Server architecture

- Variable Assignments

- Dynamic Filtering

  - Local & global

- Sorting

- History

Werner A. Koenig
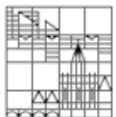WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005
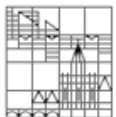
```
public interface View {
  public abstract void initView();
  public abstract void updateView();
  public abstract void repaintView();
  public abstract void repaintSelection(int id);
  public abstract void repaintFocus(Document doc);
  public abstract void repaintZoom(int[] ids, int diff);
  public abstract void finish();
}

public class ViewNotifier {
  private static View[] views;
  public static void attach(View view) {}
  public static void notifyInit() {}
  public static void notifyUpdate() {}
  public static void notifyRepaint() {}
  public static void notifySelection(View source, int id) {}
  public static void notifyFocus(View source, int id,
      String hash) {}
  public static void notifyZoom(View source, int[] ids,
      int diff){}
  public static void notifyFinish() {}
  …
}
```

```
public class HyperGrid extends JPanel
    implements View {
  private static HyperGrid instance;
  private HyperGrid() {
    ViewNotifier.attach(this);
    …
  }

  public static HyperGrid get() {
    if(instance == null) instance = new HyperGrid();
    return instance;
  }

  public void initView() {}
  public void updateView() {}
  public void repaintView() {}
  public void repaintSelection(int id) {}
  public void repaintFocus(Document doc) {}
  public void repaintZoom(int[] id, int diff) {}
  public void finish() {}
  …
}
```

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

# Data Handling

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

# D.M. Table

```java
public class DocManager implements DataTypes, Serializable {
    private Document[] DOCS = new Document[capacity];
    private int[] sortedIDs = new int[capacity];
    private int[] sortedRows = new int[capacity];
    …
    public Document getDoc(int id) {
        return DOCS[id];
    }
    public int getID(Document doc) {
        return HASH.getID(doc);
    }
    public int getID(int row) {
        return sortedIDs[row];
    }
    public int getRow(int id) {
        return sortedRows[id];
    }
    public void sortDocs() {
        sortDocs(lastSorted, lastOrdered);
        FileManager.preloading(this);
    }
}


public class Document implements Serializable {
    private Data[] data;
    private boolean selected, focused;
    private boolean availableHGrid, availablePlot;
…
    public Data getData(int field) {
        return data[field];
    }
}
```

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

---

**DataType**
*Serializable*
**DocManager**

+RESULTS:int=0
+BASKET:int=1
+NUMMODES:int=2
-manager:DocManager[]=ne
-SORTLEVELS:int=3
-currentMode:int=RESULTS
-managerMode:int
-capacity:int=10
-DOCS:Document[]=new D
-HASH:DocHash
-lastOrdered:boolean[]
-lastSorted:int[]
-mins:int[]
-maxs:int[]
-size:int=0
-available:int=0
-selected:int=0
-sortedIDs:int[]=new int[cap
-sortedRows:int[]=new int[ca
-sortOrder:int=-1

+clear():void
+DocManager(mode:int)
+get():DocManager
+get(mode:int):DocManage
+getViewMode():int
+reset():void
+setViewMode(mode:int):v
+add(data:Data[]):void
+add(doc:Document,reset:b
+addDocs(newDocs:java.ut
-compare(field:int,i:int,j:int):i
+countSelected():void
#createMinMax():void
-exchange(i:int,j:int):void
+exists(doc:Document):boo
+getDoc(hash:String):Docu
+getFocusedDoc():Docume
+getID(doc:Document):int
+getMax(field:int):Data
+getMin(field:int):Data
+getRootDoc(doc:Documer
+getRootID(doc:Document)
+init():void
+initSort(field:int,asc:boolea
+isMissing(field:int):boolean
-levelSort(fields:int[],level:in
+remove(id:int):void
+sameFocus(id:int):boolean
+setFocusedDoc(doc:Docu
+setFocusedDoc(id:int,hash
+setMax(field:int,max:int):v
+setMin(field:int,min:int):voi
+size():int
+sizeAvailable():int
+sizeSelected():int
-sort(field:int,s:int,e:int):void
+sortDocs():void
+sortDocs(field:int):void
+sortDocs(fields:int[],asc:bo
+sortedAscending():boolean
+update():void

---

**Serializable**
**Document**

-hashcode:int=0
-isAnimated:boolean
-docProps:DocProperties[]=

+Document(data:Data[])
+checkBasketSelection():bc
-calcHash():void
+hashCode():int
+setData(data:Data,field:int
+isAvailable(mode:int):bool
+isSelected(mode:int):boole
+addChild(child:Document)
+nrSubDocs():int
+searchByHash(hash:String
+sortSubs(field:int,order:int)
+hasMoreExemplars():bool

hash:String
size:int
data:Data[]
availablePlot:boolean[]
availableHGrid:boolean[]
selected:boolean[]
focused:boolean
color:Color
childsLength:int
child:Document[]
allDocuments:Document[]
availableDocs:Document[][
parent:Document
showingChilds:boolean
moreExemplars:boolean
animated:boolean

---

**DataType**
**FieldManager**

+CAPACITY:int=50
+DBFIELDS:StringHash
+types:int[]=new int[CAPACIT
+fields:String[]=new String[CA
+names:String[]=new String[C
+size:int=0
-segmentKeys:String[]
-segmentSizes:int[]
-sortOrder:int

+reset():void
+add(id:int,field:String,type:int
+getDBField(id:int):String
+getDBType(field:int):int
+getDBFieldID(field:String):int
+getName(id:int):String
+size():int
+getID(name:String):int
+calcSegments(field:int,limit:b
+sortSegmentsByOccurancy()
+sortSegmentsByKeys(asc:bo
+limitSegments(max:int):void
+getSegmentKeys():String[]
+getSegmentSizes():int[]
-sort(s:int,e:int):void
-compare(i:int,j:int):int
-exchange(i:int,j:int):void

---

**DataTypes**
+BooleanDat
+*Data*
+DateData
+DocHash
+DocTree
+DoubleData
+IDList
+ImageData
+IntegerData
+LongData
+StringData
+StringHash
+StringList
+Text

**Constants**
+Coding
+*DataTypes*
+Fields
+*Library*
+Preference
+Texts

0..*

**Several Copies**

**Multipart Items**

A
B
C
D

A
B
C
D

A    A
Ai  Aii  Aiii

| Title | Author | Year | Lang | Type | | Call No. | Status |
|---|---|---|---|---|---|---|---|
| Der **Brockhaus** Philosophie | Conradi, Elisabeth | 2004 | dt. | | Buch | inf 425:s/b76... | ▼ mehrere Exemplare |
| Der **Brockhaus** Psychologie | Starke-Perschke, Susanne; Bliesene... | 2001 | dt. | | Buch | inf 435:s/s91 | nicht ausleihbar |
| Der **Brockhaus** Zeitgeschichte | Hotz, Jürgen; Schaaf, Michael | 2003 | dt. | | Buch | gsx 10:c/h69 | ■ verfügbar |
| Der **Brockhaus**-Atlas zur Geschichte | Hotz, Jürgen; Engehausen, Frank | 2005 | dt. | | Buch | gsb 860/h69 | ■ verfügbar |
| Der grosse **Brockhaus** | | | dt. | | diverse | N 78/207...<br>N 78/207<br>inf 171:c/f16 | mehrere Exemplare<br>mehrteiliges Werk<br>■ unbekannt |
| A - Beo | | 1952 | dt. | | Buch | N 78/207-1<br>N 78/207-1<br>inf 171:c/f16-1 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Atlas : Erdkunde, Wirtschaft, Geschichte | | 1960 | dt. | | Buch | N 78/207-Atlas...<br>N 78/207-Atlas<br>inf 171:c/f16-15 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Ber - Cz | | 1953 | dt. | | Buch | N 78/207-2...<br>N 78/207-2<br>inf 171:c/f16-2 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| D - Faz | | 1953 | dt. | | Buch | N 78/207-3...<br>N 78/207-3<br>inf 171:c/f16-3 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Ergänzungsband | | 1958 | dt. | | Buch | N 78/207-13...<br>N 78/207-13<br>inf 171:c/f16-13 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Ergänzungsband : 2, A - Z | | 1963 | dt. | | Buch | N 78/207-14...<br>N 78/207-14<br>inf 171:c/f16-14 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Fba - Goz | | 1954 | dt. | | Buch | N 78/207-4...<br>N 78/207-4<br>inf 171:c/f16-4 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| Gp - Iz | | 1954 | dt. | | Buch | N 78/207-5...<br>N 78/207-5<br>inf 171:c/f16-5 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| J - Kz | | 1955 | dt. | | Buch | N 78/207-6...<br>N 78/207-6<br>inf 171:c/f16-6 | mehrere Exemplare<br>■ verfügbar<br>nicht ausleihbar |
| L - Mij | | 1955 | dt. | | Buch | N 78/207-7...<br>N 78/207-7 | mehrere Exemplare<br>■ verfügbar |

# D.M. Tree

```java
public class DocManager implements DataTypes, Serializable {
    private Document[] DOCS = new Document[capacity];
    private int[] sortedIDs = new int[capacity];
    private int[] sortedRows = new int[capacity];
    …
        public Document getRootDoc(Document doc) {
}

public class Document implements Serializable {
    private Data[] data;
    private Document[] childs = null;
    private Document parent = null;
    private boolean showingChilds = false;
    private DocProperties[] docProps;
    …
    public Document[] getAllDocuments() {…}
    public Document[] getAvailableDocs(int mode) {…}
    public Document searchByHash(String hash){…}
}

public class DocProperties implements Serializable {
    private boolean availablePlot = true;
    private boolean availableHGrid = true;
    private boolean selected = false;
    private Document doc = null;
    …
}
```

→ Strong use of recursion
→   Pretend flat array for non-hierarchical Vizs

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

---

**DataTypes**
**Serializable**
**DocManager**

+RESULTS:int=0
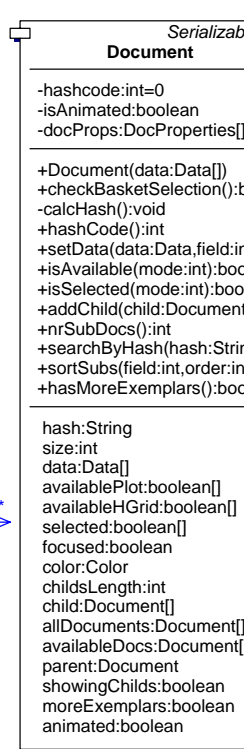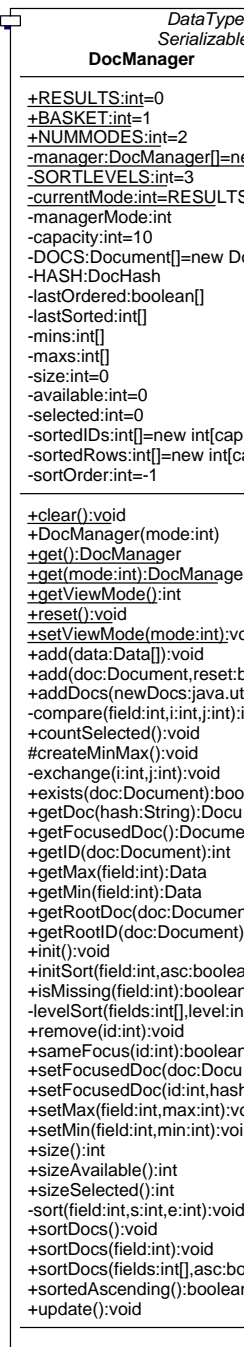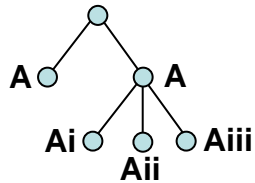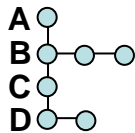+BASKET:int=1
+NUMMODES:int=2
-manager:DocManager[]=ne
-SORTLEVELS:int=3
-currentMode:int=RESULTS
-managerMode:int
-capacity:int=10
-DOCS:Document[]=new D
-focusedDoc:Document=nu
-HASH:DocHash
-lastOrdered:boolean[]
-lastSorted:int[]
-mins:int[]
-maxs:int[]
-size:int=0
-available:int=0
-selected:int=0
-sortedIDs:int[]=new int[cap
-sortedRows:int[]=new int[ca
-sortOrder:int=-1

+clear():void
+DocManager(mode:int)
+get():DocManager
+get(mode:int):DocManage
+getViewMode():int
+reset():void
+setViewMode(mode:int):vo
+add(data:Data[]):void
+add(doc:Document,reset:b
+addDocs(newDocs:java.ut
+compare(field:int,i:int,j:int):i
+countSelected():void
#createMinMax():void
-exchange(i:int,j:int):void
+exists(doc:Document):boo
+getDoc(hash:String):Docu
+getFocusedDoc():Docume
+getID(doc:Document):int
+getMax(field:int):Data
+getMin(field:int):Data
+getRootDoc(doc:Documer
+getRootID(doc:Document)
+init():void
+initSort(field:int,asc:boolea
+isMissing(field:int):boolear
-levelSort(fields:int[],level:in
+remove(id:int):void
+sameFocus(id:int):boolear
+setFocusedDoc(doc:Docu
+setFocusedDoc(id:int,hash
+setMax(field:int,max:int):v
+setMin(field:int,min:int):voi
+size():int
+sizeAvailable():int
+sizeSelected():int
-sort(field:int,s:int,e:int):void
+sortDocs():void
+sortDocs(field:int):void
+sortDocs(fields:int[],asc:bo
+sortedAscending():boolear
+update():void

---

**Serializable**
**Document**

-hashcode:int=0
-childs:Document[]=null
-isAnimated:boolean
-docProps:DocProperties[]=

+Document(data:Data[])
+checkBasketSelection():bo
+calcHash():void
+hashCode():int
+setData(data:Data,field:int
+isAvailable(mode:int):boole
+isSelected(mode:int):boole
+addChild(child:Document)
+nrSubDocs():int
+searchByHash(hash:String
+sortSubs(field:int,order:int)
+hasMoreExemplars():boole

hash:String
size:int
data:Data[]
availablePlot:boolean[]
availableHGrid:boolean[]
selected:boolean[]
focused:boolean
color:Color
childsLength:int
child:Document[]
allDocuments:Document[]
availableDocs:Document[][
parent:Document
showingChilds:boolean
moreExemplars:boolean
animated:boolean

---

**Serializable**
**DocProperties**

-doc:Document=null

+DocProperties(doc:Docum

available:boolean
availablePlot:boolean
availableHGrid:boolean
selected:boolean
select:boolean

---

**DataType**
**FieldManager**

+CAPACITY:int=50
+DBFIELDS:StringHash
+types:int[]=new int[CAPACIT
+fields:String[]=new String[CA
+names:String[]=new String[C
+size:int=0
-segmentKeys:String[]
-segmentSizes:int[]
-sortOrder:int

+reset():void
+add(id:int,field:String,type:int
+getDBField(id:int):String
+getDBType(field:int):int
+getDBFieldID(field:String):int
+getName(id:int):String
+size():int
+getID(name:String):int
+calcSegments(field:int,limit:b
+sortSegmentsByOccurancy()
+sortSegmentsByKeys(asc:bo
+limitSegments(max:int):void
+getSegmentKeys():String[]
+getSegmentSizes():int[]
-sort(s:int,e:int):void
-compare(i:int,j:int):int
-exchange(i:int,j:int):void

---

**DataTypes**
+BooleanDat
+Data
+DateData
+DocHash
+DocTree
+DoubleData
+IDList
+ImageData
+IntegerData
+LongData
+StringData
+StringHash
+StringList
+Text

---

**Constants**
+Coding
+DataTypes
+Fields
+Library
+Preference
+Texts

Temporary and persistent data (Serialization)

   → Independent data, identical interface

- More instances of the data model

- Save switching by well defined interfaces

- Identification by unique hash value

- Easy transfer between instances (garbage management)

- Selection and Focus Synchronization

- Memory optimization by the use of references

- Hierarchical Tree Support

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

# Data Model Objects

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005
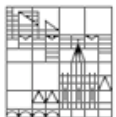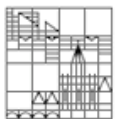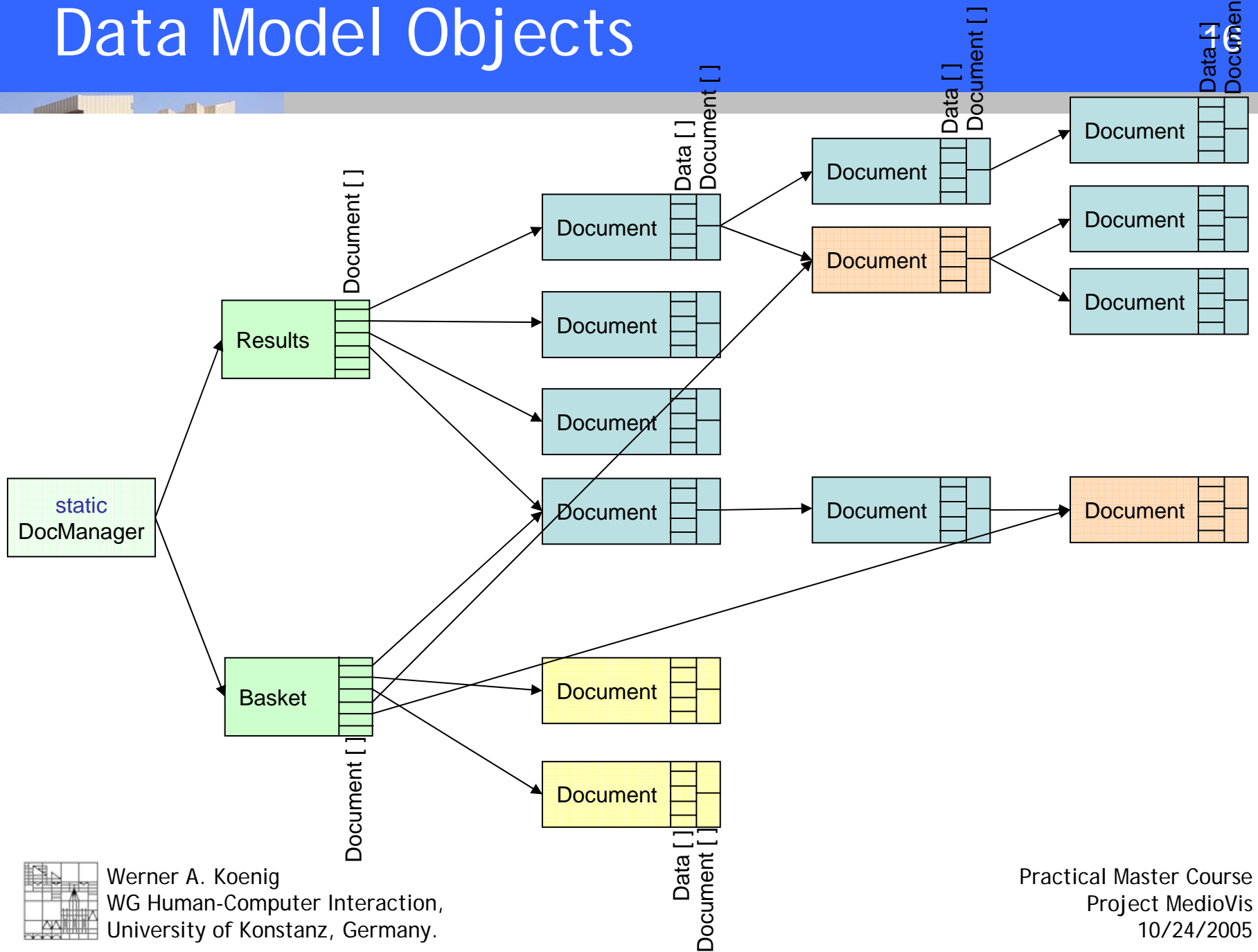
Where to save availability and selection?

- DocManager ? : Independent tree structure needed

- Document ? : Independent variables needed

→ DocProperties for each DocManager at each Document
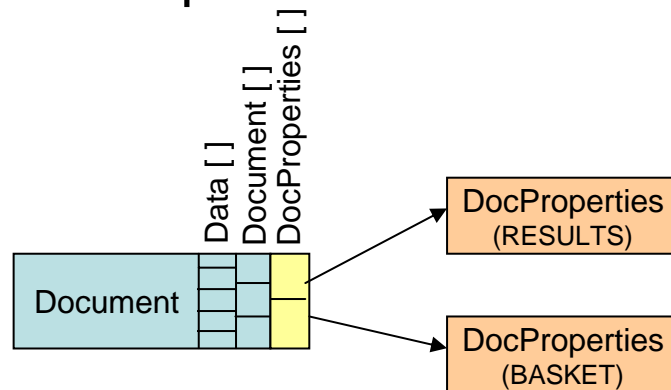


```
public class DocProperties implements Serializable
{
    private boolean availablePlot = true;
    private boolean availableHGrid = true;
    private boolean selected = false;
    private Document doc = null;
    …
}
```

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005
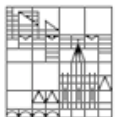
# D.M. Tree

public class **DocManager** implements DataTypes, Serializable {
   private Document[] DOCS = new Document[capacity];
   private int[] sortedIDs = new int[capacity];
   private int[] sortedRows = new int[capacity];
   …
      public Document getRootDoc(Document doc) {
}

public class **Document** implements Serializable {
   private Data[] data;
   private Document[] childs = null;
   private Document parent = null;
   private boolean showingChilds = false;
   private **DocProperties[] docProps**;
   …
   public Document[] getAllDocuments() {…}
   public Document[] getAvailableDocs(int mode) {…}
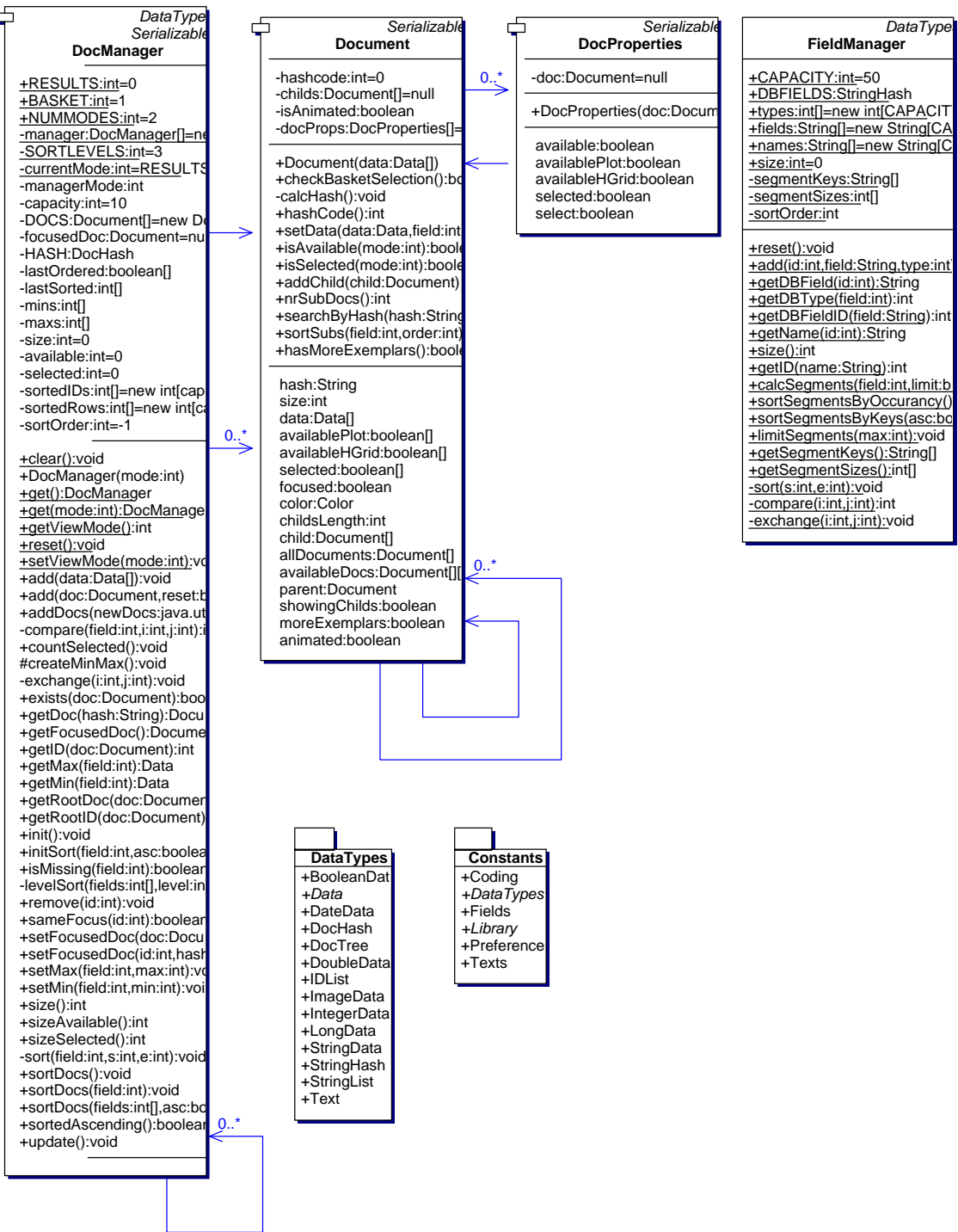   public Document searchByHash(String hash){…}
}

public class **DocProperties** implements Serializable {
   private boolean availablePlot = true;
   private boolean availableHGrid = true;
   private boolean selected = false;
   private Document doc = null;
   …
}

→ Object Oriented Hierarchical Structure

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

## DocManager
*DataType*
*Serializable*

+RESULTS:int=0
+BASKET:int=1
+NUMMODES:int=2
-manager:DocManager[]=ne
-SORTLEVELS:int=3
-currentMode:int=RESULTS
-managerMode:int
-capacity:int=10
-DOCS:Document[]=new Do
-focusedDoc:Document=nu
-HASH:DocHash
-lastOrdered:boolean[]
-lastSorted:int[]
-mins:int[]
-maxs:int[]
-size:int=0
-available:int=0
-selected:int=0
-sortedIDs:int[]=new int[cap
-sortedRows:int[]=new int[ca
-sortOrder:int=-1

+clear():void
+DocManager(mode:int)
+get():DocManager
+get(mode:int):DocManage
+getViewMode():int
+reset():void
+setViewMode(mode:int):vo
+add(data:Data[]):void
+add(doc:Document,reset:b
+addDocs(newDocs:java.ut
-compare(field:int,i:int,j:int):i
+countSelected():void
#createMinMax():void
-exchange(i:int,j:int):void
+exists(doc:Document):boo
+getDoc(hash:String):Docu
+getFocusedDoc():Docume
+getID(doc:Document):int
+getMax(field:int):Data
+getMin(field:int):Data
+getRootDoc(doc:Documen
+getRootID(doc:Document)
+init():void
+initSort(field:int,asc:boolea
+isMissing(field:int):boolean
-levelSort(fields:int[],level:in
+remove(id:int):void
+sameFocus(id:int):boolean
+setFocusedDoc(doc:Docu
+setFocusedDoc(id:int,hash
+setMax(field:int,max:int):vo
+setMin(field:int,min:int):voi
+size():int
+sizeAvailable():int
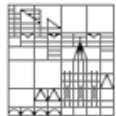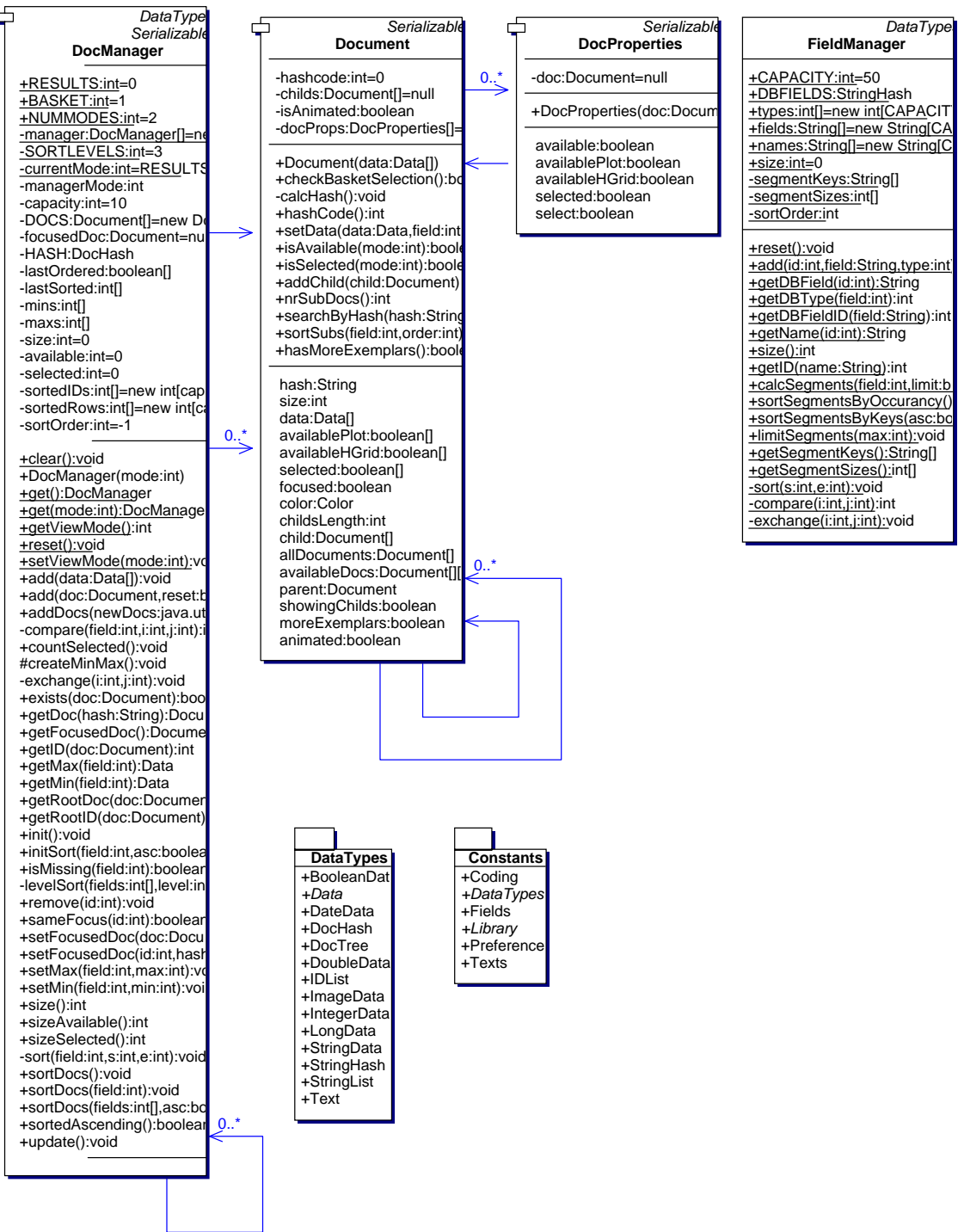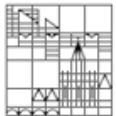+sizeSelected():int
-sort(field:int,s:int,e:int):void
+sortDocs():void
+sortDocs(field:int):void
+sortDocs(fields:int[],asc:bo
+sortedAscending():boolean
+update():void

## Document
*Serializable*

-hashcode:int=0
-childs:Document[]=null
-isAnimated:boolean
-docProps:DocProperties[]=

+Document(data:Data[])
+checkBasketSelection():bo
-calcHash():void
+hashCode():int
+setData(data:Data,field:int
+isAvailable(mode:int):boole
+isSelected(mode:int):boole
+addChild(child:Document)
+nrSubDocs():int
+searchByHash(hash:String
+sortSubs(field:int,order:int)
+hasMoreExemplars():boole

hash:String
size:int
data:Data[]
availablePlot:boolean[]
availableHGrid:boolean[]
selected:boolean[]
focused:boolean
color:Color
childsLength:int
child:Document[]
allDocuments:Document[]
availableDocs:Document[][
parent:Document
showingChilds:boolean
moreExemplars:boolean
animated:boolean

## DocProperties
*Serializable*

-doc:Document=null

+DocProperties(doc:Docum

available:boolean
availablePlot:boolean
availableHGrid:boolean
selected:boolean
select:boolean

## FieldManager
*DataType*

+CAPACITY:int=50
+DBFIELDS:StringHash
+types:int[]=new int[CAPACIT
+fields:String[]=new String[CA
+names:String[]=new String[C
+size:int=0
-segmentKeys:String[]
-segmentSizes:int[]
-sortOrder:int

+reset():void
+add(id:int,field:String,type:int
+getDBField(id:int):String
+getDBType(field:int):int
+getDBFieldID(field:String):int
+getName(id:int):String
+size():int
+getID(name:String):int
+calcSegments(field:int,limit:b
+sortSegmentsByOccurancy()
+sortSegmentsByKeys(asc:bo
+limitSegments(max:int):void
+getSegmentKeys():String[]
+getSegmentSizes():int[]
-sort(s:int,e:int):void
-compare(i:int,j:int):int
-exchange(i:int,j:int):void

## DataTypes
+BooleanDat
+*Data*
+DateData
+DocHash
+DocTree
+DoubleData
+IDList
+ImageData
+IntegerData
+LongData
+StringData
+StringHash
+StringList
+Text

## Constants
+Coding
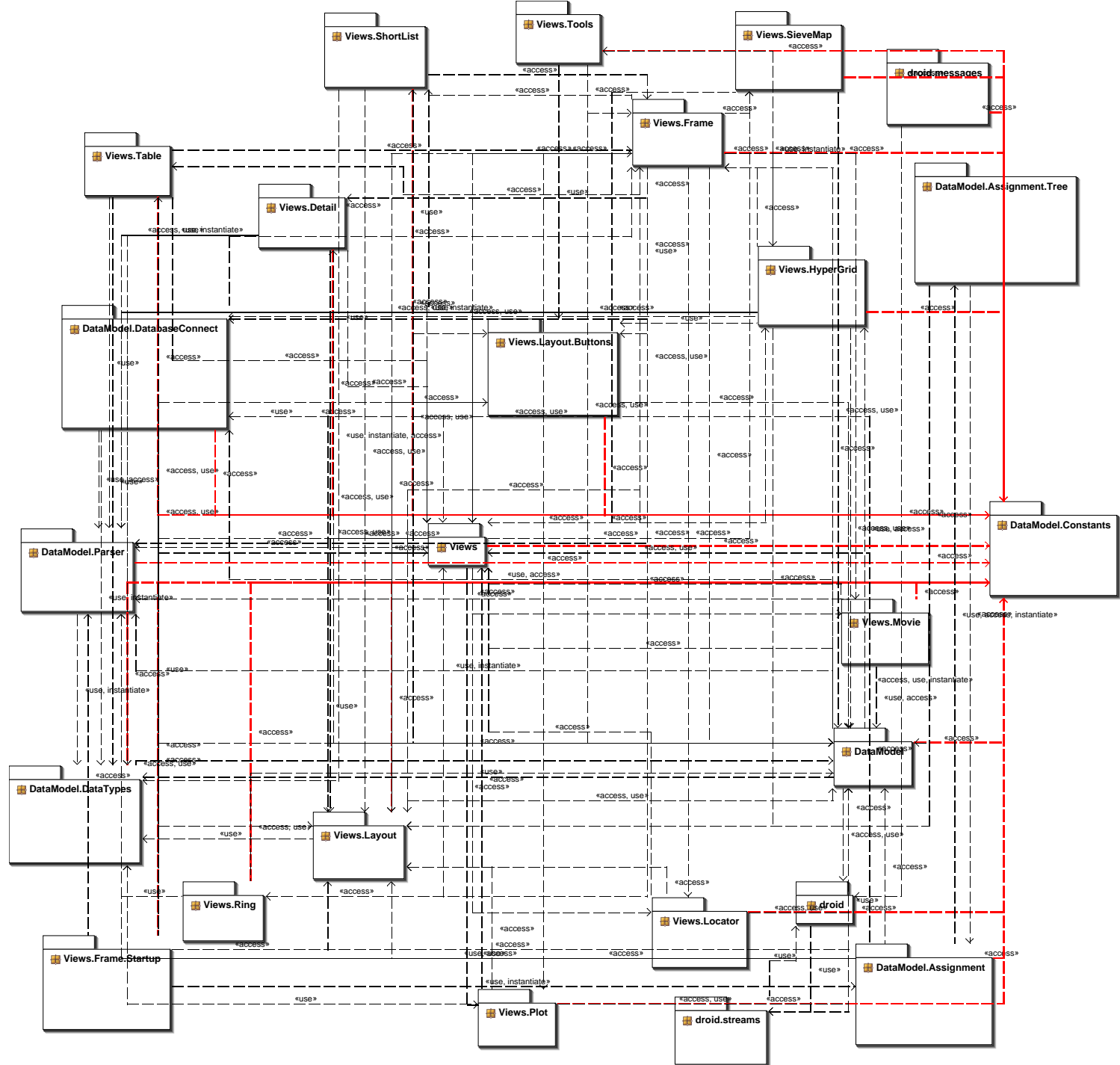+*DataTypes*
+Fields
+*Library*
+Preference
+Texts

0..*

```java
public class ViewNotifier {
…
  public static void notifyFocus(View source, int id, String hash) {}
…
}

public class ShortListView extends JPanel implements View {

    public void repaintFocus(Document doc) {
        DocManager basket = DocManager.get(DocManager.BASKET);
        Document sub = basket.getDoc(doc.getHash());
        if(sub != null) {
            basket.setFocusedDoc(basket.getRootID(sub), sub.getHash());
            panel.jumpTo(sub);
        }
    }

    public void checkFocus(final int yPosition) {
        Document document = getCurrentDoc(yPosition);
        int resultID = DocManager.get(DocManager.RESULTS).getRootID(document);
        ViewNotifier.notifyFocus(view, resultID, document.getHash());
    }
}
```

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005

- Performance Optimization:

    Caching of unchanged tree structure

    Initiate Basket Properties only when needed

- Extend DocManager to "FavoriteManager" for Personalization

- Integration of Graph Visualizations

- Level based zooming through document hierarchy

- Visual and Interaction Concept for hierarchical Data in

    HyperGrid, Scatterplot,…

Werner A. Koenig
WG Human-Computer Interaction,
University of Konstanz, Germany.

Practical Master Course
Project MedioVis
10/24/2005