



OpenDECL – Open Display Environment Configuration Language

XML-basierte Konfiguration von Multi-Display-Umgebungen

Dresden, 1. Jun 2015



Gliederung:

1. Aufgabenstellung

- Das Problem
- Bestehende Lösungsansätze
- Die Idee dieser Arbeit

2. Aktueller Stand

- Hardwareanalyse
- XML-Spezifikation

3. Probleme & Weiteres Vorgehen

4. Fragen

Aufgabenstellung – Das Problem

- Verschiedene Multi-Display Installationen
- Verschiedene Software → meist sehr komplex
- Komplexe Infrastruktur für Betrieb
- Software-Konfiguration → Aufwändig
- Kein Standard für Beschreibung



TU Dresden, 01.06.15

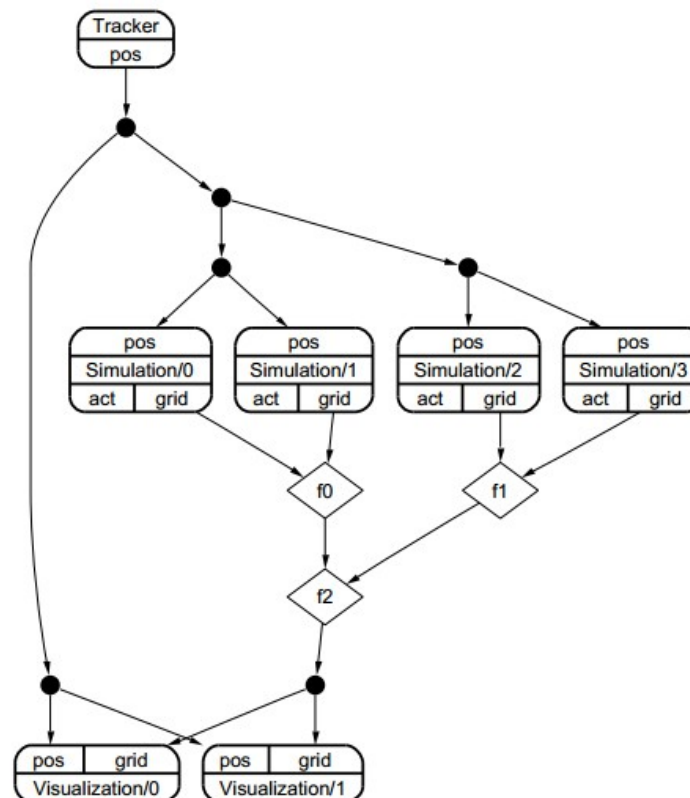


OpenDECL – Open Display Environment
Configuration Language

Folie 3 von XYZ

Aufgabenstellung – Lösungsansatz FlowVR

- Middleware für Clusterbased Computing
- Framework f. Verbindung bestehender Softwarekomponenten
- Komplexe Kommunikation im Cluster
- Applicationmodel:



Aufgabenstellung – Lösungsansatz VR Juggler

- Entwicklung v. VR-Anwendung für CAVE
- Wegkapsleung d. Implementierungsprobleme bezüglich Peripherie (Input, Output
- Kommunikation d. Anwendung mit Kernel anhand von Interfaces
- Anwendungen sind Objekte die ausgeführt werden
- Kernel mit verschiedenen Modulen (Managern) f. Entsprechende Aufgaben
- Für neue Anwendungen
- Ähnlichkeit zu JVM

Aufgabenstellung – Lösungsansatz VRUI & middleVR

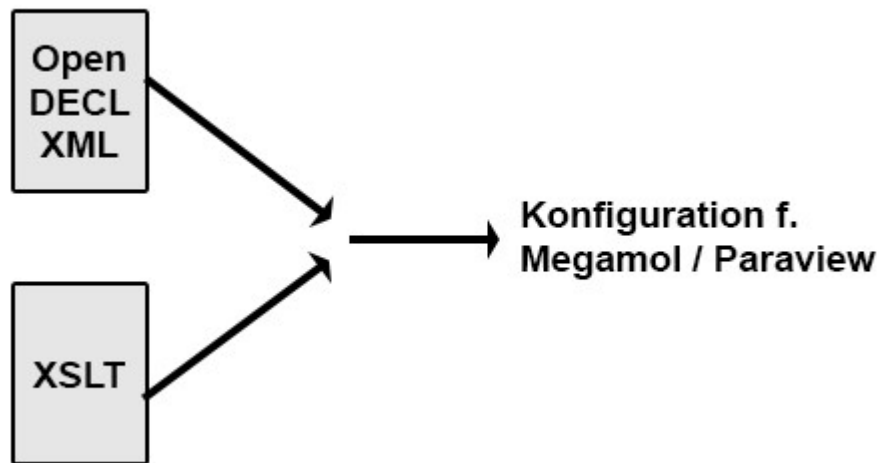
- Toolkits für Entwicklung von VR-Anwendungen
- Getrenntes Beschreiben von Viewer & Display
- Beschreibung der Displayumgebung in Konfigurationen

```
24 section Vrui
25
26 # Configuration for desktop environments:
27 section Desktop
28     inchScale 1.0
29     displayCenter (0.0, 0.0, 0.0)
30     displaySize 10.0
31     forwardDirection (0.0, 1.0, 0.0)
32     upDirection (0.0, 0.0, 1.0)
33     floorPlane (0.0, 0.0, 1.0), -48.0
34     glyphSize 0.333
35     newInputDevicePosition (0.0, -1.0, 0.0)
36     inputDeviceAdapterNames (MouseAdapter)
37     updateContinuously false
38     viewerNames (Viewer)
39     listenerNames (Listener)
40     screenNames (Screen)
41     windowNames (Window)
42     soundContextName SoundContext
43     frontplaneDist 1.0
44     backplaneDist 1000.0
45     backgroundColor (0.0, 0.0, 0.0, 1.0)
```

```
</DeviceManager>
<DisplayManager Fullscreen="0" WindowBorders="0" ShowMouseCursor="0" VSync="1" SaveRende:
    <Node3D Name="CenterNode" Parent="VRRootNode" Tracker="0" PositionLocal="0.000000,0.0
    <Node3D Name="HandNode" Tag="Hand" Parent="CenterNode" Tracker="0" PositionLocal="0.0
    <Node3D Name="HeadNode" Tag="Head" Parent="CenterNode" Tracker="0" PositionLocal="0.0
    <CameraStereo Name="CameraStereo0" Parent="HeadNode" Tracker="0" PositionLocal="0.00
    "2" InterEyeDistance="0.065" />
    <Screen Name="Screen0" Parent="CenterNode" Tracker="0" PositionLocal="0.000000,2.000
    <Viewport Name="Viewport0" Left="0" Top="0" Width="1280" Height="720" Camera="Camera
```

Aufgabenstellung – Die Idee dieser Arbeit

- Umgebungsbeschreibung in XML
- Abbildung der Infrastruktur
- Namespaces zur Abtrennung und Erweiterung
- Generieren der Konfigurationen Mittels XSLT



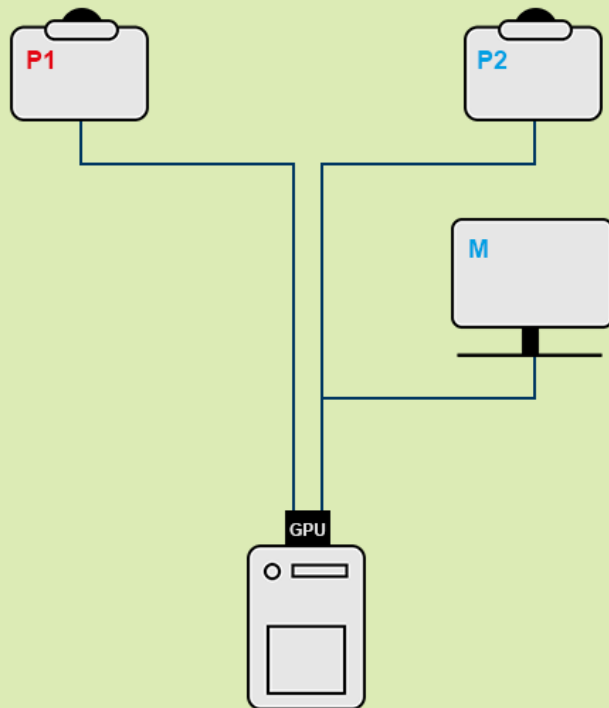
Aktueller Stand – Hardwareanalyse

- Verschiedene Szenarien:
 - Desktop mit mehreren Displays
 - CGV Stereo Powerwall
 - MT LCD Powerall
 - 5 Seiten CAVE CAD-Labor

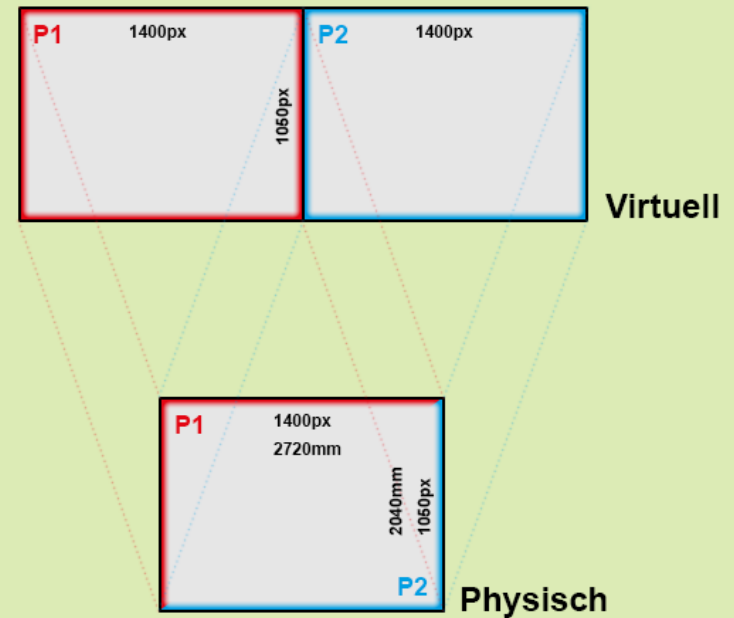
Aktueller Stand – Hardwareanalyse CGV Stereowall

CGV Stereowall

Aufbau Hardware



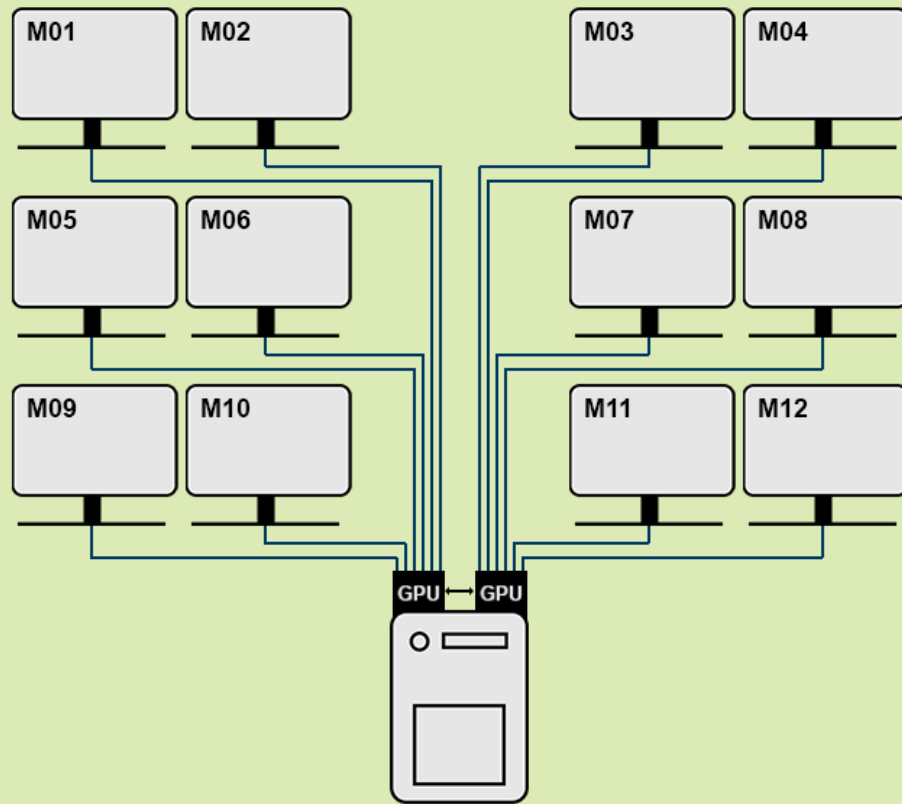
Aufbau Display



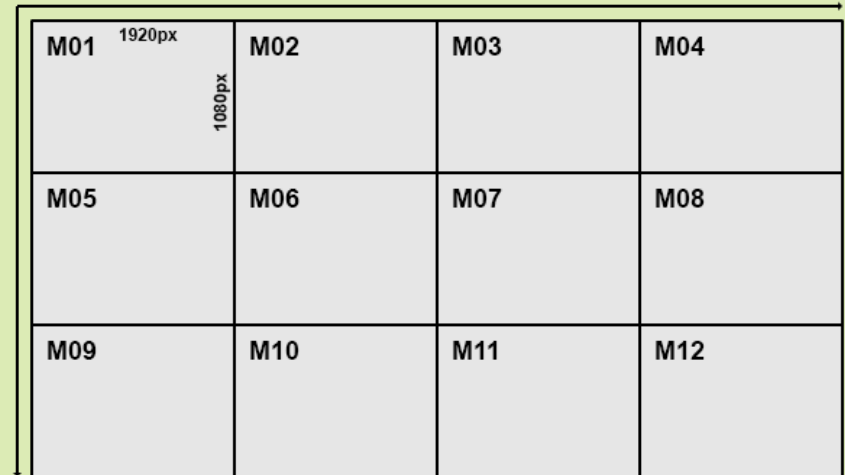
Aktueller Stand – Hardwareanalyse MT Powerwall

MT Powerwall (AMD Radeon HD 7970)

Aufbau Hardware



Aufbau Display

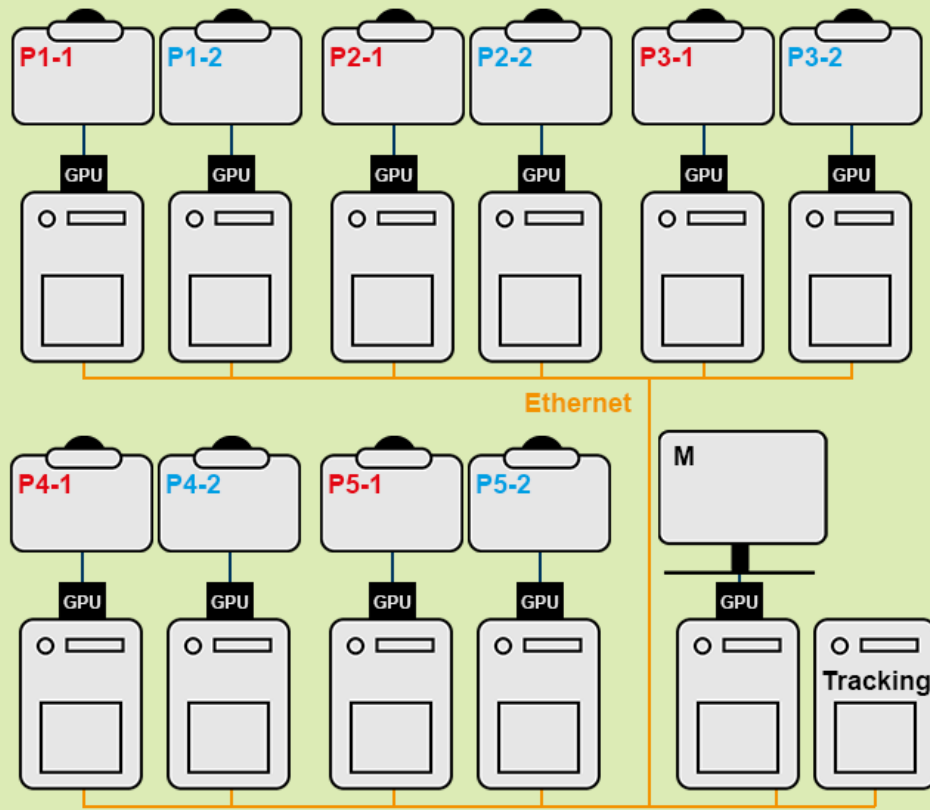


Aktueller Stand – Hardwareanalyse CAVE

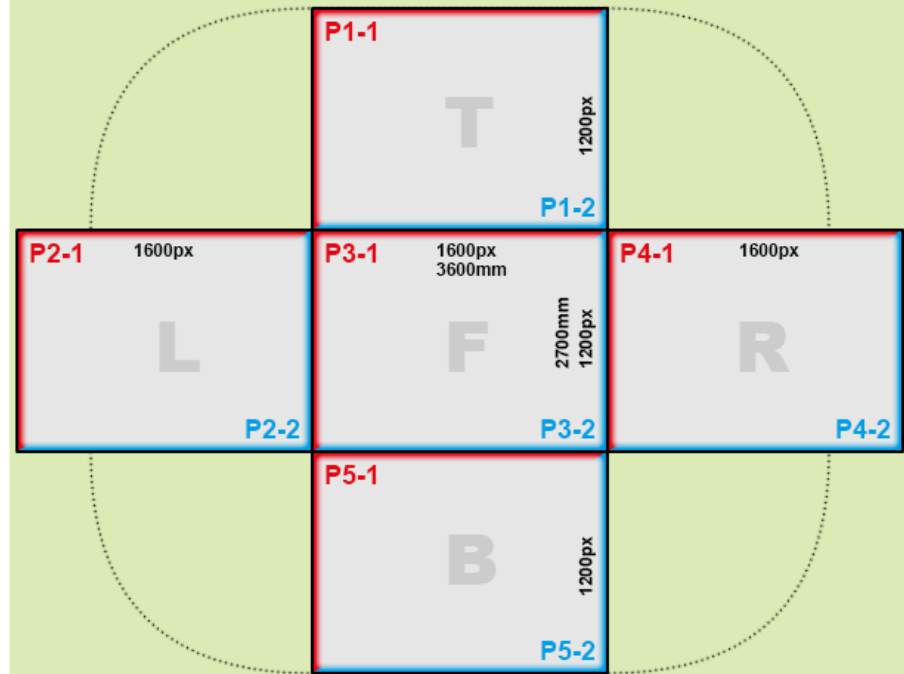
CAVE

(Nvidia quadro 4600)

Aufbau Hardware



Aufbau Display



Aktueller Stand – XML Spezifikation

- Stark beeinflusst von Paraview .PVX-Dateien
- Beschreibung mittels XML-Schema
- Ziele:
 - Gute Abbildung der Umgebung
 - Trotzdem Lesbarkeit erhalten

```
1  <?xml version="1.0" ?>
2  <pvx>
3    <Process Type="client" />
4    <Process Type="render-server">
5      <Machine Name="Links">
6        Environment="DISPLAY=:0"
7        Geometry="1600x1200+0+0"
8        FullScreen="0"
9        ShowBorders="0"
10       LowerLeft="-1.8 -1.35 1.8"
11       LowerRight="-1.8 -1.35 -1.8"
12       UpperRight="-1.8 1.35 -1.8">
13     </Machine>
14     <Machine Name="Vorn">
15       Environment="DISPLAY=:0"
16       Geometry="1600x1200+0+0"
17       FullScreen="0"
18       ShowBorders="0"
19       LowerLeft="-1.8 -1.35 -1.8"
20       LowerRight="1.8 -1.35 -1.8"
21       UpperRight="1.8 1.35 -1.8">
22     </Machine>
23   </Process>
24 </pvx>
```

Aktueller Stand – XML Spezifikation - Aufbau

Grober Aufbau in zwei Teile:

- Oberer Teil Beschreibung der Infrastruktur:
 - Rechner, Nodes → Cluster
 - Netzwerk
- Unterer Teil Beschreibung der Displayumgebung:
 - Display – Rechner - Zuordnung
 - Physikalische & Virtuelle Größe
 - Stereoeigenschaften
 - Verschiedene Setups möglich

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <openDECL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.tu-dresden.de" xsi:
3  <node name="Ronald-PC" address="127.0.0.1" purpose="render">
12 <network id="e1" bandwidth="10 Gbit/s" subnet-mask="255.255.255.0"></network>
13 <display-setup name="flat">
75 </openDECL>
```

Aktueller Stand – XML Spezifikation - Infrastruktur

```
<xs:complexType name="nodetype">
  <xs:sequence>
    <xs:element name="device" type="devicetype" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:ID" use="required" />
  <xs:attribute name="address" type="xs:string" use="required" />
  <xs:attribute name="purpose" type="xs:string" use="optional" />
</xs:complexType>

<xs:complexType name="devicetype">
  <xs:sequence>
    <xs:element name="port" type="porttype" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="type" type="nodeDeviceType" use="required" />
  <xs:attribute name="gpu-count" type="xs:integer" use="optional" />
  <xs:attribute name="vram" type="xs:integer" use="optional" />
  <xs:attribute name="model-name" type="xs:string" use="optional" />
</xs:complexType>

<xs:complexType name="porttype">
  <xs:attribute name="id" type="xs:ID" use="required" />
  <xs:attribute name="type" type="xs:string" use="optional" />
  <xs:attribute name="slot" type="xs:string" use="optional" />
</xs:complexType>

<xs:simpleType name="nodeDeviceType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="graphics-card" />
    <xs:enumeration value="network-card" />
    <xs:enumeration value="other" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="networktype">
  <xs:attribute name="id" type="xs:IDREFS" use="required" />
  <xs:attribute name="bandwidth" type="xs:string" use="optional" />
  <xs:attribute name="subnet-mask" type="xs:string" use="optional" />
</xs:complexType>
```

The diagram illustrates the XML schema definitions for the OpenDECL infrastructure. It shows five main components: **nodetype**, **devicetype**, **porttype**, **nodeDeviceType**, and **networktype**. Arrows indicate dependencies: **nodetype** depends on **devicetype**; **devicetype** depends on **porttype** and **nodeDeviceType**; **porttype** depends on **nodeDeviceType**; and **networktype** depends on **nodeDeviceType**.

Aktueller Stand – XML Spezifikation - Infrastruktur

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <openDECL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.tu-dresden.de" xsi:schemaLocation="http://www.tu-dresden.de openDECL.xsd">
3   <node name="Ronald-PC" address="127.0.0.1" purpose="render">
4     <device type="graphics-card" gpu-count="1" vram="512" model-name="ATi Radeon HD 3650 mobility">
5       <port type="display" id="M1" slot="0"></port>
6       <port type="display" id="M2" slot="HDMI 1"></port>
7     </device>
8     <device type="network-card">
9       <port type="ethernet" id="e1"></port>
10    </device>
11  </node>
12 <network id="e1" bandwidth="10 Gbit/s" subnet-mask="255.255.255.0"></network>
```

Aktueller Stand – XML Spezifikation – Displayumgebung

The diagram illustrates the XML schema definitions for a display environment, showing four complex types and one simple type with their internal structure and cross-references.

```
<xs:complexType name="display-setuptype">
  <xs:sequence>
    <xs:element name="display" type="displaytype" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:ID" use="required" />
  <xs:attribute name="eye-distance" type="xs:decimal" default="0.0" />
</xs:complexType>

<xs:complexType name="displaytype">
  <xs:sequence>
    <xs:element name="physical" type="physicaltype" minOccurs="1" maxOccurs="1" />
    <xs:element name="virtual" type="virtualtype" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:IDREF" use="required" />
  <xs:attribute name="type" type="xs:string" use="optional" />
  <xs:attribute name="stereo" type="stereoType" use="required" />
  <xs:attribute name="pixel-size-x" type="xs:positiveInteger" use="required" />
  <xs:attribute name="pixel-size-y" type="xs:positiveInteger" use="required" />
  <xs:attribute name="metric-size-x" type="xs:decimal" use="required" />
  <xs:attribute name="metric-size-y" type="xs:decimal" use="required" />
</xs:complexType>

<xs:complexType name="physicaltype">
  <xs:sequence>
    <xs:element name="upper-left" type="upper-lefttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="lower-left" type="lower-lefttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="lower-right" type="lower-righttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="upper-right" type="upper-righttype" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

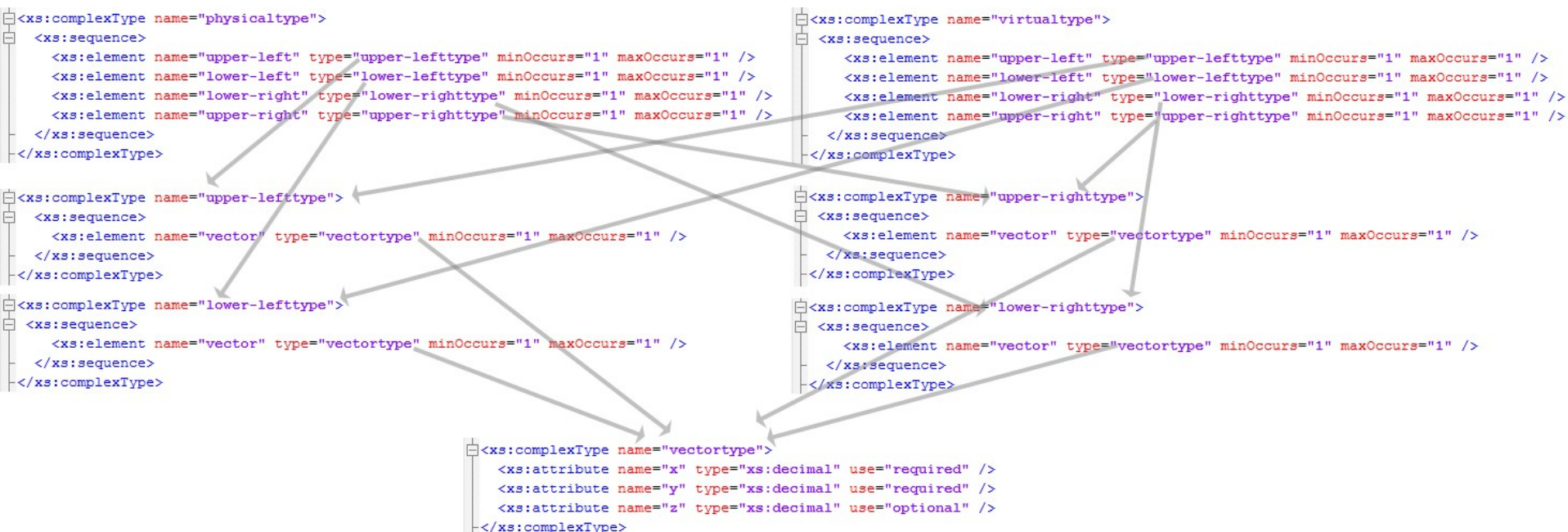
<xs:complexType name="virtualtype">
  <xs:sequence>
    <xs:element name="upper-left" type="upper-lefttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="lower-left" type="lower-lefttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="lower-right" type="lower-righttype" minOccurs="1" maxOccurs="1" />
    <xs:element name="upper-right" type="upper-righttype" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="stereoType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="left-eye" />
    <xs:enumeration value="right-eye" />
  </xs:restriction>
</xs:simpleType>
```

Arrows in the original image indicate the following relationships:

- From `display-setuptype` to `displaytype` (via the `display` element).
- From `displaytype` to `physicaltype` (via the `physical` element).
- From `displaytype` to `virtualtype` (via the `virtual` element).
- From `displaytype` to `stereoType` (via the `stereo` attribute).

Aktueller Stand – XML Spezifikation – Displayumgebung



Aktueller Stand – XML Spezifikation – Displayumgebung

```
<display-setup name="flat">
  <display id="M1" type="monitor" stereo="none" pixel-size-x="1280" pixel-size-y="800" metric-size-x="0.32" metric-size-y="0.2">
    <physical>
      <upper-left>
        <vector x="-0.32" y="0.1" z="-0.5"></vector>
      </upper-left>
      <lower-left>
        <vector x="-0.32" y="-0.1" z="-0.5"></vector>
      </lower-left>
      <lower-right>
        <vector x="0" y="-0.1" z="-0.5"></vector>
      </lower-right>
      <upper-right>
        <vector x="0" y="0.1" z="-0.5"></vector>
      </upper-right>
    </physical>
    <virtual>
      <upper-left>
        <vector x="0" y="0"></vector>
      </upper-left>
      <lower-left>
        <vector x="0" y="800"></vector>
      </lower-left>
      <lower-right>
        <vector x="1280" y="800"></vector>
      </lower-right>
      <upper-right>
        <vector x="1280" y="0"></vector>
      </upper-right>
    </virtual>
  </display>
</display-setup>
```

Aktueller Stand – XML Spezifikation – Displayumgebung

```
<display id="M2" type="monitor" stereo="none" pixel-size-x="1920" pixel-size-y="1080" metric-size-x="0.464" metric-size-y="0.26">
  <physical>
    <upper-left>
      <vector x="0" y="0.13" z="-0.5"/>
    </upper-left>
    <lower-left>
      <vector x="0" y="-0.13" z="-0.5"/>
    </lower-left>
    <lower-right>
      <vector x="0.464" y="-0.13" z="-0.5"/>
    </lower-right>
    <upper-right>
      <vector x="0.464" y="0.13" z="-0.5"/>
    </upper-right>
  </physical>
  <virtual>
    <upper-left>
      <vector x="1280" y="0"/>
    </upper-left>
    <lower-left>
      <vector x="1280" y="1080"/>
    </lower-left>
    <lower-right>
      <vector x="3200" y="1080"/>
    </lower-right>
    <upper-right>
      <vector x="3200" y="0"/>
    </upper-right>
  </virtual>
</display>
</display-setup>
```

Aktueller Stand – XML Spezifikation

```
<xs:complexType name="openDECLtype">
  <xs:sequence>
    <xs:element name="node" type="nodetype" minOccurs="1" maxOccurs="unbounded" />
    <xs:element name="network" type="networktype" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="display-setup" type="display-setuptype" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:element name="openDECL" type="openDECLtype" />
```

Beispiel einer anderen OpenDECL, zum Beispiel die für Stereo Powerwall

Aktueller Stand – XML Spezifikation – Beispiel

Beispiel einer anderen OpenDECL, zum Beispiel die für Stereo Powerwall

Probleme & Weiteres Vorgehen – Probleme

- Hardware – Software Inkompatibilitäten:
 - Megamol ohne Clustersupport
 - Paraview nur Singleviewport → limitiertes Stereo
 - Paraview zwar für CAVE geeignet → trotzdem Problem in CAVE des CAD-Labors
 - Paraview Interaktion

Probleme & Weiteres Vorgehen – Weiteres Vorgehen

- Lösen der Probleme bezügl. CAVE
- Entwicklung eines Editors (mit Namespaceunterstützung)
- Erstellen von entsprechenden XSLTs
- Evaluierung der Ergebnisse

Vielen Dank für die Aufmerksamkeit.

Fragen?