

Institut für Visualisierung und Interaktive Systeme (VIS)

Abteilung Grafisch-Interaktive Systeme

Universität Stuttgart

Universitätsstraße 38

D - 70569 Stuttgart

Fachstudie

Open Display Environment
Configuration Language

(OpenDECL)

Simon Lehmann, Tobias Linn, Daniel Wuttke

Studiengang: Softwaretechnik

Prüfer: Prof. Thomas Ertl

Betreuer: Christoph Müller, Guido Reina

begonnen am: 12.12.2014

beendet am: 12.06.2015

CR-Klassifikation: D.2.9

INHALT

Einleitung	4
Vorgehensweise	5
Begriffserklärung.....	5
Existierende Software	6
MegaMol	6
XML-Format	6
CFG-Format	7
Equalizer.....	7
Leinwand.....	8
Bilderzeugung.....	8
ParaView	9
PVX-Format	9
COVISE.....	11
Pipe-Config	11
Screen-Config	11
Window-Config	13
Channel-Config.....	14
Visit.....	14
IceT	14
Open Display Environment Configuration Language	15
Erläuterung.....	15
Das XSD-Format	15
Aufbau der XSD-Datei	15
Übertragung der Globalen Datei auf die spezifischen Konfigurationsdateien.....	18
Übersicht.....	18
<Wall>	19
<Tile> - Name	19
<Tile> - Comment.....	19
<Tile> - WidthPixel	20
<Tile> - HeightPixel.....	20
<Tile> - Width-mm	21
<Tile> - Height-mm.....	21
<Tile> - <Position> - <LeftBottom>	21
<Tile> - <Position> - <VectorHeight>	22
<Tile> - <Position> - <VectorWidth>	22
<Tile> - <Position> - <Angle>.....	22

<Computer> - Name.....	23
<Adapter> - Name.....	23
<Window> - Name	24
<Window> - Comment	24
<Window> - Height	24
<Window> - Width	25
<Window> - LeftOffset.....	25
<Window> - BottomOffset.....	25
<Window> - Decoration	26
<Output> - Name	26
<Output> - Comment.....	27
<Output> - Height	27
<Output> - Width	28
<Output> - StereoChannel	28
<Output> - LeftOffset.....	28
<Output> - BottomOffset.....	29
Transformationsdateien.....	30
Spezifischer Teil.....	30
Allgemeiner Teil	30
MegaMol Transformation	30
Tracking System und Eingabegeräte.....	34
Stereoskopie	35
Aktives Stereo	35
Passives Stereo.....	35
Fazit.....	36
Referenzen	37
Anhänge.....	38
OpenDECL.xsd	38
OpenDECL.xml.....	41
OpenDECL – MegaMol.xsl	46
MegaMol.xml	48

Das Bedürfnis mehrere Bildschirme für ein Endgerät zu benutzen wird immer relevanter, da die Vorteile gravierend herausstechen, egal in welcher Form die Benutzung aussieht. Sei es das Arbeiten an einem Computer und dadurch, dass mehr als ein Bildschirm benutzt wird, sich die Produktivität des Anwenders erhöht. Liegt es nicht nur an der Möglichkeit einzelne Programme parallel zu öffnen und zu überblicken, sondern auch daran, dass die individuelle Anordnung der Bildschirme gewährleistet ist. So kann jeder seine eigene ideale Ordnung der Monitore definieren.

Um diese Fülle an Möglichkeiten wirklich zu beherrschen gibt es verschiedene Softwareanwendungen. Diese sind dafür da, um das Bild, welches angezeigt werden soll, auf die Projektoren bzw. Monitore richtig zu verteilen. Dazu müssen viele Konfigurationsparameter im Programm festgelegt werden. Nicht nur auf der Softwareseite, welche Auflösung das Bild haben soll oder wie die Bilderzeugung aussieht, sondern auch auf der Hardwareseite, wie die Bildschirme angeordnet sind und welche Maße diese haben. Problematisch ist dabei die Verschiedenheit der Anwendungen. Denn es gibt keine einheitliche Konfiguration, sondern jede Software hat ihre eigene Formatierung der Konfigurationsdateien.

Wir wollen genau dafür eine Vereinfachung finden. Es soll ein Konfigurierungsformat sein, das alle wichtigen Parameter beinhaltet. Dazu haben wir die Konfigurierung der Programme MegaMol, Equalizer, COVISE, ParaView und VisIt analysiert. Das neue Format ist XML basiert und soll durch Programmspezifische XSL-Transformationen zur gewünschten Konfigurierungsdatei umgewandelt werden. Da es nicht bei den bisherigen Programmen bleiben soll, steht natürlich die Erweiterbarkeit auch im Mittelpunkt bei der Erstellung des neuen Konfigurierungsformats.

VORGEHENSWEISE

Um das Ziel zu erreichen, eine große Anzahl an Programmen zu unterstützen, war es nötig sich in die Dokumentation der einzelnen Programme einzulesen und diese zum Teil auch zu installieren und sich in die Einstellungen, die sich mit den anzuzeigenden Elementen befassen, einzuarbeiten. Hierzu wurden die Programme MegaMol, Equalizer, ParaView, COVISE und VisIt begutachtet. Zusätzlich wurde sich in die Dokumentation der Image Composition Engine for Tiles (IceT) eingearbeitet, da diese von einigen Programmen zur graphischen Darstellung verwendet wird. Da nicht bei allen Programmen eine erklärende Dokumentation bzw. eine Dokumentation vorhanden war gestaltete sich dieser Teil relativ schwierig. Aufgrund der Tatsache, dass die Programme einzelne wichtige Dinge wie z.B. Grafikkarte, Display, Leinwand, etc. unterschiedlich benennen, war der Aufbau der einzelnen Konfigurationsdateien nicht immer offensichtlich. Dazu war bei anderen Programmen, die z.B. IceT zur Darstellung verwenden nicht klar, dass diese IceT verwenden, da dies nicht aus der Dokumentation hervor ging.

Nachdem sich in die Programme eingearbeitet wurde, konnten dann relativ schnell die relevanten Werte extrahiert werden und sich Überlegungen zu einer alternativen universellen Konfigurationsdatei gemacht werden. Die Gestaltung eben jener universellen Konfigurationsdatei war einfach und war mit nur wenigen Iterationsschritten möglich.

BEGRIFFSERKLÄRUNG

In dem Folgenden Dokument werden Dateien beschrieben, die in einer bestimmten Art formatiert wurden. Da bei der Erklärung des Öfteren Bezug auf diese Dateien genommen werden und um Verwirrung zu vermeiden, werden Elemente aus den Dateien mit einem „<“ und einem „>“ umschlossen. Das zeigt an, dass im Moment genau von diesem Element gesprochen wird.

EXISTIERENDE SOFTWARE

Es wurden verschiedene Tools evaluiert um ein einheitliches Format zu erstellen. Im Folgenden werden diese Tools beschrieben, um einen Überblick der Funktionen zu schaffen.

MEGAMOL

MegaMol ist vom Visualisierungsinstitut der Universität Stuttgart entwickelt worden, um die eigene Powerwall zu betreiben. Die Konfiguration kann mit unterschiedlichen Formaten festgelegt werden. Zum einen in einem XML-Format und zum anderen in einem CFG-Format.

XML-FORMAT

```
<Machines>
    ...
    <Machine>
        <Identity>Machine01</Identity>

        <Tiles>
            <Tile>
                <Name>(0, 0), [2400, 4096] auf Machine01</Name>
                <WindowHeight>4096</WindowHeight>
                <WindowWidth>2400</WindowWidth>
                <LeftOffset>8400</LeftOffset>
                <TopOffset>0</TopOffset>
                <StereoChannel>Right</StereoChannel>
            </Tile>
        </Tiles>
    </Machine>
</Machines>

<Name>Vwand</Name>
<Width>10800</Width>
<Height>4096</Height>
```

LEINWAND

Die Leinwand hat drei Definitionen. Der Name, die Breite und die Höhe. Die beiden letzteren werden in Pixel angegeben.

BILDERZEUGUNG

Die Anzeigemedien werden innerhalb der <Machines> festgehalten. Es kann mehrere Rechner geben, welche in der XML-Datei <Machine> genannt werden. Jeder Rechner hat eine bestimmte Identität und kann eine oder mehrere Ausgänge haben welche als <Tile> beschrieben werden. Diese Ausgänge tragen die kompletten Informationen bezüglich des Anzeigens. Jeder Ausgang hat einen Namen, dazu eine Höhe, Breite, linken Abstand und oberen Abstand, damit das Bild auch an der gewünschten Position richtig angezeigt werden kann. Diese vier Angaben sind, wie bei der Leinwand, alle in Pixel anzugeben. Als letztes Attribut hat jeder Ausgang einen <StereoChannel> dieser besagt ob das Bild für das rechte oder das linke Auge angezeigt werden soll. Hiermit können nur flache Leinwände konfiguriert werden.

CFG-FORMAT

LEINWAND

Genau wie bei dem XML-Format hat hier die Leinwand drei Attribute. Einen Namen, die Höhe und die Breite. Die Parameter Höhe und Breite sind in Pixel anzugeben.

```
...  
<if cond="(computer==Machine01) || (computer==Machine02)">  
  <set name="tvview" value="10800;4096" />  
  
  <if cond="computer==Machine01">  
    <set name="tveye" value="left" />  
  </if>  
  
  <if cond="computer==Machine02">  
    <set name="tveye" value="right" />  
  </if>  
  
  <set name="tvproj" value="stereo" />  
  
  <set name="w3-tvtile" value="0;0;1200;2048" />  
  <set name="w4-tvtile" value="1200;0;1200;2048" />  
  <set name="w1-tvtile" value="0;2048;1200;2048" />  
  <set name="w2-tvtile" value="1200;2048;1200;2048" />  
</if>  
  
...
```

BILDERZEUGUNG

Durch mehrere „If“-Konditionen werden klar gestellt, welcher Zielcomputer konfiguriert werden soll. Es gibt für jeden Ausgang einen Namen, in **Error! Reference source not found.** z.B. „w3-tvtile“, und vier Werte, die als einen zusammengefasst sind. Von links nach rechts sind die vier Werte der X-Koordinate, die Y-Koordinate, die Breite und die Höhe. Im Vergleich zu dem XML-Format ist die X-Koordinate der Linke Abstand und die Y-Koordinate der obere Abstand. Alle vier Angaben sind in Pixel angegeben. Dazu kommt noch der Parameter für das rechte oder das linke Auge, das wird hier direkt in eine „If“-Abfrage gepackt somit für zwei verschiedene Rechner sichergestellt. Der Name zu diesem Parameter ist <tveye>.

EQUALIZER

Equalizer ist von „Eyescale Software GmbH“ entwickelt worden. Es ist von den fünf Programmen das umfangreichste, da ist für sehr viele unterschiedliche Einsatzmöglichkeiten erstellt worden ist. Dadurch können im Laufe dieser Fachstudie nicht alle Möglichkeiten von Equalizer ausgeschöpft werden, aber die wichtigsten zur Bilderzeugung, welche auch mit den anderen Programmen Überschneidungen hat, können verarbeitet werden.

LEINWAND

```
Canvas
{
  Wall
  {
    Bottom_left [ -.5 -.5 -1]
    Bottom_right [ .5 -.5 -1]
    top_left [ -.5 .5 -1]
  }

  Segment
  {
    Viewport [0 .5 .5 .5 ]
    Channel "channel1"
  }

  Segment
  {
    Viewport [.5 .5 .5 .5 ]
    Channel "channel2"
  }
}
```

Die Konfiguration der Leinwand ist bei dem Programm Equalizer in zwei Teile aufgeteilt. Zum einen werden bei <wall> die Eckpunkte festgelegt, welche zur gesamten Leinwand gehören. Dazu werden nur drei Ecken definiert, da die vierte daraus resultierend ist. Die Werte der Ecken sind zwischen null und eins relativ angegeben. Größere Zahlen sind Absolut Werte, die in Pixel angegeben werden. Zum anderen werden die einzelnen <channel> an die jeweiligen <segment> angeheftet. Dazu haben diese <segment> ihren eigenen <viewport> der folgende Werte hat, X-Koordinate, Y-Koordinate, Breite und Höhe. Genau wie den Eckpunkten sind die Werte relativ, wenn sie kleiner eins sind und absolut, wenn sie größer als eins sind. Die X- und Y-Koordinate können auch als linker und rechter Abstand zum Rand betrachtet werden. Sind die Parameter in Absolut Werten angegeben, sind es Pixel.

BILDERZEUGUNG

```
Config
{
  appNode
  {
    Pipe
    {
      Name "GPU 1"
      Window
      {
        Name "window1"
        Viewport [ 25 25 400 400 ]
        Channel { name "channel1" }
      }
    }
  }
}
```

```

Pipe
{
    Name "GPU 2"
    Window
    {
        Name "window2"
        Viewport [ 450 25 400 400 ]
        Channel { name "channel2" }
    }
}
}

```

Bei der Konfiguration von Equalizer ist der strukturelle Aufbau so gehalten, dass es vier Ebenen gibt. Es beginnt mit den Rechnern, die in der Datei als <appNode> bzw. <node> bezeichnet werden. Darin liegen die Grafikkarten, sie werden <pipe> genannt. Ein Rechner kann mehrere Grafikkarten beinhalten. Eine Ebene tiefer ist der Kontext der Grafikkarte, bei Equalizer wird diese als <window> bezeichnet. Dieses Fenster kann mehrere Ausgänge haben, hier <channel>. Auf diesem Channel wird dann die Ansicht ausgegeben. Jede Ebene braucht einen Namen, damit die <segmente> mit den <channel> verknüpft werden können.

PARAVIEW

ParaView ist eine Open-Source Anwendung, welche extrem große Datensätze verarbeiten und darstellen kann. Die Größe der zu verarbeitenden Datenmengen hängt ihr mehr oder weniger nur von der Leistungsfähigkeit der Hardware ab. So lassen sich laut Website bis zu Petabyte große Datensätze verarbeiten und darstellen.

Um diese extrem großen Datensätze darstellen zu können, bietet ParaView unter anderem auch eine Konfigurationsmöglichkeit für zusammengesetzte Bildschirme bzw. Powerwalls an.

PVX-FORMAT

ParaView verwendet als Konfigurationsdatei einen PVX-Dateityp. Dieser Dateityp ist basiert auf XML und bietet dem Anwender die Möglichkeit eine Displaykonfiguration zu erstellen, welche dann die ausgewerteten Daten anzeigt.

Displaykonfigurationen werden im Server festgelegt.

```

<?xml version="1.0" ?>
<pvx>
  <Process Type="client" />

  <Process Type="server">
    <EyeSeparation Value="0.065" />

    <Machine
      Name="Machine01"
      Geometry="1200x2048+0+0"
      FullScreen="0"
      ShowBorders="0"
      LowerLeft="-1.000000 0.000000 -1.000000"
      LowerRight="-0.777778 0.000000 -1.000000"
      UpperRight="-0.777778 1.000000 -1.000000"
    />

```

```

<Machine
  Name="Machine02"
  Geometry="1200x2048+1200+0"
  FullScreen="0"
  ShowBorders="0"
  LowerLeft="-0.777778 0.000000 -1.000000"
  LowerRight="-0.555556 0.000000 -1.000000"
  UpperRight="-0.555556 1.000000 -1.000000"
/>

<Machine
  Name="Machine03"
  Geometry="1200x2048+0+2048"
  FullScreen="0"
  ShowBorders="0"
  LowerLeft="-1.000000 -1.000000 -1.000000"
  LowerRight="-0.777778 -1.000000 -1.000000"
  UpperRight="-0.777778 0.000000 -1.000000"
/>

<Machine
  Name="Machine04"
  Geometry="1200x2048+1200+2048"
  FullScreen="0"
  ShowBorders="0"
  LowerLeft="-0.777778 -1.000000 -1.000000"
  LowerRight="-0.555556 -1.000000 -1.000000"
  UpperRight="-0.555556 0.000000 -1.000000"
/>
</Process>
</pvx>

```

LEINWAND

Die Definition der Leinwand hat drei bzw. vier Parameter. *LowerLeft*, *LowerRight* und *UpperRight*. Es ist anzunehmen, dass der *UpperLeft* Parameter bei Angabe von nur drei Parametern berechnet wird. Die Werte werden in Prozent angegeben, abhängig von der Position der Mitte der Leinwand.

BILDERZEUGUNG

Die Bilderzeugung wird im Attribut *Geometry* festgehalten. Dieser Wert setzt sich aus vier Teilen zusammen, welcher im ersten Fall mit einem „x“ und in den beiden weiteren Fällen durch ein „+“ getrennt wird.

Die vier Einzelwerte sind:

- Höhe des Teildisplays
- Breite des Teildisplays
- Horizontales Offset
- Vertikales Offset

Es ist zu vermuten, dass die Offsets durch ein „-“ als Trennzeichen auch negativ sein können.

COVISE

COVISE bzw. Collaborative Visualization and Simulation Environment ist eine erweiterbare verteilte Softwareumgebung zur nahtlosen Integration von Simulations-, Nacharbeitungs- und Visualisierungsfunktionalitäten welches vom High Performance Computing Center Stuttgart (HLRS) entwickelt wurde.

COVISE benutzt OpenCover zur Darstellung der Oberfläche. Im Folgenden wird näher auf die XML- und CONFIG-Datei eingegangen, welche die Einstellungen speichert. Dabei gibt es vier relevante Abschnitte (*PipeConfig*, *ScreenConfig*, *WindowConfig* und *ChannelConfig*) zur Konfiguration der Darstellung.

<Input> enthält die Konfiguration für alle Eingabegeräte. Hier wird mit `<TrackingSystem value="[System]"/>` festgelegt, welche Eingabegeräte unterstützt werden.

PIPE-CONFIG

Die *PipeConfig* dient der Konfiguration der Grafikkarten. Dabei werden die Attribute *name*, *screen* und *server* verwendet. *Name* und *screen* haben den gleichen Namen und werden von 0 aufsteigend durchnummeriert. In *Server* wird das dazugehörige *display* bzw. der dazugehörige *screen* gespeichert. Bevor die einzelnen Grafikkarten spezifiziert werden, muss die Anzahl der verwendeten Grafikkarten festgelegt werden. Dies geschieht im folgenden Code:

```
COVERConfig
{
    ...
    NUM_PIPES <number of graphics boards used for COVER>
    ...
}
```

Anschließend können die Grafikkarten in der *covise.config* genauer spezifiziert werden:

```
PipeConfig
{
    <index> <screen> <:server.display>
    <index> <screen> <:server.display>
    ...
}
```

SCREEN-CONFIG

Nach spezifizieren der Grafikkarten kann der Bildschirm bzw. die Projektionsfläche festgelegt werden. Zuvor muss auch hier wieder die Anzahl der physikalischen Bildschirme bzw. Wände festgelegt werden:

```
COVERConfig
{
    ...

    NUM_SCREEN 

    ...
}
```

In *ScreenConfig* werden anschließend die Größe und die Orientierung im dreidimensionalen Raum definiert. Dabei enthält ein Screen die Attribute *name*, *screen*, *width*, *height*, *originX*, *originY*, *originZ*, *h*, *p*, *r* und *comment*. *Name* und *screen* haben den gleichen Namen und werden von 0 aufsteigend durchnummeriert. Die Größe des Bildschirms bzw. der Projektionsfläche werden in Millimeter mit *width* und *height* festgelegt. Der Screen an sich wird mit den drei Origin-Attributen im dreidimensionalen Raum positioniert und kann mithilfe der Werte *h*, *p* und *r* im Raum gedreht werden. Bei diesen drei Werten handelt es sich um eulersche Winkel. Die COVISE Konfiguration dazu sieht wie folgt aus:

```
ScreenConfig
{
    <index> <name> <width [mm]> <height[mm]> <origin [mm]>
    <euler angles [degree]>

    <index> <name> <width [mm]> <height[mm]> <origin [mm]>
    <euler angles [degree]>

    ...
}
```

Folgend ist eine beispielhafte XML-Datei, die entsprechende Covise.config illustriert:

```
<ScreenConfig>
  <Screen
    name="0"
    screen="0"
    width="420"
    height="310"
    originX="0"
    originY="0"
    originZ="0"
    h="0.0"
    p="0.0"
    r="0.0"
    comment="Bildschirm"
  />
</ScreenConfig>
```

WINDOW-CONFIG

<WindowConfig> konfiguriert das Ausgabefenster. Auch hier muss vorher festgelegt werden, wie viele Fenster vorhanden sind:

```
COVERConfig
{
    ...
    NUM_WINDOWS <number of windows>
    ...
}
```

Ein *Window* enthält die Attribute *name*, *window*, *width*, *height*, *left*, *bottom*, *comment* und *decoration*. Hierbei müssen *name* und *window* den gleichen Wert besitzen und werden von 0 aufsteigend durchnummeriert. Mit *width* und *height* wird die Fenstergröße absolut in Pixel angegeben und kann mit *left* und *bottom* auf dem Display positioniert werden. Auch hier erfolgt die Angabe wieder absolut in Pixeln. Das Attribut *decoration* gibt an, ob das Fenster mit oder ohne Fensterdekoration erzeugt werden soll. *comment* bietet die Möglichkeit Anmerkungen zu speichern, der Wert kann frei gewählt werden. Es können mehrere Fenster gleichzeitig in der WindowConfig angelegt werden. Die Covise.config dazu sieht wie folgt aus:

```
WindowConfig
{
    <index> <name> <pipe index> <origin [pixels]> <size [pixels]>
    <index> <name> <pipe index> <origin [pixels]> <size [pixels]>
    ...
}
```

Folgend ist eine beispielhafte XML-Datei, die entsprechende Covise.config illustriert:

```
<WindowConfig>
  <Window
    name="0"
    window="0"
    width="1600"
    height="1200"
    left="0"
    bottom="0"
    comment="Desktop-Konfiguration"
    decoration="false"
  />
</WindowConfig>
```

CHANNEL-CONFIG

ChannelConfig konfiguriert einen Bildausschnitt (Channel) in einem vorher festgelegten Ausgabefenster. Ein *Channel* enthält die Attribute *name*, *channel*, *width*, *height*, *left*, *bottom*, *windowIndex*, *stereoMode* und *comment*. Hier gibt es die Möglichkeit anstelle von *width* und *height* auch die Attribute *right* und *top* zu verwenden. *Name* und *channel* müssen wieder gleich heißen und werden von 0 aufsteigend durchnummeriert. *Width* und *height* legen die Größe des Bildausschnittes fest und können entweder relativ (0-1) zur Ausgabefenstergröße oder absolut (1-Ausgabefenstergröße) in Pixeln angegeben werden. Verwendet man stattdessen *right* und *top*, kann in Kombination mit *left* und *bottom* der Bildausschnitt als gesamtes fest definiert werden. Mit *left* und *bottom* kann der entsprechende Bildausschnitt in dem Ausgabefenster positioniert werden. Die Angaben von *left*, *bottom*, *right*, *top* sind absolut in Pixeln. Der *windowIndex* gibt an, zu welchem Ausgabefenster der Channel gehört. Mithilfe von *stereoChannel* kann angegeben werden, ob die Ausgabe stereoskopisch oder monoskopisch erfolgt. *comment* bietet wieder die Möglichkeit Anmerkungen zu speichern. Es können mehrere Channels gleichzeitig in der *ChannelConfig* angelegt werden.

```
<ChannelConfig>
  <Channel
    name="0"
    channel="0"
    width="1"
    height="1"
    left="0"
    bottom="0"
    windowIndex="0"
    stereoMode="MONO"
    comment="Main Channel"
  />
</ChannelConfig>
```

VISIT

Visit ist eine vom Energieministerium der Vereinigten Staaten (AOE) und der Advanced Simulation and Computing Program (ASC) entwickelte Software zur Visualisierung und Analyse teraskalarer Simulationen. Sie steht per Open Source Lizenz zum Download zur Verfügung und läuft auf allen gängigen Desktop bzw. Server Betriebssystemen.

Visit verwendet zur Darstellung die der Image Composition Engine for Tiles (IceT) sowie, laut Dokumentation, die C++ Bibliothek Qt zur Darstellung der Benutzeroberfläche.

ICET

Die Image Composition Engine for Tiles ist eine hochleistungs-rendering Bibliothek für gekachelte Displays. Die Bibliothek steht als Open Source Lizenz zum Download bereit und wurde in den Sandia National Laboratories entwickelt.

OPEN DISPLAY ENVIRONMENT CONFIGURATION LANGUAGE

ERLÄUTERUNG

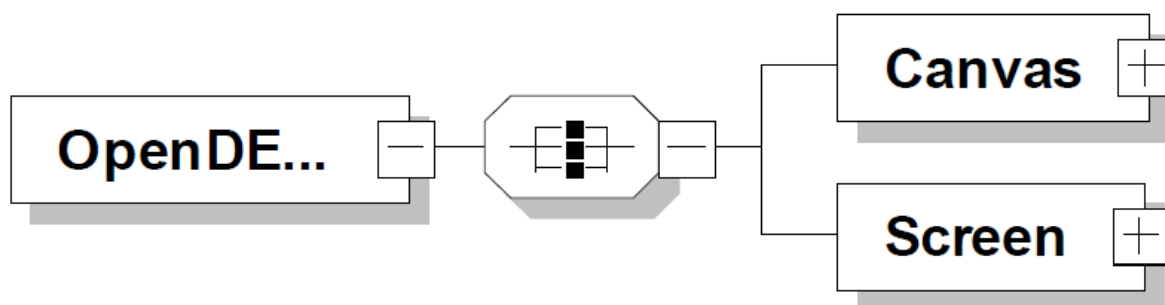
Aus den oben genannten Programmen, wird nun eine zusammengefasste XML-basierte Konfigurationsdatei, die alle grundlegenden Parameter beinhaltet erstellt. Damit kann aus einer XML-Datei die jeweilig benötigte Konfigurationsdatei eines Programms erzeugt werden. Mit Hilfe von XSLT wird aus einer globalen Konfiguration eine spezifisch, an ein bestimmtes Programm angepasste, Datei.

DAS XSD-FORMAT

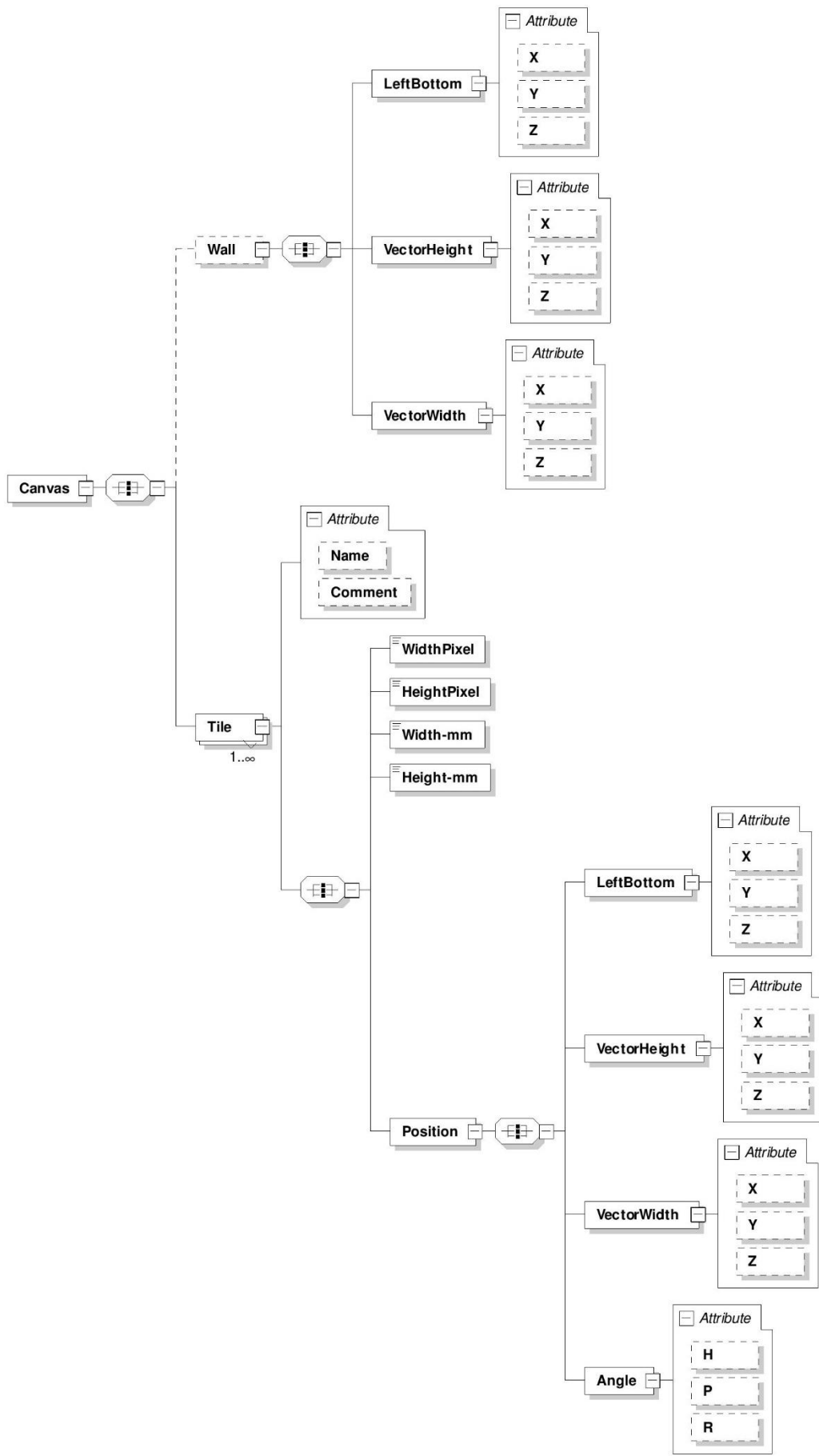
XSD heißt, XML Schema Definition. Es ist dafür ausgelegt eine Struktur einer XML-Datei festzulegen, ohne direkt Werte miteinzubinden. Somit ist sichergestellt, dass das Format eingehalten wird.

AUFBAU DER XSD-DATEI

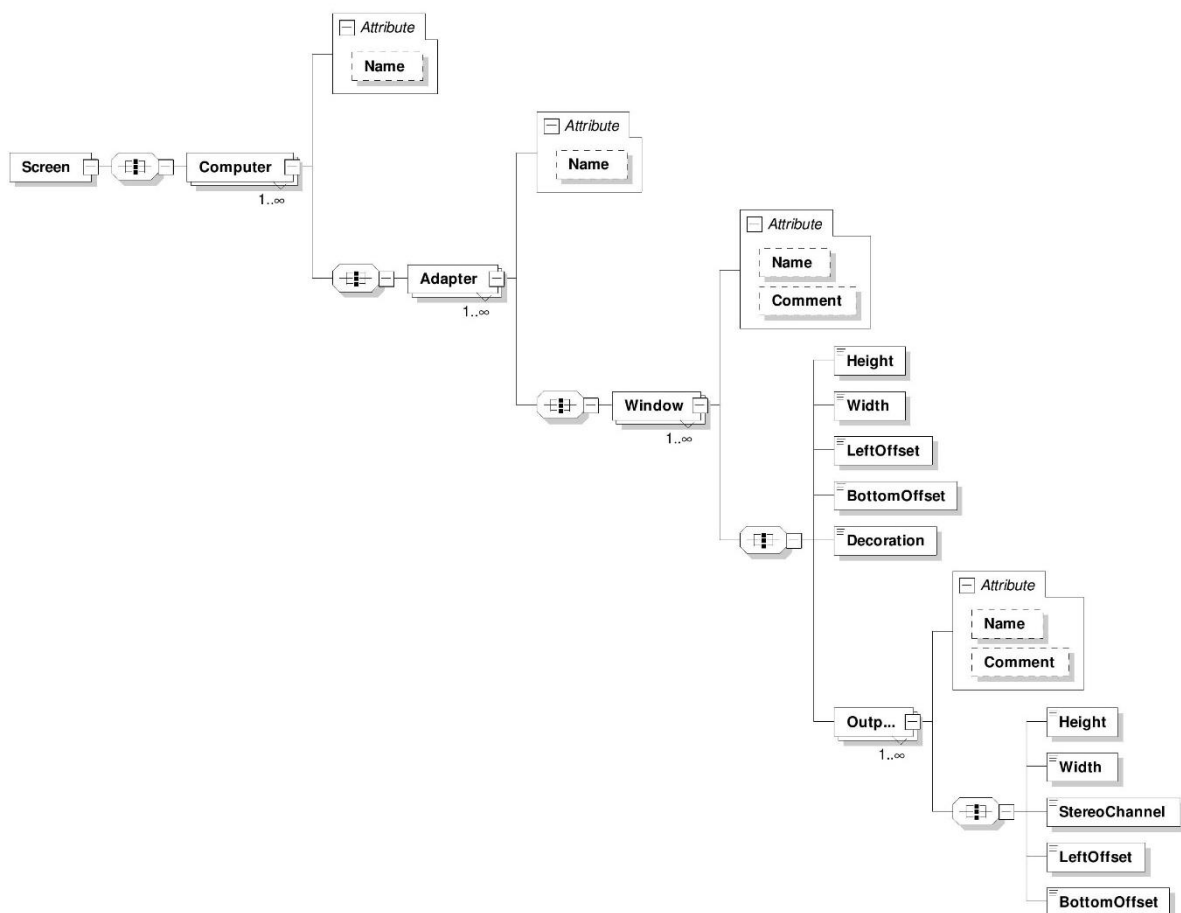
In der XSD-Datei ist die Struktur hierarchisch. Es beginnt mit `<OpenDECL>`, darin gibt es die Konfigurationen der Leinwand welche `<Canvas>` genannt wird und die Konfiguration der Hardware, also Computer, Grafikkarte und die Ausgabe welche auf die Leinwand projiziert wird. Diese wird `<Screen>` genannt.



In `<Canvas>` gibt es die `<Wall>`, diese beinhaltet die Positionskoordinaten der linken unteren Ecke der Leinwand und zwei Vektoren, wobei der eine horizontal zur rechten unteren Ecke und der andere vertikal zur linken oberen Ecke verläuft. Es kann eine oder keine `<Wall>` geben. Der zweite Teil der Leinwandkonfigurationen sind die `<Tile>`. Davon muss es mindestens eine geben aber es gibt keinen oberen Grenzwert. Hier wird konkret festgehalten, wie groß die Leinwand ist. Je nach Zielprogramm für die XSL-Transformation müssen nur die benötigten Parameter ausgefüllt werden. Dadurch, dass die Überschneidungen der Programme bei diesem Teil so gering sind, gibt es für die Größenparametrierung verschiedene Möglichkeiten. Es gibt die Höhen- und Breitenangabe in Pixel, `<WidthPixel>` und `<HeightPixel>`, sowie in Millimeter, `<Width-mm>` und `<Height-mm>`. Genauso können die Positionierungsparameter ausgefüllt werden. Dazu gibt es in `<Position>` die Koordinaten der linken unteren Ecke und dazu einen vertikalen und einen horizontalen Vektor. Daraus lassen sich alle Eckpunkte und den Mittelpunkt eines `<Tile>` definieren. Mit den Winkeln `<H>`, `<P>` und `<R>` in `<Angle>` kann die Leinwand rotiert werden. Als letztes hat `<Tile>` zwei Parameter unabhängig von der Position so wie der Größe. Das sind `<Name>`, darin wird der Name des `<Tile>` festgehalten und `<Comment>`. Dort kann ein kurzer Kommentar oder eine Anmerkung bezüglich des `<Tile>` vermerkt werden.



Die Hardwareparametrierung, welche <Screen> genannt wird, ist hierarchisch in vier Ebenen unterteilt. Das liegt daran, dass in jedem Element einer Ebene mehrere Kind-Elemente der unteren Ebene enthalten sein können. Diese Unterelemente gehören dann auch immer zum übergeordneten Element. Die oberste Ebene sind Einzelsysteme und werden hier als <Computer> definiert. Das einzige Attribut was diese bis jetzt haben ist der Name. Darin liegen die Grafikkarten, genannt <Adapter>, welche auch nur einen Namen als Attribut haben. Eine Grafikkarte kann mehrere <Window> bereithalten. Das ist die dritte Ebene. Nicht jede dieser drei Ebene werden von allen Programmen benötigt. Die Window-Ebene ist nicht direkt eine Hardware-Ebene sondern mehr der Kontext für die darin liegenden Elemente. Ein <Window> hat deswegen auch mehr Attribute als ein <Computer> oder ein <Adapter>. Es gibt Größen- und Positionierungsparameter wie die Höhe und die Breite in Pixel. Dazu gibt es einen linken Abstand zum Rand und einen unteren Abstand zum Rand welche die Positionen angeben. Sonstige Angaben sind der Name, ein Kommentar und eine Dekoration welche den Rahmen des <Window> beeinflusst. Der <Output> ist das, was alle Programme brauchen, denn damit wird die Ausgabe des Bildes festgelegt. Dazu gibt es, wie bei den Elementen <Window>, die Größen welche mit <Height> und <Width> festgelegt wird. Die Positionierung, welche durch <LeftOffset> und <BottomOffset>, also durch den linken Abstand und den unteren Abstand zum Rand definiert wird. Dazu gibt es für jeden Ausgang bzw. <Output> einen Namen, sowie einen Kommentar. Zu aller Letzt gibt es den <StereoChannel>, damit wird festgelegt, für welches Auge des Betrachters das Bild projiziert wird. Dafür gibt es drei Möglichkeiten, das rechte Auge, das linke Auge und Mono, das heißt nur für ein Auge.



ÜBERTRAGUNG DER GLOBALEN DATEI AUF DIE SPEZIFISCHEN KONFIGURATIONSDATEIEN

ÜBERSICHT

Ein „X“ in der Tabelle bedeutet, dass das Attribut bzw. das Element für genau diese Software benötigt wird. Ein „-“ dagegen heißt, dass es für das Programm nicht benötigt wird.

GLOBAL	SPEZIFISCH			
OpenDECL	MegaMol	Equalizer	ParaView	Covise
<Wall>	-	X	-	-
<Tile> - Name	X	X	-	X
<Tile> - Comment	-	-	-	X
<Tile> - WidthPixel	X	X	-	-
<Tile> - HeightPixel	X	X	-	-
<Tile> - Width-mm	-	-	-	X
<Tile> - Height-mm	-	-	-	X
<Position> - <LeftBottom>	-	X	X	X
<Position> - <VectorHeight>	-	-	X	X
<Position> - <VectorWidth>	-	-	X	X
<Position> - <Angle>	-	-	-	X
<Computer> - Name	X	X	-	-
<Adapter> - Name	-	X	-	-
<Window> - Name	-	X	-	X
<Window> - Comment	-	-	-	X
<Window> - Height	-	X	-	X
<Window> - Width	-	X	-	X
<Window> - LeftOffset	-	X	-	X
<Window> - BottomOffset	-	X	-	X
<Window> - Decoration	-	-	-	X
<Output> - Name	X	X	X	X
<Output> - Comment	-	-	-	X
<Output> - Height	X	-	X	X
<Output> - Width	X	-	X	X
<Output> - StereoChannel	X	-	X	X
<Output> - LeftOffset	X	-	X	X
<Output> - BottomOffset	X	-	X	X

<WALL>

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Equalizer hat bei der Leinwand-Konfiguration einen Abschnitt bei dem die Position der rechten und linken unteren Ecke und der linken oberen Ecke, der Wand, definiert sind. Diese Parameter sind eins zu eins übertragbar von OpenDECL zu Equalizer.

PARAVIEW

Wird nicht benötigt.

COVISE

Wird nicht benötigt.

<TILE> - NAME

MEGAMOL

In MegaMol gibt es nur eine Angabe der Leinwand. Es gibt also nur ein <Tile> und deren Namen ist bei MegaMol unter <TiledDisplay> der <Name>. Dieser Fall ist aber nur für die XML-Datei von MegaMol zu betrachten. Bei der CFG-Datei wird der Name nicht notwendigerweise benötigt.

EQUALIZER

Bei der Konfiguration von Equalizer gibt es bei der Parametrierung das <segment>. Das ist das <Tile> aus OpenDECL. In deinem <segment> gibt es aber kein direkten Namen, sondern ein Attribute, dass „Channel“ heißt. Dieses Attribut wird benutzt um die Verbindung, wenn man das aus Sicht von OpenDECL betrachtet, von einem <Tile> zu einem <Output> zu schaffen. Da der <Output> bei Equalizer <Channel> heißt, wird der Parameter in <segment> genauso genannt. Also muss hier im Namen von <Tile> der Name des <Channel> stehen.

PARAVIEW

Wird nicht benötigt.

COVISE

In Covise muss der Name von 0 an aufsteigend durchnummeriert bzw. benannt werden. Daraus werden bei der Transformation die beiden Attribute <name> und <screen> initialisiert welche sich in der <ScreenConfig> befinden. Diese müssen auf jeden Fall gleich benannt werden.

<TILE> - COMMENT

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise bietet hier die Möglichkeit eine Beschreibung zur Konfiguration des Bildschirms bzw. der Projektionsfläche hinzuzufügen. Dabei gibt es keinerlei Vorgaben wie diese aussehen muss.

<TILE> - WIDTHPIXEL

MEGAMOL

Da es bei MegaMol nur eine Parametrierung der Größe bedarf, muss es nur ein <Tile> in der OpenDECL geben. Dieses <Tile> hat eine Breite in Pixel. Diese Breite wird in MegaMol unter <TiledDisplay> in <Width> angegeben. Bei der CFG-Datei ist der erste Wert bei <tvview> die Angabe zur Breite der Leinwand.

EQUALIZER

Nach der Transformation wird aus <Tile> in OpenDECL das <segment> in Equalizer. Darin gibt es ein Parameter namens <viewport>. Dieser enthält vier Werte und davon ist der dritte Wert die Breite des <segment>. Diese Angabe muss durch die Transformation mit der Angabe aus der Breite des <Tile> beschrieben werden.

PARAVIEW

Wird nicht benötigt.

COVISE

Wird nicht benötigt.

<TILE> - HEIGHTPIXEL

MEGAMOL

Bei MegaMol gibt es nur eine Fließe, also ein <Tile>, welches die gesamte Leinwand in der Größe Definiert. Deswegen ist die Höhe in <Tile> die Höhe, in der XML-Datei, des <TiledDisplay>. Dagegen ist in der CFG-Konfigurationsdatei von MegaMol die zweite Zahl, also die Zahl nach dem Semikolon unter <tvview> die Angabe zur Höhe der Leinwand.

EQUALIZER

Genau wie bei der Breite des <Tile> ist die Höhe, die Höhe des <segment> bei Equalizer. Dieser Parameter ist der vierte Wert des <viewport> des <segment>.

PARAVIEW

Wird nicht benötigt.

COVISE

Wird nicht benötigt.

<TILE> - WIDTH-MM

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise benötigt zur Definition der Breite des Bildschirms bzw. der Projektionsfläche eine Angabe in Millimetern. In der Covise-spezifischen Konfigurationsdatei selber wird dies in dem Attribut <width> gespeichert.

<TILE> - HEIGHT-MM

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise benötigt zur Definition der Höhe des Bildschirms bzw. der Projektionsfläche eine Angabe in Millimetern. In der Covise-spezifischen Konfigurationsdatei selber wird dies in dem Attribut <height> gespeichert.

<TILE> - <POSITION> - <LEFTBOTTOM>

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Das <segment> des Equalizers hat innerhalb des <viewport> eine X- und Y-Koordinate. Beide können aus den X- und Y-Positionswerten aus <LeftBottom> übertragen werden.

PARAVIEW

Die Position der physikalischen Wand im OpenDECL wird in ParaView zu einer <Machine> und deren *LowerLeft*, *LowerRight* und *UpperRight* Attributen. Da die Position im OpenDECL aus *LeftBottom*, *VectorHeight*, *VectorWidth* und *Angle* besteht, wird die Transformation an dieser Stelle ein wenig aufwendiger. Hier müssen die korrekten Werte berechnet werden.

COVISE

In Covise wird die physikalische Wand mit den Werten <originX>, <originY> und <originZ>, sowie den Winkeln <h>, <p> und <r> positioniert. Entsprechende Werte müssen aus den OpenDECL Attributen LeftBottom, VectorHeight, VectorWidth und Angle bei der Transformation berechnet werden.

<TILE> - <POSITION> - <VECTORHEIGHT>

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Die Position der physikalischen Wand im OpenDECL wird in ParaView zu einer <Machine> und deren *LowerLeft*, *LowerRight* und *UpperRight* Attributen. Da die Position im OpenDECL aus *LeftBottom*, *VectorHeight*, *VectorWidth* und *Angle* besteht, wird die Transformation an dieser Stelle ein wenig aufwendiger. Hier müssen die korrekten Werte berechnet werden.

COVISE

In Covise wird die physikalische Wand mit den Werten <originX>, <originY> und <originZ>, sowie den Winkeln <h>, <p> und <r> positioniert. Entsprechende Werte müssen aus den OpenDECL Attributen LeftBottom, VectorHeight, VectorWidth und Angle bei der Transformation berechnet werden.

<TILE> - <POSITION> - <VECTORWIDTH>

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Die Position der physikalischen Wand im OpenDECL wird in ParaView zu einer <Machine> und deren *LowerLeft*, *LowerRight* und *UpperRight* Attributen. Da die Position im OpenDECL aus *LeftBottom*, *VectorHeight*, *VectorWidth* und *Angle* besteht, wird die Transformation an dieser Stelle ein wenig aufwendiger. Hier müssen die korrekten Werte berechnet werden.

COVISE

In Covise wird die physikalische Wand mit den Werten <originX>, <originY> und <originZ>, sowie den Winkeln <h>, <p> und <r> positioniert. Entsprechende Werte müssen aus den OpenDECL Attributen LeftBottom, VectorHeight, VectorWidth und Angle bei der Transformation berechnet werden.

<TILE> - <POSITION> - <ANGLE>

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Die Position der physikalischen Wand im OpenDECL wird in ParaView zu einer <Machine> und deren *LowerLeft*, *LowerRight* und *UpperRight* Attributen. Da die Position im OpenDECL aus *LeftBottom*, *VectorHeight*, *VectorWidth* und *Angle* besteht, wird die Transformation an dieser Stelle ein wenig aufwendiger. Hier müssen die korrekten Werte berechnet werden.

COVISE

In Covise wird die physikalische Wand mit den Werten <originX>, <originY> und <originZ>, sowie den Winkeln <h>, <p> und <r> positioniert. Entsprechende Werte müssen aus den OpenDECL Attributen *LeftBottom*, *VectorHeight*, *VectorWidth* und *Angle* bei der Transformation berechnet werden.

<COMPUTER> - NAME

MEGAMOL

Innerhalb von MegaMol ist der Name eines <Computer> ein Attribut einer <Machine>. Diese <Machine> spiegelt im Allgemeinen ein Einzelsystem, bzw. Rechner, wieder. Für das PVX-Dateiformat ist ein <Computer> - Name des OpenDECL ein <Machine> - Name in ParaView.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Innerhalb von ParaView ist der Name eines <Computer> ein Attribut einer <Machine>. Diese <Machine> spiegelt im Allgemeinen ein Einzelsystem, bzw. Rechner, wieder. Welches mehrere Grafikkarten enthalten kann. Für das CFG-Dateiformat ist ein <Computer> des OpenDECL auch ein <Computer> in MegaMol. Der Name wird in einer Bedingung abgefragt. Ein <Computer> - <Adapter> - <Window> kann allerdings nur einen <Output> haben.

COVISE

Wird nicht benötigt.

<ADAPTER> - NAME

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Der <Adapter> ist die <Pipe> von Equalizer. Eine <Pipe> kann einen Namen haben, das ist aber nur optional.

PARAVIEW

Wird nicht benötigt.

COVISE

Der Name kann in Covise aus der OpenDECL übernommen werden, muss allerdings von 0 aufsteigend durchnummeriert bzw. benannt werden. Bei der Transformation werden daraus die beiden Attribute <name> und <pipe> initialisiert.

<WINDOW> - NAME

MEGAMOL

Wird nicht benötigt.

EQUALIZER

In der Equalizer-Konfiguration gibt es, identisch zur OpenDECL Konfiguration, das <Window>. Beide haben einen Namen und dieser kann bei der XSL-Transformation einfach übertragen werden.

PARAVIEW

Wird nicht benötigt.

COVISE

Der Name kann in Covise aus der OpenDECL übernommen werden, muss allerdings von 0 aufsteigend durchnummeriert bzw. benannt werden. Bei der Transformation werden daraus die beiden Attribute <name> und <window> initialisiert.

<WINDOW> - COMMENT

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise bietet die Möglichkeit in comment eine Beschreibung zur Fensterkonfiguration hinzuzufügen. Hierbei gibt es keinerlei Vorgaben wie die Beschreibung aussehen muss.

<WINDOW> - HEIGHT

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Die Höhe ist bei OpenDECL ein einfaches Attribut. Dagegen wird der Wert des <window> in Equalizer als vierter Parameter des <viewports>, in <window>, gehalten.

PARAVIEW

Wird nicht benötigt.

COVISE

Der Wert height kann in Covise direkt aus der OpenDECL übernommen werden und kommt in der Covise-spezifischen Konfiguration in ein gleichnamiges Attribut.

<WINDOW> - WIDTH

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Exakt wie bei der Höhe wird die Breite des <Window> im <viewport> vom Equalizer-<window> parametrieret. Es ist der dritte Wert des <viewport>.

PARAVIEW

Wird nicht benötigt.

COVISE

Der Wert width kann in Covise direkt aus der OpenDECL übernommen werden und kommt in der Covise-spezifischen Konfiguration in ein gleichnamiges Attribut.

<WINDOW> - LEFTOFFSET

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Der linke Abstand aus <Window>, also hier „LeftOffset“, ist der erste Wert, der im <viewport> steht. Eigentlich ist das eine Koordinate im <viewport> aber da es relativ oder in Pixel angegeben wird, kann Wert bei der Transformation benutzt werden.

PARAVIEW

Wird nicht benötigt.

COVISE

Auch in Covise kann das Fenster mithilfe des LeftOffset Wertes auf der Projektionsfläche bzw. dem Bildschirm positioniert werden. Dazu ist in der Covise-spezifischen Konfiguration ein Attribut <left> vorgesehen. Hier erfolgt die Angabe in absoluten Pixeln.

<WINDOW> - BOTTOMOFFSET

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Der untere Abstand ist die Y-Koordinate des <viewport>, also der zweite Wert. Da Equalizer auch von links unten mit dem Koordinatensystem beginnt und die Werte relativ oder absolut in Pixel angegeben werden, kann dieser „BottomOffset“ direkt übertragen werden zur Y-Koordinate.

PARAVIEW

Wird nicht benötigt.

COVISE

Auch in Covise kann das Fenster mithilfe des BottomOffset Wertes auf der Projektionsfläche bzw. dem Bildschirm positioniert werden. Dazu ist in der Covise-spezifischen Konfiguration ein Attribut <bottom> vorgesehen. Hier erfolgt die Angabe in absoluten Pixeln.

<WINDOW> - DECORATION

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise bietet die Möglichkeit das Anzeigefenster zu „dekorieren“, was so viel bedeutet wie beispielsweise einen sichtbaren Rahmen um das definierte Fenster herum anzuzeigen.

<OUTPUT> - NAME

MEGAMOL

Der Name des <Output> in der OpenDECL-Konfiguration ist bei der MegaMol-Konfiguration der Name eines <Tile>. Der Name kann frei gewählt werden und unterliegt keinerlei Einschränkungen. Bei der CFG-Konfigurationsdatei wird durch ein <set> ein Element definiert, dort gibt es ein Attribut Name, welches bei der Transformation beschrieben werden kann.

EQUALIZER

Der Name des <Output> aus OpenDECL ist der Name des <Channel> in Equalizer. Das wichtige hier, ist das der Name gleich ist, wie der Parameter <channel> aus <segment>. Damit Die Verbindung zwischen <Output> und <Tile> bzw. <segment> und <channel> hergestellt werden kann.

PARAVIEW

Wird nicht benötigt.

COVISE

Auch beim der Name beim Output muss bei Covise wieder von 0 an aufsteigend durchnummeriert bzw. benannt werden. Im Gegensatz zur OpenDECL heißt Output in der Covise-spezifischen Konfiguration allerdings <channel>. Auch hier wird bei der Transformation name wieder in <name> und <channel> umgeändert. Diese müssen auch wieder gleich heißen müssen.

<OUTPUT> - COMMENT

MEGAMOL

Wird nicht benötigt.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Covise bietet die Möglichkeit die Output Konfiguration zu kommentieren bzw. eine Beschreibung hinzuzufügen. Dabei gibt es keine Vorschriften, wie dieser aussehen muss.

<OUTPUT> - HEIGHT

MEGAMOL

Die Höhe wird in OpenDECL und bei MegaMol in Pixel angegeben, deswegen können bei der Transformation, die Werte eins zu eins übernommen werden. Diese Höhe ist die Höhe des <Tile> in MegaMol. In einem <set> im CFG-Format können in einem Attribut „Value“, durch Semikolon getrennt vier Werte stehen. Davon ist der vierte die Höhe.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Die Höhe wird in OpenDECL und bei ParaView in Pixel angegeben, deswegen können bei der Transformation, die Werte eins zu eins übernommen werden. Diese Höhe ist die Höhe der <Machine> in ParaView. In einem <Geometry> Attribut in der PVX Konfiguration können vier Werte stehen. Davon ist der zweite die Höhe.

COVISE

In der OpenDECL wird die Höhe in absoluten Pixeln angegeben, was auch in Covise möglich ist. Hierbei kann der wert einfach übernommen werden. Zusätzlich bietet Covise die Möglichkeit die Höhe auch relativ (0-1) zur Höhe des Anzeigefensters anzugeben, wobei bei der Transformation entsprechend umgerechnet werden muss.

<OUTPUT> - WIDTH

MEGAMOL

Genau wie bei der Höhe, wird die Breite in Pixel angegeben. Da für MegaMol das gleiche gilt, kann der Wert ohne Umwandlung übertragen werden. Bei MegaMol ist die Breite der <Tile> die Breite des <Output> in OpenDECL. In der CFG-Datei ist es der dritte Wert in „Value“.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Die Breite wird in OpenDECL und bei ParaView in Pixel angegeben, deswegen können bei der Transformation, die Werte eins zu eins übernommen werden. Diese Breite ist die Breite der <Machine> in ParaView. In einem <Geometry> Attribut in der PVX Konfiguration können vier Werte stehen. Davon ist der erste die Breite.

COVISE

In der OpenDECL wird die Breite in absoluten Pixeln angegeben, was auch in Covise möglich ist. Hierbei kann der wert einfach übernommen werden. Zusätzlich bietet Covise die Möglichkeit die Breite auch relativ (0-1) zur Breite des Anzeigefensters anzugeben, wobei bei der Transformation entsprechend umgerechnet werden muss.

<OUTPUT> - STEREOCHANNEL

MEGAMOL

Der „StereoChannel“ des <Output> ist der Parameter für welches Auge das Bild dargestellt werden soll. Übertragen auf MegaMol, ist das der „StereoChannel“ eines <Tile>. Die Werte die es bei MegaMol gibt, sind „left“, „right“ und „mono“. Dafür wird innerhalb dem CFG-Format ein eigenes <set> erstellt, wobei der Name „tveye“ heißt und in „value“ der Wert steht.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Wird nicht benötigt.

COVISE

Der StereoChannel, welche in der Covise-spezifischen Konfiguration <stereoMode> benannt ist, gibt an, ob das Bild Stereoskopisch oder Monoskopisch angezeigt werden soll. D.h. es ist anzugeben, ob das Bild für das rechte bzw. linke Auge oder für beide Augen vorgesehen ist.

<OUTPUT> - LEFTOFFSET

MEGAMOL

Das ist der linke Abstand zum Rand. Genau wie bei OpenDECL heißt das Attribut bei MegaMol „LeftOffset“. Es ist in <Tile>. Der Wert wird bei beiden, MegaMol und OpenDECL, in Pixel angegeben. Das ist in der CFG-Konfiguration innerhalb das Positionierungs- und Größen-<set> an der ersten Position.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Das linke Offset wird in OpenDECL und bei ParaView in Pixel angegeben, deswegen können bei der Transformation, die Werte eins zu eins übernommen werden. Dieses linke Offset ist das linke Offset der <Machine> in ParaView. In einem <Geometry> Attribut in der PVX Konfiguration können vier Werte stehen. Davon ist der dritte das linke Offset.

COVISE

In OpenDECL wie auch in der Covise-spezifischen Konfiguration werden die Offsetwerte in absoluten Pixeln angegeben. Der zu übernehmende Wert steht in der Covise-spezifischen Konfiguration in dem Attribut, welches schlicht <left> heißt.

<OUTPUT> - BOTTOMOFFSET

MEGAMOL

Bei OpenDECL wird die linke untere Ecke als Ausgangspunkt verwendet. Deswegen gibt es einen „BottomOffset“. Dagegen ist bei MegaMol die linke obere Ecke als Startkoordinate festgelegt. Damit der richtige Wert von „BottomOffset“ aus <Output> nach „TopOffset“ aus <Tile> transformiert wird, müssen bei der XSL-Transformation die Werte anhand von der Höhe und des unteren Abstandes berechnet werden.

EQUALIZER

Wird nicht benötigt.

PARAVIEW

Das untere Offset wird in OpenDECL und bei ParaView in Pixel angegeben, deswegen können bei der Transformation, die Werte eins zu eins übernommen werden. Dieses untere Offset ist das untere Offset der <Machine> in ParaView. In einem <Geometry> Attribut in der PVX Konfiguration können vier Werte stehen. Davon ist der vierte das untere Offset.

COVISE

In OpenDECL wie auch in der Covise-spezifischen Konfiguration werden die Offsetwerte in absoluten Pixeln angegeben. Der zu übernehmende Wert steht in der Covise-spezifischen Konfiguration in dem Attribut, welches schlicht <bottom> heißt.

TRANSFORMATIONSDATEIEN

Um die vereinheitlichte XML-Datei für die jeweiligen Tools nutzbar zu machen, ist eine XML-Transformation pro Tool vorgesehen. Gegebenenfalls kann eine solche Transformation sehr kompliziert werden, allerdings ist diese auch nur genau einmal pro Tool zu erstellen und danach auf alle Konfigurationsdateien anwendbar.

Eine Transformation besteht im Wesentlichen aus zwei Teilen. Ein Teil, welcher nur Informationen enthält, welche das spezifische Tool betreffen. Und ein weiterer Teil, welcher die Daten aus der Open DECL transformiert.

SPEZIFISCHER TEIL

Hier werden Daten hinzugefügt, welche das Tool, für welches die Transformation gedacht ist im speziellen benötigt. Bei einer Transformation für MegaMol sind das im speziellen das XML-Rootelement und dessen Attribute.

ALLGEMEINER TEIL

Hier werden die Daten aus dem OpenDECL in die jeweilige Struktur eines der Tools transformiert.

MEGAMOL TRANSFORMATION

Im Folgenden wird eine Transformation von OpenDECL auf MegaMol beschrieben:

Der Anfang der XSL-Datei ist mehr oder weniger Standard mit ein paar wenigen Output-Optionen. Unter anderem wird hier festgelegt, dass die Formatierung der Datei nicht verloren geht. Dies verbessert die Lesbarkeit und vereinfacht dadurch auch das Erstellen einer anderen Transformation.

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="xml" encoding="utf-8" indent="yes"/>
```

Dann wird der toolspezifische Teil erstellt. Bei MegaMol beschränkt sich das hier auf ein Root-Element und ein paar Namespaces.

```
  <xsl:template match="/">
    <xsl:element name="TiledDisplay">
      <xsl:attribute name="xmlns:spacei">
        http://www.w3.org/2001/XMLSchema-instance
      </xsl:attribute>

      <xsl:attribute name="xmlns:space">
        http://www.visus.uni-stuttgart.de/vvand/2012/description
      </xsl:attribute>

      <xsl:apply-templates />
    </xsl:element>
  </xsl:template>
```

Nun werden die für das jeweilige Tool relevanten Elemente transformiert:
Für <Tile> ist das sehr einfach, weil nur der name übernommen werden muss.

```

<xsl:template match="Tile">
  <xsl:element name="Name">
    <xsl:value-of select="@Name"/>
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Für <WidthPixel> wird der Text Wert übernommen.

```

<xsl:template match="WidthPixel">
  <xsl:element name="Width">
    <xsl:value-of select="text ()"/>
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Für <HeightPixel> wird der Text Wert übernommen.

```

<xsl:template match="HeightPixel">
  <xsl:element name="Height">
    <xsl:value-of select="text ()"/>
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Ein <Screen> im OpenDECL wird in ein <Machines> in der MegaMol Konfigurationsdatei umgewandelt.

```

<xsl:template match="Screen">
  <xsl:element name="Machines">
    <xsl:apply-templates />
  </xsl:element>
</xsl:template>

```

Der Name vom <Computer> im OpenDECL wird zum Text Wert in <Machine> - <Identity> in der MegaMol Konfigurationsdatei.

```

<xsl:template match="Computer">
  <xsl:element name="Machine">
    <xsl:element name="Identity">
      <xsl:value-of select="@Name"/>
    </xsl:element>

    <xsl:apply-templates />
  </xsl:element>
</xsl:template>

```

Das <Output> Element des OpenDECL wird hier in ein <Tile> Element in der MegaMol Konfigurationsdatei transformiert.

```

<xsl:template match="Output">
  <xsl:element name="Tiles">
    <xsl:element name="Tile">
      <xsl:element name="Name">
        <xsl:value-of select="@Name"/>
      </xsl:element>

```



```

        <xsl:apply-templates />
      </xsl:element>
    </xsl:element>
  </xsl:template>

```

Der <Height> Text Wert im OpenDECL wird zu einem <WindowsHeight> Wert in der MegaMol Konfigurationsdatei.

```

<xsl:template match="Height">
  <xsl:element name="WindowHeight">
    <xsl:value-of select="text()" />
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Der <Width> Text Wert im OpenDECL wird zu einem <WindowsWidth> Wert in der MegaMol Konfigurationsdatei.

```

<xsl:template match="Width">
  <xsl:element name="WindowWidth">
    <xsl:value-of select="text()" />
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Der <StereoChannel> Text Wert im OpenDECL wird direkt in die MegaMol Konfigurationsdatei übernommen.

```

<xsl:template match="StereoChannel">
  <xsl:element name="StereoChannel">
    <xsl:value-of select="text()" />
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Der <LeftOffset> Text Wert im OpenDECL wird direkt in die MegaMol Konfigurationsdatei übernommen.

```

<xsl:template match="LeftOffset">
  <xsl:element name="LeftOffset">
    <xsl:value-of select="text()" />
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Der <BottomOffset> Text Wert im OpenDECL wird zum <TopOffset> in der MegaMol Konfigurationsdatei. In diesem Beispiel ist das möglich die Werte direkt zu übernehmen, da diese immer 0 sind. Falls die Werte nicht 0 sind, muss an dieser Stelle die Transformation erweitert werden und der Wert entsprechend berechnet werden.

```

<xsl:template match="BottomOffset">
  <xsl:element name="TopOffset">
    <xsl:value-of select="text()" />
  </xsl:element>

  <xsl:apply-templates />
</xsl:template>

```

Die letzte Transformationszeile dient dazu, dass Alle überflüssigen Text Werte der OpenDECL Datei verschwinden. Andernfalls würden sich diese noch in der Transformierten Datei befinden, was diese ungültig für MegaMol machen würde.

```
<xsl:template match="text()"/>
</xsl:stylesheet>
```

Wie man sehen kann, ist die Transformation auf MegaMol relative einfach und überschaubar. Für weitere Tools kann es hier zu komplizierteren Transformationen kommen. Allerdings ist diese auch nur genau einmal nötig. Daher ist es den Aufwand wert.

TRACKING SYSTEM UND EINGABEGERÄTE

OpenDECL ist so aufgebaut, dass es ohne großen Aufwand möglich ist Einstellungen für Tracking Systeme und Eingabegeräte hinzuzufügen. Hierzu kann neben den beiden Punkten <Canvas> und <Screen> ein weiterer Punkt eingefügt werden, welcher die zur Spezifizierung eben jener Tracking Systeme und Eingabegeräte benötigten Attribute beinhaltet.

Im Folgenden wird mit Blick auf die COVISE Konfiguration für Tracking Systeme und Eingabegeräte eine kurze Einführung in die Idee einer Konfigurationsmöglichkeit gegeben:

COVISE sieht hierfür den Punkt <Input> vor in dem die Einstellungen vorgenommen werden. Dabei wird in <TrackingSystem> festgelegt welches Tracking System verwendet wird. In diesem Beispiel wird das Tracking System auf DTRACK festgelegt. Dabei handelt es sich um ein optisches Tracking System, dass via Ethernet mit einem PC verbunden ist. Um entsprechendes System zu starten wird in <DTrack> unter <Startup> ein PC mit entsprechender Ethernet Adresse angegeben. Da dieses System nur Positionsdaten liefert, wird zusätzlich unter <ButtonSystem> ein Eingabegerät festgelegt, in diesem Beispiel wird hier MOUSE festgelegt. Hierdurch wird allerdings nur irgendeine Maus als Eingabegerät festgelegt. Die genauere Definition einer speziellen Maus wird unter <ButtonConfig> in <ButtonDevise> festgelegt. Um die entsprechenden eingebundenen Tracking Systeme zu kalibrieren gibt es unter <TrackingSystem> die Punkte <Offset> und <Orientation>. Hier können die Objekte den Werten x, y und z bzw. den Winkeln h, p und r kalibriert werden.

STEREOSKOPIE

Der Punkt Stereo Modus ermöglicht das Festlegen des Stereomodus. Hier wird zwischen monoskopischer und stereoskopischer Darstellung unterschieden. Bei der monoskopischen Darstellung handelt es sich um ein normales zweidimensionales Bild wohingegen bei der stereoskopischen Darstellung zwei Bilder gleichzeitig angezeigt werden. Jeweils eins für das rechte bzw. linke Auge, wodurch das Bild dreidimensional erscheint. Die stereoskopische Darstellung kann mit `<Stereo value="ON" />` ein bzw. abgeschaltet werden. Da die Augen von Mensch zu Mensch unterschiedlich weit auseinander sind, bietet `<Stereo seperation="[int]" />` die Möglichkeit einen individuellen Wert für den Abstand der Augen anzugeben. Der Wert 0 gleicht hier dem Abschalten des Stereomodus.

Es gibt verschiedene Methoden zur Darstellung von Stereoskopie. Beispielsweise Anaglyphenbilder, aktive und passive Stereoskopie. Letztere werden im Folgenden genauer beschrieben.

AKTIVES STEREO

Bei dieser Art der Stereoskopie werden zwei verschiedene Bilder gerendert, eines für jedes Auge. Diese werden abwechselnd nacheinander für das jeweilige Auge angezeigt. Um schließlich ein dreidimensionales Bild zu sehen wird eine Shutterbrille benötigt.

PASSIVES STEREO

Bei der passiven Stereoskopie werden die Bilder für das jeweils rechte und linke Auge gleichzeitig angezeigt. Hierzu werden allerdings zwei Ausgänge und Projektoren benötigt, die die Bilder ausgeben. Jeder dieser Projektoren hat eine polarisierte Linse. Zusätzlich wird eine spezielle Brille gebraucht, die entsprechend den Linsen ebenfalls polarisierte Gläser hat, was dazu führt, dass jedes Auge nur das für sich vorgesehene Bild sieht.

FAZIT

Zusammenfassend lässt sich sagen, dass die evaluierten Tools zum Großteil sehr ähnlich aufgebaut sind, was ein Einheitliches Konfigurationsformat leichter macht, als erwartet.

Wesentlich schwieriger war es allerdings brauchbare Informationen über die untersuchten Tools zu finden. Die Dokumentationen waren oft sehr spärlich und Tests in der praktischen Anwendung waren eher schwer umzusetzen.

Nichtsdestotrotz ist eine Vereinheitlichung der Konfigurationsdateien dieser Tools ein Sinnvoller Schritt und sollte in Zukunft einiges an Arbeitsaufwand ersparen. Des Weiteren können dadurch Tool-Wechsel einfacher durchgeführt werden, da die Konfiguration nur einmal vorgenommen werden muss.

REFERENZEN

<https://svn.vis.uni-stuttgart.de/trac/megamol>

<http://www.equalizergraphics.com/>

<http://www.paraview.org/>

<http://www.hlr.de/organization/av/vis/covise>

<https://wci.llnl.gov/simulation/computer-codes/visit/>

<http://icet.sandia.gov/>

ANHÄNGE

Die Anhänge bieten einen Überblick über die entwickelten Dateien.

Aus der „OpenDECL.xsd“ Datei lässt sich eine Standard „OpenDECL.xml“ Datei generieren, welche dann mit Daten gespeist werden kann. Diese XML Datei kann dann zum Beispiel mit Hilfe der „OpenDECL – MegaMol.xsl“ Datei zu einer „MegaMol.xml“ Datei transformiert werden.

OPENDECL.XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  vc:minVersion="1.1">

  <xs:element name="OpenDECL">
    <xs:complexType>
      <xs:all>
        <xs:element name="Canvas">
          <xs:complexType>
            <xs:all>
              <xs:element name="Wall" minOccurs="0">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="LeftBottom">
                      <xs:complexType>
                        <xs:attribute name="X"/>
                        <xs:attribute name="Y"/>
                        <xs:attribute name="Z"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:all>
                </xs:complexType>
              </xs:element>

              <xs:element name="VectorHeight">
                <xs:complexType>
                  <xs:attribute name="X"/>
                  <xs:attribute name="Y"/>
                  <xs:attribute name="Z"/>
                </xs:complexType>
              </xs:element>

              <xs:element name="VectorWidth">
                <xs:complexType>
                  <xs:attribute name="X"/>
                  <xs:attribute name="Y"/>
                  <xs:attribute name="Z"/>
                </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>

  <xs:element name="Tile">
    <xs:complexType>
      <xs:all>
        <xs:element name="WitdhPixel"/>
        <xs:element name="HeightPixel"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="Width-mm"/>
<xs:element name="Height-mm"/>
<xs:element name="Position">
  <xs:complexType>
    <xs:all>
      <xs:element name="LeftBottom">
        <xs:complexType>
          <xs:attribute name="X"/>
          <xs:attribute name="Y"/>
          <xs:attribute name="Z"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="VectorHeight">
        <xs:complexType>
          <xs:attribute name="X"/>
          <xs:attribute name="Y"/>
          <xs:attribute name="Z"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="VectorWidth">
        <xs:complexType>
          <xs:attribute name="X"/>
          <xs:attribute name="Y"/>
          <xs:attribute name="Z"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="Angle">
        <xs:complexType>
          <xs:attribute name="H"/>
          <xs:attribute name="P"/>
          <xs:attribute name="R"/>
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:all>
<xs:attribute name="Name"/>
<xs:attribute name="Comment"/>
</xs:complexType>
</xs:element>

<xs:element name="Screen">
  <xs:complexType>
    <xs:all>
      <xs:element name="Computer">
        <xs:complexType>
          <xs:all>
            <xs:element name="Adapter" maxOccurs="unbounded">
              <xs:complexType>
                <xs:all>
                  <xs:element
                    name="Window"
                    maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:all>

```



```

<?xml version="1.0" encoding="utf-8"?>

<OpenDECL>
  <Canvas>
    <!--
    <Wall>
      <LeftBottom X="???" Y="???" Z="???" />
      <VectorHeight X="???" Y="???" Z="???" />
      <VectorWidth X="???" Y="???" Z="???" />
    </Wall>
    -->

    <Tile Name="Vvand" Comment="">
      <WidthPixel>10800</WidthPixel>
      <HeightPixel>4096</HeightPixel>

      <!--
      <Witdh-mm></Witdh-mm>
      <Height-mm></Height-mm>

      <Position>
        <LeftBottom X="???" Y="???" Z="???" />
        <VectorHeight X="???" Y="???" Z="???" />
        <VectorWidth X="???" Y="???" Z="???" />
        <Angle H="???" P="???" R="???" />
      </Position>
      -->
    </Tile>
  </Canvas>

  <Screen>
    <Computer Name="Test01">
      <Adapter Name="">
        <Window Name="" Comment="">
          <!--
          <Height></Height>
          <Width></Width>

          <LeftOffset></LeftOffset>
          <BottomOffset></BottomOffset>

          <Decoration></Decoration>
          -->

          <Output
            Name="(0, 0), [2400, 4096] auf Maschine Test01" Comment="">
            <Height>4096</Height>
            <Width>2400</Width>
            <StereoChannel>Left</StereoChannel>
            <LeftOffset>0</LeftOffset>
            <BottomOffset>0</BottomOffset>
          </Output>
        </Window>
      </Adapter>
    </Computer>

    <Computer Name="Test02">
      <Adapter Name="">

```

```

<Window Name="" Comment="">
  <!--
  <Height></Height>
  <Width></Width>

  <LeftOffset></LeftOffset>
  <BottomOffset></BottomOffset>

  <Decoration></Decoration>
  -->

  <Output
Name="(0, 0), [2400, 4096] auf Maschine Test02" Comment="">
  <Height>4096</Height>
  <Width>2400</Width>
  <StereoChannel>Right</StereoChannel>
  <LeftOffset>0</LeftOffset>
  <BottomOffset>0</BottomOffset>
  </Output>
</Window>
</Adapter>
</Computer>

<Computer Name="Test03">
  <Adapter Name="">
    <Window Name="" Comment="">
      <!--
      <Height></Height>
      <Width></Width>

      <LeftOffset></LeftOffset>
      <BottomOffset></BottomOffset>

      <Decoration></Decoration>
      -->

      <Output
Name="(0, 0), [2400, 4096] auf Maschine Test03" Comment="">
      <Height>4096</Height>
      <Width>2400</Width>
      <StereoChannel>Left</StereoChannel>
      <LeftOffset>2100</LeftOffset>
      <BottomOffset>0</BottomOffset>
      </Output>
    </Window>
  </Adapter>
</Computer>

<Computer Name="Test04">
  <Adapter Name="">
    <Window Name="" Comment="">
      <!--
      <Height></Height>
      <Width></Width>

      <LeftOffset></LeftOffset>
      <BottomOffset></BottomOffset>

      <Decoration></Decoration>
      -->

      <Output

```

```

Name="(0, 0), [2400, 4096] auf Maschine Test04" Comment="">
    <Height>4096</Height>
    <Width>2400</Width>
    <StereoChannel>Right</StereoChannel>
    <LeftOffset>2100</LeftOffset>
    <BottomOffset>0</BottomOffset>
</Output>
</Window>
</Adapter>
</Computer>

<Computer Name="Test05">
    <Adapter Name="">
        <Window Name="" Comment="">
            <!--
            <Height></Height>
            <Width></Width>

            <LeftOffset></LeftOffset>
            <BottomOffset></BottomOffset>

            <Decoration></Decoration>
            -->

            <Output
Name="(0, 0), [2400, 4096] auf Maschine Test05" Comment="">
                <Height>4096</Height>
                <Width>2400</Width>
                <StereoChannel>Left</StereoChannel>
                <LeftOffset>4200</LeftOffset>
                <BottomOffset>0</BottomOffset>
            </Output>
            </Window>
        </Adapter>
    </Computer>

<Computer Name="Test06">
    <Adapter Name="">
        <Window Name="" Comment="">
            <!--
            <Height></Height>
            <Width></Width>

            <LeftOffset></LeftOffset>
            <BottomOffset></BottomOffset>

            <Decoration></Decoration>
            -->

            <Output
Name="(0, 0), [2400, 4096] auf Maschine Test06" Comment="">
                <Height>4096</Height>
                <Width>2400</Width>
                <StereoChannel>Right</StereoChannel>
                <LeftOffset>4200</LeftOffset>
                <BottomOffset>0</BottomOffset>
            </Output>
            </Window>
        </Adapter>
    </Computer>

<Computer Name="Test07">

```

```

<Adapter Name="">
  <Window Name="" Comment="">
    <!--
    <Height></Height>
    <Width></Width>

    <LeftOffset></LeftOffset>
    <BottomOffset></BottomOffset>

    <Decoration></Decoration>
    -->

    <Output
Name="(0, 0), [2400, 4096] auf Maschine Test07" Comment="">
      <Height>4096</Height>
      <Width>2400</Width>
      <StereoChannel>Left</StereoChannel>
      <LeftOffset>6300</LeftOffset>
      <BottomOffset>0</BottomOffset>
    </Output>
  </Window>
</Adapter>
</Computer>

<Computer Name="Test08">
  <Adapter Name="">
    <Window Name="" Comment="">
      <!--
      <Height></Height>
      <Width></Width>

      <LeftOffset></LeftOffset>
      <BottomOffset></BottomOffset>

      <Decoration></Decoration>
      -->

      <Output
Name="(0, 0), [2400, 4096] auf Maschine Test08" Comment="">
        <Height>4096</Height>
        <Width>2400</Width>
        <StereoChannel>Right</StereoChannel>
        <LeftOffset>6300</LeftOffset>
        <BottomOffset>0</BottomOffset>
      </Output>
    </Window>
  </Adapter>
</Computer>

<Computer Name="Test09">
  <Adapter Name="">
    <Window Name="" Comment="">
      <!--
      <Height></Height>
      <Width></Width>

      <LeftOffset></LeftOffset>
      <BottomOffset></BottomOffset>

      <Decoration></Decoration>
      -->

```

```

        <Output
Name="(0, 0), [2400, 4096] auf Maschine Test09" Comment="">
        <Height>4096</Height>
        <Width>2400</Width>
        <StereoChannel>Left</StereoChannel>
        <LeftOffset>8400</LeftOffset>
        <BottomOffset>0</BottomOffset>
    </Output>
</Window>
</Adapter>
</Computer>

<Computer Name="Test10">
    <Adapter Name="">
        <Window Name="" Comment="">
            <!--
            <Height></Height>
            <Width></Width>

            <LeftOffset></LeftOffset>
            <BottomOffset></BottomOffset>

            <Decoration></Decoration>
            -->

            <Output
Name="(0, 0), [2400, 4096] auf Maschine Test10" Comment="">
            <Height>4096</Height>
            <Width>2400</Width>
            <StereoChannel>Right</StereoChannel>
            <LeftOffset>8400</LeftOffset>
            <BottomOffset>0</BottomOffset>
        </Output>
        </Window>
    </Adapter>
</Computer>
</Screen>
</OpenDECL>

```

```

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="xml" encoding="utf-8" indent="yes"/>

  <xsl:template match="/">
    <xsl:element name="TiledDisplay">
      <xsl:attribute name="xmlns:cei"
        http://www.w3.org/2001/XMLSchema-instance
      </xsl:attribute>

      <xsl:attribute name="xmlns"
        http://www.visus.uni-stuttgart.de/vvand/2012/description
      </xsl:attribute>

      <xsl:apply-templates />
    </xsl:element>
  </xsl:template>

  <xsl:template match="Tile">
    <xsl:element name="Name">
      <xsl:value-of select="@Name"/>
    </xsl:element>

    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="WidthPixel">
    <xsl:element name="Width">
      <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="HeightPixel">
    <xsl:element name="Height">
      <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="Screen">
    <xsl:element name="Machines">
      <xsl:apply-templates />
    </xsl:element>
  </xsl:template>

  <xsl:template match="Computer">
    <xsl:element name="Machine">
      <xsl:element name="Identity">
        <xsl:value-of select="@Name"/>
      </xsl:element>
    </xsl:element>
  </xsl:template>

```

```

        <xsl:apply-templates />
    </xsl:element>
</xsl:template>

<xsl:template match="Output">
    <xsl:element name="Tiles">
        <xsl:element name="Tile">
            <xsl:element name="Name">
                <xsl:value-of select="@Name"/>
            </xsl:element>

            <xsl:apply-templates />
        </xsl:element>
    </xsl:element>
</xsl:template>

<xsl:template match="Height">
    <xsl:element name="WindowHeight">
        <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
</xsl:template>

<xsl:template match="Width">
    <xsl:element name="WindowWidth">
        <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
</xsl:template>

<xsl:template match="StereoChannel">
    <xsl:element name="StereoChannel">
        <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
</xsl:template>

<xsl:template match="LeftOffset">
    <xsl:element name="LeftOffset">
        <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
</xsl:template>

<xsl:template match="BottomOffset">
    <xsl:element name="TopOffset">
        <xsl:value-of select="text()"/>
    </xsl:element>

    <xsl:apply-templates />
</xsl:template>

<xsl:template match="text()"/>
</xsl:stylesheet>

```



```

<?xml version="1.0" encoding="utf-8"?>
<TiledDisplay
xmlns:spacei="http://www.w3.org/2001/XMLSchema-instance"
xmlns:space="http://www.visus.uni-stuttgart.de/vvand/2012/description">
  <Name>VVand</Name>
  <Width>10800</Width>
  <Height>4096</Height>
  <Machines>
    <Machine>
      <Identity>Test01</Identity>
      <Tiles>
        <Tile>
          <Name>(0, 0), [2400, 4096] auf Maschine Test01</Name>
          <WindowHeight>4096</WindowHeight>
          <WindowWidth>2400</WindowWidth>
          <StereoChannel>Left</StereoChannel>
          <LeftOffset>0</LeftOffset>
          <TopOffset>0</TopOffset>
        </Tile>
      </Tiles>
    </Machine>
    <Machine>
      <Identity>Test02</Identity>
      <Tiles>
        <Tile>
          <Name>(0, 0), [2400, 4096] auf Maschine Test02</Name>
          <WindowHeight>4096</WindowHeight>
          <WindowWidth>2400</WindowWidth>
          <StereoChannel>Right</StereoChannel>
          <LeftOffset>0</LeftOffset>
          <TopOffset>0</TopOffset>
        </Tile>
      </Tiles>
    </Machine>
    <Machine>
      <Identity>Test03</Identity>
      <Tiles>
        <Tile>
          <Name>(0, 0), [2400, 4096] auf Maschine Test03</Name>
          <WindowHeight>4096</WindowHeight>
          <WindowWidth>2400</WindowWidth>
          <StereoChannel>Left</StereoChannel>
          <LeftOffset>2100</LeftOffset>
          <TopOffset>0</TopOffset>
        </Tile>
      </Tiles>
    </Machine>
    <Machine>
      <Identity>Test04</Identity>
      <Tiles>
        <Tile>
          <Name>(0, 0), [2400, 4096] auf Maschine Test04</Name>
          <WindowHeight>4096</WindowHeight>
          <WindowWidth>2400</WindowWidth>
          <StereoChannel>Right</StereoChannel>
          <LeftOffset>2100</LeftOffset>
          <TopOffset>0</TopOffset>
        </Tile>
      </Tiles>
    </Machine>
  </Machines>
</TiledDisplay>

```

```

</Tiles>
</Machine>
<Machine>
  <Identity>Test05</Identity>
  <Tiles>
    <Tile>
      <Name>(0, 0), [2400, 4096] auf Maschine Test05</Name>
      <WindowHeight>4096</WindowHeight>
      <WindowWidth>2400</WindowWidth>
      <StereoChannel>Left</StereoChannel>
      <LeftOffset>4200</LeftOffset>
      <TopOffset>0</TopOffset>
    </Tile>
  </Tiles>
</Machine>
<Machine>
  <Identity>Test06</Identity>
  <Tiles>
    <Tile>
      <Name>(0, 0), [2400, 4096] auf Maschine Test06</Name>
      <WindowHeight>4096</WindowHeight>
      <WindowWidth>2400</WindowWidth>
      <StereoChannel>Right</StereoChannel>
      <LeftOffset>4200</LeftOffset>
      <TopOffset>0</TopOffset>
    </Tile>
  </Tiles>
</Machine>
<Machine>
  <Identity>Test07</Identity>
  <Tiles>
    <Tile>
      <Name>(0, 0), [2400, 4096] auf Maschine Test07</Name>
      <WindowHeight>4096</WindowHeight>
      <WindowWidth>2400</WindowWidth>
      <StereoChannel>Left</StereoChannel>
      <LeftOffset>6300</LeftOffset>
      <TopOffset>0</TopOffset>
    </Tile>
  </Tiles>
</Machine>
<Machine>
  <Identity>Test08</Identity>
  <Tiles>
    <Tile>
      <Name>(0, 0), [2400, 4096] auf Maschine Test08</Name>
      <WindowHeight>4096</WindowHeight>
      <WindowWidth>2400</WindowWidth>
      <StereoChannel>Right</StereoChannel>
      <LeftOffset>6300</LeftOffset>
      <TopOffset>0</TopOffset>
    </Tile>
  </Tiles>
</Machine>
<Machine>
  <Identity>Test09</Identity>
  <Tiles>
    <Tile>
      <Name>(0, 0), [2400, 4096] auf Maschine Test09</Name>
      <WindowHeight>4096</WindowHeight>
      <WindowWidth>2400</WindowWidth>
      <StereoChannel>Left</StereoChannel>

```

```
        <LeftOffset>8400</LeftOffset>
        <TopOffset>0</TopOffset>
    </Tile>
</Tiles>
</Machine>
<Machine>
    <Identity>Test10</Identity>
    <Tiles>
        <Tile>
            <Name>(0, 0), [2400, 4096] auf Maschine Test10</Name>
            <WindowHeight>4096</WindowHeight>
            <WindowWidth>2400</WindowWidth>
            <StereoChannel>Right</StereoChannel>
            <LeftOffset>8400</LeftOffset>
            <TopOffset>0</TopOffset>
        </Tile>
    </Tiles>
</Machine>
</Machines>
</TiledDisplay>
```

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 11.06.2015

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, 11.06.2015