

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK

INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK

PROFESSUR FÜR COMPUTERGRAPHIK UND VISUALISIERUNG

PROF. DR. STEFAN GUMHOLD

## Diplomarbeit

zur Erlangung des akademischen Grades  
Diplom-Medieninformatiker

# Open Display Environment Configuration Language

Ronald Großmann

(Geboren am 6. April 1990 in Sebnitz, Mat.-Nr.: 3507432)

Betreuer: Dr. rer. nat. Sebastian Grottel

Dresden, 6. Juli 2015

---

## Aufgabenstellung

Die Anzahl der Multi-Display-Installationen nimmt in beinahe jeder Umgebung zu: angefangen von Desktopsystemen mit mehreren Monitoren, über Projektionsflächen mit mehreren Projektoren (aka Powerwalls) bis hin zu komplexen VR-Installationen wie Caves. Oft benötigen solche Display-Systeme leistungsstarke, parallel GPU-Cluster zur Bilderzeugung, allein um der notwendigen Pixelfüllrate gerecht werden zu können. Zusätzlich kommen in solchen Anlagen üblicherweise auch Tracking-Systeme zum Einsatz, die den Benutzer, also die physische Welt, mit den dargestellten Szenen, der künstlichen Welt, verbinden; auch im Desktop-Bereich, z.B. durch Windows Kinect oder Leap Motion. Um solche Systeme zu betreiben bedarf es komplexer Software. Diese ist oft im akademischen Umfeld entwickelt. Solche Software muss umfassend konfiguriert werden, einerseits was die verfügbare Rechnerinfrastruktur betrifft (welche Computer sind mit welchen Ausgabegeräten verbunden, welche Computer dienen rein zur entfernten Bilderzeugung und wie sind die Rechner miteinander vernetzt), andererseits auch was die logischen und physikalischen Parameter der Ausgabegeräte betrifft (virtuelle Desktop-Größen und Teile einzelner Beamer, physikalische Anordnung von Display oder Projektoren und Abgleich mit den Raumkoordinaten eines Benutzer-Trackings). Allerdings hat sich für diese Konfigurationen bisher kein Standard entwickelt.

In dieser Arbeit soll ein Vorschlag für so ein standardisiertes Konfigurationsformat auf Basis von XML entwickelt werden. Mittels XSLT soll es möglich sein, aus einer XML-Datei Konfigurationsdateien für unterschiedliche Programme zu erzeugen. Ein graphischer interaktiver Editor soll das Erstellen und Bearbeiten der XML-Dateien, sowohl der strukturellen Eigenschaften (Compute-Cluster-Architektur, inklusive GPUs und Display-Anschlüssen) als auch der 3D physikalischen Eigenschaften (Display-, Projekt-Setup) anschaulich und einfach ermöglichen. Das XML-Format muss ?sauber? durch Namespaces aufgeteilt und erweiterbar sein, was auch durch entsprechende Funktionen im graphischen Editor reflektiert werden muss (z.B. muss es im Editor möglich sein, eigentlich nicht unterstützte Tags editieren und beim Abspeichern erhalten zu können). Die Erstellung von XSLT-Dateien muss durch den Editor NICHT unterstützt werden, ihre Anwendung zum Export der Konfiguration in entsprechende andere Formate jedoch schon.

Folgende Hardware-Installationen müssen unterstützt werden:

1. Desktop-Computer mit mehreren Monitoren (mindestens zwei) die nicht in einer gemeinsamen Ebene stehen.
2. Stereo-Powerwall durch zwei Beamer betrieben an einem Rechner (Powerwall an der Professur

---

CGV)

3. Großfläche Displaywand mit mehreren Panels (Displaywand an der Professur Multimedia-Technologie)
4. Fünf-Wand-CAVE mit zehn Beamer (im VR-Labor des Lehrstuhls Konstruktionstechnik / CAD)

Hierbei müssen die physikalischen Display- und Projektionsanordnungen unterstützt werden, und zusätzliche Infrastruktur, wie z.B. Computer, GPUs, Tracking-Systeme etc., sollen so weit wie möglich unterstützt werden.

Die Hardwareinstallationen sollen in ihrem physikalischen Raum, in Metern, frei definierbar sein. Ist kein Benutzertracking vorhanden, so muss eine Standardposition für den Benutzer (Blickpunkt) konfigurierbar sein.

Die Konfiguration der Rechnerinfrastruktur muss mindestens die Computer enthalten, die direkt an die Ausgabegeräte angeschlossen sind. Ihre Netzwerkverbindungen untereinander sollten enthalten sein. Eventuelle Compute-Cluster zur Bilderzeugung und ihre Netzwerkverbindungen untereinander, sowie zu den Ausgaberechnern sollten ebenfalls konfigurierbar sein.

Folgende Software muss unterstützt werden, indem ihre Konfigurationsdateien, mindestens aber der entsprechend dieser Arbeit relevante Teil der Konfigurationsdateien, erzeugt werden kann:

- a,** MegaMol, bzw. mittels einer kleinen Bibliothek jede an der TUD selbst entwickelte Software
- b,** Paraview
- c,** Equalizer (optional)

Weitere Software soll nach Absprache mit dem Betreuer ebenfalls unterstützt werden.

Die Bearbeitung erfolgt mit diesen Teilzielen:

- Anforderungsanalyse auf Basis der vorgegebenen Display-Hardware und Konfigurationsspezifikationen der einzusetzenden Software
- Literaturrecherche zur Large-Display- und VR-Software-Middleware und Konfigurationen. Auch zu allgemeinen Arbeiten zur Kalibrieren und Konfiguration solcher Hardware-System
- Spezifikation der XML-basierten Konfiguration
- Umsetzung des geforderten Editor-Prototypens inklusive XSLT-basiertem Export der Konfigurationen

- 
- Evaluierung im Kontext der vorgegebenen Display-Systeme durch Darlegung und Durchführung des kompletten Konfigurationsprozesses anhand der vorgegebene Software
  - Optional: Erweiterung des Editors um weitere Funktionalitäten zur semi-automatischen Erzeugung der Konfigurationsdateien
  - Optional: Code-Bibliothek zur direkten Nutzung der Konfigurations-Xml-Datei
  - Optional: Untersuchung weiterer Display-Konfigurationen (z.B. gekrümmter Projektionen) und weiterer Visualisierungs- und VR-Software

---

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Arbeit zum Thema:

*Open Display Environment Configuration Language*

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 6. Juli 2015

Ronald Großmann

---

## **Kurzfassung**

Zusammenfassung Text Deutsch

## **Abstract**

abstract text english

# Inhaltsverzeichnis

<b>1 Problemstellung</b>	<b>3</b>
1.1 Problemfelder . . . . .	3
<b>2 Verwandte Arbeiten</b>	<b>4</b>
2.1 Problemfeld 1 . . . . .	4
2.2 Problemfeld 2 . . . . .	4
2.3 Problemfeld 3 . . . . .	4
<b>3 Anforderungsanalyse</b>	<b>5</b>
3.1 Hardwareanalyse . . . . .	5
3.2 Softwareanalyse . . . . .	5
<b>4 Konzeption</b>	<b>6</b>
<b>5 OpenDECL Spezifikation</b>	<b>7</b>
5.1 Ansatz . . . . .	7
5.2 Aufbau . . . . .	7
5.2.1 node . . . . .	7
5.2.1.1 graphics-device . . . . .	7
5.2.1.2 network-device . . . . .	7
5.2.2 network . . . . .	7
5.2.3 display-setup . . . . .	7
5.2.3.1 user . . . . .	7
5.2.3.2 display . . . . .	7
<b>6 XSLT Konfigurationsgenerierung</b>	<b>8</b>
6.1 Beispiel . . . . .	8
<b>7 Editor</b>	<b>9</b>
7.1 Funktionen . . . . .	9

7.2 Umsetzung . . . . .	9
<b>8 Evaluation</b>	<b>10</b>
<b>9 Diskussion</b>	<b>11</b>
<b>10 Ausblick</b>	<b>12</b>
<b>Literaturverzeichnis</b>	<b>13</b>



# **1 Problemstellung**

## **1.1 Problemfelder**

## **2 Verwandte Arbeiten**

### **2.1 Problemfeld 1**

### **2.2 Problemfeld 2**

### **2.3 Problemfeld 3**

## **3 Anforderungsanalyse**

### **3.1 Hardwareanalyse**

### **3.2 Softwareanalyse**

## **4 Konzeption**

## **5 OpenDECL Spezifikation**

### **5.1 Ansatz**

### **5.2 Aufbau**

#### **5.2.1 node**

##### **5.2.1.1 graphics-device**

**port**

##### **5.2.1.2 network-device**

#### **5.2.2 network**

#### **5.2.3 display-setup**

**vector**

##### **5.2.3.1 user**

##### **5.2.3.2 display**

**physical**

**virtual**

## **6 XSLT Konfigurationsgenerierung**

### **6.1 Beispiel**

## **7 Editor**

### **7.1 Funktionen**

### **7.2 Umsetzung**

## 8 Evaluation



## 9 Diskussion

## **10 Ausblick**

## **Literaturverzeichnis**

## **Danksagung**

Die Danksagung...

## **Erklärungen zum Urheberrecht**

Hier soll jeder Autor die von ihm eingeholten Zustimmungen der Copyright-Besitzer angeben bzw. die in Web Press Rooms angegebenen generellen Konditionen seiner Text- und Bildübernahmen zitieren.