DOI: 10.22032/dbt.49324

XRchitecture – Remote Control of Virtual DMX/Art-Net for a Real Light-Console

Tom Milter, M.Sc., Anke von der Heide (HAW Hamburg), Lukas Runge, Lukas Riffel (BHT), Lukas Lobmann (HAW Hamburg), Clara Grebenstein (TU Kaiserslautern)

Hamburg University of Applied Science, Research and Transfer Center Digital Reality, Am Sandtorkai 27, 20456 Hamburg
Beuth University of Applied Sciences Berlin, Event Technology and -Management, Haus Beuth, Luetticher Str. 38, 13343 Berlin
Technical University Kaiserslautern, Computer Science, Gottlieb-Daimler-Str., 67663 Kaiserslautern

Abstract

The XRchitecture event platform serves as a pilot project to explore hybrid event solutions. The aim is to create an experience for multiusers to attend an event not only on site, but also in virtual space (VR). This could result in event formats where part of the audience participates on site and another part virtually. The sub-project presented here, a virtual light editor, enables the simultaneous manipulation of light and media architecture installations via the same real interface, a light console into both worlds. In order to test a hybrid event format, a historical facade at the Carl-Bosch Gymnasium in Ludwigshafen was illuminated in cooperation with students from the Beuth University of Applied Sciences Berlin, while the transmission into VR was controlled and monitored via servers at Hamburg University of Applied Science (HAW).

Index Terms: Cross Reality, Hybride Event, Light Control System

1 Introduction

The XRchitecture event platform creates new ways of communication by interweaving real performances with virtual multi-user events. Any event can be experienced simultaneously in the real and virtual world. Artists can broadcast their performances live in both worlds. Synchronized manipulations of performances for hybrid events become possible via common interfaces for controlling sound, light and video.

Since 2020 and most likely 2021, we live in a reality where social events, festivals, concerts and conferences are cancelled and cannot be easily repeated. Clubs and other nightlife venues will remain closed indefinitely. This poses social and financial challenges for visitors as well as artists, event organizers and their teams. With the XRchitecture project, we wanted to find an answer to this and immersed ourselves from our original work for real events into an unknown virtual one.

This paper presents a sub-project of the XRchitecture platform, a virtual event lighting toolkit that converts DMX to Art-Net and optimises the data rate for a fast connectivity via the internet. An existing solution for Art-Net integration in Unity was adapted to control virtual luminaires with a light-console (GrandMA) in the local network. During the process, an editor was created in which virtual luminaires can be freely placed. These luminaires can be rearranged in the virtual space until shortly before the event. At the same time, the editor serves as a control room to monitor both virtual and real event formats.

In March 2021 the editor was tested in a hybrid event format. Therefore, a historical facade at the Carl-Bosch Gymnasium in Ludwigshafen was illuminated, while the transmission into VR was controlled and monitored via servers in Hamburg. The test event was developed in cooperation with students from the Hamburg University of Applied Science, Beuth University of Applied Sciences Berlin and the Technical University Kaiserslautern.

2 Introduction to DMX and Art-Net

The “DMX” Protocol is a standard for wiring Lights and their controllers and officially named “Entertainment Technology—USITT DMX512-A—Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories” from 2004, and renamed "E1.11 – 2008, USITT DMX512-A" in 2008 [1].

It functions as a “Controller/Responder” configuration, where, for example, a light-console is the “Controller” and one or more “Responder” devices are connected to it. Such devices could be dimmers, fog machines, motors for kinetic installations, even more complex lights such as moving heads, etc. They can be wired in a daisy chain (connecting the next device to the one before) instead of running a wire to each individual lamp.

The DMX standard allows to control 512 parameters per DMX-Output (also called “Universe”). Each parameter is bound to a “DMX-Channel” and can control for example a dimmer or a fog machine. The resolution of a channel is 8 bit, enabling 256 steps with 0 to 255 values [1].

More complex controls, colours for example, need more than one channel (e.g. red green and blue). This means that complex lamps like Moving-Heads with Gobos, Colour Wheels, and two axes for movements will need many channels. Therefore, bigger Consoles have multiple DMX-Outputs for multiple universes, and even bigger ones can transmit “Art-Net” via an ethernet-port.

Art-Net is used as a protocol to transmit DMX protocols over UDP (User Data Protocol) into an IP-network. Since the release of Art-Net III in 2011, it has been possible to transmit up to 32768 (!) universes over sufficiently fast networks, typically private local area networks. The Art-Net specification reads: “The use of class A addressing is allowed within a closed network. It is important to ensure that Art-Net data is not routed onto the internet” [2].

In the case of the XRchitecture, it is not possible to stay in a local network environment because end-users devices, e.g. VR-hardware is distributed all over the world. Therefore, it becomes necessary to capture the data, repack it and transmit the signals via the internet.

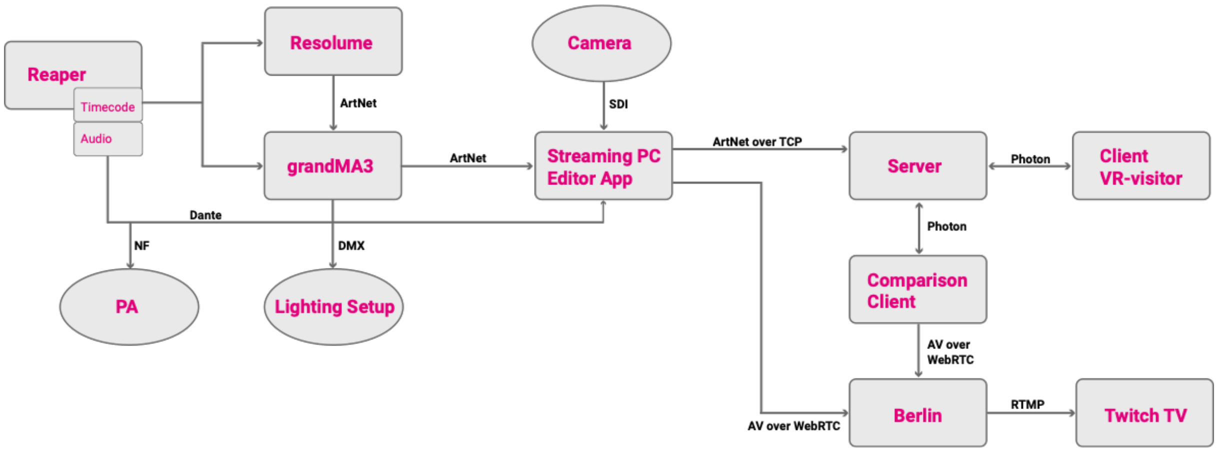
3 Technical Concept of a virtual light-editor

To cope with this problem an editor/control-room software was developed. It receives control packets via the light-console network and sends them to the platform's system. The editor also functions as a light-show visualizer for designers. An additional operator can engage as a controller to receive diagnostic data, while configuring the data rate or other parameters in real time. In order to develop such an editor it becomes necessary to access Art-Net packets to be able to transmit them. To send them via Wi-Fi it becomes necessary to reduce their size.

3.1 Access to Art-Net

Accessing the Art-Net packets becomes possible by linking a device to the Art-Net network. However, in a real lighting setup this can be anywhere in the network but it is recommended to place the streaming PC close to the light-console, since it can be also used as a visualizer of the VR-world.

Issues arise when the Art-Net network uses the “Unicast" mode. In this mode the packets are only sent to their destination and the streaming PC will not receive them. While newer versions of Art-Net recommend this mode [2], on most light-consoles it can be switched to "Broadcast" to allow each device to receive all packets.



Graphic 1: Streaming Scheme

To support "Full Package Capture" an additional device is needed. This would allow it to capture all data, even in "Unicast" mode.

3.2 Connection and Repackaging

As soon as the packets are received by the streaming PC a C# library (pre-build toolset for developers), to make these packets usable for the editor software, is used. The packets are converted to an object in order to be understood by the Unity-Game Engine [3], [4].

Because of the unstable nature of the internet and especially Wi-Fi (used by the Oculus Quest VR-glasses) we opted to create TCP-packets to transmit the packets over the Internet. To further cope with this challenge, a TCP-server was set up. The idea was to make it easy for the user to connect to the server and therefore to the content. Unfortunately, on the user-side a simple "read what you recieve" function did not work. TCP-packets are sent as a byte stream [5] and the fast and simple Unity implementation started to randomly split the packets. These split packages could not be read by the virtual DMX-controller. To circumvent this problem, a buffer was implemented and filled with data until a whole DMX-packet was received. The packets were sent in a json format that extended the package to a "human readable format". Unfortunately this package almost doubled in size.

But this split version became easily detectable through a defined beginning and ending of the packet ("and"). Additionally the buffer was able to search for these apprentices and to reconstruct the packet. This worked in theory, but sometimes the buffer was not cleared correctly or parts of a packet got lost, causing the connection to freeze until the packet was received again (TCP function) [6]. This caused the lamps to stand still until the connection was re-established. This happened mainly when the connection was unstable or the performance of the user was too low.

3.3 Performance

In the later described test scenario several "Facebook Oculus Quest VR-glasses" were used as receiving devices. These Android based, head mounted displays

(HMD) use “mobile phone-hardware”. This means, they are very limited in their processing power compared to a regular computer, while all of the processing power is built into the device. Additionally they receive data via WLAN or the mobile-data connection.

As soon as the Art-Net packets are received by the HMD the Unity client creates an array from it. This array is then parsed, and in case a virtual device (e.g. lamp) exists at the address, the value is used to perform an action. The action depends on the configuration of the DMX-channel within the virtual device. E. g. a rotation value is calibrated into the maximum and minimum range and the lamp is given a target to move to.

> ***Important note:***
> *The movement of the virtual lamp cannot be changed immediately. While the real lamp has a maximum acceleration and turn-rate, the position of the virtual lamp is calculated by each frame until the lamp meets its target point. The destination can be changed as soon as a new Art Net packet is received, and the movement to the new destination is also changed. In the proposed framework this feature can be disabled. This enables scenes with only one lamp that completes a 180° turn in less than 1/30 of a second.*

Using only a limited number of lamps (in the test two universes of parameters) the headsets have enough processing power to go through each received packet and to control all parameters. For larger festival installations often 40 or more universes are used [7]. This means that in a hybrid event setting, a few real but 100 times more virtual lights can be controlled simultaneously.

Therefore, optimizing the packet transmission and handling is the main goal to make a smooth operation possible. In this regard two existing Unity frameworks could help to optimize XRchitecture system further: the VideoCodecs a huge data saver for video-transmission, and a multiplayer game framework.

4 Visual Concept and Implementation for a Test Event

To test the editor a hybrid cross-reality test event, with a storytelling for the facade of the Carl-Bosch Gymnasium in Ludwigshafen was developed within one week. Students from different universities met at their former school (Carl-Bosch Gymnasium in Ludwigshafen) and revived their old school-event-workshop. Due to the closing of the school during the pandemic they could turn the whole building into a five minute looped light show.

The show was looped for 30 minutes and additionally streamed into Twitch TV. There, both events, the real and the virtual, were transmitted simultaneously in two live images placed next to each other.

An intensive study of the building conditions and possibilities for the placement of lighting fixtures was necessary before the creative work could begin. First, Google Maps images were examined and 3D data of the building was extracted (Fig. 1).

Afterwards a 3D model of the building was created based on accurate dimensions from 2D plans (Fig. 2). This was then used to place the virtual lights for the virtual event in Unity, as well as in grandMA3 for the visualization in the 3D view.

Fig.1: Perspective of a drone image, similar to a Google Map

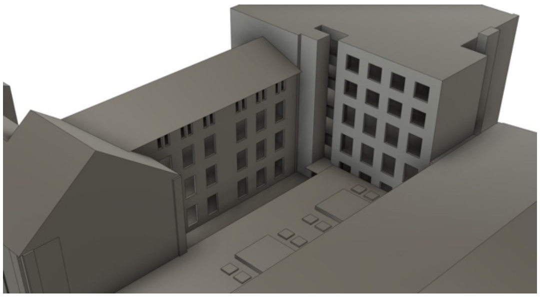


Fig. 2: 3D model of the school building

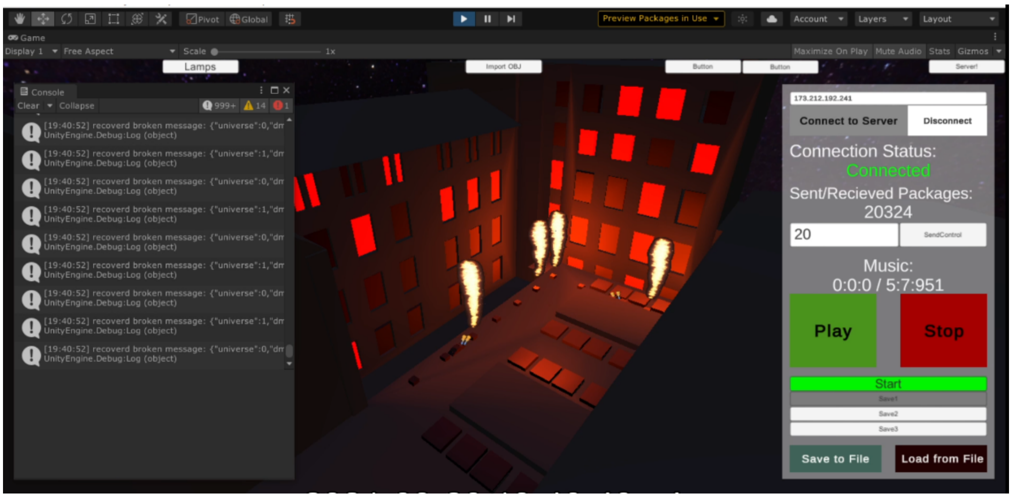


Fig.3: Screenshot Lighting Editor and Control Data

For the virtual setup the same 3D modeled building, as described above, was used in the lighting-editor. As shown in the screen shot (Fig. 3), in the middle part of the editor window, 3D modeled lamps can be placed freely. The functional characteristics of the luminaire are also added here. On the right side of the editor the system functions, like connection to the server, lighting-console, the synchronization to music as well as the play, pause and stop buttons are placed. The control functions for the operator of the live and virtual light-show are at the left side of the window.

Playing with the conditions of the building, e.g. backlighting of windows and direct lighting of the façade, was the main focus. The concept was realized with 65x LED Pars/Bars/Floods, 4x Martin Mac 2000 Profile II, 10x TBF Spraymaster, 2x MagicFX Flamaniac and a control mix of grandMA3 onPC CommandWing, Resolume Arena, Reaper, Midas M32C + DL16 (Fig. 4). The story started with a slow lighting sequence followed by a harder rhythm of red and yellow lights and flamethrowers, a middle cheerful section that ends in a crescendo.

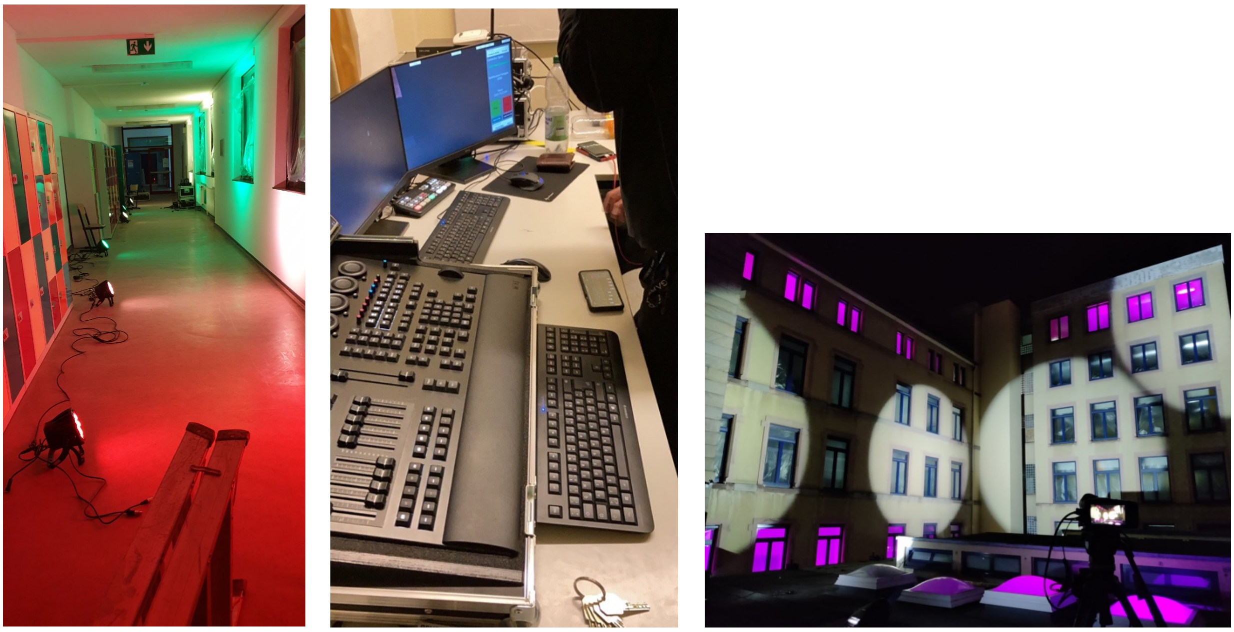


Fig. 4: Lighting setup, illuminating and programming of the real show

5 Test-Event and Findings

The hybrid test-event showed clearly that the predicted challenges really happen, while using the editor in a real event environment, mainly the connection and data transmission issues. While most users had a good experience, others had problems. This was not only communicated directly by the user, but also shown in the diagnostic packages including “Connection Time” and “Packages Received” data, sent by the HMD’s .

However, the collected data revealed that most failures were rather random and only some had a bad connection, most probably because of their own WIFI connection. Overall the 25 test users send 2.000.000 packages and 100.000 diagnostic packages. This makes it clear that it is possible to send more than 100 packets per second, but most light-console manufacturers use 30 packets per second. It is therefore not recommended to go below 25 packets per second, as the transmission becomes asynchronous.

The test event also allows insights into the usual workflow of real-life events. For example, changes to DMX addresses or fixture positions are not uncommon in the real world until a few hours before the show. However, this is no longer possible in an app that is downloaded by users in advance. In order to enable the event technology to run naturally, an update option has been created for the end devices. This way, the editor can send an update to the end devices even five minutes before the show starts.

6 Conclusion and Outlook

With this flexible modular-system, the XRchitecture platform can be used for various event formats. Exhibitions, conferences, concerts, light or video mapping shows, club nights or festivals are possible. The lighting editor is an essential part of this platform and is used for each of these events.

It allows technicians to use their usual equipment, such as light-consoles, and follow their usual workflows to set up an event. In addition, the editor allows last-minute changes to the virtual setup and forwards these to the visitor devices as an update of the application.

After this proof of concept, a new version of the editor will include the same possibilities for video mapping and sound equipment. An organizer tool will serve the responsible party as a plug-and-play control-centre for the virtual event.

Additionally, the control-centre mode could make it possible to use the VR-system to control a real event remotely.

7 Bibliography

[1] Ruling, K.G. and Nix, R.J. 2018. *American National Standard Entertainment Technology — USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*. Entertainment Services and Technology Association (ESTA).

[2] Award, I. 2016. Art-Net 4 Specification for the Art-Net 4 Ethernet Communication Protocol Winner of the PLASA 2016 Innovation Award. (2016).

[3] Hironori Sugino, “sugi-cho/Art-Net.Unity: Art-Net library for Unity(C#). Based on An Art-Net library for C# and VB.Net developers. Based on the ACN project codebase.” https://github.com/sugi-cho/Art-Net.Unity (accessed May 29, 2021).

[4] Mike James, “MikeCodesDotNET/Art-Net.Net: An Art-Net library for C# and VB.Net developers. Based on the ACN project codebase.” https://github.com/MikeCodesDotNET/Art-Net.Net (accessed May 29, 2021).

[5] Suznjevic, M. et al. 2014. Analyzing the Effect of TCP and Server Population on Massively Multiplayer Games. *International Journal of Computer Games Technology*. 2014, (2014), 602403. DOI:https://doi.org/10.1155/2014/602403.

[6] G.Fiedler, UDP vs. TCP | Gaffer On Games: 2008. *https://gafferongames.com/post/udp_vs_tcp/*. Accessed: 2021-06-04.

[7] How does a big DMX setup look like (event/festival) : lightingdesign.: 2020. *https://www.reddit.com/r/lightingdesign/comments/gjv3s5/how_does_a_big_dmx_setup_look_like_eventfestival/fqn90mw/?utm_source=share&utm_medium=web2x&context=3*. Accessed: 2021-06-04.

8 Illustrations, Graphs and Photographs