

Concept for KWlive Architecture

Contents

1	Concept for KWlive Architecture	3
1.1	Preface	3
1.1.1	Use for other projects	3
1.1.2	Users	3
1.2	Introduction	4
1.3	Components	4
1.3.1	Realizer	4
1.3.2	Keyworx server	5
1.3.3	Traffic Control Centre (TCC)	5
1.3.4	GUI	6
1.3.5	Live Inputs	6
1.3.6	Streaming server	6

Chapter 1

Concept for KWlive Architecture

1.1 Preface

KeyWorx is a software platform that aims to enable developers to invent, develop, integrate and deploy applications with multi-user/multimedia features. A common aspect of these applications is that they aim to empower users to be the (joint) creators of multimedia content. Multimedia content may be of any form, dependent on the application services reused and/or developed for that application. Examples of services range from simple file uploading/publishing through newspaper-type of multimedia document editing, drawing services and real-time audio/video sharing.

KWlive is the (working) name for the set of applications within the KeyWorx platform suiting "live" multi-user audio/visual performance. The KWlive project is the successor of the original Keystroke project, that may also be known under the name "KeyWorx Classic" or just "the Realizer and the Patcher". The original application targeted to the media/performing artist user group began development in 1998 and completed development in February, 2004.

New thoughts about the KeyWorx architecture and the protocols and as a result new requirements of the KWlive components were the main reasons to start development all over again. Within the framework of the MultimediaN project, research has been conducted for the development of KWlive. For this research reflected user experiences will be taken in account. This document is a summary of the concepts for the KWlive architecture.

1.1.1 Use for other projects

Artist/programmer participation in the project will bring added value to the platform and increase its overall use through third party development of interfaces. Artist participation is vital to the continuing innovation of the platform. It is one of the most important aspects of open sourcing and needs support from the entire KWlive project. It is very important for the R&D activities and goals defined in the MultimediaN project. And last, it will be used in the Sensing Presence projects.

1.1.2 Users

Current KeyWorx user group and beta-testers need technical, educational and ÒemotionalÓ guidance through the transitional process. Some will immediately begin to design personal interfaces, others, with limited programming skills, will want to use an Òout of the boxÓ interface. The main goal of the transitional period and the beyond is to create an awareness among old and new KWlive users that they are responsible to maintain and enhance their community. Waag Society will assist and support by maintaining

mailing lists, offering some tech support and providing dissemination events and educational workshops, for example, the public dissemination event at the Píksel05 conference in Bergen, Norway in October, 2005.

1.2 Introduction

This document describes the upcoming KWlive version and the main differences with the "Classic" version.

One of the big changes is the introduction of a new component, the TCC. Instead of having both the Realizer and the Patcher connect to the server, now only the TCC communicates directly with the server. In theory the Realizer, TCC and GUI can all be running on different machines as long as they are on the same subnet. This allows for a much simpler, yet more powerful configuration. You could even have multiple Realizers for one user. Or multiple users connected to the same Realizer.

Another important change is the way live inputs are dealt with. The only way to get live data into the Realizer is via Open Sound Control. As a result of this approach some extra tools are needed to convert data from live inputs to OSC. One of the advantages is that it no longer matters whether you are using midi, a joystick or even other software (like max, PD) to send this data. All data is treated the similar.

1.3 Components

The whole architecture consists of more components than the old architecture. The idea is having each component to only do one thing, and do it well. This makes configuration a lot easier and more flexible. Each component is described in this section. The illustration below shows how all components interact.

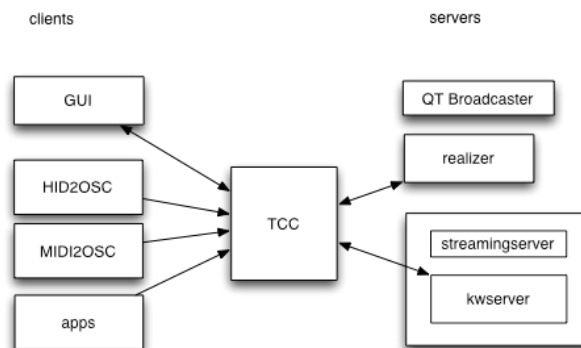


Figure 1.1: KWlive architecture

1.3.1 Realizer

The Realizer is responsible for rendering audio and video. The new version will be less complex than the previous version. It is not connected to the KeyWorx server and is not responsible for downloading files. Instead, it acts as a server which a Patcher can connect to via TCP. Because it is a server, it should also be able to run as a background process and send its output to a media stream.

- Open Sound Control:

OSC plays an important part in the Realizer. In fact, there are no more MIDI or joystick modules. Instead, the Realizer only listens to OSC. To use any of these interfaces an extra piece of software is



used (see **Live Inputs**(p. 6)) in combination with a generic module for receiving OSC in the Realizer. The reason for this is that it is now much easier to distribute live data, because the Realizer doesn't need to have any knowledge of other users. That part is handled by the **Traffic Control Centre (TCC)**(p. 5).

- **Multiple outputs:**

Instead of having a single output window, it is possible to have multiple outputs. These can be windows, or streams or both. There is a compositor module that renders the image output, that can be connected to any number of output modules. In a multi-user setting this means that when a patch contains one or more image outputs, it is allowed for each user to configure those outputs differently (window, stream, ignore).

- **Muti-channel sound:**

Just like it is possible to have multiple image outputs it is also possible to have multiple audio outputs. Each audio output is a stereo channel and there can be any amount of audio outputs in a patch. Each user can configure for each output where to send the audio to (mix it all down to 2 channels, send it to multiple outputs, send it to a file?).

- **Media:**

All media files can come from anywhere (not only the media folder). All media are referenced to by an URL, and downloaded by the TCC if needed.

- **Stability:**

The new Realizer is currently being completely rewritten, a lot of effort is going into making it more stable than the previous version.

1.3.2 Keyworx server

The KeyWorx server is basically the same as before but it should now also be possible for media to be associated with a space.

1.3.3 Traffic Control Centre (TCC)

The TCC is a new component that has several functions. It's main function is to handle all communication between all components.

- it talks to the Realizer
- it talks to the KeyWorx server
- it makes sure all live data gets sent to the right place.
- file sharing, it makes sure media files are downloaded from the server if necessary. Bittorrent will probably be used for the file sharing.

Because the TCC is the only component connected to the server, it is in fact the representation of the user on the server.



1.3.4 GUI

The GUI is both interface for patching as for configuring the Realizer, live inputs etc. It connects to the TCC which in turn makes sure that other users in the "space" receive all patch commands. The GUI is from a developer's point also less complex than before because most functionality is already in the TCC. Therefore it should become much easier to build your own GUI. An "out of the box" GUI will probably be made in Director of Flash. Some functionality that previously belonged to the Realizer as well will now belong exclusively to the Patcher i.e. saving and opening patches. Also there doesn't necessarily need to be a one-on-one relation between how the Realizer sees the patch and how the GUI sees it. There might be macro's that can expand into multiple modules for instance. There might be a joystick module even though the Realizer only has an OSC input. It would then be the responsibility of the Patcher to start the external tool that reads the joystick and sends its data as OSC.

1.3.5 Live Inputs

- live input data from devices like joysticks, midi and mice will be converted to OSC by separate applications i.e. hid2osc. The GUI will be responsible for launching (whenever needed) and configuring these live input applications. The OSC data is sent to the TCC that makes sure that the Realizer and other users(via the KeyWorx server) receive the data.
- live video is sent by Quicktime-broadcaster and distributed by a streaming server.

1.3.6 Streaming server

The streaming server (Darwin streaming server?) is responsible for distributing streaming media to all users (Realizers). It will probably run on the same machine as the KeyWorx server.

