TINE VIDEO SYSTEM: PROCEEDINGS ON REDESIGN

Stefan Weisse#, David Melkumyan†, DESY, Zeuthen, Germany
Philip Duval‡, DESY, Hamburg, Germany

Abstract

Experience has shown that imaging software and hardware installations at accelerator facilities needs to be changed, adapted and updated on a semi-permanent basis. The past TINE-based Video System [1], initiated at PITZ [2], has undergone a thorough redesign after years of operation and upgrading in order to extend interoperability and cope with new challenges while still staying flexible. Emphasis was placed on flexibility, avoiding redundancies, good documentation, component-based architecture, multi-platform capability and ease of use as well as reuse. This contribution will show the current status of the redesign as well as the near and far future outlook. The main focus is put on the Java-based TINE ACOP Video Bean and its integration into jDDD and COMA, PNG file format support, core applications and services as well as universal slow control for cameras. Moreover, the Video System acts as a basis for advanced high-level software applications such as semiautomatic Emittance Measurement Wizard (EMWiz). Although the outlined Video System implementation is integrated into TINE control system [3], it is modular enough so that integration into other control systems can be considered.

INTRODUCTION

The origin of the featured video system is the **P**hoto **I**njector **T**est Facility **Z**euthen (PITZ). PITZ is a test facility at DESY for research and development on laser driven electron sources for Free Electron Lasers (FEL) and linear colliders [4, 5].

Currently, the TINE Video System is undergoing a redesign, which was started in 2008 [6]. At that time the rapid rate of evolution at PITZ and consequently the PITZ Video System exposed several weak points of the original design [1], nowadays frequently referred to as Video System 2 (VSv2). Started in the year 2002, it was never designed to grow as big as it finally got. The ensuing redesign of the Video System has now given rise to what is known as Video System 3 (VSv3). It is necessary to mention that fundaments of VSv2 date back to the year 2000, giving in total more than 9 years of experience on video controls.

As of October 2009, certain aspects of the redesign are in operation whereas others are currently being implemented or are still in a planning stage. At PITZ, the video system was and is a central tool for monitoring, optimizing and measuring the electron beam and its main performance characteristic, the minimum transverse emittance at 1 nC bunch charge.

stefan.weisse@desy.de
† david.melkumyan@desy.de
‡ philip.duval@desy.de

STEP BY STEP TRANSITION

Certain finished parts of the Video System 3 were installed at PITZ in 2008 and 2009. The initial upgrade offered support for JAI Pulnix Gigabit Ethernet cameras which were added at PITZ. Cameras from JAI Pulnix were necessary to be installed in order to improve measurements of laser beam characteristics. The new front-end server required for this was implemented as Small Grabber Part (SGP) component. To provide backward compatibility, a so-called GrabServer 2 compatibility layer, pre- and test-stage of the Core Provider component, was set up between SGP for JAI Pulnix and an existing Video System 2 installation.

As a second step, existing legacy slow control was exchanged by a more flexible Java-based universal slow control solution, explained in detail later on. In addition, a special installation of Video System 3 components was done for colour imaging of the photo cathode.

CHANGES TO FRONT END

The Small Grabber Part (SGP) is in brief the front-end server component of VSv3 which takes images from hardware via API, puts them into a defined frame container and provides VSv3 transport layer output interface. It has now been extended to receive update notifications from a central timing source which multicasts event numbers throughout the control network. TINE features mechanisms to provide low-latency delivery of such time-critical data, making this scheme possible to be implemented. Having video images tagged with a unique event number, next step is going to be performed: storing video images into DAQ/Archive System.

In addition, SGP servers were extended to provide simplified access to camera slow control properties, exposed by hardware API which is only accessible from within SGP component. This is necessary groundwork for a universal slow control solution. In addition, the flexibility of the SGP server configuration for certain SDK APIs (e.g. Prosilica and JAI) was improved.

CORE PROVIDER

A major design goal of Video System 3 is to split existing monolithic functionality into flexible pieces. SGP, unlike its grandfather GrabServer 2 of VSv2, contains just the interface to hardware/API and does not perform any device-independent pre-processing of video. This will be performed in the Core Provider component instead.

As a first stage, the Core Provider contains an input interface where the video images are taken over from SGP connected in front. After that, each image travels through a pipeline of image and metadata pre-processing tasks which can be enabled, disabled or tuned on a permanent or temporary basis. After pre-processing tasks have been performed, the resulting video image(s) can be requested on the output side. Three different image output feeds are foreseen. By splitting video data delivery into different modes of operation (live vs. data taking), special requirements for each mode can be fulfilled, which are mutually exclusive if provided in just one feed.

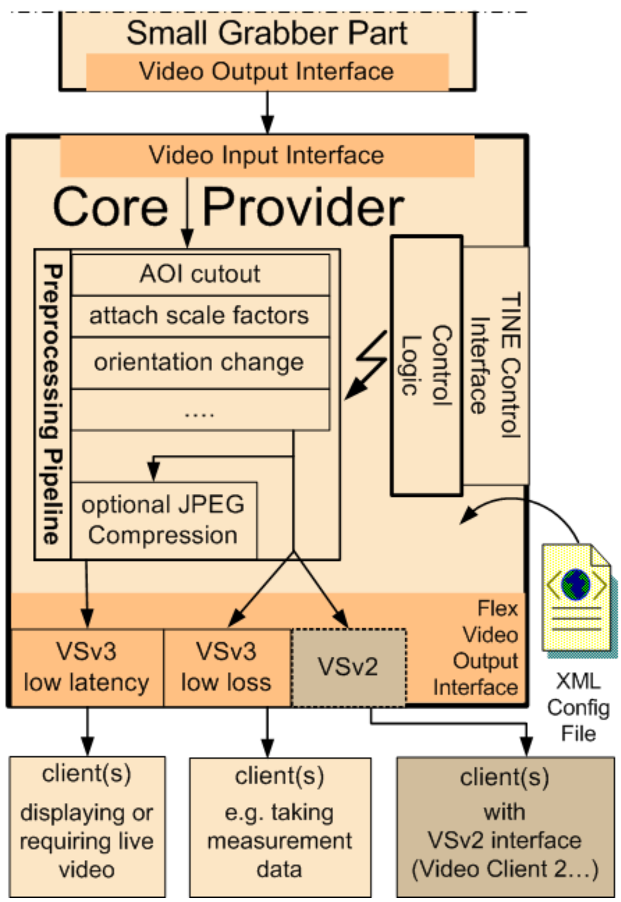

Figure 1: Core Provider schematics.

STANDARD IMAGE FORMATS

For historical reasons, VSv2 used proprietary, limited file formats to store images and image sequences. VSv3 will change this. The PNG file format was selected for single image and metadata storage. Certain characteristics of PNG make it the format of choice for image data storage to file:

- open-source, open standard
- widely used in industry
- lossless compression of image bits
- greyscale and RGB image support
- good multiplatform C library, native Java support
- extensible by optional chunks and text tags
- no patent issues, community maintained
- LGPL, fair licensing terms of C library

Extensible optional chunks and text tags are chosen to contain image metadata in a way that is much more flexible than it was before. Practice shows that image bits are not complete without appropriate metadata, because it contains indispensable additional information.

Image sequences, proprietary in the past, are foreseen to be stored as PNG in a plain and simple ZIP directory container (PNGZIP).

Once having integrated this new way of storing images and image sequences into VSv3 applications, it is expected that exchanging images with other tools frequently used by physicists like MatLab, ROOT, LabVIEW etc. will become much easier. The same is true for the export of video images in order to use them for presentations.

JAVA VIDEO ON CLIENT-SIDE

In 2008, the completion of the Java ACOP video bean has proven that Java GUI clients receiving and displaying live video with satisfactory performance are possible to be implemented [6]. The bean is now ready to be used as standalone application, inside Java DOOCS Data Display client GUI toolkit (jDDD) [7, 8], as ACOP bean inside of Java applications, and via ACOP beans-based Component Object Manager (COMA) class [9]. Progress has been made on interoperability of DOOCS camera server [10] and TINE-based Video System. Based on user feedback, bean has been extended while sharp corners of original design could be identified and made round. Surprisingly, performance figures were also improved by this. Last but not least, the integration of an interface to intermediate video analysis server, initially created by CosyLab for video readout and control of Petra III pre-accelerators, was performed.

UNIVERSAL SLOW CONTROL

Universal slow control (USC) is meant to replace all slightly different but comparable camera slow control solutions by a more general approach of slow control. Functionality is divided into server and client-side. Communication between server and client is performed by TINE. A dedicated specification text has been created showing details on how the control system property interface needs to look like. USC was made in a way that implementation of any slow controllable device can be considered. To provide stable access to special camera characteristics which might be differently accessed in each distinct piece of hardware, a so-called generic slow control (GSC) was invented. This becomes important if interfacing video technology with (semi-)automatic measurement clients.

The implementation has been done in native Java, which makes it easily possible to use the server and, much more important, the client on all available platforms/consoles. The only dependency is that Java6 must be available. The USC client is a Java GUI application meant for general access to slow control properties. Effort has been put to avoid redundancy in configuration and settings. Nearly all information that the client needs during runtime it gets by examining USC server's TINE properties.

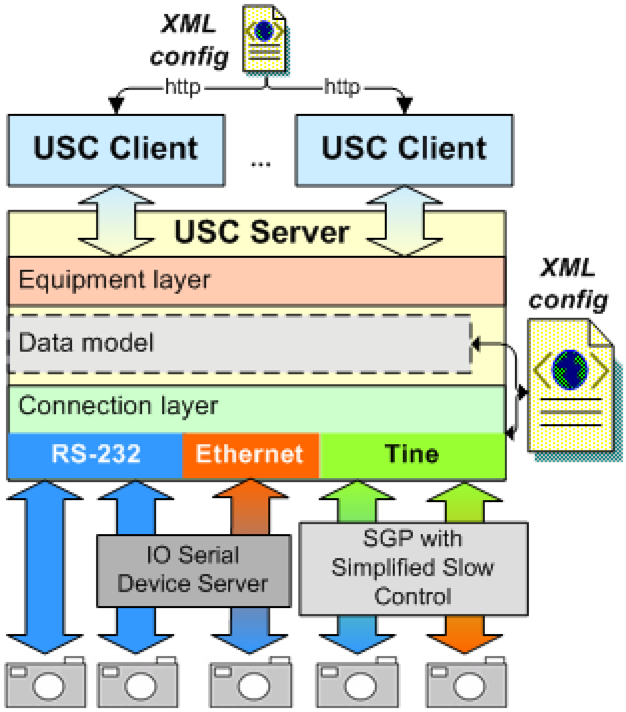

Figure 2: Schematic diagram of the USC solution.

CURRENT STATUS OF REDESIGN

The ACOP video bean is already in use at PITZ (on a colour camera setup), HASYLAB (Petra III beamlines) and at Petra III pre-accelerators (integrated into a dedicated video application called SchirmMonitor). As jDDD and COMA integration is ready, use of the Video bean as a video display component should be considered. The Java-based universal slow control is now installed at PITZ. There, the USC client is made available by Java Web Start. USC setup will be extended step by step until all camera models used at PITZ are controllable by it.

The PITZ Emittance Measurement Wizard (EMWiz), a semiautomatic emittance measurement client application, strongly interfacing with the machine control and video system, is used frequently at PITZ. The outcome is that the final emittance value, key characteristic of PITZ, can be measured much quicker, requiring less attention and user-interaction by operators. This is a big step forward in comparison to the emittance measurement solutions which have been used before.

Important characteristics when it comes to video data transfer are latency, performance and adequate network bandwidth. Effort has been made to provide strong figures on Gigabit Ethernet using TINE data transport. Looking closer, this required more work than expected. Current tests have been successful transferring stable rates of 30 MB/s of multicasting video frames (each about 2.8 MB) via TINE on Gigabit Ethernet with less than 1 frame drop per minute. Available bandwidth is not the only limiting factor. Frame rate and frame size also need to be considered, too. The frame rate was also verified to work from 1 up to 30 Hz. The frame size can be guaranteed to work from a few kilobytes to 3 megabytes per frame. If very special state-of-the-art setups are tuned, 50 MB/s and 45 frames per second are possible to reach with less than one frame drop each minute.

When it comes to standard image formats, the specification of saving VSv3 video frames (including header) to PNG as well as resulting reference implementation for C++ and Java are finished. PNGZIP exists as draft-type specification and prototype tests of it have been successfully finished. Although PNGZIP it is still in early phase of development, it is not expected that roadmap will be changed significantly. Several SGPs have been finished so far:

- Prosilica GigE-Vision cameras (Prosilica API)
- JAI Pulnix GigE-Vision cameras (JAI SDK)
- Dalsa PCVision analogue PCI framegrabber card
- National Instruments IMAQ interface
- Microsoft DirectShow API
- Animation file playback from disk

PERSPECTIVE

The highest priority is finishing the core components of VSv3. DAQ/Archive integration of video streams, considering Extensible Metadata Platform (XMP) for storing metadata into PNG files and completing design, API of PNG VSv3 images and PNGZIP image sequences is in the pipeline already. Unleashing the full power of Gigabit Ethernet for video transport, continuing to spend effort to switch to Java on GUI client side and providing flexible APIs on many platforms are certainly worth pursuing.

REFERENCES

[1] S. Weisse et al., "Status of a versatile Video System at PITZ, DESY-2 and EMBL Hamburg", ICALEPCS 2007, Knoxville, TN, USA.
[2] http://pitz.desy.de.
[3] http://tine.desy.de.
[4] C. Boulware et al., "Latest Results at the Upgraded PITZ Facility", FEL 2008, Gyeongju, S. Korea.
[5] F. Stephan et al., "New Experimental Results from PITZ", LINAC 2008, Victoria, British Columbia, Canada.
[6] S. Weisse, "TINE Video System: A Modular, Well-Defined, Component-Based and Interoperable TV System Undergoing A Redesign", PCaPAC 2008, Ljubljana, Slovenia.
[7] E. Sombrowski et al., "jDDD in Action", ICALEPCS 2009, Kobe, Japan.
[8] http://jddd.desy.de.
[9] G. Grygiel, V. Rybnikov, "DOOCS Camera System", ICALEPCS 2007, Knoxville, TN, USA.
[10] R. Bacher, P.K. Bartkiewicz, P. Duval, „The Runtime Customization of Java Rich Clients With The COMA Class", ICALEPCS 2007, Knoxville, TN, USA.