An Overview of The Khronos Group and OpenML™

This white paper describes the motivation for the formation of the Khronos Group and provides an overview of the OpenML digital content creation and playback environment that the Khronos Group is developing.

Special Interest Group (SIG)

The following companies are represented in the Khronos Special Interest Group:

3dfx Inc.
3Dlabs Inc.
ATI Technologies Inc.
Compaq Computer Corp.
Discreet Logic Inc.
Evans and Sutherland Computer Corporation
Hewlett-Packard Inc.
IBM Inc.
Intel Corporation
Matrox Inc.
S3 Inc.
SGI
Sun Microsystems, Inc.

Contributors

Roi Agneta	Evans and Sutherland Computer Corporation
Matthew Allard	Intel Corporation
William Clifford	Intel Corporation
Randi Rost	3Dlabs Inc.

Disclaimer and Copyright Notice

Table of Contents

1. INTRODUCTION.. 4

2. MOTIVATION 5

2.1 Example of OpenML™ Standards 5

3. FORMATION OF THE KHRONOS GROUP 8

4. DIGITAL CONTENT CREATION- A BRIEF HISTORY OF VIDEO GRAPHICS....... 9

5. AN OVERVIEW OF OPENML™ 12

5.1 Synchronizing Audio, Video and Graphics 13

5.2 kDM Features 14

5.3 OpenGL® Features 15

5.4 Video Back-end Control 16

5.5 OpenML™ Application Profiles 16

6. SUMMARY 17

List of Figures

Figure 1. Early Graphics Architecture 9

Figure 2. Second Generation Architecture 10

Figure 3. Current Generation Architecture 11

Figure 4. Programmer's View of the OpenML™ Environment...................................... 13

Figure 5. Data Flow in the OpenML™ Environment.. 15

1. Introduction

The development of media authoring systems has evolved from early, highly customized, monolithic approaches. Today's systems are assembled from a diverse set of standard components. However, while the burden of hardware development has been eased, system-level software problems have been compounded, especially when the system is required to interoperate with other media devices.

The goal of the Khronos Group is to develop and manage OpenML™, a standard set of open application programming interfaces (APIs) for media content creation and playback.

It is the intent of the Khronos Group that OpenML will:

- Foster a cross-platform, cross-OS development environment. The group will drive open standards between platform, hardware, and application vendors to enable a seamless interoperability to customers for transparent migration of content creation and playback across a variety of platforms and devices.

- Enable integration and synchronization of Video, Audio and 2D/3D graphics to deliver compelling content through media-rich interactive applications.

- Enable hardware and software providers to produce a larger number of standardized, transportable, and compelling media products to be brought to market in a more timely fashion. This in turn will foster user acceptance and market growth with customers benefiting from a larger selection of systems, applications, and peripherals to choose from.

- Establish synergy to multi-purpose and re-purpose content for a variety of distribution mediums such as broadcast and the internet.

- Build on the strengths of OpenGL® and work with the OpenGL ARB to strengthen OpenGL.

2. Motivation

There are numerous examples of how establishment of industry standards has helped to accelerate market growth and acceptance of new technology. Industry standards exist for many technologies, such as 3D graphics programming APIs, web page programming APIs, network protocols, and high speed bus interfaces.

Standards are defined to establish a common ground for developers looking at a problem from two or more directions. For instance, a standard for 3D graphics programming benefits both 3D graphics application developers and 3D graphics hardware developers by defining a common interface that both sides can implement. When completed, a 3D graphics application written to this interface will run on any hardware that supports the interface. Conversely, hardware developed to support the interface can support any application written to the standard interface. In a similar manner, both computer manufacturers and peripheral manufacturers benefit from having a common interconnection standard and bus protocol. Finally, end users benefit from standards as the market grows and costs decline.

The members of the Khronos Group SIG believe that a standard is necessary to accelerate the development of both digital media hardware and application software. With the formation of the Khronos Group, and the development of the OpenML™ specification, the following example may be only a couple of years away, rather than 5 or 6 years away.

2.1 Example of OpenML™ Standards

Vacation Research using Multimedia Application

You purchase the 2003 edition of Encyclopedia Geographica. It is distributed on a DVD-ROM and comes with 1 year's unlimited access to the multimedia library on Encyclopedia Geographica's website. When the application starts, a spinning globe appears as a 3D texture-mapped sphere. The major tourist attractions for your upcoming vacation in south central Africa can be found by clicking on that part of the map. From the menu that pops up, you select Victoria Falls and watch a short video clip on the computer monitor. "Looks like an awesome place to visit," you think to yourself. Unfortunately, the low-resolution video clip on your monitor doesn't really do justice to the grandeur of Victoria Falls.

Multimedia Portability from Computer System to Home Theater

To get a better feel for what Victoria Falls is really like, you request the high quality presentation from the Encyclopedia Geographica website. You are asked for your language preference, and you select "English" from the menu of choices. Three- (3) files, an HDTV video clip and two- (2) CD-quality audio tracks, are streamed to your 6 terabyte hard drive across your 1 Gigabit/second internet connection. One- (1) of the audio files contains an English narration track and the other contains a surround-sound recording made at Victoria Falls. On your computer display, you bring up your multimedia device control application and select "Home Theater Video" as the destination for the playback of the HDTV video clip, then "Home Theater Audio" as the destination for the two- (2) audio tracks. (Incidentally, this device control application is completely independent of the hardware connected to your system and was purchased at the local computer store for $29.95.) It was able to query the devices connected to your system and develop a command and control panel unique to your setup without any intervention on your part.

Device Control and Media Synchronization

When you click the play button on your computer, the room lights are dimmed, the HDTV video clip is played on your home theater HDTV screen, and the two- (2) synchronized audio tracks are played over your THX[*]-equipped surround sound stereo system. As the camera pans across the panoramic vista of Victoria Falls, you hear the monkeys squawking behind you and the birds chirping overhead. You decide that Victoria Falls is definitely a place you have to experience in person on your upcoming vacation.

Multimedia Editing

Your parents are also planning to accompany you on your African vacation. Since their native language is French, you request the French narration track from the Encyclopedia Geographica website. You then use your media authoring application to splice together the audio and video clips of Victoria Falls, Mount Kilimanjaro, the Maasai Mara National Reserve, and Nairobi National Park that you captured. In between the clips, you decide to put a title page that includes the destination name and the dates that you propose for visiting each of them. This is done with a handy little title-generation shareware program that you downloaded from the web.

Interoperability of Multimedia

You turn on your video camera and record a live video greeting to your parents and an introduction to the media clip that follows. This is spliced onto the beginning of your multimedia masterpiece. The entire editing and recording operation has taken less than 10 minutes. You email your finished multimedia masterpiece to your parents and they are able to view and hear the multimedia presentation on their multimedia system that contains completely different hardware and OS and costs about 1/10th of yours.

This example demonstrates that the standards created by the Khronos Group SIG will allow companies such as Encyclopedia Geographica to produce their software, video, and audio, regardless of the hardware an end-user has set up at home. The user might have any combination of computer, OS, stereo system, and display device. Assuming all the pieces were OpenML-compliant, this scenario could be repeated in any household, in any country, and the Encyclopedia Geographica product would work right out of the box. The device control application, the media authoring application, and the title-generation software used in the example are similarly independent of the hardware that is installed on the system. A user is also able to choose from a large variety of audio input, audio output, video input, and video output equipment as they assemble the system that meets their needs, as well as the needs of their pocketbook.

Although this example describes several benefits of OpenML technology for the consumer market, OpenML is intended to simultaneously address the needs of both the professional content creation market and the consumer multimedia market. No API standard (or standards) prior to OpenML has been able to achieve this. OpenML is intended to be an industrial strength specification for industrial strength applications, as required by companies developing professional quality media authoring tools. The emphasis on quality, precision, and robustness will benefit the consumer as well, since features previously available only in professional products will also be available in consumer products. This includes features such as the quality of the playback experience, perfect synchronization of video/audio, tone reproduction, and ancillary data.

3. Formation of the Khronos Group

It is typical of advanced technology fields that the integration of groups of technologies into a coherent, cost-effective environment lags the development of the individual components. In the drive toward e-commerce and internet content delivery, the need for tools and a framework for creating content to meet the growing demands of the information age is sometimes lost. Observing this wild and woolly world (www), the founders of the Khronos Group recognized the need for an open, cross platform framework for the creation and playback of digital media content. The Khronos Group SIG was formed to develop a specification that addresses this need.

The rapid expansion of the Open Source model and the Linux operating system has amplified this need. A major issue in the Linux environment is the lack of cross-platform standards for multimedia application development. OpenGL® is the only truly open, cross platform standard in this area. For this reason, OpenGL is a key component of the OpenML specification.

A full multimedia solution, encompassing the integration and synchronization of 2D and 3D graphics, audio and video codecs, and I/O, is necessary. Whether working on a video production, developing a game, or constructing an interactive disc, creators are currently forced to use many different APIs and SDKs, from numerous vendors and organizations, to simply tie their multiple applications and hardware peripherals together. The Khronos Group will create a common specification for interoperability across platforms, operating systems, hardware devices and software applications.

It is the intention of the Khronos Group to produce the OpenML, Version 1 specification by the end of 2000, with the first product implementations appearing in the first half of 2001. The Khronos Group will create the spec and several sample implementations. These items will be licensed on a royalty free basis. Other entities will be encouraged to create their own implementations using their own business models.

Creating such a specification requires the input and expertise of a diverse group of organizations. This group evolved, over time, from a core of platform, application, and hardware vendors. SIG has 3Dlabs, Evans & Sutherland, Matrox, ATI, S3/Diamond Professional, IBM, 3dfx, and SGI as contributing members for graphics and video hardware technology. On the platform side, the SIG has the involvement of Intel, Hewlett-Packard, Compaq, SGI and Sun Microsystems. Finally, Discreet Logic is a content creation application provider.

To ensure that the efforts of the Khronos SIG align with other initiatives in the marketplace, SIG members are also members of organizations such as SMPTE, ISO/MPEG, the OpenGL® ARB, and the AAF Association. The SIG will maintain relationships with these and other industry groups where appropriate.

The Khronos Group is actively seeking contributing members to round out creation of the specification. Areas of prime interest include audio technology and Open Source product providers. The group will also be looking for organizations that wish to adopt the final specification to create their own implementations.

4. Digital Content Creation- A Brief History of Video Graphics

Video Graphics have been around for longer than many people suspect. The earliest mainstream examples were simple text generators used in live television production. They were, in effect, typewriters that generated a video signal output instead of ink on paper, and they found a ready market in the technology-hungry broadcast business.

These devices came on the scene (pun intended) around the late 1960's. They were completely custom – virtually every system component, including Operating System, File System, CPU, peripheral bus, and video generator were designed by those intrepid pioneers. There were few alternatives in the late '60's, since the cost of a commercial "mini-computer" would have driven the overall system price out of range. Because these devices were meant to operate in a video environment, the system architecture was video-centric, as shown in Figure 1.

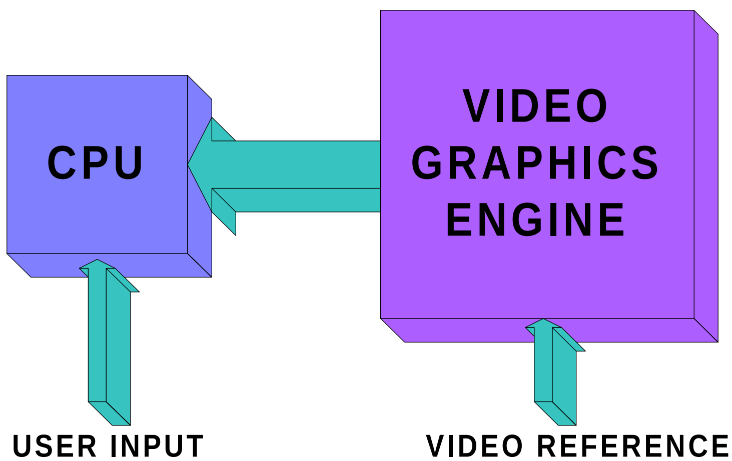

Figure 1. Early Graphics Architecture

Although the architecture is, ostensibly, the classic CPU host with a video graphics peripheral, in fact, it was the video graphics engine that was the center of the universe. This was necessary, as low performance CPUs of the time could not keep up with the real time demands of video. A new task had to be serviced every 64 microseconds (one scan line), and the CPU was running at a 1 MHz clock rate!

The entire system was effectively a State Machine, which was the only practical approach to guaranteed system response time. As you might imagine, programming these beasts was more witchcraft than science. Adding a single line of code at the wrong point could throw the entire delicate structure out of whack. Debugging was primitive – break points, which were effected by replacing an instruction with a halt command. This may seem unimaginable today, but the systems were relatively simple.

Consider that the Chyron* II, which was the leading character generator of its day, ran on 16 Kb of system memory (yes, that is kilo).

Despite its shortcomings, the early graphic architecture filled a need and allowed the broadcast industry to thrive. This general concept continued to evolve and keep pace with advances in memory and processor technology until the next generation architecture displaced it in the late '70s.

Second generation video graphics architectures were based on a marriage of two- (2) technologies – high performance, single-board computers, and the frame buffer. With this generation, the center of the universe switched to the host. The rationale is obvious. It is much better to program graphics operations in software than to hardwire them. Using standard components on the software side greatly eased the burden of application development through high-level languages and sophisticated development tools.

The system response problem was partially addressed by the use of frame buffers. The buffers effectively isolated the non-deterministic behavior of the CPU from the demands of the real time video reference signal. This worked, as long as the user did not break the rules by dealing with "too many" objects in a frame. The problem, of course, is that the rules were too limiting from the user perspective.

This shortcoming was overshadowed by the many benefits of frame buffer technology and graphics programming that, in effect, created a platform for virtually limitless capability. The catch-22 was response time. Given enough time, you could create any effect, but within the boundaries of a single frame, you could only do…well, actually it was hard to know what you could do! This led to various clever approaches to optimization, leading directly back to the major problem that encumbered the previous generation: the system became so highly tuned that it became a custom system. Change any one component and the system became unpredictable. Porting the application to another environment was out of the question.

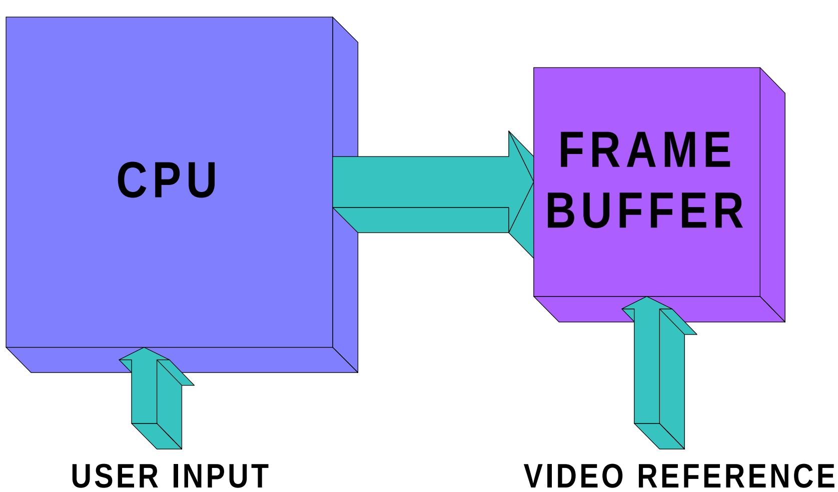

Figure 2. Second Generation Architecture

The current generation offers much promise. As seen in the previous two- (2) architectures, most problems originate from the relationship between the host and the Video Graphics Engine. When the Video Graphics Engine is favored, performance is deterministic, but the system tends to be inflexible and difficult to program. The inverse is true when the CPU is favored. The ideal approach is to achieve some balance between the two while preserving the inherent benefits of previous generations.

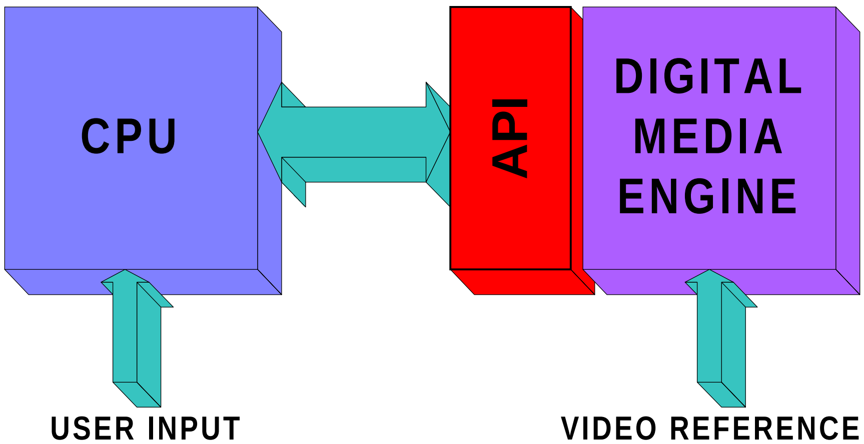

Figure 3. Current Generation Architecture

This ideal has been partially realized through the use of standardized APIs, such as OpenGL®, and architectures that are peer-to-peer, rather than master-slave. In this case, the Digital Media Engine is a relatively autonomous component that sometimes includes its own real time Operating System. The use of simple I/O buffers effectively isolates the non-deterministic behavior of the application from the real time demands of video, and the use of standardized APIs means that applications can, theoretically, move freely to other environments. The reality is somewhat different. Three major issues remain:

- Digital Media device vendors have inconsistent implementations of standard APIs.
- Digital Media devices have a wide variety of feature sets.
- Standard graphic APIs do not include video or audio controls.

These issues conspire to put the application developer back into the mode of creating custom systems, since any system variable must be handled as a special case.

5. An Overview of OpenML™

The primary goal of the Khronos Group is to specify a standard, cross-platform environment that supports the creation and display of digital media, including audio, video, and graphics. We call the specification of this media-rich environment OpenML. During calendar year 2000, the focus of the Khronos Group is to develop a thorough specification of this environment. The final specification will provide guidance to multimedia hardware developers regarding the functionalities that are important to digital media applications in future generations of hardware. Additionally, the specification will define a set of application programming interfaces (API's) that are guaranteed to exist in an OpenML compliant environment. This will allow multimedia application developers to focus on application development and spend less time on mundane details such as device control, synchronization, and buffering issues. It will also free them from the unenviable task of dealing with different API's for video, graphics, and audio for each operating environment. The OpenML environment provides standard API's for dealing with video, graphics, and audio and it is expected that the standard OpenML API's will be supported on major operating environments, including Microsoft Windows*, Linux, UNIX*, and Mac OS*.

Figure 4 is a pictorial representation of the OpenML environment from a programmer's perspective. It identifies the major components of the OpenML environment. The two- (2) main API's available to applications in the OpenML environment are OpenGL® and kDM. OpenGL is the natural choice for a cross-platform standard for 3D graphics. OpenGL is a mature API that is already supported by every major OS. kDM is an API that fills the need for a cross-platform standard dealing with media input, output, and device control. These two- (2) libraries communicate with each other at the device driver level in order to efficiently utilize system resources, to achieve maximum performance and throughput, and to properly synchronize digital media streams with graphics output.

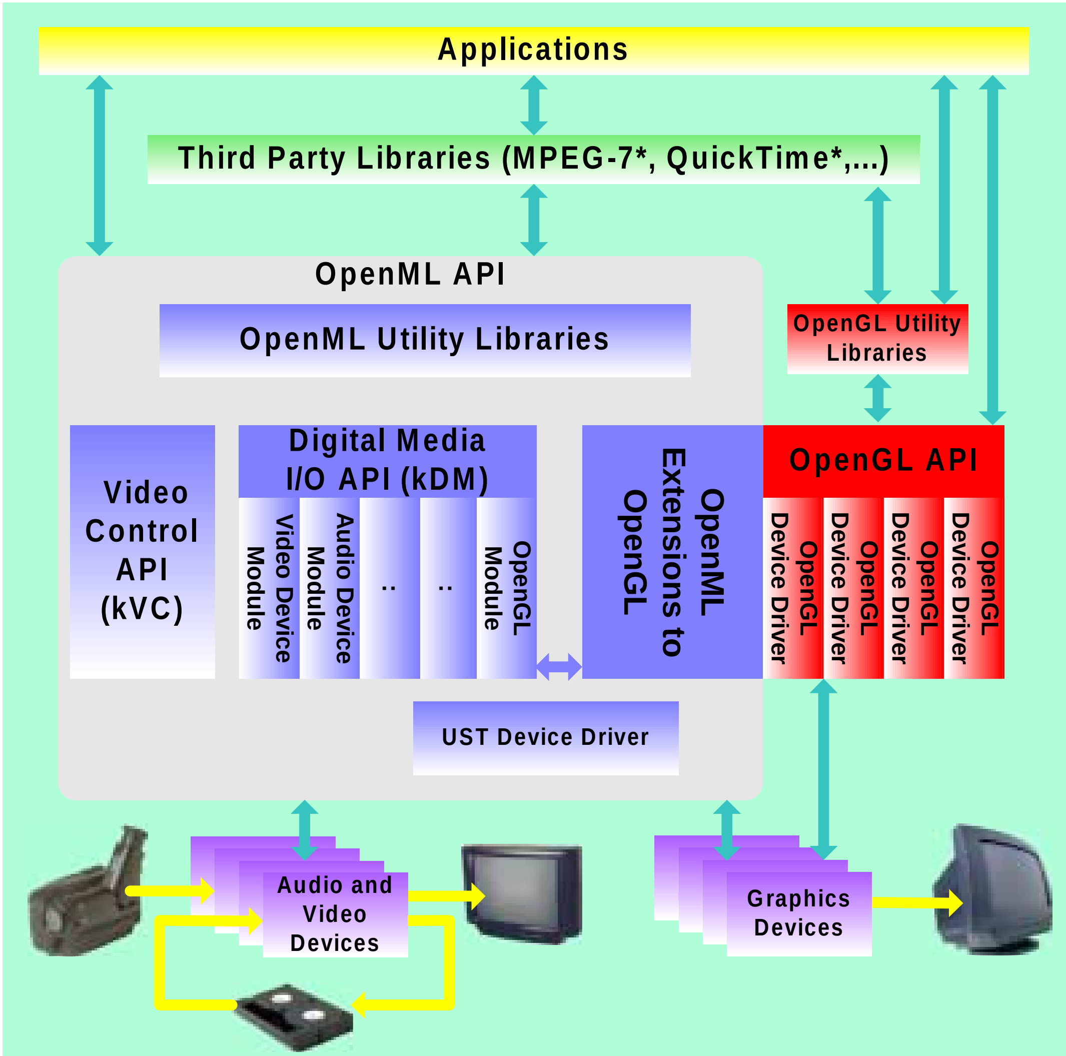

Figure 4. Programmer's View of the OpenML™ Environment

5.1 Synchronizing Audio, Video and Graphics

OpenML also defines facilities that provide precise timing and synchronization information. The Unadjusted System Time (or UST) is a high-resolution, 64-bit, monotonically increasing counter that is available throughout the system. In addition, each media device in the OpenML environment is expected to maintain a Media Stream Counter (or MSC). This type of counter is incremented for each media event that occurs for that device. For instance, a video input device will maintain an MSC that is incremented for each video frame or field of the attached device. A graphics accelerator will maintain an MSC that is incremented for each vertical retrace that occurs. By using UST/MSC pairs, an application can accurately control and synchronize media streams between different devices in the system. UST values are available to applications through the kDM API.

5.2 kDM Features

kDM is a new API based on dmSDK* 2.0 from SGI. It represents the culmination of several generations of API development aimed at supporting digital media in a hardware and OS-independent fashion. kDM is a low-level API in the same sense that OpenGL is considered a low-level API; it exposes the capabilities of the underlying hardware in a way that imposes little policy. Policy decisions can be made by higher-level software such as utility libraries or toolkits, or left up to the application itself.

The primary functions of kDM are to:

- Support asynchronous communication between an application and media devices such as video input, audio output, and graphics.
- Provide synchronization primitives that give applications the ability to correlate multiple digital media streams and coordinate their presentation to an end user.
- Provide processing capabilities (transcoders) for digital media streams.
- Provide device control and device capability queries.
- Provide buffering mechanisms that support the smooth delivery of digital media and obtain the best possible performance on a given system.

An underlying paradigm of kDM is the concept of a path, which is a directed connection between a device and a buffer, or between two- (2) buffers. A path may operate on data as it is moved from the source to destination. A buffer may simply be a user allocated and managed region of system memory.

More interestingly, for paths that transport video images, such a buffer can be an OpenGL pbuffer, a hardware accelerated pixel buffer for rendering. The ability to move video data into and out of OpenGL managed buffers is the basis for the integration of video and graphics in OpenML. Once a video image has been moved to a pbuffer, it can be efficiently used in a variety of OpenGL operations. In particular, it is possible to use the video image in texturing operations. This is the basis for the implementation of the compositing operations common to digital content creation and playback.

Figure 5 is a simplified representation of the possible data flow paths in the OpenML environment.

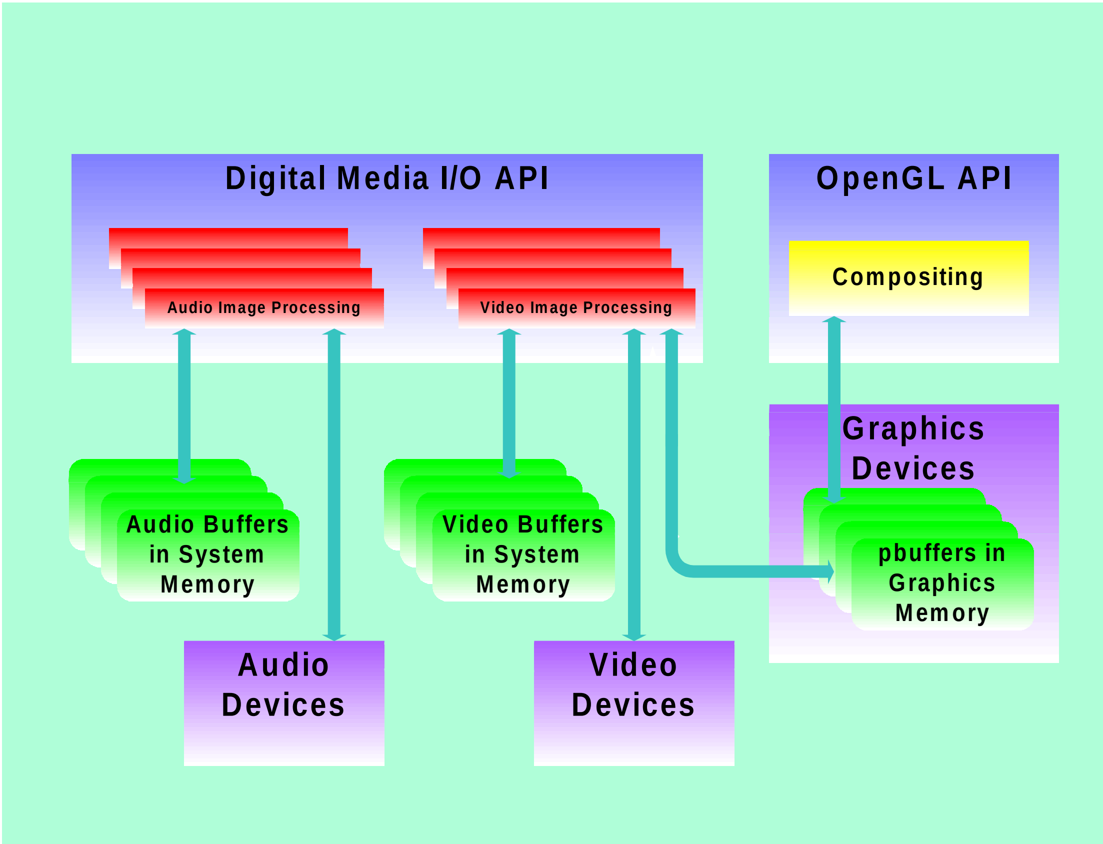

Figure 5. Data Flow in the OpenML™ Environment

5.3 OpenGL® Features

OpenGL is known as an API with a rich and robust set of features for 3D graphics programming. OpenGL also has an extensive set of capabilities for dealing with pixel data (images), both on their way into and out of the frame buffer. OpenGL has achieved a level of standardization and popularity that no other 3D graphics API has ever achieved. This makes it a natural choice as the API that provides graphics and access to the frame buffer. OpenML compliance requires support of OpenGL 1.2.1. It is also expected that OpenML compliance will require the OpenGL 1.2.1 Imaging Subset.

OpenGL has a well-defined extension mechanism that has led to the definition of more than 200 unique extensions. Some of these extensions were developed to address the needs of multimedia application developers. The existing OpenGL extensions that address the needs of digital media creation or playback are being evaluated for their applicability to the OpenML environment. These extensions will become a required part of an OpenML-compliant environment. In addition, the Khronos Group is developing new OpenGL extensions to strengthen OpenGL's rendering and video integration capabilities. This set of extensions is an integral part of OpenML and are required as part of an OpenML compliant implementation.

The list of such extensions is expected to include:

- Synchronization using UST/MSC information
- Reading and writing of interlaced video images
- Direct support for video pixel formats such as YCrCb
- Synchronized usage of pbuffers as a target for streaming video
- Video input to texture memory
- Enhanced texture memory management
- Extended texturing functionality in support of compositing
- Higher quality texturing
- Full scene anti-aliasing
- Access to programmable geometry and shading hardware

5.4 Video Back-end Control

One thing that differs dramatically from one operating environment to another is the windowing environment. Rather than attempt to achieve cross-platform standardization in this area, the OpenML environment will simply coexist with the native windowing environment. This results in some platform dependencies for application writers, but many applications have already been modified to modularize the user interface and windowing operations, making them as portable as possible to different operating environments.

An example of this principle is found in the kVC API. kVC permits user control of video (output) operation on graphics devices for Microsoft Windows* platforms. kVC is designed to be the equivalent of the proposed Xdc Extension to the X11 Window System*. These libraries provide a range of services to permit programmatic user control of video output operations on graphics devices. Examples of control include loading video formats, setting video output gain, setting pedestal, and changing H-phase (horizontal genlock phase).

5.5 OpenML™ Application Profiles

OpenML is intended for use in a range of application scenarios, from professional content authoring through playback on desktops, in set top boxes, and even in such devices as PDAs. We believe that all these domains require essentially the same functionality but with different performance profiles. For example, professional content authoring typically requires massive bandwidth, the ability to composite many layers in real time, and hardware support for full scene anti-aliasing, among other requirements. Playback usually involves only modest bandwidth utilization, compositing of just one- (1) or two- (2) layers and may not utilize anti-aliasing at all.

To help users make purchasing decisions, the Khronos Group expects to develop a suite of performance benchmarks. These benchmarks will be designed to provide performance information for a variety of application profiles.

6. Summary

Rich media content has already found its way into our everyday lives. Rapid advances in personal appliances, wireless devices, and high bandwidth delivery vehicles will create even more demand. This, in turn, creates an unprecedented opportunity for application developers.

The challenges have increased with the opportunities. Developing for a single platform or a single display format is a risky proposition when the landscape is in flux. Porting applications to multiple environments minimizes the risk, but is a daunting task given the number of potential permutations of platform and delivery formats. Ironically, the very success of media-rich content hinges on this diversity – everything from wall size display monitors to PDAs to cell phones should be considered as a potential target.

OpenML™ provides the abstraction layer that allows application developers to focus on their value proposition, enables their products to move easily across the new range of platforms, and let's them move rapidly, reaping the rewards of emergent opportunities.