Bionic Buffalo Tech Note #104

Summary of Pegasus/ISA versus DSM-CC Specification Coverage

last revised Thursday 2003.04.17

Pegasus and ISA are *Not* DSM-CC

There is a misconception among some that the Time-Warner Pegasus architecture, which includes the Pegasus Interactive Services Architecture (ISA), is a form of DSM-CC. In fact, although there is some overlap and there are a few similarities, the two specifications are quite different. In fact, for the most part, they don't even cover the same problem space. The problem is compounded by their use in some instances of the same names for different things.

What is DSM-CC?

DSM-CC, or Digital Storage Media Command and Control, is one of the MPEG-2 family of specifications. Formally designated as ISO/IEC 13818-6, the DSM-CC specification is owned by the International Organization for Standardization (ISO) and the International Electrotechnical Commision (IEC). (The other MPEG-2 specifications are designated 13818-1, 13818-2, and so on. DSM-CC is part 6 of 10 parts.)

Revision 1 of the draft specification is dated November 1995.

In its own words, DSM-CC “consists of a modular set of protocols that may be combined or used individually to provide a wide range of functionality which may be used to support emerging mulitmedia technologies... The concepts and protocols of DSM-CC provide the general capability to browse, select, download, and control a variety of bit stream types. DSM-CC also provides a mechanism to manage network and application resources...”. There is also an abstraction layer, so that “a suite of uniform interfaces are visible to the application, shielding it from the details of inter-working among heterogeneous networks... a server may simultaneously and uniformly interact through a single network interface with clients connected to different network types, without requiring a separate network interface to each client.” (Quoted from the Introduction of the DSM-CC specification.)

In other words, DSM-CC is a set of protocols and interfaces (APIs), and is equally focused on servers and on clients.

The high-level abstraction seen by client and server applications is defined in Interface Definition Language (IDL), part of the Common Object Request Broker Architecture (CORBA), defined by the Object Management Group (OMG). IDL has mappings to various programming langugages and to an over-the-wire protocol (GIOP, the General Inter-ORB Protocol, and its internet specialization, IIOP, the Internet Inter-ORB Protocol).

What are Pegasus and ISA?

Pegasus is a program initiated by Time-Warner to support the deployment of the Time Warner Cable Full Service Network™ architecture. The Phase I RFP, which described the architecture, was published in March 1996. Pegasus defines a client (the Pegasus Set-Top Terminal), a delivery network. network, and servers.

The Pegasus Set-Top Terminal includes certain specific software components (browsers and applications), hardware interfaces, and protocol support. The applications rely heavily on HTML, the basic document language of the internet. The RFP calls out a specific operating system API (the PowerTV™ OS v1.5 API), and specific hardware configuration.

The Pegasus specification also defines some spectral allocations, protocols, topologies, and headend components.

In September 2000, Time Warner published the first version of the Interactive Services Architecture (ISA) specification. ISA is "intended to define the server side interfaces required for a service provider (application developer) to implement a service for the Pegasus platform. It does not describe interfaces on the client (set top or PC). It does not define any specific services..." However, ISA is supplemented by various other specifications, for services (such as Movies on Demand) subsystems (such as Asset Distribution).

The interfaces in ISA and related specifications are defined in IDL, as are the high-level client and server interfaces of DSM-CC.

Contrasting DSM-CC against Pegasus

Object System. DSM-CC defines a distributed object system, so clients can invoke most object methods on the server. By contrast, Pegasus objects are not visible on the client. Pegasus objects are available only on the server side of the network. Both architectures define interfaces in terms of CORBA and IDL. In DSM-CC, client applications can use CORBA interfaces for distributed objects. In Pegasus, client applications do not have access to CORBA interfaces.

Network. DSM-CC anticipates a wide range of network topologies, technologies, and lower-level protocols. Pegasus is specifically intended for the network architecture of Time Warner, although it may be used in compatible environments.

Platform. DSM-CC doesn't specify any particular platform. Pegasus expects certain characteristics in the client hardware and software.

Protocols. At the transport and presentation layer, DSM-CC defines several protocols, and defines specializations on several others. DSM-CC is agnostic with regard to the network layer and beneath. Pegasus specifies lower layer protocols, the specialization of a DSM-CC protocol (User-Network Session), and creates a new protocol (LSCP) for controlling video-on-demand playback.

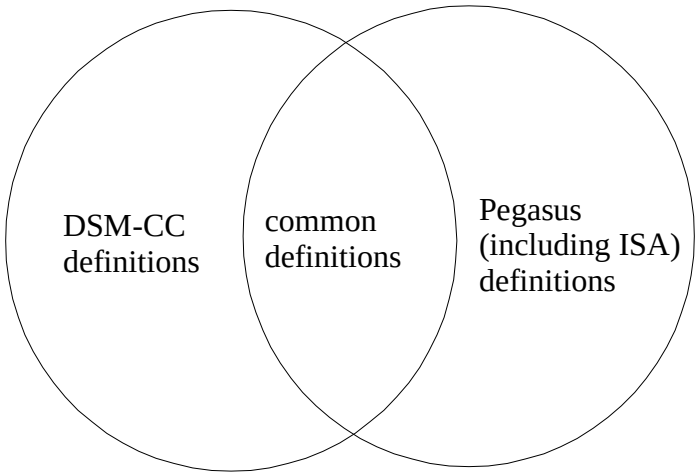
Interfaces and APIs. DSM-CC specifies the same API for the client and server, based on the CORBA mapping of the IDL. Pegasus has one API for the client (based on the PowerTV™ OS API), and another for the server (based on the CORBA IDL mapping). Although some of the interfaces have the same names, t*here is not a single interface definition in common between Pegasus and DSM-CC.*

Level of Abstraction. At the level of fudamental services (stream control, file access, and so on), DSM-CC has a complete abstraction layer, where Pegasus is does not. At the higher application level, DSM-CC is silent while Pegasus specifies several services (billing, asset management, and so on).

Motivation and Goals. One of the main goals of DSM-CC is to foster interoperability among a wide variety of software components from diverse sources. Pegasus has a more specific purpose: to foster interoperability among the products of suppliers to Time Warner.

Overlap and Separation

Instead of viewing Pegasus as “DSM-CC compliant”, or looking at the two architectures as having a great deal in common, it is clear that they are largely disjoint. If we view the subject matter of the architectures' definitions in the following way, the dissimilarity is clear. Consider the sets of definitions:



The definitions specific only to DSM-CC include:

Interfaces:
`DSM::Base` and `DSM::Access`, for common operations and attributes
`DSM::Stream`, to interactively control MPEG continuous media streams
`DSM::File`, to read and write data files
`DSM::Directory` and `DSM::BindingIterator`, to access object directories
`DSM::Session`, `DSM::SessionUU`, `DSM::SessionSI`, `DSM::ServiceGateway`, `DSM::ServiceGatewayUU`, `DSM::ServiceGatewaySI`, and `DSM::First`, for connection to service domains
`DSM::Event`, to subscribe or unsubscribe to events (which are sent in the MPEG transport stream)
`DSM::Download`, `DSM::DownloadSI` to manage U-N download operations
`DSM::Composite`, to handle multiple objects associated as a set
`DSM::View`, to apply a relational model to objects in directories, or to access relational databases
`DSM::State`, to suspend and resume application state
`DSM::Interfaces`, to examine or define interface definitions
`DSM::Config`, to define invocation behaviour
`DSM::LifeCycle`, to create interoperable object references
`DSM::Kind`, to determine the interfaces an object supports
`DSM::Security`, to provide authentication information to gain access to an object

Protocols:
User-Network Configuration Protocol, to provide clients or servers with the configuration parameters required for the device to operate on the network
User-Network Session Protocol, the subset of 29 message types and 22 resource types not supported by Pegasus SSP, for setting up and tearing down sessions
User-User Protocol, an enhancement and extension of CORBA IIOP, by which client applications communicate with server objects
Switched Digital Broadcast Channel Change Protocol, allowing a client to remotely switch from channel to channel in a broadcast environment
User-Network Passthrough Protocol, for sending messages among users (clients or servers)
User-User Object Carousel and the *Broadcast Inter-ORB Protocol*, for downloading objects to the client from the server
MPEG-2 Transport Protocol extensions: various descriptors defined to support object carousels and events

The definitions specific only to Pegasus include:

Interfaces:
`Asset`, `AssetEventChannel`, `AssetFactory`, to implement primary management abstraction for media assets

`Content`, `ContentEventChannel`, `ContentStore`, `ContentStoreFactory`, to represent contents of media files associated with assets
`Customer`, `CustomerEventChannel`, `CustomerFactory`, representing the home user of service
`Equipment`, `EquipmentFactory`, to abstract equipment such as a terminal
`Manager`, part of base module for component architecture
`MetadataEventChannel`, `MetadataList`, `MetadataListFactory`, for abstracting contained-by-value attributes of other classes
`Offering`, an offer to sell to a customer
`Package`, `PackageEventChannel`, `PackageFactory`, a specialization of asset, to manage the relationship among assets in a distribution
`Product`, `ProductEventChannel`, `ProductFactory`, to represent the sum total of assets required to implement a service
`Provider`, `ProviderEventChannel`, `ProviderFactory`, representing the source of assets delivered into a headend
`Purchase`, `PurchaseEventChannel`, `PurchaseFactory`, to represent the contract between a customer and an offering
`Resource`, to represent reserved or assigned network connections, channels, addresses, capacity, or other services
`ServantBase`, `ServantBaseIterator`, `ServantFactory`, providing a base class for entities which do actual work
`Service`, `ServiceEventChannel`, to represent a customer-consumable invocation of an application
`ServiceGateway`, to route transactions from a session gateway to a service
`Session`, `SessionEventChannel`, to represent an associated collection of resources required to deliver a service
`SessionGateway`, which negotiates with the network for resources, and creates sessions
`SessionNegotiationCallback`, providing a mechanism for notification of resource negotiation status
`SessionUU`, a base interface providing attach and detach operations for objects participating in sessions
`Stream`, `StreamEventChannel`, `StreamService`, to implement control of MPEG-2 streams
`Terminal`, `TerminalFactory`, representing a type of equipment which participates in sessions

Protocols:
User-Network Session Protocol, the additional 5 resource types and 6 resource descriptors defined by SSP as an extension to DSM-CC, for setting up and tearing down sessions
Lightweight Stream Control Protocol (*LSCP*), for controlling MPEG-2 streams

The definitions commmon to both architectures include:

> Protocols:
> *User-Network Session Protocol*, the subset of 21 common message types, and 4 common resource types, for setting up and tearing down sessions (this is the subset of DSM-CC U-N Session supported by Pegasus SSP)
> *User-Network Download Protocol*, including the data carousel

In other words, DSM-CC and Pegasus have very little in common. By rough page count, only about 15% of the DSM-CC specification (mostly portions relating to User-Network Session and Download protocols) is applicable to the Pegasus architecture.

(Note: The lists above were taken from the 1998 revision of the DSM-CC specification, and from the Pegasus IDL version 1.4 as published by Time Warner around August 2002.)

Potential Interoperability

Pegasus is not DSM-CC. However, the two architectures can be interoperable. Bionic Buffalo has written Tech Note #103, *Introduction to Pegasus Client Support*, to describe how interoperability is achieved on the client. A future document will provide an equivalent discussion of interoperability on the server.
