Annex: Relation of DSM-CC to MHEG-5

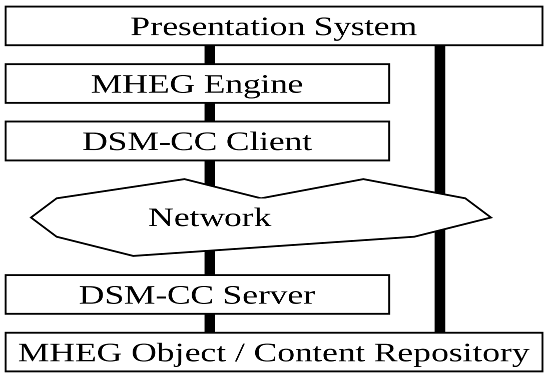
This annex does not form an integral part of this document

MHEG-5 (ISO/IEC SC29WG12 13522-5) provides a framework for the distribution of interactive multimedia applications across minimal resource platforms of different types. A MHEG-5 application resides on a server, and as portions of the application are needed, they will be downloaded to the client. In a broadcast environment, this download mechanism could rely, for instance, on cyclic rebroadcasting of all portions of the application.

A minimal MHEG-5 runtime environment has to provide an entity for decoding of the MHEG data structures and an entity called MHEG engine, which parses and interprets the MHEG-5 objects. The engine also communicates with the local presentation environment and the MHEG-5 objects. It responds to the events initiated by the application or the user (for example timer events, or "button pressed") in the application specific way. A MHEG application is always event driven.

An MHEG-5 application consists of Scenes objects and objects that are common to all Scenes within an Application object. At most one Scene is active at any one time. This is the part of the application that has to loaded on the clients' system. Navigation in an application is done by transitioning between Scenes. A Scene contains a group of objects, called Ingredients, that represent information (graphics, sound, video, etc.). The content data is typically not part of the encoded Scene object. Instead, content data can be referenced and stored externally.

A MHEG-5 runtime system can utilize DSM-CC for access to MHEG objects and content data. Such a system is sketched below (The design of the system shown below is not normative by MHEG-5):



Example of MHEG-5 runtime system

The following sections provide examples and hints, how DSM-CC facilities can be used by an MHEG-5 runtime system. Further information on details of MHEG-5 facilities are available in the MHEG-5 standard.

Name Space

When an application starts, it is assumed that a service gateway has been located and attached to, so that there is exactly one name space within which the application objects are located. Within that name space, a service has also been located. That service is a DSM-CC directory; within it, there can be other directories, files, and streams.

Furthermore, it is assumed that each object belongs to exactly one application. This assumption is necessary to allow for unambiguous object references. Three types of retrieved data can be differentiated:

objects that comply to MHEG-5 (Scenes or Applications),
the content (such as bitmaps or text) of those objects, and
streams (such as video and audio).

For accessing the various objects of an application on the server side, the DSM-CC Directory, File and Stream objects are used. Note that the server, in this context, does not have to be a physical server, but could be implemented, for example, as a broadcast carousel in a pure-broadcast topology.

Each file is either a Scene object, an Application object, or the content data of an Ingredient object, belonging to a Scene or an Application. Each Scene object, Application object and content data is stored in a separate file.

Object References

MHEG objects and content data can be exchanged in two ways: either the object is exchanged as a DSM-CC file object or within another object. The former method is used for Applications and Scenes, the latter for all other objects contained in a scene or application object. The MHEG objects are identified by an `ObjectReference`, consisting of an optional byte string `GroupIdentifier`, followed by an integer, the `ObjectNumber`. For the `GroupIdentifier`, the following rules may be defined by the application domain:

All `GroupIdentifiers` are ASCII strings. A `GroupIdentifier` reference is mapped on a DSM-CC name within the name space of the service gateway to which the runtime is attached. Within the `GroupIdentifier`, the character '/' (standard ASCII slash) is used to delimit directory references (of the 'depth' type); for instance, if the `GroupIdentifier` is 'apps/otherAppl', it is mapped on the DSM-CC name 'otherAppl' in the directory 'apps' of the service gateway to which the runtime has attached.

Additional rules may be defined (e.g. for shortcut/wildcards of application identifiers or references to the current direcotry root) if required. For the mapping on DSM-CC, the following additional rules may be used:

Each Application and Scene object shall have in its `GroupIdentifier` a byte string which maps on the name of the DSM-CC file which contains that object. These objects have their ObjectNumber set to 0 (normative by MHEG-5).

Each application shall have exactly one Application object. That object shall be contained in a DSM-CC File object with the special name, e.g. 'startup' (normative by the application domain).

References to content data objects may (as defined by the application domain)

a) either leave out the `GroupIdentifier`, in which case it is assumed to be a string which maps on the name of a DSM-CC file which contains the object (Application or Scene) of which this object is a part,

b) or fill in the `GroupIdentifier` with such a string.

Such objects shall have their ObjectNumber set to a value which is unique within that file (normative by MHEG-5).

Content References

The high-level API has a separate way of referencing the external content of objects belonging to the Ingredient class. This is done by way of a `ContentReference`. The `ContentReference` consists of a byte string. The following rule may be defined by the application domain:

The exact same mapping shall be used as for the `GroupIdentifier` above;

the relative name of the content object file is appended to the `GroupIdentifier`, separated with a '/'.

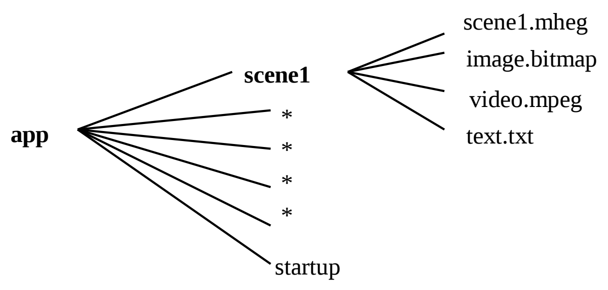
Stream Events and Normal Play Time

The DSM-CC StreamEvent interface provides the possibility to carry private data in the data structure for the event, in the form of the PrivateDataByte field. These bytes shall be mapped one-to-one on the `StreamEventTag` of the MHEG-5 event `StreamEvent`.

The MHEG-5 internal attribute `CounterPosition` of the Stream class shall also be mapped one-to-one on the value of the DSM-CC Normal Play Time of the corresponding stream.

Example of DSM-CC file structure for an application

This section show how MHEG-5 objects may be mapped to DSM-CC file structure.



DSM-CC file structure example

Below is one textual representation of the code for accessing the different objects depicted in the figure above. The first is an Application Object, performing a transition to the first scene. The Scene object identifies the content objects which belong to the scene by referencing the content files.

```
{:application
      :object-identifier ( "app/startup" 0 )
      :on-startup ( :transition-to ( app/scene1/scene1.mheg)
}

{:scene
      :object-identifier ( "app/scene1/scene1.mheg" 0 )
      (:bitmap 1
            (:content-data
             :referenced-content "app/scene1/image.bitmap") ...)
      (:video 2
            (:content-data
             :referenced-content "app/scene1/video.mpeg") ...)
      (:text 3
            (:content-data
             :referenced-content "app/scene1/text.txt") ...))
}
```

Example of Mapping High-Level API Actions on DSM-CC Primitives

Below is a possible example of "translation" of MHEG-5 actions to DSM-CC primitives.

MHEG-5 Behavior	*Object Type*	*DSM-CC U-U Function*
Launch/Spawn	Application	DirectoryOpen*(app.fileid) -> FileObRef* FileRead*(FileObRef)*
Prepare	Scene, content object and stream	DirectoryOpen*(scene.fileid) -> FileObRef* FileRead *(FileObRef)*
Run	Video and Audio	DirectoryOpen*(stream.file) ->StreamObRef* StreamResume(*StreamObRef, starttime, 1/1*)
SetSpeed(0)	Stream	StreamPause(*StreamObRef*, x80000000.x00000000)
Stop	Stream	StreamClose(StreamObRef)
StreamMarker	Stream	StreamSubscribe (*StreamObRef, marker*) StreamNotify (*StreamObRef, marker, call back function*) StreamUnSubscribe (*StreamObRef, marker)*
StreamTimer	Stream	StreamStatus -> Gets normal playtime
Call and Fork	Application	RPC - UNO
OpenConnection	Application	Attach (*ID)*