Application Definition
Blu-ray Disc Format

BD-J Baseline Application and Logical Model Definition for BD-ROM

March 2005

Table of Contents:

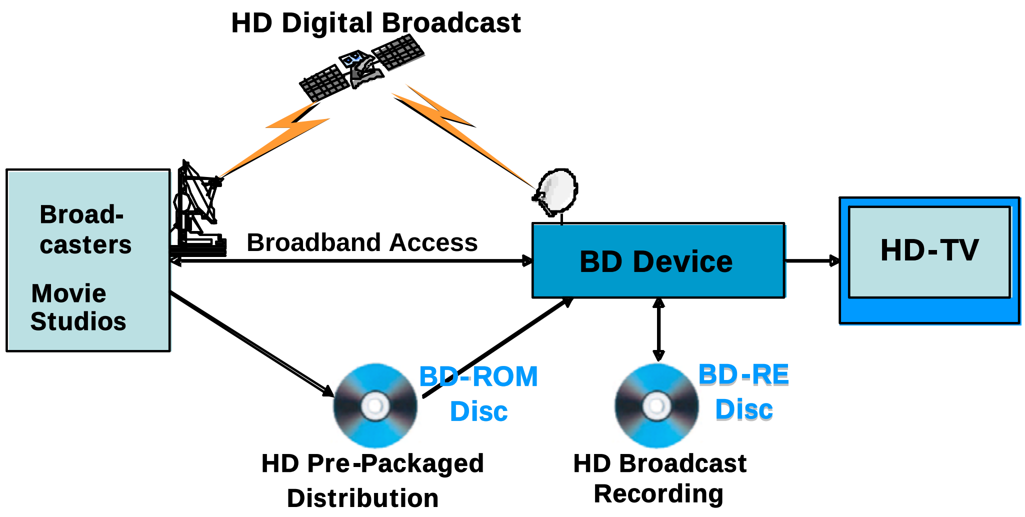
1 Introduction

This document is intended to provide content authors with the information needed to produce BD-J enhancements and other BD-J applications for the Blu-ray Disc Prerecorded application format (BD-ROM). The application format is based on the J2ME™[1] Personal Basis Profile (PBP) version 1.0 Java™ platform from Sun Microsystems and Globally Executable MHP™[2] (GEM) version 1.0.2 from DVB™.

1.1 Background

1.1.1 Aims of BD-ROM

BD-ROM is designed not only for pre-packaged High Definition (HD) movie content but also as a key component of a consumer HD platform. As shown in Figure 1-1 below, the Blu-ray Disc (BD) platform is designed to provide access to HD content throughout the home via HD digital broadcast recording and HD playback functions.



We aim to accelerate home HD compatibility by making HD digital broadcast recording and HD pre-packaged playback applications possible on BD devices. By supporting both recording and pre-packaged playback functions, the need to support two different media, as was the case for SD images (recording: VHS, package playback: DVD-Video) is eliminated.

Figure 1-1 – HD Content Distribution as envisioned by the BD Application Standard

[1] Java, J2ME and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. For more information please refer to: http://www.sun.com/suntrademarks/

[2] DVB and MHP are registered trademarks of the DVB Project

When DVD-Video was developed, Standard Definition (SD) image recording devices were already present in the market (VHS). The user therefore had to contend with two different forms of media for SD: VHS for recording and DVD-Video for package media playback. For HD however, we are at the initial rollout stage for HD broadcast receivers, HD recording functions via packaged media, and HD pre-packaged content distribution. By combining all of these functions into one platform, BD provides a one-stop-shop to fulfill the desires of users who want to use all of these functions. Simply put, the user will be HD compatible by using BD.

1.1.2 Understanding Basic Issues (Required Specifications)

Understanding 1: Market Trends (DVD is still going as strong as ever)

1. When comparing VHS with DVD, there are clear differences: (1) improvement in maneuverability by going from tape to optical disc, (2) improvements in image quality via MPEG-2, (3) going from linear movie viewing to viewing multimedia titles including special features.
2. Popularity of DVD-Video business. US retail sales for 2004: 37.125 million DVD players sold, 1,518.3 million titles sold (Source: DVD Entertainment Group).
3. It will therefore be extremely important to design BD-ROM so that there is clear integration/continuity with DVD-Video, but also with sufficient differentiation.

Understanding 2: Consumer Perspective

1. Consumers expect high image quality with large-screen displays. Plasmas, LCDs, rear-projection televisions, digital projectors and other large-screen, HD capable display devices are now becoming popular. DVD image resolution (SD) is insufficient on these types of large-screen, HD capable displays.
2. Special Features for Movies are Important: US consumers are especially drawn to special features (production shots, outtakes, scenes cut from the movie release, etc.) and cite these special features as one of the major reasons for purchasing DVD-Video titles.
3. Need for new ways to view Movies: Consumers may not pay attention to BD unless we can provide new, revolutionary ways to view movies and movie related content in addition to providing a HD experience e.g. through services provided via broadband.
4. User Interface Integration is important: We need to keep in mind that users are familiar with (and are comfortable with) the DVD-Video User Interface but that they also desire a richer interaction experience.

Understanding 3: Studio Perspective

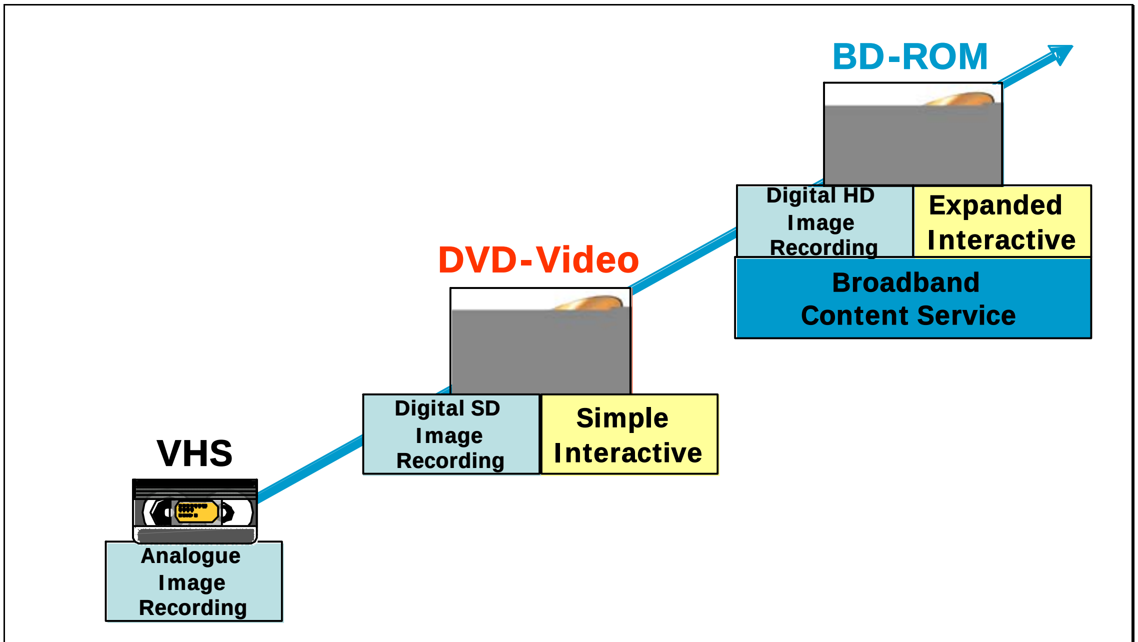
1. The DVD-Video business is going extremely well, but competition between titles has also increased dramatically.
   - There is a need to include many special features and games to clearly distinguish titles.
   - Increased Profit by releasing the DVD-Video title quickly after the theatrical release (In the US, DVD-Video titles are released about 5 months after the theatrical release of the movie).
   - Production schedules are becoming shorter and title programs are becoming more complex – the burden on the production is getting heavier. Production needs to become more efficient.
2. BD-ROM Production:
   - The DVD-Video production area is shouldering a large burden. Assuming that current DVD-Video producers will also take on BD-ROM production, a smooth integration of the production process is essential.
   - We would like to make BD-ROM easier to manufacture than DVD-Video (and capable of doing more, easily).

- Desire for New Type of Titles & Services: We would like to promote the wide use of BD by providing new services and functions through BD-ROM titles. This will be another means of re-selling catalogue titles (i.e., continue to ride the wave of DVD sales).

3. Desire to Implement New Business Models:
   - The introduction of DVD-Video allowed the industry to move from the “rental” driven VHS business model to a “sales” driven DVD business model with a higher profit margin.
   - Studios are looking forward to gaining a new source of income through the introduction of new package media. To prepare for this eventuality, they are looking to new means of distribution including those that use broadband technology.

1.1.3 Evolution of Package Media for Movies

BD-ROM design is conducted based on the probable evolution of package media as envisioned in Figure 1-2 below, with due consideration to the issues (usage requests) outlined previously.



DVD-Video usage spread quickly via the implementation of MPEG and IT technologies, namely the achievement of high quality SD images and the use of simple interactive functions typified in special features. BD-ROM not only displays HD images and supports expanded interactive functions on its own, but by providing broadband content service functions through broadband technology, will lead the user to the HD/BD World.

Figure 1-2 –Evolution of Movie Package Media

The use of DVD-Video spread quickly because of high quality SD image technology and simple interactive functions implemented through MPEG and IT technologies.

BD-ROM is being designed with an architecture for a “BD World” that clearly distinguishes itself from DVD by (1) realizing High Definition images, (2) creating an “expanded” interactive environment, and (3) creating a platform for broadband content service functions that will expand the viewer's way of enjoying content.

1.2 Overview

For the purposes of providing the reader with an understanding of BD-ROM features, this specification categorizes related features into "modes" of which two are defined ("HDMV" and "BD-J"). The categorization used in this document does not represent the actual structure of BD-ROM nor does it provide a description of features that are not relevant to content authors.

Introduction to HDMV mode

BD-ROM provides an easy-to-author framework for creation of High Definition movie experiences known as "HDMV" (High Definition Movie) mode. HDMV has been designed from the ground up to support a feature set that supersedes DVD-Video while emphasizing production continuity with existing media formats. HDMV supports all of the well known DVD-Video features such as seamless multi-angle and multi story, Language Credits (dynamic selection of a credits sequence depending on the users Language choice), Director's cuts, Trilogy collections etc.

Here are some of the key features offered by HDMV:

- Industry Standard High Definition Video and Surround Sound Audio:
  - MPEG-2, MPEG-4 AVC and SMPTE VC-1 video formats.

LPCM as well as Dolby®[3] Digital' Dolby Digital Plus, Dolby Lossless, DTS digital surround®[4] and DTS-HD® audio formats.

- Independent High Definition Graphic planes:

Two independent High Definition graphic planes and one High Definition video plane simplifies the process of Authoring both Menu and Subtitle graphics compared to DVD-Video.

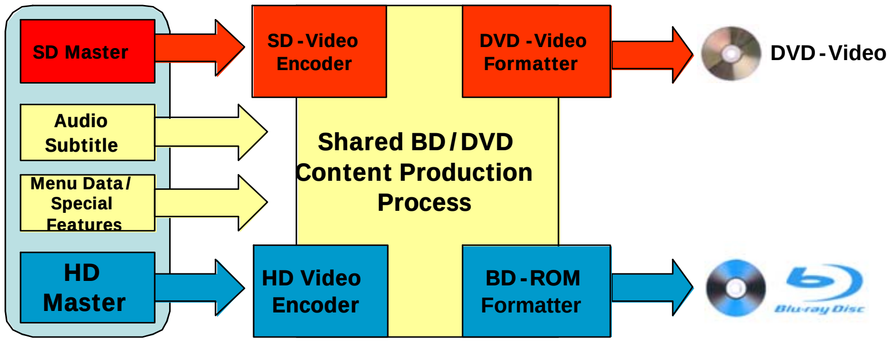
- Improved Menu features:
  - "Multi-page Menus" - Menu presentations can be changed with no interruption to AV playback.
  - "Pop-up Menus" – Menus can be shown or removed from display based on User request.
  - Full color High Definition animated Buttons and animated Menu transition effects.
  - "Button-sounds" – sounds can be presented when Menu Buttons are selected or activated.
- Improved Subtitle features:
  - High Definition "Bitmap Subtitles" supporting full color images with frame-accurate animation effects up to full video frame rate.
  - High Definition "Text Subtitles" utilizing vector-based fonts, encoded text data and multiple style definitions. Text Subtitles do not affect the bandwidth available for a BD-ROM Title's AV stream.
- Additional Features:
  - "Browsable Slideshow" - still images may be presented and changed without interruption to audio playback.

HDMV has been developed with a focus on ease-of-authoring, ease-of-verification and reduced content production cost. This has been achieved in three ways: 1) ground-up modern design of flexible data

[3] Dolby and the double-D symbol are registered trademarks of Dolby Laboratories.

[4] DTS, DTS digital surround and DTS-HD are registered trademarks of Digital Theater Systems, Inc.

structure definitions; 2) integration into the current content production and material preparation process used for DVD-Video titles; 3) enabling the possibility of simultaneous production of DVD-Video and BD-ROM titles as shown in Figure 1-3 below.



BD-ROM is designed to implement new functions that differentiate it from existing media and to ensure compatibility with BD-RE. Additionally, the design has an emphasis on BD-ROM content production. From the perspective of content production/raw material preparation, title production is made that much more difficult if the design demands a completely different process in creating an HD movie mode-led title. (Impediments would include higher costs for new BD-ROM content production tools, new specialized equipment, new tool implementation, operator training, etc.) Therefore, as shown above, the HD movie mode has been designed with emphasis on compatibility with DVD in the content production process that will allow simultaneous DVD and BD-ROM title production.

Figure 1-3 – Example of HD Movie mode content production process

If the BD-ROM standard is based solely on the assumption of a content production and material preparation process that is entirely different from DVD-Video, then the cost of Title production becomes prohibitively more expensive (costs for new development of BD-ROM content production tools, exclusive material necessary for production, new tool implementation, operator training, etc.), thereby potentially hindering the spread of BD-ROM.

Details of the HDMV platform are given in Section 5 along with more detailed examples of HDMV applications.

Introduction to BD-J mode

BD-ROM also provides a fully programmable application environment with network connectivity thereby enabling the Content Provider to create highly interactive, updateable BD-ROM titles. This mode is based on a platform that is fully compliant with the J2ME Personal Basis Profile Java platform (PBP 1.0) and the Globally Executable MHP specification (GEM 1.0.2). It is known as “BD-J”. Content Providers are able to include interactive Java applications on a BD-ROM disc in various ways (one application for the entire

[5] Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. For more information please refer to: http://www.sun.com/suntrademarks/

disc, one application per Title, etc.).

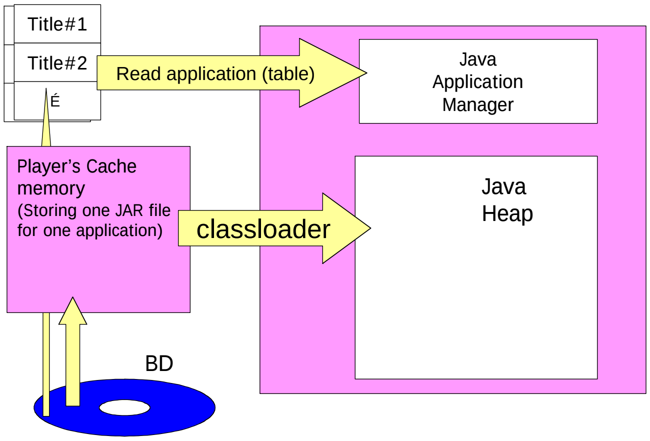


Figure 1-4 – Overview of Java application tables in BD-ROM

Possible BD-J applications include:

- A BD-ROM Title that supports downloading trailers for a sequel from a Content Provider's website and playback under application control.
- A BD-ROM disc with a set of games, each game associated with a Title in the disc's table of content. The main Menu of the disc allows downloading subsequent games from a Content Provider's website under certain conditions, like solving a puzzle for example.
- A BD-ROM Title is distributed supporting only a small number of languages. Later support for more languages (i.e. subtitle and or audio streams) can be downloaded by the BD-J application on the disc.

Java is a platform independent programming environment deployed in a wide verity of environments: Server based applications can be supported through the Java 2 platform Enterprise Edition (J2EE), while Desktop based applications can be supported through the Java 2 platform Standard Edition (J2SE), and Consumer Electronics based applications (for devices like cell-phones and interactive digital receivers) can be supported through the Java 2 platform Micro Edition (J2ME). Personal Basis Profile is a member of the J2ME family of platform profiles. Java provides an open and flexible programming environment for BD-ROM.

BD-J provides a Java UI & graphics framework along with support for Local Storage and Internet connectivity features thereby creating a complete and future proof solution. A BD-ROM disc can contain a mix of titles based on HDMV and BD-J. Details of the BD-J platform are given in Sections 6 along with more detailed examples of BD-J applications. Section 7 contains details of the relationship between BD-J and GEM, and further details about available functions.

2 Referenced Specifications

[PBP] JSR 129: J2ME™ Personal Basis Profile Specification (1.0), available from JCP - http://jcp.org/en/jsr/detail?id=129

[GEM] ETSI TS 102 819 V1.3.1: Digital Video Broadcasting (DVB™); Globally Executable MHP™ (GEM) version 1.0.2

[OTF] OpenType – OpenType™ Specification version 1.4, http://www.microsoft.com/typography/otspec/default.htm

3 Database information

3.1 BD-ROM data structure

Figure 3-1 below provides a simplified overview of the BD-ROM data structure. BD-ROM has four layers for managing AV stream files as follows: ***Index table***, ***Movie Object/BD-J Object***, ***PlayList*** and ***Clip***.

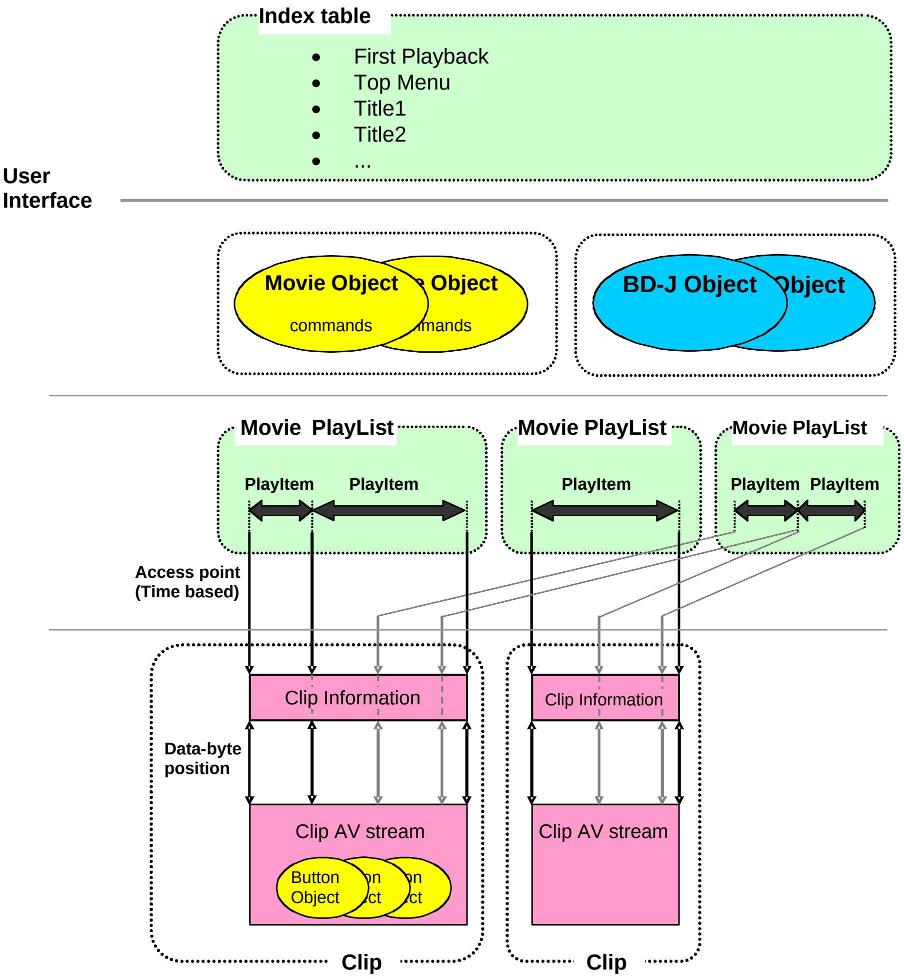


Figure 3-1 – Simplified structure of BD-ROM

3.1.1 Index table

The "Index Table" is a top-level table that defines the Titles and the Top Menu of a BD-ROM disc. This table contains entry points for all of the Titles and the Top Menu. The Player references this table whenever a Title or Menu is to be executed e.g. whenever the Title Search or Menu Call operation is called, the player refers to this table to determine the corresponding Movie Object/BD-J Object that is to be executed.

The Index Table also has an entry to a Movie Object/BD-J Object designated for “First Playback” – this can be used by Content Providers to perform automatic playback. When the disc is loaded, the player refers to the “First Playback” entry to determine the corresponding Movie Object or BD-J Object that shall be executed.

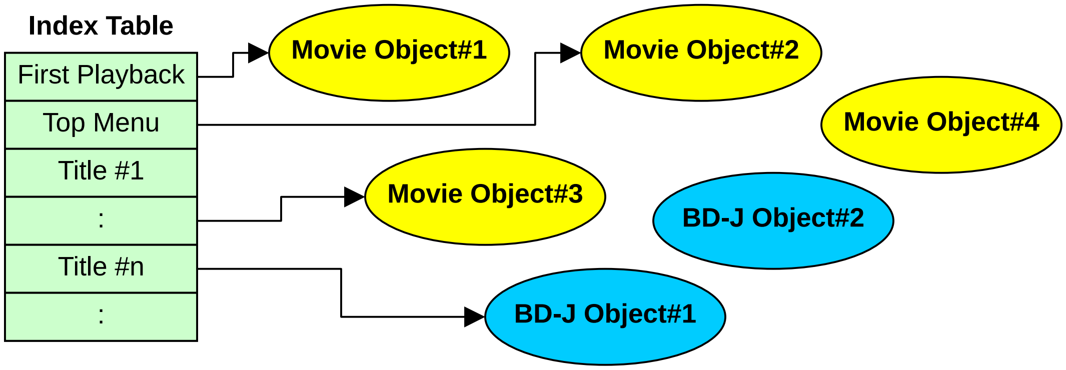


Figure 3-2 – An illustration of Index Table

3.1.2 Movie object

A “Movie Object” consists of an executable navigation command program (HDMV program). This enables dynamic scenario description. Movie Objects exist in the layer above PlayLists.

Navigation commands in a Movie Object can launch PlayList playback or another Movie Object. This enables the Content Provider to define a set of Movie Objects for managing playback of PlayLists in accordance with a user’s interaction and preferences.

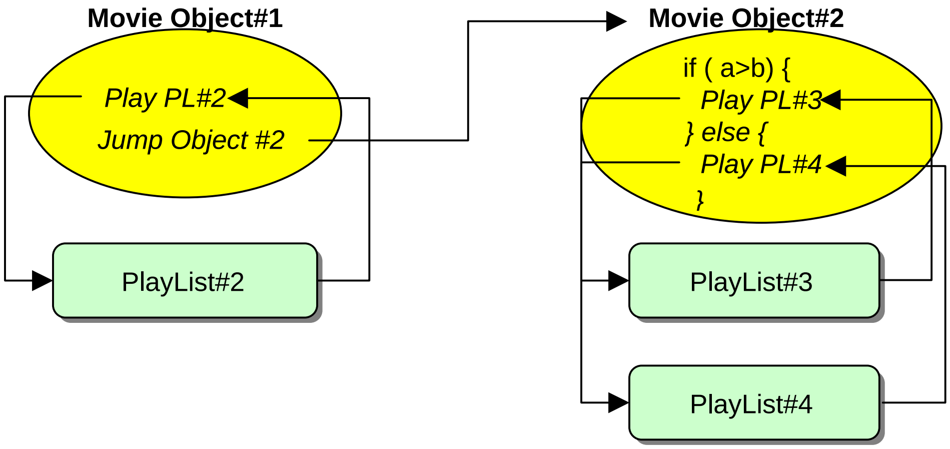


Figure 3-3 – An illustration of Movie Object and PlayList

3.1.3 Java Xlet

When a Title associated with a BD-J Object in the Index Table on disc is selected, the corresponding application is automatically launched and its lifecycle is bound to the Title. A BD-J application is a Java Xlet which is controlled by the BD-ROM player's Application Manager through its Xlet interface. The Xlet interface has four states as follows: loaded, paused, active and destroyed. Once a BD-J application is destroyed, any resources allocated to it, such as memory and AV control, shall be released.

3.1.4 Movie PlayList

A "Movie PlayList" is a collection of playing intervals in the Clips. One such playing interval is called a *PlayItem* and consists of an IN-point and an OUT-point, each of which refers to positions on a time axis of the Clip. Therefore a PlayList is a collection of PlayItems. Here the IN-point means a start point of a playing interval, and the OUT-point means an end point of the playing interval.

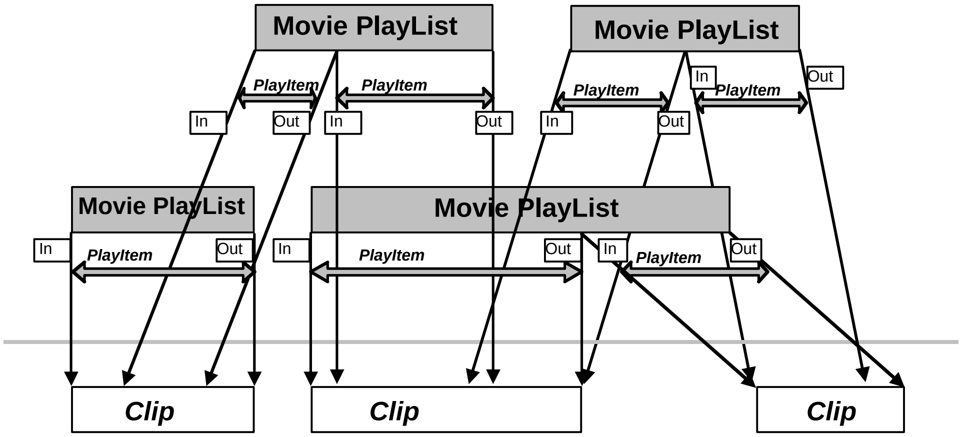


Figure 3-4 – An illustration of Movie PlayList

3.1.5 Clip

An AV stream file together with its associated database attributes is considered to be one object. The AV stream file is called a "*Clip AV stream file"*, and the associated database attribute file is called a "*Clip Information file"*.

An object consisting of a Clip AV stream file and its corresponding Clip information file is called a ***Clip***.

(1) Clip AV stream file

A Clip AV stream file stores an MPEG-2 Transport Stream (ISO/IEC 13818-1) contained in a structure compliant with the BD-ROM AV specification. This structure is called the "*BDAV MPEG-2 Transport Stream"*, an overview of which is provided in Section 4.1.

(2) Clip Information file

The Clip Information file stores the time stamps of the access point into the corresponding AV stream file. The Player reads the Clip Information to find out the position where it should begin to read the data from the AV stream file.

There is a one-to-one relationship between a Clip AV stream file and a Clip Information file.

3.2 Mechanism to realize “Browsable Slideshow”, “Pop-up Menu” and “Text subtitle” applications

BD-ROM provides a framework to realize “Browsable Slideshow”, “Pop-up Menu” and “Text subtitle” applications by providing a mechanism called “sub-path”.

The sub-path is a structure that enables the author to easily add a separate (out-of-mux) stream to the main Clip on the main-path.

Figure 3-5 shows the relation between the main-path and the sub-path in case of Text subtitle application. As shown in the figure, a SubPlayItem is logically added to the PlayItem in the PlayList to associate a separate Clip (SubPath) to the PlayItem on the Main path. The text subtitle presentation path using the SubPlayItem is synchronized with the main path using PlayItems in the PlayList.

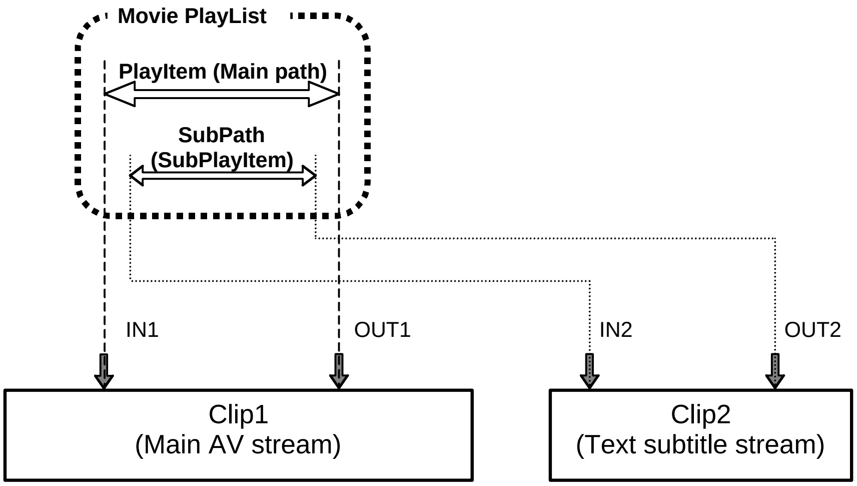
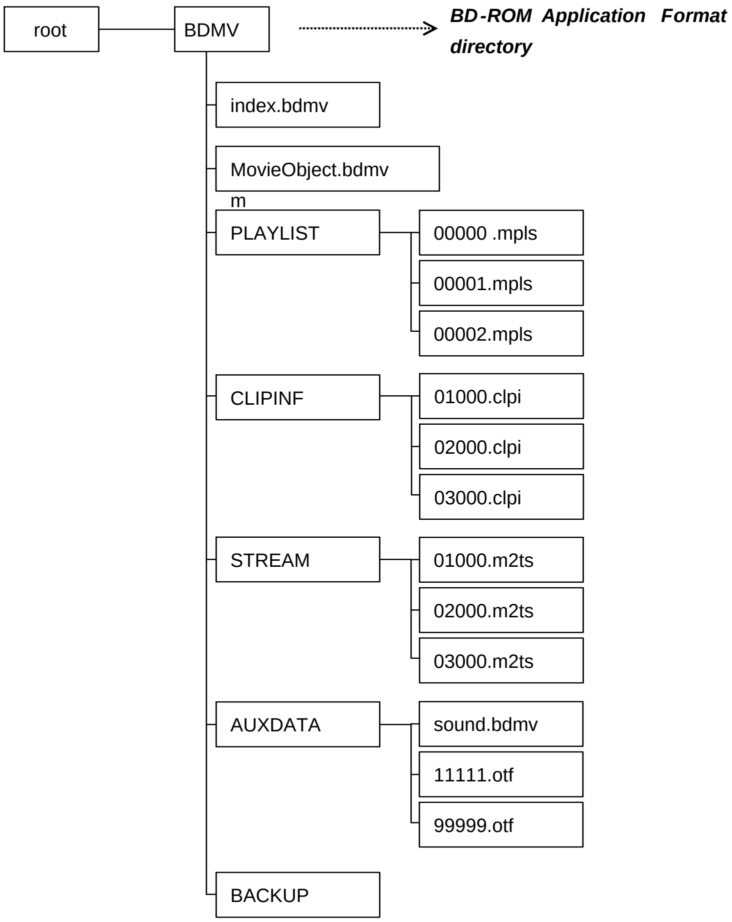


Figure 3-5 - An example of Main path and Sub path structure with Text subtitle presentation path

3.3 Directory and File structure

All BD-ROM application files are stored under a “BDMV” directory.



1) “BDMV” directory:
The BDMV directory contains the PLAYLIST, CLIPINF, STREAM, AUXDATA and BACKUP directories.

2) “PLAYLIST” directory:
The PLAYLIST directory contains the Database files for Movie PlayLists.

3) “CLIPINF” directory
The CLIPINF directory contains the Database files for Clips.

4) “STREAM” directory
The STREAM directory contains AV stream files.

5) “AUXDATA” directory
The AUXDATA directory contains Sound data files and Font files.

6) “BACKUP” directory
The BACKUP directory contains copies of the "index.bdmv” file, the “MovieObject.bdmv” file, all the files in the PLAYLIST directory and all files in the CLIPINF directory.

7) “index.bdmv” file

The “index.bdmv” file stores information describing the contents of the BDMV directory. There is only one index.bdmv file under the BDMV directory and its filename is fixed to “index.bdmv”.

8) “MovieObject.bdmv” file

The “MovieObject.bdmv” file stores information for one or more Movie Objects. There is only one MovieObject.bdmv under the BDMV directory and its filename is fixed to “ MovieObject.bdmv “.

9) “xxxxx.mpls” file

The “xxxxx.mpls” files store information corresponding to Movie PlayLists. One file is created for each Movie PlayList. The filenames of these files are in the form “xxxxx.mpls”, where “xxxxx” is a 5-digit number corresponding to the Movie PlayList.

10) “zzzzz.clpi” file

The “zzzzz.clpi” files store Clip information associated with a Clip AV stream file. The filenames of these files are in the form “zzzzz.clpi”, where “zzzzz” is a 5-digit number corresponding to the Clip.

11) “zzzzz.m2ts” file

The “zzzzz.m2ts” files contains a BDAV MPEG-2 transport stream. The names of these files are in the form “zzzzz.m2ts”, where “zzzzz” is a 5-digit number corresponding to the Clip. The same 5-digit number “zzzzz” is used for an AV stream file and its associated Clip information file.

12) “sound.bdmv” file

The “sound.bdmv” file stores data relating to one or more sounds associated with HDMV Interactive Graphic streams applications. This file may or may not exist under the AUXDATA directory. If it exists, there shall be only one sound.bdmv file and its filename is fixed to “sound.bdmv “.

13) “aaaaa.otf” file

The “aaaaa.otf” file stores the font information associated with Text subtitle applications. The names of these files are in the form “aaaaa.otf”, where “aaaaa” is a 5-digit number corresponding to the Font.

4 MPEG2 Transport stream for BD-ROM

The Blu-ray Disc Prerecorded application format (BD-ROM) and the Blu-ray Disc Recordable application format (BD-RE) use a common format for stream multiplexing – this format is based on the MPEG-2 Transport Stream industry standard (ISO/IEC 13818-1).

4.1 BDAV MPEG-2 Transport Stream

A MPEG-2 Transport Stream is stored in a Clip AV stream file in a structure known as the *"BDAV MPEG-2 Transport Stream"*. A BDAV MPEG-2 Transport Stream conforms to the data structure illustrated in Figure 4-1. The BDAV MPEG-2 Transport Stream is constructed from one or more "Aligned units", where:

1) The size of an Aligned unit is 6144 bytes (2048*3 bytes).
2) An Aligned unit starts from the first byte of the source packets.
3) The length of a source packet is 192 bytes. One source packet consists of one TP_extra_header structure and one MPEG2 transport packet structure. The length of the TP_extra_header structure is 4 bytes and the length of the transport packet structure is 188 bytes.
4) One Aligned unit consists of 32 source packets.

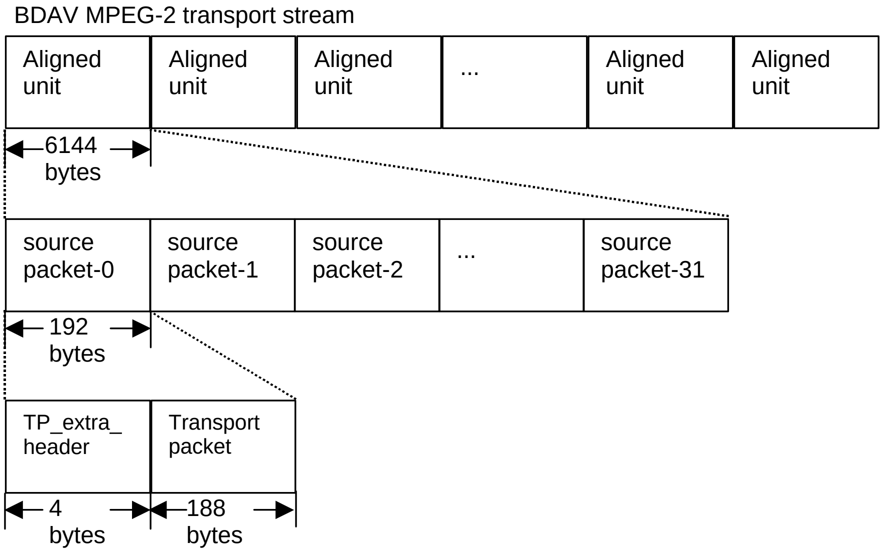


Figure 4-1 – Structure of BDAV MPEG-2 transport stream

Aligned units are recorded in three consecutive logical sectors on the BD-ROM disc. The size of one logical sector is 2048 bytes. The maximum multiplex rate of the BDAV MPEG-2 Transport Stream is 48Mbps.

The decoder model of the BDAV MPEG-2 Transport Stream is described in the “White Paper: BD RE - Logical and Audio Visual Application Format Specifications” available at:
“http://www.blu-raydisc.com/”.

4.2 Elementary streams in BDAV MPEG-2 Transport Stream

Video, audio, graphics and text elementary streams are coded in the PES packet payload of the BDAV MPEG-2 Transport Stream. The coding method for each of these elementary streams is specified in Figure 4-2 below.

Name of elementary stream	**Coding method of elementary stream**
Video stream	SMPTE VC-1 video stream
	MPEG-4 AVC video stream
	MPEG-2 video stream
Audio stream	Linear PCM audio stream
	Dolby Digital audio stream
	Dolby Digital Plus audio stream
	Dolby Lossless audio stream
	DTS digital surround audio stream
	DTS-HD audio stream
Graphics stream	Presentation graphics stream
	Interactive graphics stream
Text subtitle stream	HDMV Text subtitle stream

Figure 4-2 - Elementary streams in the BDAV MPEG2 Transport Stream

4.3 Video streams

Video streams shall be MPEG-2 video format (ISO/IEC 13818-2) compliant, MPEG-4 AVC video format (ISO/IEC 14496-10) compliant or SMPTE VC-1 video format compliant.

The video formats shown in Figure 4-3 can be used for BD-ROM video streams.

Video	CODECS	MPEG-2: MP@HL and MP@ML
		MPEG-4 AVC: MPEG-4 AVC: HP@4.1/4.0 and MP@4.1/4.0/3.2/3.1/3.0
		SMPTE VC-1: AP@L3 and AP@L2
	Max. bitrate	40Mbps
	HD	1920x1080x59.94-i, 50-i (16:9) 1920x1080x24-p, 23.976-p (16:9) 1440x1080x59.94-i, 50-i (16:9) MPEG-4 AVC / SMPTE VC-1 only 1440x1080x24-p, 23.976-p (16:9) MPEG-4 AVC / SMPTE VC-1 only 1280x720x59.94-p, 50-p (16:9) 1280x720x24-p, 23.976-p (16:9)
	SD	720x480x59.94-i (4:3/16:9) 720x576x50-i (4:3/16:9)

Figure 4-3 - Specification of BD-ROM Video streams

4.4 Audio streams

BD-ROM supports six types of audio stream formats ranging from a low bit rate to high audio quality, as shown in Figure 4-4 below.

Audio	CODEC	LPCM	Dolby Digital	Dolby Digital Plus	Dolby Lossless	DTS digital surround	DTS-HD
	Max.bitrate	27.648Mbps	640kbps	4.736Mbps	18.64Mbps	1.524Mbps	24.5Mbps
	Max.ch	8(48kHz, 96kHz), 6(192kHz)	5.1	7.1	8(48kHz, 96kHz), 6(192kHz)	5.1	8(48kHz, 96kHz), 6(192kHz)
	bits/sample	16, 20, 24	16 - 24	16 - 24	16 - 24	16, 20, 24	16 - 24
	Sampling frequency	48kHz, 96kHz, 192kHz	48kHz	48kHz	48kHz, 96kHz, 192kHz	48kHz	48kHz, 96kHz, 192kHz

Figure 4-4 – Specification of BD-ROM Audio streams

4.5 Presentation Graphics and Interactive Graphics streams

BD-ROM provides two types of graphics streams as shown in Figure 4-5 below. The Presentation Graphics stream (available in HDMV and BD-J) is intended for Subtitles and Animated Graphics, and the Interactive Graphics (available only in HDMV) is intended for Menu Graphics.

Graphics	Plane size	1920x1080/1280x720/720x480/720x576	
	Color	8bit Index lookup table(24 bit color + 8 bit alpha)	
	Compression	Run Length Encoding	
	Presentation planes	2 planes	
	Presentation Plane name	Presentation Graphics	Interactive Graphics
	Main usage	Subtitles	Menus
	Animation Effects	Fade In/Out, Color changing, Wipe In/Out, Scrolling	

Figure 4-5 – Specification of BD-ROM Graphics streams

4.6 Text subtitle streams

BD-ROM also supports Text subtitle streams. A Text subtitle is defined by a series of character codes plus font and style information. Text subtitles are available in addition to the Bitmap based Subtitles provided by BD-ROM Presentation Graphics streams.

Text Subtitle	Plane size	1920x1080/1280x720/720x480/720x576
	Character encoding	Unicode2.0 (UTF-8 and UTF-16BE), Shift-JIS, KSC 5601-1987 (including KSC 5653), GB18030-2000, GB2312, BIG5
	Presentation Plane name	Presentation Graphics
	Color	8bit Index lookup table(24 bit color + 8 bit alpha)
	Effect	Fade In/Out, Color changing
	Presentation style	Text position/flow/alignment, font style/size/color
	User change-able style	Text position, font size, line space

Figure 4-6 – Specification of BD-ROM Text subtitle streams

5 HDMV mode

The BD-ROM HDMV platform provides a flexible, simple framework for creation of interactive High Definition and Standard Definition movie experience applications. This section will provide an overview of some of the key features provided in HDMV.

5.1 Core functions

5.1.1 Out-of-Mux stream Framework

HDMV provides a framework for individual stream handling. An Out-of-Mux stream is an additional stream which is decoded while the main MPEG stream is decoding. The Out-of-Mux framework provides support for new applications such as Pop-Up Menus, Browsable Slideshow with background music, Button click sound and Text subtitle display.

Decoder model

The HDMV decoder model is equipped with two read buffers, two preloading buffers and two switches. The second read buffer enables the supply of an Out-of-Mux audio stream to the decoder even while the main MPEG stream is being decoded. The preloading buffers cache Text subtitles, Interactive Graphics and sounds effects (which are presented at Button selection or activation). The preloading buffer stores data before movie playback begins and supplies data for presentation even while the main MPEG stream is being decoded.

This switch between the ECC decoder and buffers selects the appropriate buffer to receive demodulated packet data from any one of read buffers or preloading buffers. Before starting the main movie presentation, effect sounds data (if it exists), text subtitle data (if it exists) and Interactive Graphics (if preloaded Interactive Graphics exist) are preloaded and sent to each buffer respectively through the switch. The main MPEG stream is sent to the primary read buffer (RB1) and the Out-of-Mux stream is sent to the secondary read buffer (RB2) by the switch.

The audio decoder also has a switch to select a read buffer for source audio data. In the case of a Browsable Slideshow with background music, the switch selects the secondary read buffer (RB2) to store an Out-of-Mux audio stream and continue supplying the audio stream to the decoder. In all other cases the switch selects the primary read buffer (RB1).

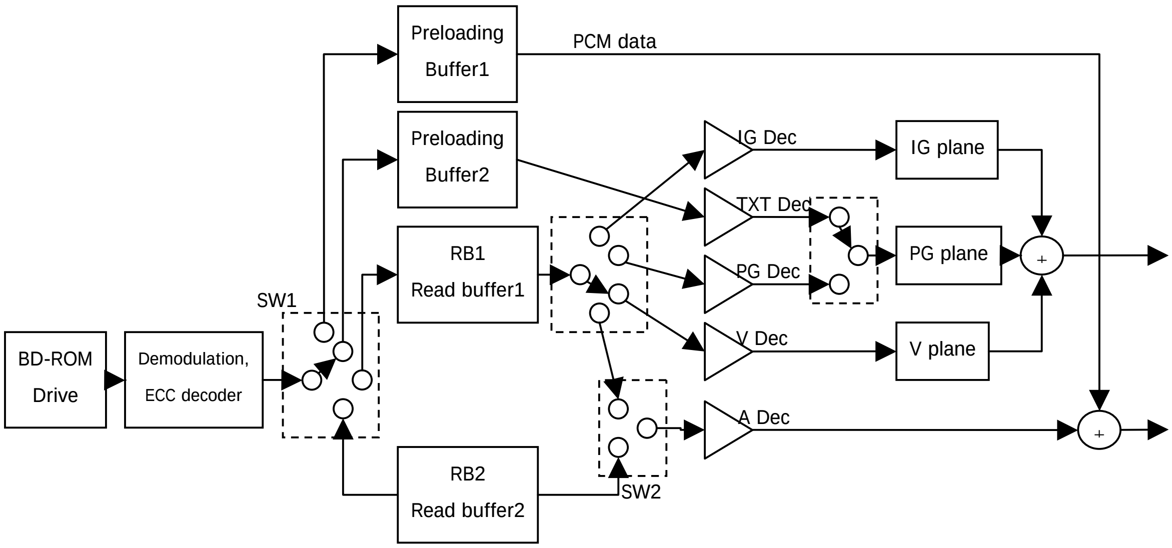


Figure 5-1 – Decoder model

5.1.2 Graphics Framework

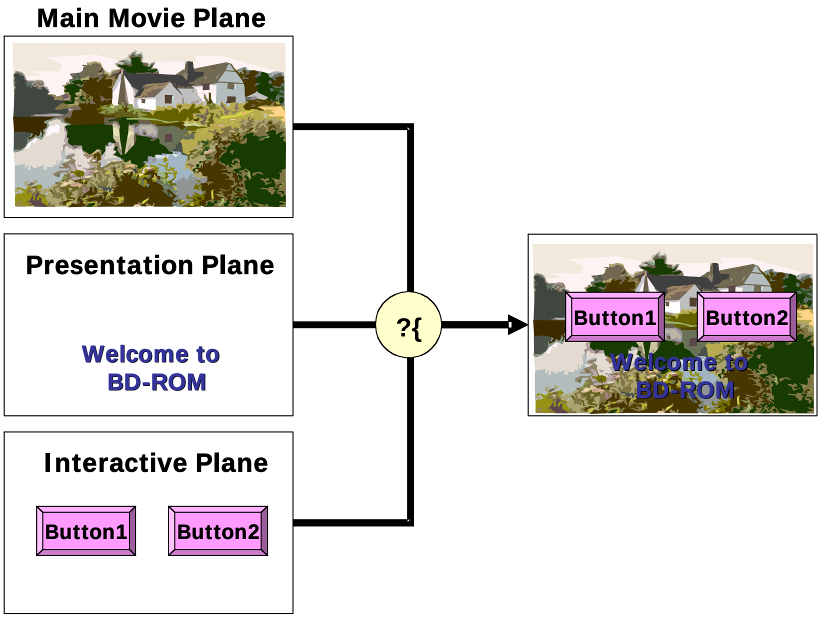
HDMV provides two graphics frameworks for compositing graphics on video: the Interactive Graphics system and the Presentation Graphics system.

A BD-ROM Interactive graphics stream contains information required to provide a series of interactive displays, which appear and disappear with frame accuracy, that are supplemental to an associated HDMV presentation. It is envisaged that Interactive graphics streams will typically be used to provide both the display and associated commands of graphical interactive displays during a HDMV presentation.

A BD-ROM Presentation graphics stream, available in both HDMV and BD-J modes, contains information required to provide non-interactive images that are supplemental to an associated BD-ROM presentation. The images described in the stream are designed for graphic overlay, with frame accuracy, on the associated video image. It is envisaged that BD-ROM Presentation graphics streams will typically be used to provide subtitle services and/or other animated graphics during a HDMV or BD-J presentation.

1) Graphics planes

As shown in Figure 5-2, HDMV defines two independent full HD resolution (1920x1080) graphics planes for graphics which are composited on the video plane. One graphics plane is assigned for subtitling applications (Presentation Graphics or Text Subtitles) and the other plane is assigned to interactive applications (HDMV or BD-J mode interactivity graphics).



BD-ROM is designed to have multiple image planes (main movie, subtitle, graphics & button) in the reference player model, which are then combined into one image at display. This allows the movie image, subtitle and buttons to be independently controlled, therefore increasing compatibility and ease of implementation compared to existing media.

Figure 5-2 – Graphics planes

Each graphics plane has 8-bits per pixel, with each pixel value referring to an index entry in a Palette for translation to YCrCb color and 8-bit (256level) alpha. This color capability offers an enhanced visual experience and allows compelling content to be displayed using the HDMV Interactive Graphics system.

2) Graphics model

The HDMV graphics systems define a flexible decoding and composition system for providing graphics displays whereby graphic images may be reused, with different effects applied, in one or more graphics displays.

A HDMV graphics stream consists of one or more “Segments” – “Segments” are the basic syntactical element of HDMV graphics streams. There are three types of Segments - Graphics Object Segment, Composition Segment and Palette Segment:

- Composition Segment – defines the appearance of a graphics display.
- Graphics Object Segment – Bitmap image data compressed with an RLE compression schema.
- Palette Segment – color and transparency data (up to 256 entries) for translation of 8bit index color to full color when compositing on the video plane.

Segments are processed by the BD-ROM HDMV graphics decoder as shown in Figure 5-3 below.

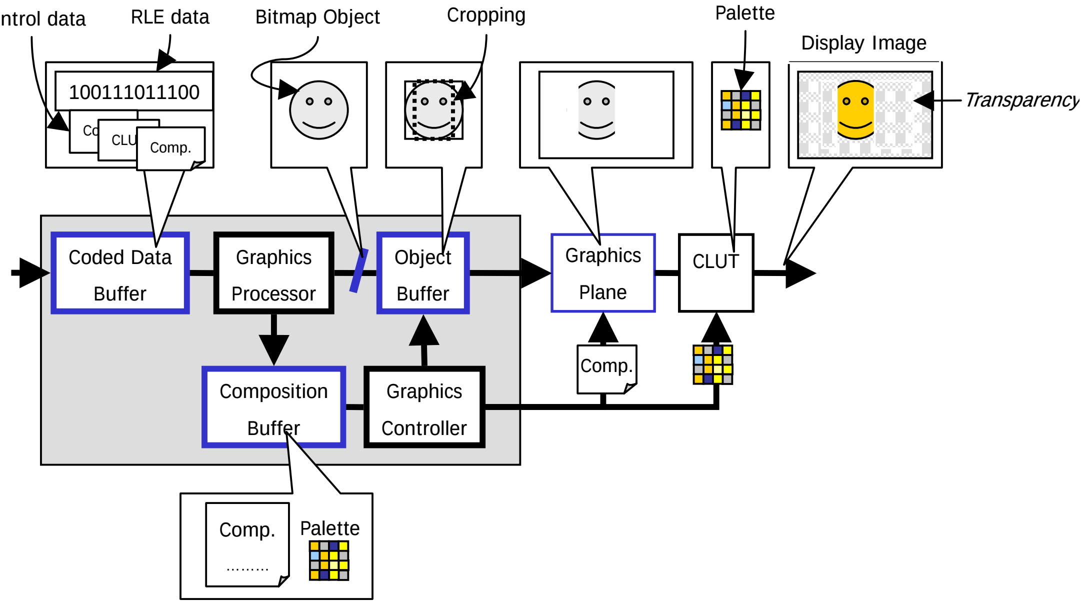


Figure 5-3 – Illustration of BD-ROM HDMV Graphics decoding

A Segment first arrives at the Coded Data Buffer. The Graphics Processor extracts the Segment at the time defined by a system time-stamp associated with the Segment. When Composition and Palette Segments arrive at the Graphics processor, they are decoded to the Composition Buffer.

When Graphics Object Segments arrive at the Graphics Processor, the Graphics Processor decodes the Graphics Object to an uncompressed 8-bit graphics object which is then stored in the Object Buffer. Once a Graphics Object has been decoded, it is available for use by one or more graphics displays as described in Composition Segment.

The Graphics Controller is responsible for compositing graphics images on to the graphics plane in accordance with the description in the Composition Segment. The composited image on the graphics plane is transformed to full color and transparency by the CLUT module and then overlaid on the video image. The decoder implements a Pipelined decoding model such that Graphics Displays may be assembled in the Graphics Plane while, at the same time, new Graphics data is decoded into the Object Buffer.

3) Graphics animations

Support for graphics effects is part of the graphics tool set for Content Providers to create rich BD-ROM Graphics Displays. Supported effects include scrolls, wipes, cuts, fades (transparency changes) and color changes. All of these effects may be utilized in both Interactive (e.g. to be used for Menu page transitions) and Presentation Graphics streams (e.g. to be used for advanced Subtitles or Karaoke).

Composition Segments indicate the Graphics Objects to be used for a graphics display and may define a cropping transform to be applied when compositing the Graphics Object. Composition Segments also indicate the Palette to be used for the graphics display. Effects are realized by providing multiple

Compositions Segments which change cropping areas of Graphics Objects (e.g. to provide wipes, scrolls and cuts) as illustrated in Figure 5-4 and/or reference different Palettes (e.g. to provide fades or color changes).

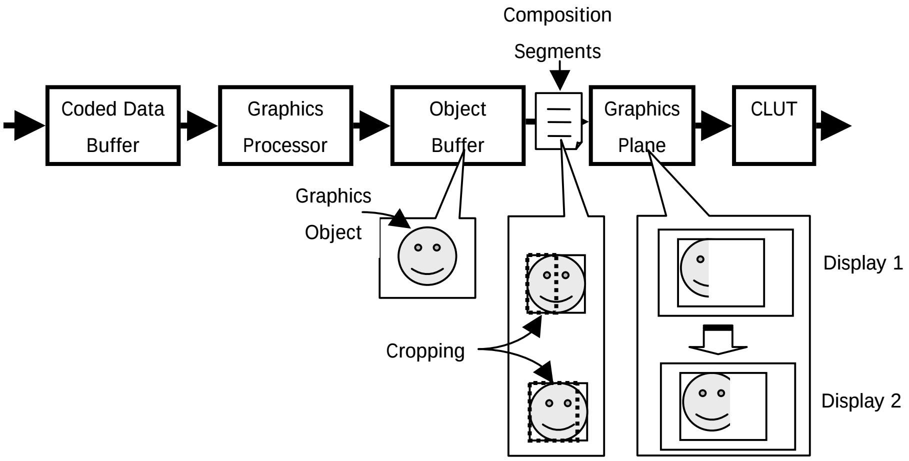


Figure 5-4 – Illustration of wipe effect

HDMV Interactive Graphics are further extended to support animated sequences of graphics for Buttons. The Normal, Selected and Activated states of a Button may be animated with a sequence of different images. With 8-bit index color and transparency support along with support for frame rates up to the underlying video frame rate, the creative possibilities are greatly expanded over existing formats.

5.1.3 Text Subtitle Framework

HDMV provides support for Text based subtitles. This framework enables the Content Provider to create Subtitling applications using character code sequences (text) plus style information. The Text subtitle stream is stored as an Out-Of-Mux stream (as described in Section 5.1.1) which does not impinge on the bandwidth of the main AV stream thereby enabling the Content Provider to supply several Subtitle streams without affecting the quality of the associated audio and video presentation. The BD-ROM player can present either a Text Subtitle stream or a Presentation Graphics Subtitle stream – it will not present both at the same time.

A Text subtitle stream consists of a sequence of “Text Dialogs”, each of which consists of a text to be presented. Text Dialogs contain up to a maximum of two Text Regions, each of which may be presented with an individual presentation Style. All text in a Text Dialog shall be presented during the same presentation time slot which is frame-accurately synchronized with the main AV stream.

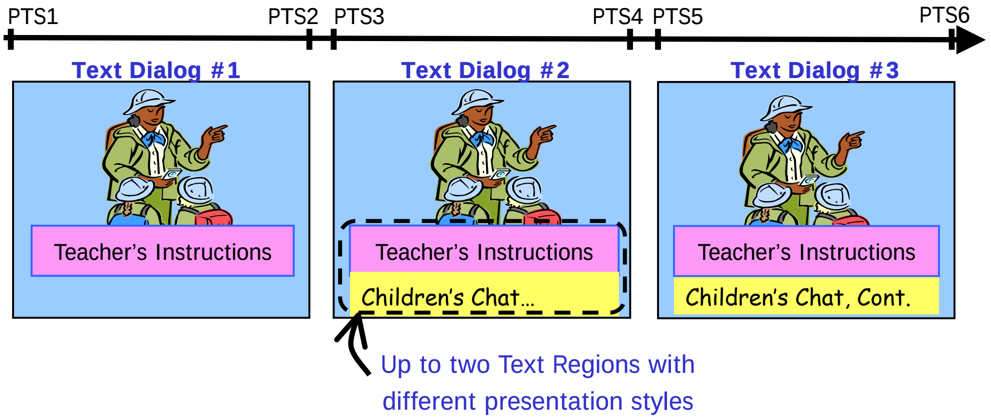


Figure 5-5 – Illustration of Text Dialogs and Text Regions

Text subtle streams are rendered to an 8-bit index image and a Palette is applied for translation to YCrCb color and 8-bit (256level) alpha prior to display.

The Text subtitle framework supports various presentation styles as shown in Figure 5-6:

- A "Text Region" is defined by position and size within the Graphic plane. Each Text Region can have a unique background color.
- A "Text box" is defined by position and size within a Text Region. Text boxes are used for the placement of text for display.
- Text boxes define the style of text for display as follows:
    - Text arrangement using text flow (e.g. left-to-top progression), text alignment (e.g. left alignment) and line space style attributes.
    - Font styles using font type (e.g. Arial, Courier), font style (e.g. bold, italic), font size and font color can be set for each Text Region.
- In addition to the Text box style, variations of font styles are allowed for character by character (known as "in-line" styles).

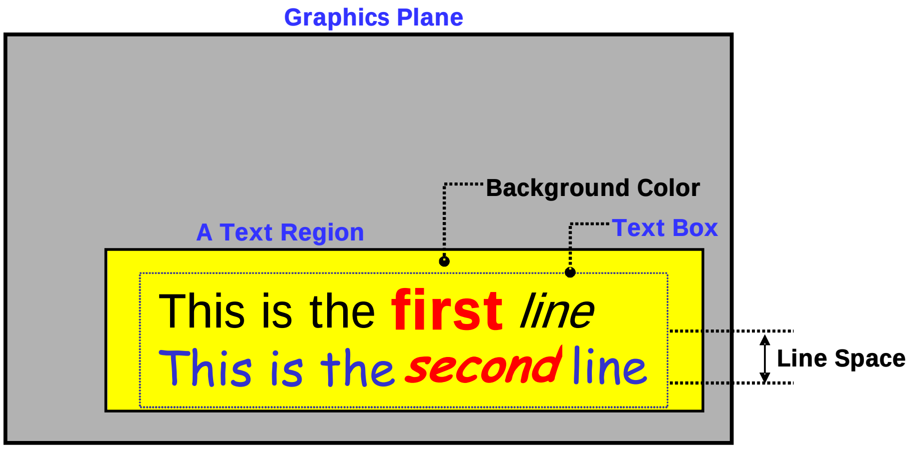


Figure 5-6 – Illustration of Text Region and Text Box

Figure 5-7 shows an example of some temporal effects which can be performed using the HDMV Text subtitle format. Fading of text is realized by simple Palette changes. Seamless presentation between individual Text subtitle displays is guaranteed.

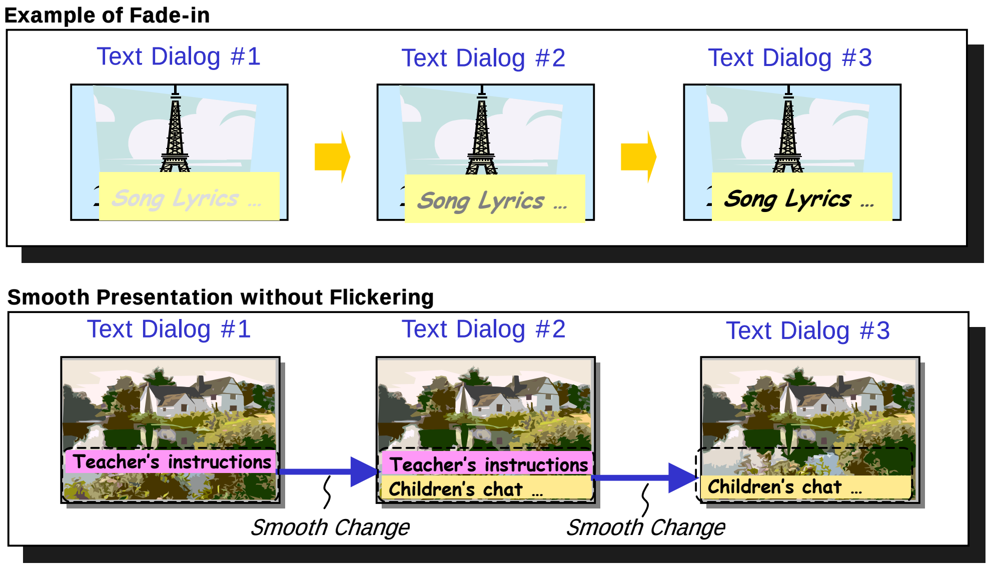


Figure 5-7 – Example of Text subtitle effects

5.1.4 Interactivity Framework

1) Pop-Up Menus

HDMV Interactive Graphics support a “Pop-Up” Menu Interface: once playback of video has begun, HDMV graphical interactive content may be activated during the playback of video by pressing a ‘Pop-Up’ Button on the remote. In this case, video playback can continue while the HDMV Interactive graphics are on the screen or video playback may be paused – this is determined by the Content Provider using navigation commands.

Menus that support a “Pop-Up” Menu Interface are always pre-loaded. As shown in Figure 5-8, several pages of HDMV Interactive Graphics data can be pre-loaded before playback starts. This Interactive Graphics data is kept in memory during playback of the AV stream and is not displayed until requested by the user.

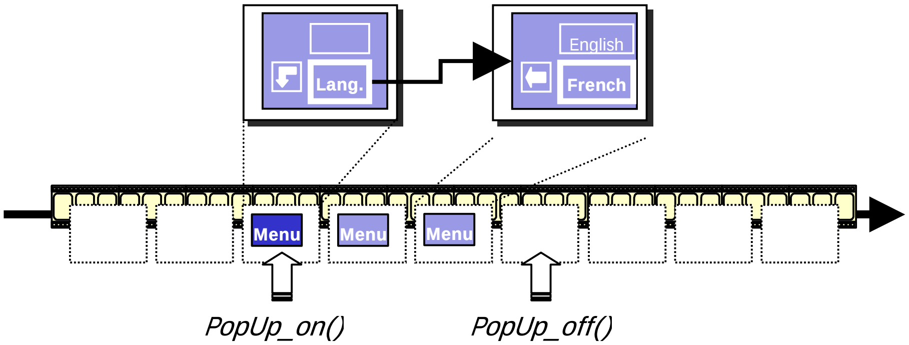


Figure 5-8 – Illustration of Pop-Up Menus

2) Always-On Menus

HDMV Interactive Graphics support an "Always-On" Menu Interface; Interactive Graphics content that cannot be removed from the screen by user request is called "Always-On". This is one of the methods provided by HDMV to present interactivity to the user and is similar to that provided by DVD-Video. For example, a Menu implemented with the Always-On interface may be presented to the user when the disc is inserted into the player.

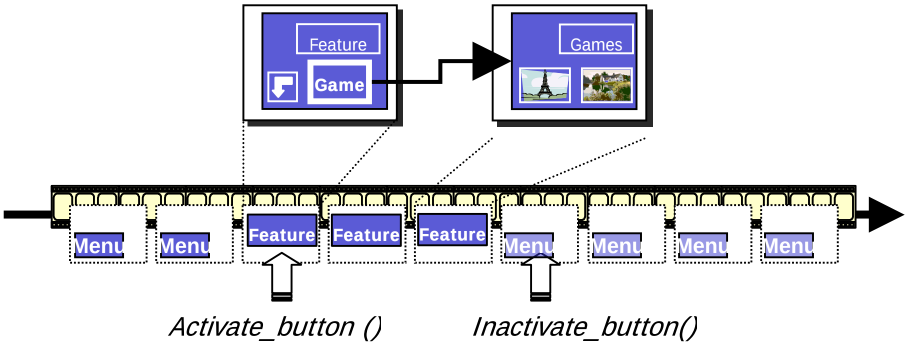


Figure 5-9 – Illustration of Always-On Menus

Menus that support an "Always-On" Menu Interface may be pre-loaded or multiplexed with video. If the HDMV Interactive Graphics stream is multiplexed with video, PTS/DTS timestamps can frame accurately determine when the Always-On Menu shall appear and disappear.

3) Multi-page Menus

The HDMV Interactive Graphics framework provides a scheme for Menu Page definition, thereby allowing a large amount of data to be presented in an organized manner with special commands available for inter-page navigation. When a Button is activated, a corresponding navigation command is executed which causes the display to change to a specified page. This action is performed with no visible interruption to the screen allowing a seamless user experience.

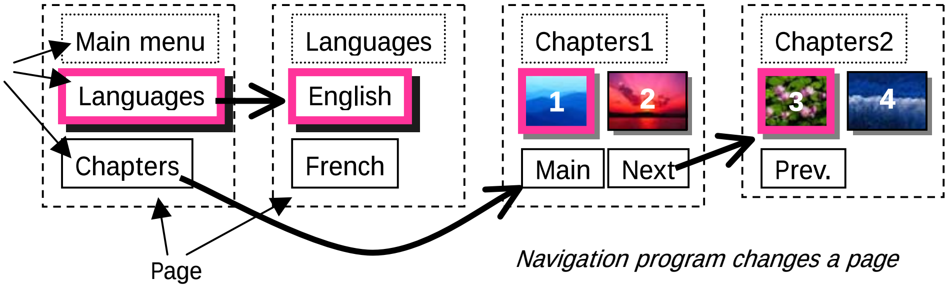


Figure 5-10 – Example of Multi-page Menu

4) Button enabling and disabling

The HDMV Interactive Graphics framework also provides a scheme for dynamic graphics display. On a single page, this enables the Content Provider to determine dynamically which Buttons are visible and invisible at any point in time. This scheme could be used, for example, to provide Buttons that present a set of options and when one of those Buttons is selected, additional Buttons appear. When a Button is enabled it becomes visible and can be navigated to. This action is performed with no visible interruption to the screen allowing a seamless user experience. The author may choose to either keep the earlier Buttons accessible or disable them which would clear them from the display.

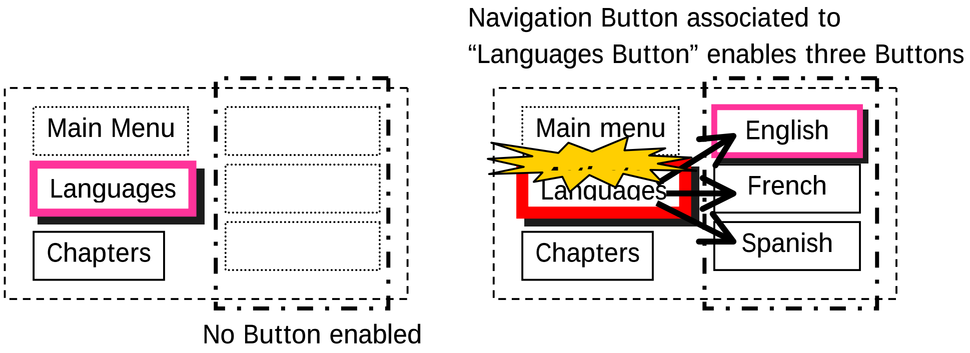


Figure 5-11 – Example of Button enabling and disabling

5.1.5 Command Framework

HDMV provides a simple programming platform to enable the Content Provider to author interactive movie contents, such as those seen in the DVD-Video market. This platform provides a scheme to manage the behavior of Menus, Browsable Slideshow pages and so on.

There are two types of Objects which contain navigation commands – the Movie Object and the Button Object. A Movie Object is executed when the Title associated with the Movie Object begins playback. Movie Object navigation commands are used to manage PlayList playback. While a PlayList is under playback, the state of a Movie Object is maintained and resumes after PlayList playback is terminated. A Button Object is an alternative programming method that is available while the PlayList is under playback and a Button Object is executed by user activation or system timer.

Programming commands and Registers

HDMV navigation commands have three operation groups: playback operation group, compare operation group, and arithmetical and bitwise operation group. The playback operation group manages PlayList playback, execution of Movie Objects, execution of Titles and control of the Graphics display (Button enabling and disabling). The comparison operation group provides comparison functions between parameters and/or given values and provides a Boolean result.

The player has two types of Registers: General Purpose Registers and Player Status Registers. General Purpose Registers provide the Content Provider with 4096 4-bytes unsigned registers. Player Status Registers represent the Player’s playback status, configuration and preferences.

5.2 Application Examples

5.2.1 Interactive Menus

The HDMV Interactive Graphics framework is used to provide interactive Menu displays. For instance, changing the display image of selected Buttons and changing the graphics display of the page (Buttons appearing and disappearing) with Button activation. This framework enables the author to provide flexible Menu navigation while the movie is presented.

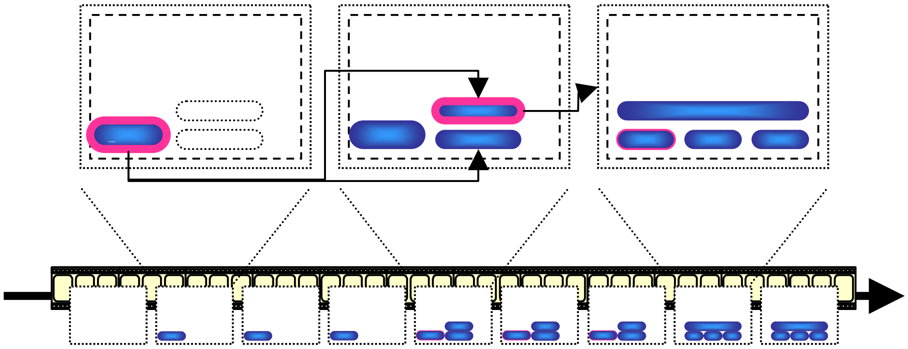


Figure 5-12 – Example of Multi-page Menu with dynamic Button display

5.2.1.1 Browsable Slideshow with background music

The HDMV decoder enables simultaneous decoding of still picture data and audio data. Since the decoding process of each is independent, the decoder enables the user to freely change (skip next or skip back) the still picture while not interrupting the audio presentation. This means that background music can continuously be presented during a Browsable Slideshow application.

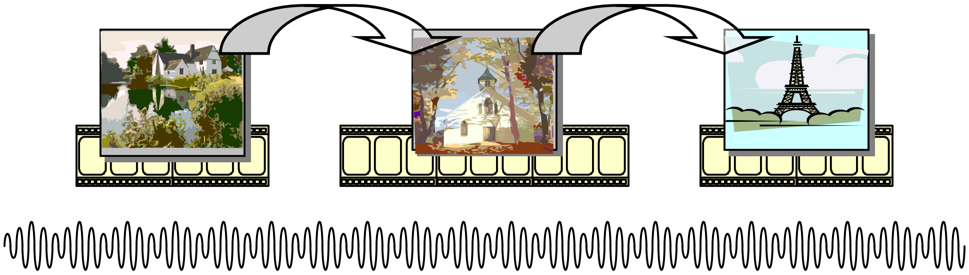


Figure 5-13 – Example of Browsable Slideshow with background music

5.2.1.2 Button sounds

The concept of Button Sounds is available in HDMV. Both the Select and Activate actions may be associated with short duration sounds which are mixed with the underlying audio.

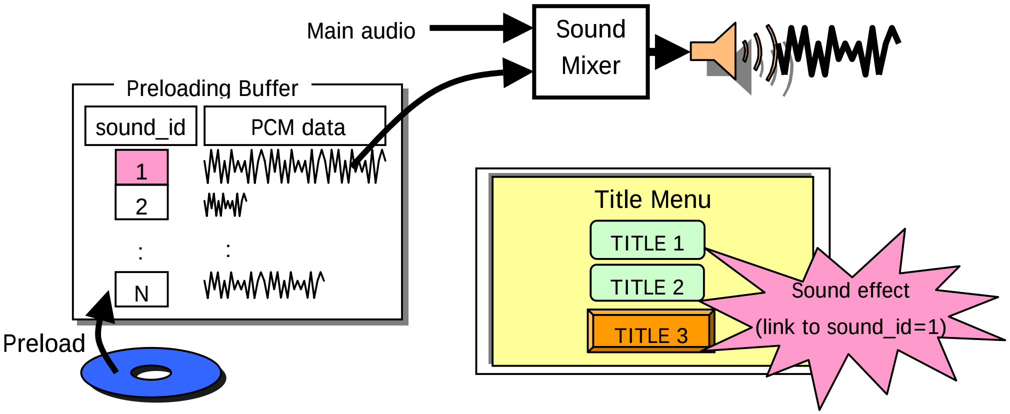


Figure 5-14 – Example of Button sounds

5.2.2 User Changeable Subtitle styles

In addition to the basic style control made available in the Text subtitle format, the Content Provider can also define a set of various presentation positions and font sizes for the display region as user selectable styles. In this case, the user can change the presentation style by selecting one style from the set of user selectable styles as shown in Figure 5-15 below.

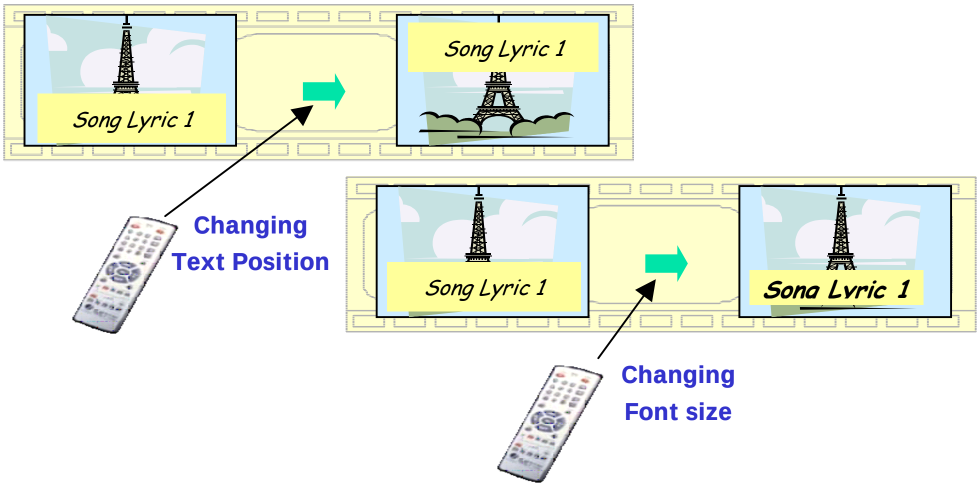


Figure 5-15 – User change to presentation styles

6 BD-J mode

This section will cover the main features of the BD-J mode platform which is shown in Figure 6-1 below.

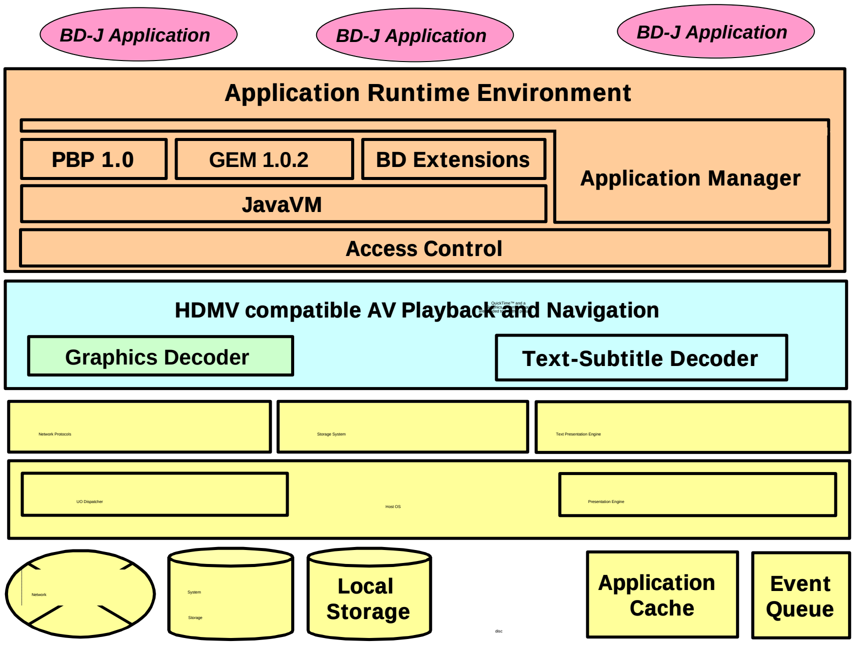


Figure 6-1 - Overall BD-J system model

BD-J is based on the Java 2 Micro-Edition (J2ME) Personal Basis Profile [PBP] – a Java profile that was developed for consumer electronics devices.

6.1 Core functions

6.1.1 Application Execution / Management

A key concept of BD-J is the BD-J Object. A BD-J Object is a Java Xlet that is registered in the Application Management Table (AMT). Each Title on a disc or even the disc itself can have an associated AMT.

At least one application in the AMT must be signaled as “autostart”. This application will be started when the corresponding Title is selected and from thereon the BD-J platform is used by the BD-J application. This could include selecting another Title and launching other applications signaled in the AMT or downloading from the Internet.

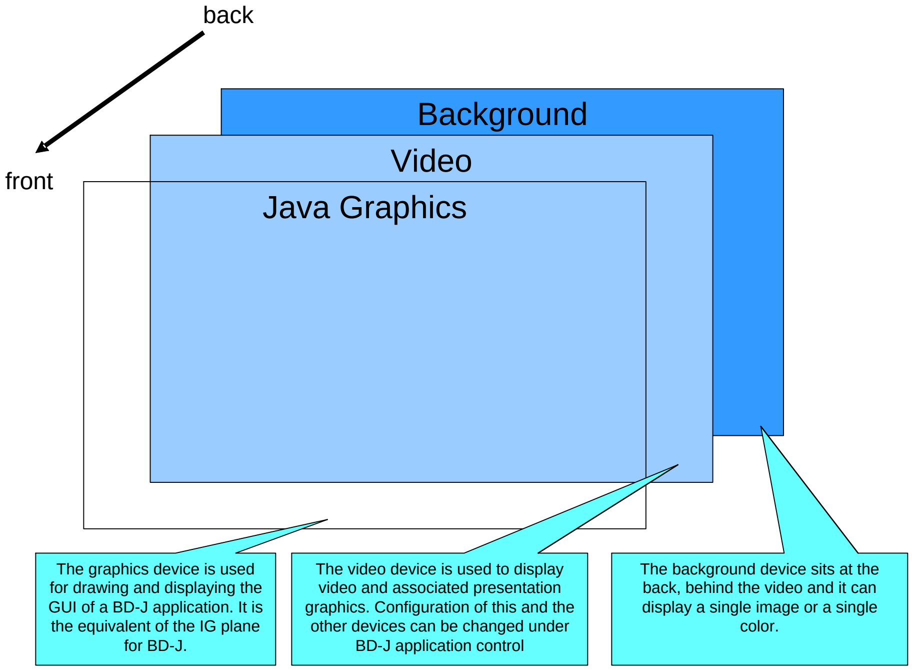
6.1.2 GUI framework and User Interface

BD-J includes a GUI framework that is suitable for a CE environment. A BD-J application’s GUI can be operated with a remote control with a required set of keys and an optional pointing device. The set of required keys includes at least the keys needed to support the User Operations in HDMV applications.

The GUI framework in BD-J includes the HAVi(6) UI framework mandated by [GEM]; it is not a desktop GUI framework like Swing or AWT. The GUI framework is based on the core of AWT as specified by PBP, but the widget set includes mechanisms for remote control navigation from GEM and easy customization of look and feel from HAVi.

6.1.3 Device model & HAVi

BD-J includes the HAVi device model that maps to the BD-ROM system resources. One of the devices supported in the model is the Screen device that is build up of a Background Device, a Video Device and a Graphics Device. The configuration of the Screen and its constituent devices is under control of the BD-J application, as shown in Figure 6-2 below.



The graphics device is used for drawing and displaying the GUI of a BD-J application. It is the equivalent of the IG plane for BD-J.

The video device is used to display video and associated presentation graphics. Configuration of this and the other devices can be changed under BD-J application control

The background device sits at the back, behind the video and it can display a single image or a single color.

Figure 6-2 – BD-J system device model

Supported resolutions of the devices are compatible with the supported video format resolutions. The Graphics Device uses a 32-bit color RGB color model.

BD-J graphics can use alpha for overlay with video. Additionally, the video can be scaled behind the BD-J graphics and the video background device can display a single image.

6.1.4 AV Playback and Navigation and Subtitle/Audio Language Control

BD-J includes a media framework similar to JMF for the playback of media content related to the BD-ROM disc. It is assumed that the BD-ROM disc will be the prime source for media files, but it will not be the only one; other sources could be the studio's web server and local storage.

[6] HAVi is owned by the HAVi Organization. For more information, please refer to: http://www.havi.org/

The unit of playback in BD-J is the Playlist, just as in HDMV. All features of HDMV, except Interactive Graphics, can be used by a BD-J Application. HDMV Interactive Graphics is replaced by BD-J graphics. Supported features include video, audio, Presentation Graphics, Text Subtitle component selection, media-time and playback-rate (trick-mode) control.

The BD-J Video Device is a combination of the HDMV Video and Presentation Graphics planes. Both Video and Presentation graphics will play back in the Video Device.

6.1.5 Other (static) content format functions (Graphics, Text, Audio Clips)

BD-J includes standard Java libraries for decoding and displaying images in JFIF (JPEG), PNG and other image formats. These images can be displayed on the Java graphics plane using standard Java graphics functions. An image can also be rendered in the background plane using a BD-J specific package.

Text can be rendered using standard Java text functions. These text-rendering functions are extended with a more advanced text layout manager that integrates with the BD-J UI framework. The text is rendered using a vector-based font either coming from the disc, the player (default font) or downloaded from the network.

Button sounds from HDMV can also be used by the Java UI framework. Sound files can be loaded and rendered as a reaction to the user pressing a key, or as a reaction on a marked event related to the movie - or as a reaction to any event generated by a BD-J Application.

6.1.6 Access control, security scheme, application authentication scheme

The BD-J environment uses the Java 2 security model to authenticate signed applications and to grant them permissions that go beyond the core functions (the BD-J defined sandbox).

The authentication scheme of BD-J applications is based on signing the JAR files that contain the applications. The relation between the authentication of BD-J applications coming from the disc and the BD Copy Protection System is out of scope for this paper, but an efficient workable scenario will be part of the BD-ROM Full Feature specification. The BD-J classloader will only load authenticated applications when the disc is in the player.

Authenticated applications can use a (signed) permission request file to acquire permissions that go beyond the BD-J sandbox. Permissions can be acquired for:

- Reading and writing to local and system storage
- Using the network connection (to connect to defined servers)
- Access of the file system on the BD-ROM disc
- Title selection of other titles on the BD-ROM disc
- Control of other running BD-J applications

6.1.7 Internet Connectivity & Download of New Contents/Applications

BD-J contains the Java network package. Java applications can use this package to connect to servers on the Internet. The physical connection might differ between implementations e.g. Ethernet, telephone line, etc. At the network level, TCP/IP is supported and the HTTP protocol may be used. Moreover, the Java package for secure connections is included (JSSE) as part of the BD-J platform. Before a BD-J application can use the network connection, it must be authenticated and have suitable permission to use the network.

The web sites to which the application will go are under full control of the Content Provider. This control is guaranteed in two ways:

- Only (disc) authenticated BD-J applications are allowed to run when the disc is played. The application controls the use of the network connection.
- In addition, permissions defined on the disc can restrict the use of the (TCP/IP) network connection to certain sites.

6.1.8 System/Local Storage

BD-J will include support for storage. Two flavors of storage are included – mandatory System Storage and optional Local Storage. All storage is accessed using methods from the Java IO package. The path for local storage is as specified by [GEM].

System storage is storage that will be present in all BD-J players. The required minimum size of this system storage will permit storage of application data like settings, high-scores etc. It will not be big enough to store downloaded AV material. For this purpose, optional local storage is available. Typically system storage will be implemented using Flash memory and the optional local storage will be implemented on a HDD.

Since storage is a shared resource between all discs played on the player, Java access control is part of BD-J. BD-J applications can only access a disc specific part of the storage space and cannot access the part belonging to other discs.

6.1.9 Binding scheme for on-the-disc and off-the-disc content

A binding scheme between media content (AV files, subtitles, Java applications files, database files) on the disc and content (related to the disc) stored on local storage is defined. This scheme enables a seamless user experience to be provided when playing back media data, regardless of the origin of the data.

6.2 Application Examples

BD-J allows many possible application types. In this section we will cover a few typical examples in more detail.

6.2.1 AV playback control

One of the main features of BD-J is playback of A/V material. A disc bound BD-J application can be created which is started when the disc is put into the player. This application could present a Menu on the screen, e.g. while playing an introduction of the movie in a scaled-down manner in a corner of the screen which allows language selection, chapter selection, and display of background information that might be retrieved from disc or from the Internet. Once the user selects playback of a Title, the disc application becomes invisible but allows the user to use trick modes with a simple on-screen GUI on top of the video (as long as the application on the disc allows this). The user also has the option of going back to the full-screen Menu of the disc application using one of the remote control keys.

BD-J features used in this example include: media control (including video scaling, playback speed, language component selection), GUI framework, and Internet connectivity.

6.2.2 Subtitle Updates

The BD-J application described above can be further extended to allow the user to obtain subtitles in a language that is not supported on the disc. The Content Provider can publish new or updated subtitle files on a website dedicated to the disc Title. The BD-J application on the disc can include the retrieval of this subtitle file and storage (in encrypted form) in the player's local storage. After storing the subtitle file, the application can select the new subtitles for a Title.

Only BD compliant players and trusted and authenticated applications will be able to do this and only from trusted and authenticated websites. The trust scheme will make use of the Java 2 security scheme and be tied to the CPS of the disc.

Additional BD-J features used in this example: downloading data into local storage, combined playback of subtitles from local storage with video from disc, merged file-system view, Java 2 security model.

6.2.3 Download new Movie trailer

When the Content Provider that published the disc is launching a sequel to the Title, they may also choose to publish a trailer for the sequel on their website, specifically for holders of the current title. A BD-J application, present on the disc, can connect to this website and see if there is new content available. The BD-J application can inform the user that a trailer for the new sequel movie is available e.g. by showing a number of (JPEG) images in the Main Menu. After the user has selected to view the trailer, the BD-J application downloads this trailer, while at the same time showing some background information on the actors in this sequel. When the download of the trailer to local storage is completed, the application plays it back, showing at the bottom of the screen the movie theatres where this movie can be seen.

Additional BD-J features used in this example: downloading A/V material to local storage, playback of A/V material from local storage, display of (JPEG) images from local storage, retrieval and usage of user information (for the display of localized information).

6.2.4 Play games on the disc and also online game

BD-J is not only a good solution for flexible media-playback from disc and from the Internet, it can also be used for games. A disc can contain, besides the movie Title, a Title that contains a set of games. The Java application associated with the Title displays the Menu of available games. The set of games can be a combination of games coming from the disc and games downloaded in JAR files from the Content Provider's website. Games can retrieve high scores from the Internet and achieving a new high-score can result in the user's alias appearing in the updated game results. Game applications can make use of the Java graphics and UI input features of the Java programming environment.

Additional BD-J features used in this example: multiple application support, Java graphics, user input (keys, optional pointing device).

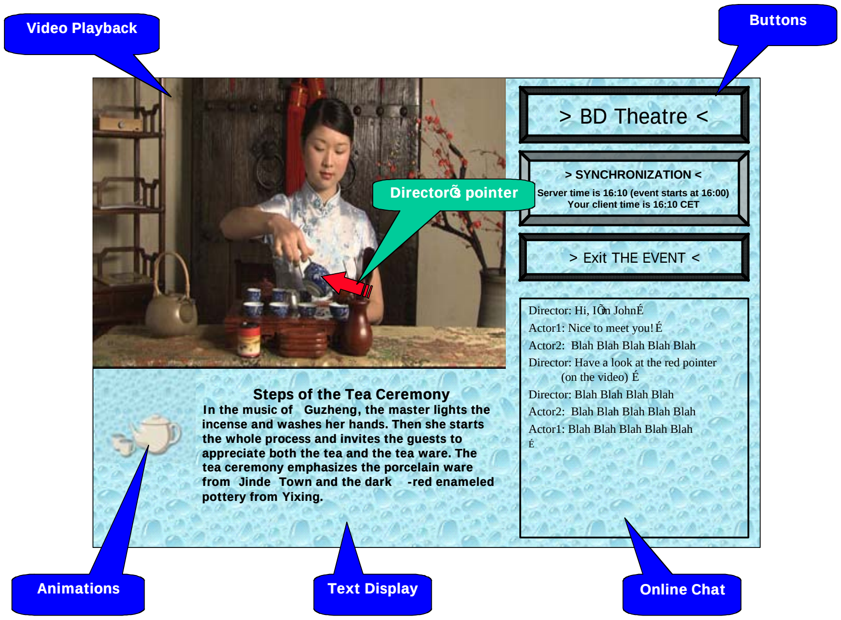
6.2.5 Advanced Applications

With the features described above it is possible to create new Advanced Applications, for example:

- An online shopping application that may allow the end-user to buy Title related merchandising.
- Chat applications that may allow on-line discussion with other purchasers of the same Title.

6.2.6 Application Illustrations

Figure 6-3 below, further illustrates potential BD-J application types. This illustration includes an application that allows a movie director to give comments on the movie, to control playback of the disc and to point to certain items in the video. Note that this does not have to be a live commentary, but can be scripted at the server side.



> BD Theatre <

> SYNCHRONIZATION <
Server time is 16:10 (event starts at 16:00)
Your client time is 16:10 CET

> Exit THE EVENT <

Steps of the Tea Ceremony

In the music of Guzheng, the master lights the incense and washes her hands. Then she starts the whole process and invites the guests to appreciate both the tea and the tea ware. The tea ceremony emphasizes the porcelain ware from Jinde Town and the dark -red enameled pottery from Yixing.

Director: Hi, IÕm JohnÉ
Actor1: Nice to meet you! É
Actor2: Blah Blah Blah Blah Blah
Director: Have a look at the red pointer
(on the video) É
Director: Blah Blah Blah Blah
Actor2: Blah Blah Blah Blah Blah
Actor1: Blah Blah Blah Blah Blah
É

Figure 6-3 – Example of BD-J application

The four pictures below illustrate the use of multiple concurrent applications. One typical example of this is a main application that controls media playback and a second application that displays some information transparently on top of the video. The main game Menu that allows launching various games is another example.

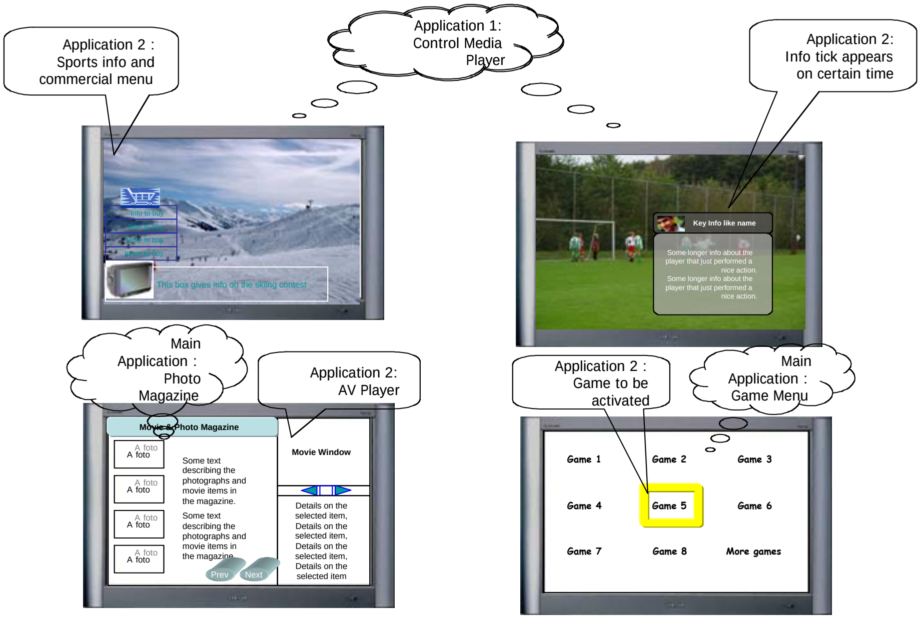

Application 2 : Sports info and commercial menu
Application 1: Control Media Player
Application 2: Info tick appears on certain time
Info to buy
Item to buy
More to buy
Items to buy
This box gives info on the skiing contest
Key Info like name
Some longer info about the player that just performed a nice action.
Some longer info about the player that just performed a nice action.
Main Application : Photo Magazine
Application 2: AV Player
Application 2 : Game to be activated
Main Application : Game Menu
Movie & Photo Magazine
A foto
A foto
A foto
A foto
Some text describing the photographs and movie items in the magazine.
Some text describing the photographs and movie items in the magazine.
Prev
Next
Movie Window
Details on the selected item, Details on the selected item, Details on the selected item, Details on the selected item
Game 1
Game 2
Game 3
Game 4
Game 5
Game 6
Game 7
Game 8
More games

Figure 6-4 – Example of multiple concurrent BD-J application

7 BD-J Application Platform

This chapter defines the capabilities of the BD-J platform as presented to applications.

7.1 Core Platform

The core platform for BD-J applications is Personal Basis Profile 1.0 [PBP]. The JSSE optional package of Personal Basis Profile 1.0 is mandatory for BD-J.

7.1.1 Compliance with GEM

BD-J is fully compliant with GEM 1.0.2 [GEM] for package media targets. This specification adopts the MHP definition of the following functional equivalent, as specified in GEM clause 15.6:

- Text Wrapping

For avoidance of doubt, in the event of a conflict between GEM and this specification, the normative guarantees of GEM shall take precedence except as detailed in section 7.1.1.1, “GEM Errata”.

7.1.1.1 GEM Errata

No errata to GEM have been identified.

7.1.2 GEM Profiles

GEM 1.0.2 [GEM] provides two profiles for packaged media targets: the Enhanced Profile and the Interactive Profile. BD-ROM players may comply with either profile, so applications must be written to react appropriately, using the GEM mechanisms to determine the supported profile. The two GEM profiles are defined in GEM clause 15.0.

7.1.3 BD-J Definitions of GEM functional equivalents

The following table lists the set of GEM functional equivalents, as defined in GEM clause 15.6. For each functional equivalent, the BD-J definition of the functional equivalent is listed. Optional GEM functional equivalents that are not defined for BDA are listed as “n/a”.

Name	BD-J Definition
Arch	See Section 7.1.3.1, “Arch Functional Equivalent”
Carousel	See Section 7.1.3.2, “Carousel Functional Equivalent”
IP MPE	n/a
SI	In BD-J, the GEM concept of service is mapped to title. The information that describes the titles is on the BD-ROM disc; see Section 3.1.1, “Index table”
Broadcast IP signaling	n/a
Audio	See Section 4.3, “Video streams”
Video	See Section 4.4, “Audio streams”
Subtitles	See Section 4.5, “Presentation Graphics and Interactive Graphics streams” and section 4.6, “Text subtitle

	streams”
Audio Clips	See Section 7.1.3.3, “Audio Clips Functional Equivalent”
Resident Fonts	See Section 7.1.3.4, “Resident Fonts Functional Equivalent”
Downloadable Fonts	See Section 7.1.3.5, “Downloadable Fonts Functional Equivalent”
Application Signaling	See Section 7.1.3.6, “Application Signaling Functional Equivalent”
Application Authentication	See Section 7.1.3.7, “Application Authentication Functional Equivalent”
Conditional Access	n/a
Content Referencing	See Section 7.1.3.8, “Content Referencing Functional Equivalent”
Graphics Resolution	See Section 7.1.3.9, “Graphics Resolution Functional Equivalent”
Text Wrapping	Adopts MHP definition; see Section 7.1.1, “Compliance with GEM”.
Minimum CLUT	See Section 7.1.3.10, “Minimum CLUT Functional Equivalent”
RCMM	See Section 7.1.3.11, “RCMM Functional Equivalent”
Active Format Descriptor	See Section 7.1.3.12, “Active Format Descriptor Functional Equivalent”

Figure 7-1 - Table of GEM Functional Equivalents

7.1.3.1 Arch Functional Equivalent

7.1.3.1.1 Context

The context of BD-J is given below in its most abstract form. BD-J is a GEM (Package Media Profile) compliant system. See GEM [GEM]. GEM does not prescribe any specific architecture. This informative section describes the architecture of a BD-J system.

BD-J has access to audio, video and other data on a BD-ROM disc. Audio and video data can be presented to the user. Other data can be used for any purpose a BD-J application wants. BD-J has access to local storage that can be used to store video, audio and other data. BD-J has access to remote servers via a network connection. The network can be used to retrieve audio, video and other data from an external server as well as for sending data to an external server.
BD-J can receive inputs from a user via some input device.

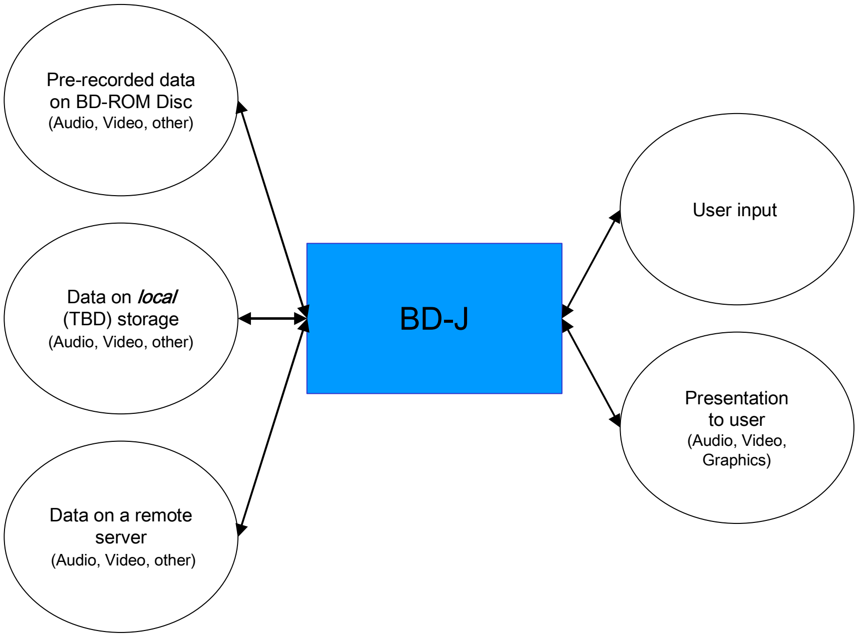


Figure 7-2 - BD-J Context Diagram

Although a device implementing BD-J might include other (non BD-ROM specific) functionality that e.g. connect it to a home network, or that supports BD-RE, this functionality is out of scope of BD-J and will not be covered in this document.

7.1.3.1.2 Architecture

The Architecture describes how the BD-J software elements are organized.
The BD-J Architecture considers 3 layers:

- Resources
- System software
- Applications

The BD-J API lies between the Applications and the BD-J System Software seen from the perspective of an application.

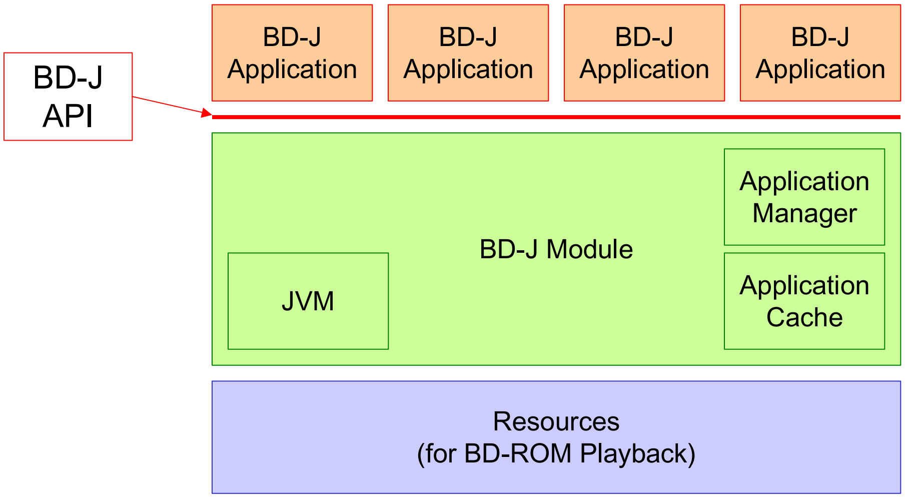


Figure 7-3 BD-J Architecture

7.1.3.1.3 Resources

The hardware entities in the platform include a number of functions. Hardware or software resources represent these functions. There is no assumption about how they are grouped. The model considers that there can be more than one hardware entity in the total Platform.

From an abstract point of view it makes no difference if the logical resources are mapped into one or several hardware entities. What is important is that resources are provided to BD-J transparently. An application should be able to access all resources as if they were elements of a single entity.

7.1.3.1.4 BD-J Module

Applications will not directly address resources. The BD-J Module consists of the BD-J specific system software in a BD-ROM system. This system software brings an abstract view of such resources. This middle layer isolates the application from the hardware, enabling portability of the application.

The implementations of the Resources and System software are not specified in this document.

7.1.3.1.5 Application Manager

The system software includes an application management function, which is responsible for managing the lifecycle of all BD-J applications.

7.1.3.1.6 Application cache

The system software includes an application cache from which the application manager can load applications. The application cache is the preload buffer for BD-J. It is read-only. Note that BD-J applications can use additional data, including class files, that is not preloaded. One example of this is the loading of data from JAR files in local storage.

The purpose of the application cache is to guarantee seamless playback of AV material from the disc during application loading and to reduce latency in loading data.

7.1.3.1.7 Interfaces Between a BD-J Application and the BD-J System

Application(s) use the API to access the actual resources of the BD-J system, including: BD-ROM file-system, BD-ROM media decoders, static content decoders, storage devices and a network connection. These resources are functional entities of the BD-J system and may be finally mapped onto the hardware of the BD-J system in some manner.

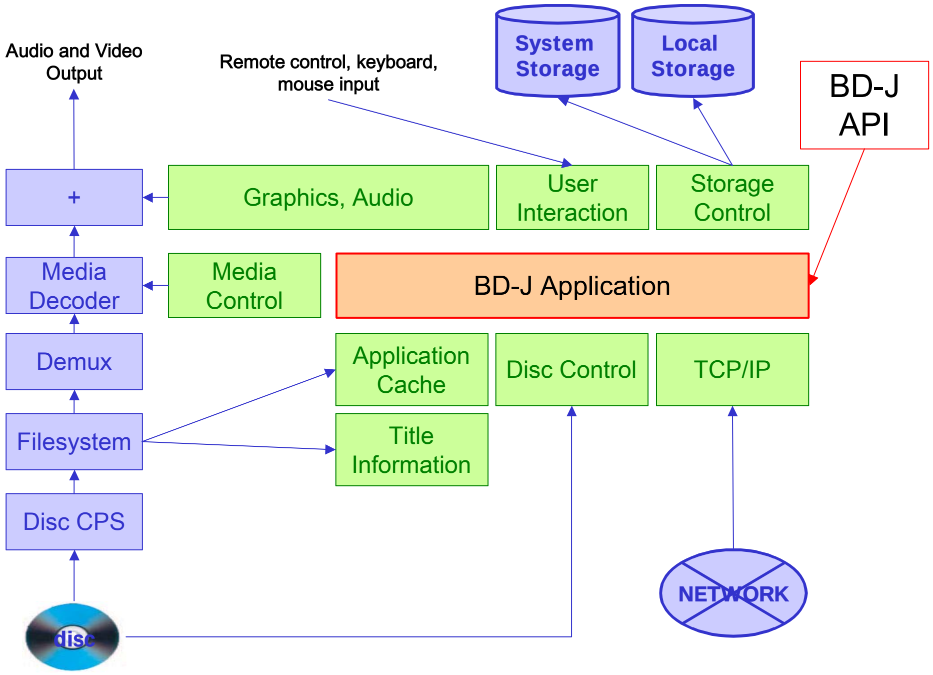


Figure 7-4 Interfaces between a BD-J application and BD-J System software

The diagram in Figure 7-4 shows these interfaces and their relationships to media and information flows within a BD-J system. The diagram shows a generic instance of BD-J system.

7.1.3.2 Carousel Functional Equivalent

In BD-J, a Service Domain is mapped to one JAR File. A JAR File can include stream description, trigger object and trigger event as special files. Stream description is mapped onto the PlayList which the BD-J application references.

BD Terminal shall detect playback events associated with the PlayList currently in play. BD Terminal shall post corresponding stream events to BD-J applications according to the trigger object and the trigger event.

The “priority value” that is defined in B.2.1.1 of GEM 1.0.2 [GEM] is mapped to the Application cache order.

7.1.3.3 Audio Clips Functional Equivalent

BD-J can use LPCM encoded audio clips for usage by instances of HSound. The clips can come from the disc or from local storage. Supported sampling frequencies are 48kHz, 96kHz, 192kHz, with 16, 20 or 24 bits per sample. On the disc they are combined so that the player similar to the application JAR files can preload them to prevent interruption of the A/V playback from the disc at the moment the clip is accessed for usage by HSound.

7.1.3.4 Resident Fonts Functional Equivalent

BD-J implementations do not have any prescribed resident font, which is compliant with GEM. A content author desiring to use a specific default font shall provide this font on the BD-ROM disc. The format of this font shall be OpenType [OTF]. This disc provided font can be used by the BD-J application without explicit loading of the font by the downloadable font API's of GEM. The usage of OpenType and associated Java font binding API is being harmonized with expected future versions of MHP.

7.1.3.5 Downloadable Fonts Functional Equivalent

In BD-J the functional equivalent for "downloadable fonts" is called "runtime installable fonts". The format for runtime installable font files is OpenType. The usage of OpenType and associated Java font binding API is being harmonized with expected future versions of MHP. These font files can be provided on the BD-ROM disc or downloaded from the network.

7.1.3.6 Application Signaling Functional Equivalent

BD-J defines an "Application Management Table" (AMT), which implements the GEM Functional Equivalent named "Application Signaling". One instance of an AMT is provided on the disc per BD-J title. The AMT identifies the BD-J applications associated with the title and provides all information as required by GEM for each, e.g. identification-, lifecycle-, parameterization-, profile & version-, user- and other information.

7.1.3.7 Application Authentication Functional Equivalent

For BD-J, authentication messages that conform to the Signed JAR File format defined in "JAR File Specification" provided by Sun Microsystems, Inc, are used. The statement in Section 12.2.1.4 of DVB-MHP (as referenced by GEM) will apply to JAR files instead of hierarchical structure.

7.1.3.8 Content Referencing Functional Equivalent

BD-J provides a textual representation for entities on the BD-ROM disc. A locator can be constructed from this textual representation. In this way the BD-J locator can be used to address the following entities:

Entity	Text Representation
Transport stream	BD-J locator indicating a disc
Title (GEM Service)	BD-J locator indicating a title
Service Domain	BD-J locator indicating a JAR file
MPEG Elementary Stream	BD-J locator indicating a TS component
File	"file:", "http:" and "https:" URLs, as referred to in MHP [MHP], clause 14.8 BD-J locator indicating a file in a BD-J application JAR file BD-J locator indicating a file in the BD-ROM file system
Directory	"file:", "http:" and "https:" URLs, as referred to in MHP [MHP], clause 14.8 BD-J locator indicating a directory in a BD-J application JAR file BD-J locator indicating a directory in the BD-ROM file system
Click sound	BD-J locator for a click-sound

7.1.3.9 Graphics Resolution Functional Equivalent

BD-J supports PAL and NTSC SD resolutions, as well as graphics resolutions for HD. For HD 1920x1080x32bpp resolution is supported in BD-J.

The video material played by the background media player by default determines the graphics resolution. If no video material is being played at the start of a title a default graphics device configuration of 1920x1080 is used.

7.1.3.10 Minimum CLUT Functional Equivalent

BD-J applications can rely on a 32 bpp true-color model of the graphics plane.

7.1.3.11 RCMM Functional Equivalent

As specified in GEM 1.0.2 [GEM] clause 12.9.2, “Root certificate management,” GEM terminal specifications for the packaged media profile are not required to implement the MHP root certificate management mechanism. BD-J terminals are not required to implement this mechanism; the security mechanisms of BD-J are out of scope of the present document.

7.1.3.12 Active Format Descriptor Functional Equivalent

In BD-J, the active format area is guaranteed to be the complete frame as present in the video. Thus, the method getActiveFormatDefinition() of the class org.dvb.media.VideoFormatControl always returns the value AFD_SAME.

7.2 BD-J Additional Packages

BD-J applications that wish to use functionality that goes beyond GEM may use extension packages specific to the BD-ROM format.

7.2.1 org.bluray.media

This package provides extra controls various BD specific functionalities to the Java Media Framework. The control for angles and controls providing functionalities for selecting, audio and subtitle by id are added into this package. These controls extend the org.davic.media.LanguageControl.

7.2.2 org.bluray.ti

In BD-J, Service is mapped to Title and this package extends the javax.tv.service.* packages.

It provides the mechanism for querying the title information (e.g playlists or playItem) from BD-ROM and selecting the new title.

7.2.3 org.bluray.application

This package contains BD-J specific APIs for lifecycle management of applications, e.g. an API to query the minimum BD-J profile & version carried in the signalling required to run the application, an event notification API for disc eject and disc insert as well as a query API to disc specific lifecycle signalling.

7.2.4 org.bluray.ui

The org.bluray.ui package defines constant for BD-J specific key events and includes a class for frame synchronous animations that takes an array of images that are rendered in sync with the video played by the background player. Methods for the application to start and stop the animation at a certain media time are provided, as well as methods that allow the application to draw safely directly into the images.

7.2.5 org.bluray.vfs

This package provides binding mechanisms between off-the-disc contents (additional contents on local storage) and the corresponding on-the-disc contents (BD-ROM contents), which enables a seamless user experience to play back media data, regardless of the origin of the data. This package contains request API to construct/cancel the binding, permission control, listen the construction status of the binding and so on.