First edition
1998-09-01

Information technology — Generic coding of moving pictures and associated audio information —

Part 6:
Extensions for DSM-CC

Technologies de l'information — Codage générique des images animées et des informations sonores associées —

Partie 6: Extensions pour DSM-CC

Reference number
ISO/IEC 13818-6:1998(E)

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 13818-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 13818 consists of the following parts, under the general title *Information technology — Generic coding of moving pictures and associated audio information*:

— *Part 1: Systems*
— *Part 2: Video*
— *Part 3: Audio*
— *Part 4: Compliance testing*
— *Part 5: Software simulation*
— *Part 6: Extensions for DSM-CC*
— *Part 7: Advanced Audio Coding (AAC)*
— *Part 9: Extension for real time interface for systems decoders*
— *Part 10: Conformance extensions for DSM-CC*

Annex A forms an integral part of this part of ISO/IEC 13818. Annexes B to N are for information only.

0. Introduction

The Digital Storage Media Command and Control (DSM-CC) specification is an integral part of ISO/IEC 13818 (MPEG-2). It 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 multimedia 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 through the concept of a "session". A Session is an associated collection of resources required to deliver a Service. Examples of resources are MPEG-2 Transport Stream packet identifiers and network bandwidth. The Session complements a "Service Domain", which is a collection of interfaces to browse and select services, and control the delivery of bit streams.

One of the strengths of DSM-CC is in its abstraction from underlying networks; a suite of uniform interfaces are visible to the application, shielding it from the details of inter-working among heterogeneous networks – e.g., Hybrid Fiber Coax (HFC), Asynchronous Transfer Mode (ATM), Asymmetric Digital Subscriber Loop (ADSL), Internet Protocol (IP), and combinations of these technologies as part of an end-to-end multimedia system. In other words, 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.

The session signaling layer provides a uniform, flexible, and extensible method for managing heterogeneous resource types. In addition to the network and service types described in this specification, DSM-CC may be extended to support other networks and services through the definition of new resource types.

In DSM-CC, a bit stream is sourced by a Server and delivered to a Client. Both the Client and the Server are logical embodiments and do not imply a singular device in an actual implementation.

Application/service examples are interactive multimedia retrieval (including video-on-demand), Internet access, digital video broadcasting, data downloading, and audio/video/graphics conferencing.

0.1 Guiding Factors in the Formulation of DSM-CC

The DSM-CC specification was influenced by the following factors:

- A wide range of network topologies may be used to deliver DSM-CC.
- Resources are finite and need to be managed.
- Latencies need to be minimized to provide (interactive) services.
- DSM-CC applications need to be supported by an underlying protocol that facilitates communications between a Server-side application and a corresponding Client-side application.

0.2 DSM-CC Client-Network-Server Model

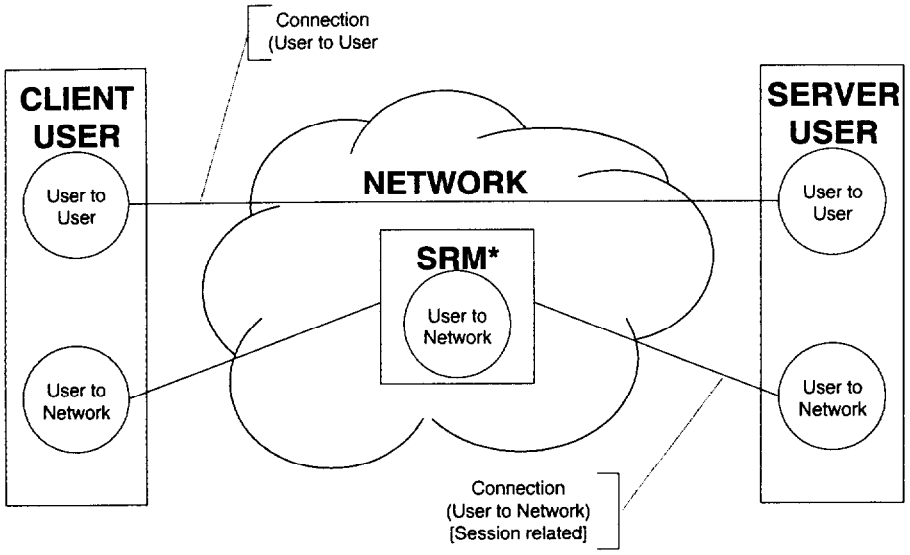

Figure 0-1 DSM-CC basic Client-Network-Server model

Figure 0-1 depicts the basic model used in DSM-CC. A Session and Resource Manager (SRM) provides logically centralized management of the DSM-CC Sessions and Resources. DSM-CC User-to-Network (U-N) messages flow between the Client and SRM and the Server and SRM. Both the Client and the Server are called Users of DSM-CC. The U-N session protocol establishes a Session and groups all the resources required for delivering a service. The service interactions are carried between the Client and the Server participating in the Session using DSM-CC User-to-User (U-U) messages. The SRM also does U-N configuration management and control of both Clients and Servers to allow their participation in the DSM-CC environment.

DSM-CC supports network topologies which consist of multiple Clients and multiple Servers. Any Client-Server pair can communicate together by establishing a Session. Each Client can have multiple simultaneous Sessions with any specific Server or any combination of Servers. For this phase of DSM-CC, a Session is typically limited to one Client, one Server, and one SRM. The exception is the case of Continuous Feed Sessions (CFS). A CFS may be used by, e.g., a stream broadcasting application, where broadcast "feeds" are established with the network with no particular Client specified. Clients may "attach" to a CFS by setting up a Session with the network to connect to the CFS and, optionally, to establish Client-unique resources (such as a return control channel that may be needed by an interactive application which shares a downstream feed, e.g., game show voting). Alternatively, Clients may "attach" to a CFS or another broadcast "feed" by using the U-N Switched Digital Broadcast Channel Change Protocol (SDB-CCP), when no Client-unique Resources are needed by the application (such as with traditional "pay-per-view").

0.3 Outline of the DSM-CC Specification

DSM-CC consists of a set of User-to-Network and User-to-User protocols. These protocols are described in the clauses listed below.

0.3.1 User-to-Network

- DSM-CC Message Header, clause 2
- U-N Configuration messages, clause 3
- U-N Session messages and flow diagrams for Session and Resource management, clause 4
- U-N Download messages, clause 7
- U-N Switched Digital Broadcast Channel Change Protocol, clause 10
- U-N Pass Thru messages, clause 12
- The transport of DSM-CC U-N messages using MPEG-2 Systems (ISO/IEC 13818-1), clause 9
- The transport of generic IP messages using DSM-CC Sections and MPEG-2 Systems, clause 9

0.3.2 User-to-User

- U-U Remote Procedure Call (RPC), clause 5
- U-U Session interface, clause 5
- U-U Download interface, clause 5
- U-U Object Carousel interface, clause 11
- U-U Local Object interface, clause 5
- U-U Stream Descriptors, clause 8

0.4 Supported Network Technologies

DSM-CC does not specify the underlying physical, data link, transport, or RPC layers of the overall protocol stack. However, DSM-CC does specify requirements for these layers in clause 9.

0.5 Supported Connection Types

DSM-CC supports the following connection types:

- Point-to-point
- Point-to-multi-point (broadcast)

User-to-User application and service exchanges are carried over point-to-point type connections.

The point-to-multi-point type connections are used to feed a single stream to multiple Clients. In this case, no single Client has control (e.g., for the purpose of pause, fast forward) of the received bit stream. However, in the case where the network (as opposed to the Client) does stream switching such as with Switched Digital Broadcast (SDB) applications, a means is provided for Clients to switch between streams using the SDB channel change protocol (SDB-CCP). The latter is useful for applications such as the so-called "enhanced pay-per-view" or "near video on demand".

0.6 DSM-CC Interfaces

The DSM-CC model (Figure 0-1) consists of three Subsystems:

- Client
- Server
- Session & Resource Manager (SRM)

Each subsystem is a logical embodiment within a DSM-CC System. It does not map directly to physical equipment. The SRM represents the DSM-CC functionality within a DSM-CC network (the Network).

In order to define interfaces, a DSM-CC System Reference Model is used to subdivide the DSM-CC environment into a hierarchy of several levels (see Figure 0-2):

- System
- Subsystem
- Entity
- Sub-entity

A Subsystem may contain more than one Entity. The types of Entities are:

- Client User-to-User Entity
- Client User-to-Network Entity
- Server User-to-User Entity
- Server User-to-Network Entity
- SRM User-to-Network Entity

DSM-CC signaling is always exchanged between specific Subsystem Entities.

From the normative perspective, the System Reference Model does not show any more detail below the Entities. However, for informative reasons, the Entities have in some cases been further divided into Sub-entities. For example, the Client and Server U-N Entities include a Session Manager, Resource Manager, and a Configuration Manager. On the Server side, the U-U Entity includes a Service Gateway and an Object Access Manager.

DSM-CC recognizes that Subsystems will require other Entities which are not specified by this part of ISO/IEC 13818. Examples are a Connection Management Entity and Application Entities.

Figure 0-2 shows the different entities and sub-entities of the DSM-CC system.

To show the boundaries between Entities, Figure 0-2 is divided into four layers:

- The Application Entity layer, which is outside the scope of this part of ISO/IEC 13818
- The User-to-User Entity, which is an Application/Service Control/Management Layer
- The User-to-Network Entity, which is a Session and Resource Control/Management Layer
- The Connection Control Entity, which is (typically part of) a Transport Layer, and is outside the scope of this part of ISO/IEC 13818.

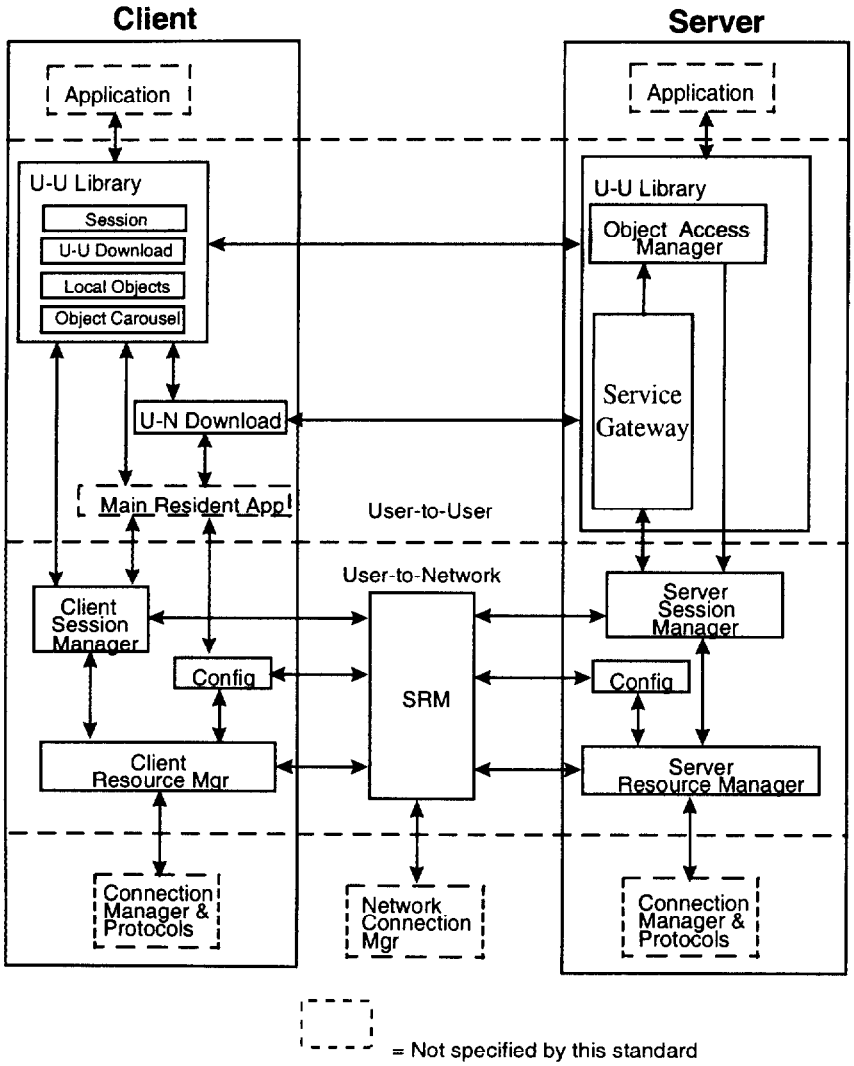

Figure 0-2 DSM-CC System Reference Model

Each of the lines with arrows in Figure 0-2 represents a logical interface. DSM-CC defines three types of interfaces:

- Inter-Entity

- Intra-Entity
- Intra-Subsystem

The Inter-Entity interfaces are between peer Entities in different Subsystems. The interfaces between the Sub-entities within a common Entity are called Intra-Entity interfaces. The interfaces between Entities within a common Subsystem are called Intra-Subsystem interfaces.

The DSM-CC System Reference Model specifies three communication paths over which DSM-CC messages are exchanged. The communication between U-U Entities are represented as the DSM-CC U-U Protocol. The communication between U-N Entities are represented as the DSM-CC U-N Protocol.

- Client U-U Entity to Server U-U Entity (U-U)
- Client U-N Entity to SRM U-N Entity (U-N)
- Server U-N Entity to SRM U-N Entity (U-N)

Table 0-1 summarizes Inter-Entity interfaces and Intra-Subsystem interfaces within the scope of DSM-CC.

Table 0-1 DSM-CC Interface Scope Summary

Peer 1	Peer 2	Protocol	Inter-Entity	Intra-Subsystem
Client U-U Library	Server Service Gateway	U-U	X	
Client U-U Library	Server Object Access	U-U	X	
Client Session Gateway	SRM	U-N	X	
Client Resource Manager	SRM	U-N	X	
Server Session Manager	SRM	U-N	X	
Server Resource Manager	SRM	U-N	X	
Client Configuration	SRM	U-N Config	X	
Server Configuration	SRM	U-N Config	X	
Server DSM Source (e.g. MPEG-2 Transport / Video / Audio)	Client DSM Consumer	(MPEG)	X[1]	
Download Server (Source)	U-N Download Client (Consumer)	Down-load	X[1]	
Object Carousel Server	Object Carousel Client	Object Carousel / Down-load	X[1]	
SDB Server	(SDB) Client	SDB-CCP	X[1]	
Client Application	Client U-U Library	U-U		X

Note 1: Interface not shown on Figure 0-2.

0.7 DSM-CC Interface Protocols

Figure 0-3 depicts DSM-CC protocols used at DSM-CC interface points. The top section of the figure contains some applications which may use DSM-CC. The middle section of the figure contains all of the DSM-CC specified protocols. The specific Transport Layers, the bottom section, are not specified by this part of ISO/IEC 13818.

Note that Figure 0-3 applies to the case where the full suite of DSM-CC protocols (except for the extended protocol groups) are employed. DSM-CC allows each protocol to be implemented without the others (see subclause 1.2 Profiles and Compliance). If the U-U Library is not used, then the implementation will not have an Application Portability Interface specified by DSM-CC.

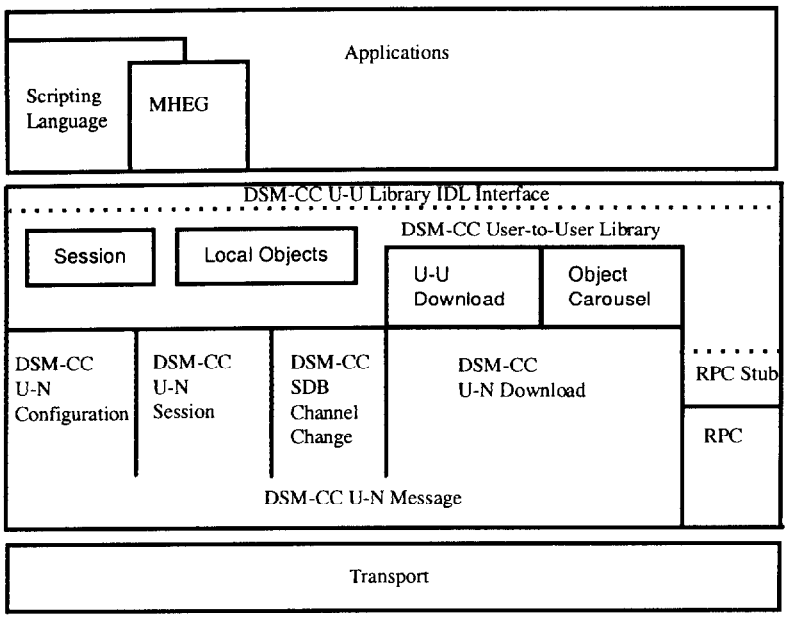

Figure 0-3 DSM-CC Interface Protocols

DSM-CC provides access to Stream and Data objects for applications (e.g., MHEG applications and scripting language applications). The primary application interface layer is the DSM-CC U-U Library Interface Definition Language (IDL), or Application Portability Interface. The U-U Library may in turn make use of the U-N Session Management, U-N Download, and U-U Object Carousel layer to establish and manage Sessions and Resources required for the management and delivery of the Stream and Data objects.

Table 0-2 lists the DSM-CC protocols. The protocols which use the DSM-CC message format are U-N Configuration, U-N Session, U-N SDB-CCP, U-N Download, and U-U Object Carousel (because it, in turn, uses U-N Download). In some cases, the use of a message passing interface is needed because the Client device may not have higher layer protocols (e.g., RPC) resident.

The U-U Library uses the services of the U-N protocols, but also adds its own on-the-wire protocol, the U-U RPC Stub Library, which are based on existing Remote Procedure Call (RPC) interfaces. The protocols which use RPC do so because it provides sophisticated object based services.

The third category is IDL, which is used in communicating within the Subsystem to applications.

Table 0-2 The DSM-CC Protocols used on the Interfaces

DSM-CC Protocols	Peer 1	Peer 2	U-N Message Format	IDL/RPC
U-N Configuration	Client / Server	SRM	x	
U-N Session	Client / Server	SRM	x	
U-N Download	Client	Download server	x	
U-N Switched Digital Broadcast Channel Change	Client	SDB server	x	
U-N Pass Thru	Client	Server	x	
U-U RPC	Client	RPC server		RPC
U-U Session	Client Application	Client U-U Library		IDL
U-U Download	Client Application	Client U-U Library		IDL
U-U Object Carousel	Object Carousel Client	Object Carousel	x	
U-U Local Objects	Client Application	U-U Library		IDL

The transport layer in Figure 0-3 may consist of any protocol which meets the transport requirements described in clause 9. Examples are, TCP or UDP over IP, AAL-5 over ATM, or DSM-CC/private_sections over MPEG-2 Transport Stream.

0.8 Communications Requirements

The DSM-CC U-N Configuration, U-N Session, U-N SDB-CCP and U-N Download messages all use the DSM-CC Message Format and are implemented using a simple message passing method; therefore, all have similar Transport Layer requirements. The U-U Object Carousel uses the U-N Download protocol and its associated transport requirements. The U-U RPC Stub Library uses RPC and its associated transport requirements.

The requirements for the underlying Transport services for all DSM-CC protocols are provided in detail in clause 9, Transport.

0.9 Methods of Specification

0.9.1 Messages

U-N messages are described in tables which list the bit or byte level assignment for all of the fields in each message. The syntactical structure of the messages are defined by Syntax Tables like the example below. Field names are shown in bold and always have an associated number of bytes indicated. All numeric values are unsigned big-endian (most significant byte first, most significant bit first) unless otherwise specified. The method of syntax description supports loops and 'procedures' using a pseudo-C syntax. In the example below, a for() loop, in normal font, indicates that the field **uuDataByte** repeats uuDataCount times. Also, the structure has been named UserData(), which now can in turn be referenced in other larger structures.

Syntax	Num. of Bytes
UserData(){	
uuDataLength	2
for(i=0;i<uuDataCount;i++) {	
uuDataByte	1
}	
privateDataLength	2
for(i=0;i<privateDataLength;i++) {	
privateDataByte	1
}	
}	

Figure 0-4 Example of U-N message syntax

The messages for U-N Configuration and U-N Session flow between Client and Network (SRM), and Server and Network (SRM). For consistency, the suffix of each of these messages use the following terminology:

Request - A message sent from a User (Client or Server) to the Network to begin a scenario.

Confirm - A message being sent from the Network to a User (Client or Server) in response to a Request message.

Indication - A message which is sent from the Network to a User.

Response - A message from a User to the Network in response to an Indication message.

Clause 9, Transport, defines the communications requirements (reliability, addressing etc.) for the delivery of these messages.

A standard programming API for the use of these messages is outside the scope of this part of ISO/IEC 13818.

0.9.2 Message Flow Diagram Scenarios

Flow diagrams have been provided to help explain the use of the DSM-CC message protocols. These diagrams show the sequence and direction of flow for the messages of a specific scenario. In these diagrams, the time axis runs vertically, with messages lower on the diagrams representing later transmission. The selected scenarios are the most typical ones and do not represent the exhaustive list of examples of scenarios. The Specification and Description Language (SDL) representations provide a more exhaustive representation, including exception cases.

0.9.3 Specification and Description Language

The SDL-language is officially defined in ITU-T recommendation Z.100. For the translation of the DSM-CC specification into SDL, SDL-88 (Z.100 blue book) is used. There are several advantages to using SDL:

- Contrary to the textual part, usage of SDL in the specification makes it unambiguous due to the fact that SDL is a formal language.
- One representation of SDL is the graphical one. This makes the language more comprehensible.
- The SDL specification can be analyzed for completeness and correctness.
- It is easy to generate executable code in order to simulate and validate the specification.
- The specification can also be used for conformance test purposes.

For simulation purposes, Message Sequence Charts (MSC), as defined in ITU-T recommendation Z.120, are used.

A model described in SDL consists of three different types of levels.

1. System level
2. Block level
3. Process level

The highest level of the SDL model is the system level. The system is surrounded by the environment represented by a rectangle in the graphical representation. On the system level, the model of the system is described in a very rough shape

divided into one or more blocks. The blocks can contain either new blocks or processes. At some block level, the content is one or several processes in each block. The process level could then describe logical parts of the system related to each other with the signals exchanged between them.

A static process is created at start-up time for the system. A dynamic process is created during runtime by another process. The number of dynamic processes which may be created is set by a constant value. A process can be stopped by the process itself at any point in time.

A process is a state machine and the only way to move from one state to another state is via a transition. One or several possible transitions can be connected to a state. A transition is always initiated by either an input signal or an enabling condition. An input signal can be generated by an output signal from an outside process, from within the same process, or by an expired timer. Here, the environment is also regarded as a process. The input signal is put in an input queue which is a common queue for the process.

When an input signal is consumed, a transition is started and the actual code defined between the state and the next state is executed. In the graphical representation, the code consists of one or several graphical symbols with some additional plain text; variables may be assigned new values in a task, questions may be answered in a decision, an output signal may be sent to another process, etc.

Figure 0-5 shows some common SDL symbols. Complete specification of SDL is outside of the scope of this specification, but may be found in ITU-T Z.100 and Z.120.

The intent is to have the message flow diagrams and prose be consistent with the SDL tables. Since the SDL is more exhaustive, if there is any form of contradiction between the prose and SDL, the SDL shall take precedence.

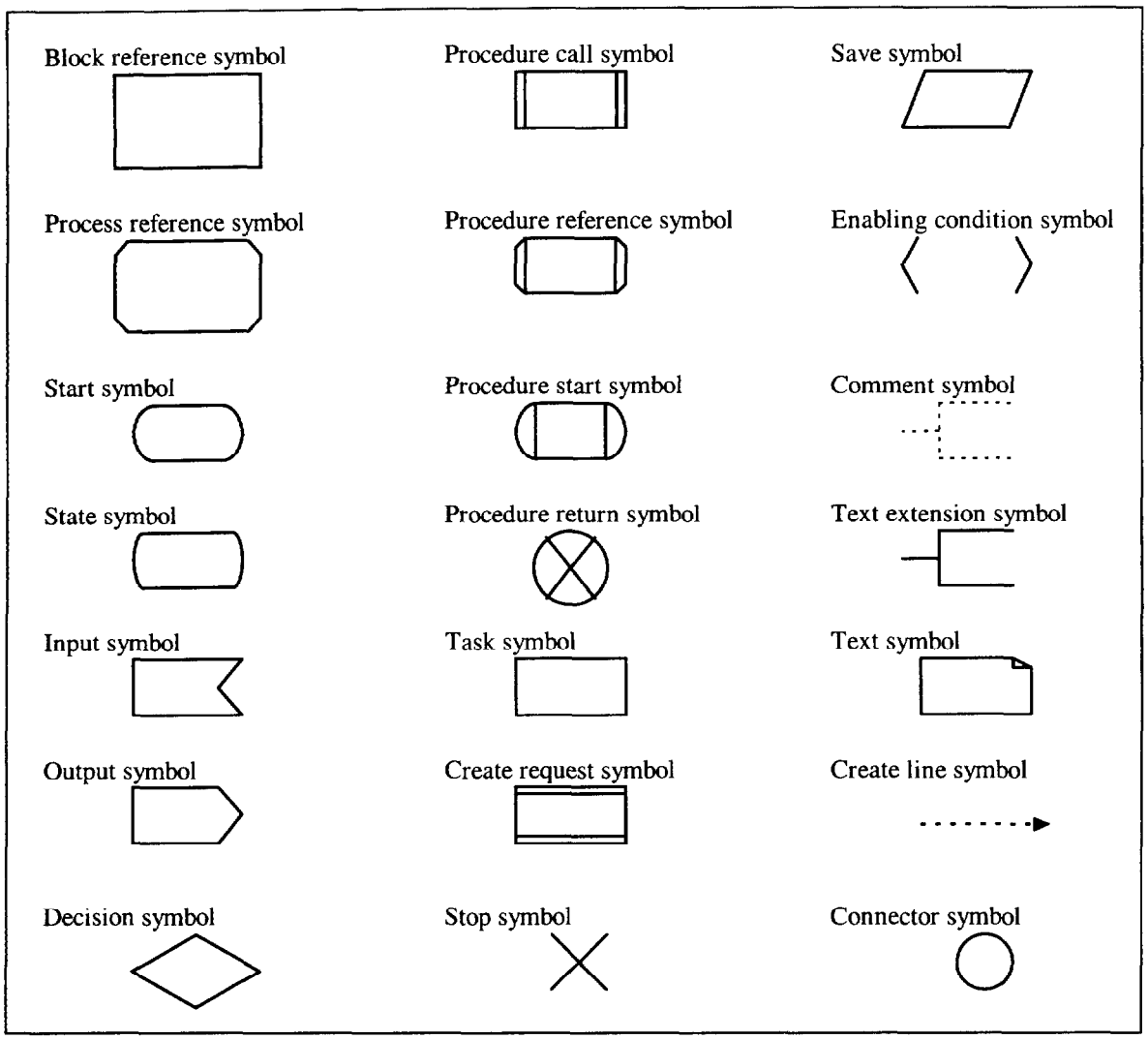

Figure 0-5 SDL Symbols

0.9.4 Interface Definition Language (IDL)

The U-U API primitives that use RPC are defined in terms of OMG Interface Definition Language (IDL), defined by ISO/IEC 14750. The IDL provides a grammar for defining the function call-like API specification for each primitive. Primitives written in the IDL are compiled by an IDL compiler to produce client and server stubs (executable code that implements packet formation, dispatch, receipt, and interpretation) and a header file used during compilation of the client and server applications.

0.9.5 Remote Procedure Call (RPC)

U-U functionality exploits a Remote Procedure Call (RPC) protocol. A RPC allows implementation of a client-server model in which applications on a client are written to call functions that are similar to those that might be used if all actions were to be executed locally. For those U-U API primitives that use the RPC, the RPC and data encoding defines the actual bits that are exchanged as primitives are executed.

The downstream reply from the Server can be delivered via encapsulation within a MPEG-2 Transport Stream. Although this part of ISO/IEC 13818 specifies how to encapsulate common protocols (e.g., IP) over MPEG-2 Transport, there is no requirement that control messages or RPC messages be delivered within MPEG-2 Transport Streams.