ETSI TS 102 818 V1.4.1 (2008-06)

Technical Specification

Digital Audio Broadcasting (DAB);
Digital Radio Mondial (DRM);
XML Specification for Electronic Programme Guide (EPG)

European Broadcasting Union Union Européenne de Radio-Télévision

EBU·UER

Reference
RTS/JTC-DAB-56

Keywords
audio, broadcasting, DAB, DRM, digital, EPG

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.
© European Broadcasting Union 2008.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE 1: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The Eureka Project 147 was established in 1987, with funding from the European Commission, to develop a system for the broadcasting of audio and data to fixed, portable or mobile receivers. Their work resulted in the publication of European Standard, EN 300 401 [18], for DAB (see note 2) which now has worldwide acceptance. The members of the Eureka Project 147 are drawn from broadcasting organizations and telecommunication providers together with companies from the professional and consumer electronics industry.

NOTE 2: DAB is a registered trademark owned by one of the Eureka Project 147 partners.

1 Scope

The present document defines the XML schema data model for an Electronic Programme Guide (EPG) for Eureka-147 Digital Audio Broadcasting (DAB) (EN 300 401 [18]) and Digital Radio Mondiale (DRM) (ES 201 980 [23]). Within the present document the term "DAB" is used to refer to the Eureka-147 Digital Audio Broadcasting standard. It is envisaged that this data format could be used both for transmitting schedule data to EPG applications on receivers and as the basis for exchanging information between broadcasters, network operators and content providers.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI TS 102 822-4: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 4: Content referencing".

[2] ISO 8601: "Data elements and interchange formats - Information interchange - Representation of dates and times".

[3] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax".

[4] WAP Forum: "Wireless Application Protocol; Wireless Markup Language Specification".

[5] W3C Recommendation: "Extensible Markup Language (XML) 1.0 (Third Edition)".

[6] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[7] IETF RFC 3066: "Tags for the Identification of Languages".

[8] PNG Development Group: "Portable Network Graphics (PNG) Specification, Version 1.1".

[9] IETF RFC 2806: "URLs for Telephone Calls".

[10] IETF RFC 3191: "Minimal GSTN address format in Internet Mail".

[11] IETF RFC 2368: "The mailto URL scheme".

[12] WAP Forum: "Wireless Application Protocol; Wireless Application Environment Specification Version 2.0".

[13] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".

[14] IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types".

[15] IETF RFC 2048: "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures".

[16] ISO/IEC 11172-3: "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio".

[17] ISO/IEC 13818-3: "Information technology - Generic coding of moving pictures and associated audio information - Part 3: Audio".

[18] ETSI EN 300 401: "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".

[19] ISO/IEC 10646: "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

[20] ISO 8859-2: "Information technology - 8-bit single-byte coded graphic character sets - Part 2: Latin alphabet No. 2".

[21] ETSI TS 102 822-3-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 1: Phase 1 - Metadata schemas".

[22] ETSI TS 102 371: "Digital Audio Broadcasting (DAB); Digital Radio Mondiale (DRM); Transportation and Binary Encoding Specification for Electronic Programme Guide (EPG)".

[23] ETSI ES 201 980: "Digital Radio Mondiale (DRM); System specification".

[24] ETSI TS 102 563: "Digital Audio Broadcasting (DAB); Transport of Advanced Audio Coding (AAC) audio".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Conditional Access (CA): mechanism by which the user access to service components can be restricted

data service: service which comprises a non-audio primary service component and optionally secondary service components

ensemble: transmitted signal, comprising a set of regularly and closely-spaced orthogonal carriers

NOTE: The ensemble is the entity that is received and processed. In general, it contains audio and data services.

Ensemble Identifier (EId): unique 16-bit code, allocated to an ensemble and intended to allow unambiguous worldwide identification of that ensemble

eXtended Programme Associated Data (X-PAD): extended part of the PAD carried towards the end of the DAB audio frame, immediately before the Scale Factor Cyclic Redundancy Check (CRC)

NOTE: Its length is variable.

Programme Associated Data (PAD): information that is related to the audio data in terms of contents and synchronization

NOTE: The PAD field is located at the end of the DAB audio frame.

secondary service component: in the case where a service contains more than the primary service component, the additional service components are secondary service components

service: in the present document the term "service" is used to refer to a "radio station" such as BBC Radio 4 or Oneword

NOTE: In strict DAB terms this is actually a service component of a service.

service component: part of a service which carries either audio (including PAD) or data

NOTE: The service components of a given service are linked together by the Multiplex Configuration Information. Each service component is carried either in a sub-channel or in the Fast Information Data Channel.

Service Identifier (SId): 16-bit, 24-bit or 32-bit code used to identify a particular service

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CA	Conditional Access
CRC	Cyclic Redundancy Check
CRID	Content Reference ID
CS	Classification Schemes
DAB	Digital Audio Broadcasting
DRM	Digital Radio Mondiale
ECC	Extended Country Code
EId	Ensemble Identifier
EPG	Electronic Programme Guide
IANA	Internet Assigned Numbers Authority
ISO	International Organization for Standardization
MIME	Multipurpose Internet Mail Extensions
MOT	Multimedia Object Transfer
PAD	Programme Associated Data
PNG	Portable Network Graphics
SCIdS	Service Component Identifier within the Service
SDARs	Satellite Digital Audio Radios
SI	Service Information
SId	Service Identifier
SMS	Short Messaging Service
UATy	User Application Type
URI	Uniform Resource Identifier
URL	Uniform Resource Location
UTC	Co-ordinated Universal Time
WAP	Wireless Access Protocol
WBMP	Wireless BitMaP
WWW	World Wide Web

XML	eXtensible Markup Language
X-PAD	eXtended Programme Associated Data

4 Introduction

It is intended that the EPG will be used to provide programme listings information for both audio and data services and as a mechanism for the user to select services, programmes and related content. A key requirement is that the EPG must work on a range of receivers with differing display capabilities, resources and back-channel capabilities. To achieve this a flexible structure has been defined, as shown in figure 1. The EPG data is broken down into service information (ensembles and services) and programme information (schedules, programmes, groups and events). Additionally programmes and events can be linked together into groups (e.g. for grouping programmes together into serials or series).

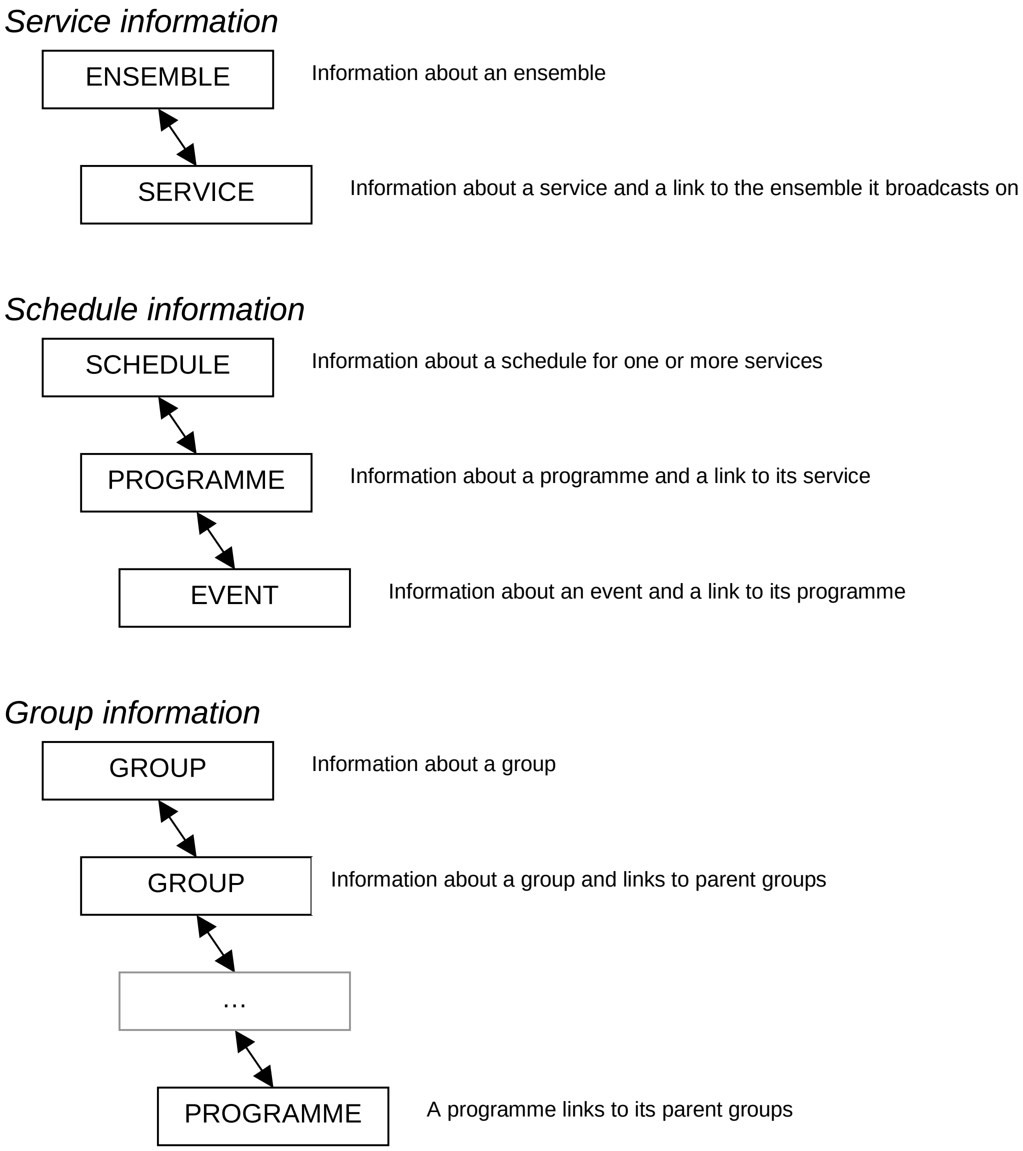
Service information
ENSEMBLE
Information about an ensemble
SERVICE
Information about a service and a link to the ensemble it broadcasts on
Schedule information
SCHEDULE
Information about a schedule for one or more services
PROGRAMME
Information about a programme and a link to its service
EVENT
Information about an event and a link to its programme
Group information
GROUP
Information about a group
GROUP
Information about a group and links to parent groups
…
PROGRAMME
A programme links to its parent groups

Figure 1

The EPG may be delivered using the DAB or DRM broadcasting systems. The philosophy is that DAB will carry a DAB EPG describing DAB services and DRM will carry a DRM EPG describing DRM services. In general, the DAB and DRM EPGs are the same; however there are slight differences to the format of certain fields to deal with the specific requirements for these two systems and signalling is provided to receivers to ensure there is no confusion.

4.1 Document structure

The EPG specification is split into 3 schemas:

- Common data types - epgDataTypes_14.xsd.
- Schedules - epgSchedule_14.xsd.
- Service information - epgSI_14.xsd.

The present document is therefore also split into three clauses with the schemas in annexes at the end of the document. Each clause defines and describes each of the entities, elements and attributes in the respective schema.

NOTE: Some of the examples use the representation "…" to indicate possible child elements, this is not valid XML.

4.2 XML information

4.2.1 Why XML?

Standards: XML is a well-established standard for describing structured information.

Future expandability and backwards-compatibility: An appropriately designed XML application can be expanded in the future without breaking any previous systems. This is particularly important in this case where we are trying to develop a specification that will be used in a large number of applications, some of which are unknown at this point in time.

Use of existing tools: Many applications and APIs already exist for manipulating XML and these would be useful in creating/editing content and writing robust software utilizing EPG documents.

4.2.2 Character encoding

The ISO/IEC 10646 [19] character set using UTF-8 character encoding must be used in all EPG XML documents where applicable.

NOTE: The ISO/IEC 10646 [19] character set contains all characters of the DAB character sets (three EBU Latin-based sets, ISO 8859-2 [20] and ISO/IEC 10646 [19] using UTF-8).

4.3 Examples

To give an idea of what can be done with this XML definition some simple and complex examples are shown in clauses 4.3.1 to 4.3.3.

4.3.1 Schedule

Schedule information describes a schedule and its programmes on one or more services for a defined time period. Programmes can also include programme events.

```
<?xml version="1.0" encoding="UTF-8"?>
<epg xmlns="http://www.worlddab.org/schemas/epgSchedule/14"
xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.worlddab.org/schemas/epgSchedule/14 epgSchedule_14.xsd" system="DAB"
xml:lang="en">
    <schedule version="1" creationTime="2001-02-28T00:00:00" originator="BBC">
        <scope startTime="2001-03-01T00:00:00" stopTime="2001-03-02T18:00:00">
            <serviceScope id="e1.ce15.c221.0"/>
            <serviceScope id="e1.ce15.c224.0"/>
        </scope>
        <!-- Comprehensive example -->
        <programme shortId="213456" id="crid://bbc.co.uk/4969758988" recommendation="yes">
            <epg:mediumName>Gilles Peterson:</epg:mediumName>
            <epg:longName>Gilles Peterson: Worldwide</epg:longName>
            <epg:location>
                <epg:time time="2003-12-18T00:00:00" duration="PT2H0M0S" actualTime="2003-12-
18T00:00:00" actualDuration="PT2H0M0S"/>
                <epg:bearer id="e1.ce15.c221.0"/>
            </epg:location>
            <epg:mediaDescription>
                <epg:shortDescription><![CDATA[Gilles Peterson brings you two hours of global beats
and the best of cool.   Including the Worldwide family. KV5 are live from Maida Vale with special
guests.]]></epg:shortDescription>
            </epg:mediaDescription>
            <epg:genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.7">
                <epg:name><![CDATA[ Rap/Hip Hop/Reggae]]></epg:name>
            </epg:genre>
            <epg:genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.8">
                <epg:name><![CDATA[ Electronic/Club/Urban/Dance]]></epg:name>
            </epg:genre>
            <epg:genre href="urn:tva:metadata:cs:FormatCS:2002:2.5">
                <epg:name><![CDATA[ ARTISTIC PERFORMANCE]]></epg:name>
            </epg:genre>
            <epg:genre href="urn:tva:metadata:cs:IntentionCS:2002:1.1">
                <epg:name><![CDATA[ ENTERTAINMENT]]></epg:name>
            </epg:genre>
            <epg:genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.9">
                <epg:name><![CDATA[ World/Traditional/Ethnic/Folk music]]></epg:name>
            </epg:genre>
            <epg:memberOf shortId="1000" id="crid://www.bbc.co.uk/WorldwideGroup"/>
            <epg:link url="mailto:gilles.peterson@bbc.co.uk" description="Email:"/>
            <epg:link url="http://www.bbc.co.uk/radio1/urban/peterson/" description="Web:"/>
            <epg:programmeEvent shortId="6353" id="crid://www.bbc.co.uk;dab/BC81123456a"
recommendation="yes">
                <epg:shortName xml:lang="en">Herbert</epg:shortName>
                <epg:mediumName xml:lang="en">Herbert Live</epg:mediumName>
                <epg:longName xml:lang="en">Live session from Herbert</epg:longName>
                <epg:location>
                    <epg:relativeTime time="PT45M" duration="PT15M"/>
                </epg:location>
                <epg:mediaDescription>
                    <epg:shortDescription xml:lang="en">Live session from Herbert, recorded at Cargo
on 24/2/01</epg:shortDescription>
                </epg:mediaDescription>
            </epg:programmeEvent>
        </programme>
        <!-- Minimum example -->
        <programme shortId="59033">
            <epg:mediumName>PM</epg:mediumName>
            <epg:location>
                <epg:time time="2003-12-18T17:00:00" duration="PT1H0M0S"/>
                <epg:bearer id="e1.ce15.c224.0"/>
            </epg:location>
        </programme>
    </schedule>
</epg>
```

4.3.2 Group information

Group information allows programmes to be put into groups. These may be series, serials or just general themes. A hierarchical approach also allows groups to belong to other groups.

NOTE: This example defines the group that is pointed to by the first programme in the previous example. This group also belongs to another group, "Radio1_Series" that is not defined here.

```
<?xml version="1.0" encoding="UTF-8"?>
<epg xmlns="http://www.worlddab.org/schemas/epgSchedule/14"
xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.worlddab.org/schemas/epgSchedule/14 epgSchedule_14.xsd" system="DAB"
xml:lang="en">
    <programmeGroups version="1" creationTime="2001-02-28T00:00:00" originator="BBC">
        <programmeGroup shortId="1000" id="crid://www.bbc.co.uk/WorldwideGroup" version="1"
type="show" numOfItems="206">
            <epg:mediumName xml:lang="en">Gilles Peterson</epg:mediumName>
            <epg:longName xml:lang="en">Gilles Peterson: Worldwide</epg:longName>
            <mediaDescription>
                <epg:shortDescription xml:lang="en">Worldwide: Global beats.</epg:shortDescription>
            </mediaDescription>
            <mediaDescription>
```

```
                <epg:longDescription xml:lang="en">Worldwide: Music from the back room of Club Radio
1.</epg:longDescription>
            </mediaDescription>
            <genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.7">
                <epg:name><![CDATA[ Rap/Hip Hop/Reggae]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.8">
                <epg:name><![CDATA[ Electronic/Club/Urban/Dance]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:FormatCS:2002:2.5">
                <epg:name><![CDATA[ ARTISTIC PERFORMANCE]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:IntentionCS:2002:1.1">
                <epg:name><![CDATA[ ENTERTAINMENT]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.9">
                <epg:name><![CDATA[ World/Traditional/Ethnic/Folk music]]></epg:name>
            </genre>
            <memberOf shortId="100" id="crid://www.bbc.co.uk/Radio1_Series"/>
        </programmeGroup>
    </programmeGroups>
</epg>
```

4.3.3 Service information

Service information includes the structure of and information about the broadcast channel and its associated services.

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceInformation xmlns="http://www.worlddab.org/schemas/epgSI/14"
xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.worlddab.org/schemas/epgSI/14 epgSI_14.xsd" version="1"
creationTime="2001-02-28T00:00:00" originator="BBC" serviceProvider="BBC" system="DAB"
xml:lang="en">
    <!-- Comprehensive example -->
    <ensemble id="e1.ce15">
        <epg:shortName xml:lang="en">BBC</epg:shortName>
        <epg:mediumName xml:lang="en">BBC National</epg:mediumName>
        <frequency type="primary" kHz="225648"/>
        <mediaDescription>
            <epg:shortDescription xml:lang="en">Digital Radio from the BBC</epg:shortDescription>
        </mediaDescription>
        <mediaDescription>
            <epg:multimedia url="http://www.bbc.co.uk/radio1/images/bbclogo.png"
type="logo_colour_rectangle"/>
        </mediaDescription>
        <mediaDescription>
            <epg:multimedia mimeValue="image/png" url="http://www.bbc.co.uk/radio/bbclogo_large.png"
type="logo_unrestricted" height="200" width="200"/>
        </mediaDescription>
        <CA type="none"/>
        <keywords xml:lang="en">Radio1, Radio2, Radio3, Radio4, Radio5,Live</keywords>
        <link url="http://www.bbc.co.uk/radio/" mimeValue="text/html" description="BBC Radio
homepage"/>
        <service format="audio" bitrate="160" version="1">
            <serviceID id="e1.ce15.c221.0" type="primary"/>
            <epg:shortName xml:lang="en">Radio 1</epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio 1</epg:mediumName>
            <mediaDescription>
                <epg:shortDescription xml:lang="en">Rock and pop music from the
BBC.</epg:shortDescription>
            </mediaDescription>
            <mediaDescription>
                <epg:multimedia url="http://www.bbc.co.uk/radio1/images/r1logo.png"
type="logo_colour_square"/>
            </mediaDescription>
            <genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.7">
                <epg:name><![CDATA[ Rap/Hip Hop/Reggae]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.8">
                <epg:name><![CDATA[ Electronic/Club/Urban/Dance]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:FormatCS:2002:2.5">
                <epg:name><![CDATA[ ARTISTIC PERFORMANCE]]></epg:name>
            </genre>
            <genre href="urn:tva:metadata:cs:IntentionCS:2002:1.1">
                <epg:name><![CDATA[ ENTERTAINMENT]]></epg:name>
            </genre>
            <epgLanguage xml:lang="en"/>
            <keywords xml:lang="en"> music, pop, rock, dance, hip-hop, soul   </keywords>
            <link url="http://www.bbc.co.uk/radio1/" mimeValue="text/html" xml:lang="en"/>
        </service>
    </ensemble>
    <!-- Minimum example -->
    <ensemble id="e1.ce15">
        <epg:shortName xml:lang="en">BBC</epg:shortName>
        <epg:mediumName xml:lang="en">BBC National</epg:mediumName>
        <service>
            <serviceID id="e1.ce15.c221.0"/>
            <epg:shortName xml:lang="en">Radio 1</epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio 1</epg:mediumName>
        </service>
        <service>
            <serviceID id="e1.ce15.c222.0"/>
            <epg:shortName xml:lang="en">BBCR2 </epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio 2</epg:mediumName>
        </service>
        <service>
            <serviceID id="e1.ce15.c223.0"/>
            <epg:shortName xml:lang="en">BBCR3</epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio 3</epg:mediumName>
        </service>
        <service>
            <serviceID id="e1.ce15.c224.0"/>
            <epg:shortName xml:lang="en">BBCR4</epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio 4</epg:mediumName>
        </service>
        <service>
            <serviceID id="e1.ce15.c225.0"/>
            <epg:shortName xml:lang="en">BBC5L</epg:shortName>
            <epg:mediumName xml:lang="en">BBC Radio Five Live</epg:mediumName>
        </service>
    </ensemble>
</serviceInformation>
```

5 Common data types

This clause describes common data types (simple and complex types) that are used throughout this XML specification.

5.1 Text

Any text sections in attributes or elements should be careful to avoid using any of the reserved XML characters:

& < > " '

These characters should be encoded using the predefined entity references (&amp; &lt; &gt; &quot; &apos;) or enclosed in a CDATA section (e.g. <![CDATA[Some text including an &]]>.

5.2 Schema simple types

5.2.1 broadcastType

```
<!-- ##################################################################### -->
<!-- Definition of broadcastType-->
<!-- ##################################################################### -->
<xs:simpleType name="broadcastType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="on-air"/>
        <xs:enumeration value="off-air"/>
    </xs:restriction>
</xs:simpleType>
```

This indicates, for the duration of this programme or event, whether the parent service is being broadcast (i.e. "on-air") or not (i.e. "off-air"). At times when a service is not being broadcast the broadcaster can use this facility to include "dummy" EPG entries that promote the service.

5.2.2 CRIDType

```
<!-- ########################################## -->
<!-- Definition of CRIDType -->
<!-- ########################################## -->
<simpleType name="CRIDType">
    <restriction base="xs:anyURI">
        <whiteSpace value="collapse"/>
        <pattern value="(c|C)(r|R)(i|I)(d|D)://.*/.*"/>
    </restriction>
</simpleType>
```

A unique identifier for a programme, programme event or programme group in the format of a Content Reference ID as defined in the TV-Anytime specification [1]. This CRID (Content Reference ID) should be in the form of *crid://<authority>/<data>*. Where *<authority>* is a registered Internet domain name that the CRID author has permission to use. The *<authority>* string is case insensitive. *<data>* is a free format string (URI compliant and case insensitive) that is meaningful to the given authority and should uniquely identify the content within that authority. Also see the shortCRIDType in clause 5.2.11.

E.g.

```
    crid://www.bbc.co.uk/811200000256129
    crid://broadcaster.co.uk/foobar
```

5.2.3 contentIDType

```
    <!-- ##################################################################### -->
    <!-- Definition of contentIDType-->
    <!-- ##################################################################### -->
    <xs:simpleType name="contentIDType">
        <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse"/>
            <xs:pattern value="(([0-9a-fA-F]{2}\.[0-9a-fA-F]{4}\.)?[0-9a-fA-F]{4,8}\.[0-9a-fA-
F]{1}(\.[0-9a-fA-F]{2})?)|([0-9a-fA-F]{6})"/>
        </xs:restriction>
    </xs:simpleType>
```

This is used to identify content elements.

For DAB, this is the content of the DAB ensemble. It shall be a string of the form:

<ECC>.<EId>.<SId>.<SCIdS>.<X-PAD> in hex

The Eureka-147 [18] Ensemble Identifier (<EId>) and the Extended Country Code (ECC) are optional. The SId (Service ID) is either a 16-bit service identifier (for audio services) or a 32-bit service identifier (for data services). The X-PAD application type (<X-PAD>) is optional.

NOTE: ECC = Extended Country Code, EId = Ensemble Id, SId = Service Id, SCIdS = Service component Id within Service, X-PAD = X-PAD application type.

E.g.

```
    e1.ce15.c221.0.1
    c224.0
```

For DRM, this is the content of the DRM channel. It shall be a string of the form:

<SId> in hex

The SId (Service ID) is the 24-bit service identifier.

E.g.

```
    e1c238
```

5.2.4 durationType

```
    <!-- ######################################### -->
    <!--  Definition of durationType  -->
        <!-- ######################################### -->
        <!-- restrictions: no leading minus sign, no year/month/day, no fractional seconds -->
        <!-- Note: maximum of 18 hours is not enforced -->
        <xs:simpleType name="durationType">
            <xs:restriction base="xs:duration">
                <xs:pattern value="PT[^\.]+"/>
            </xs:restriction>
        </xs:simpleType>
```

Duration is based on the ISO 8601 [2] extended format: PTnHnMnS, where "T" represents the date/time separator, "nH" the number of hours, "nM" the number of minutes and "nS" the number of seconds. The values of the Hour, Minutes and Seconds components are not restricted but allow an arbitrary integer. Reduced precision and truncated representations of this format are allowed provided they conform to the following:

- The lowest order items may be omitted. If omitted their value is assumed to be zero.
- If the number of hours, minutes or seconds in any expression equals zero, the number and its corresponding designator may be omitted. However, at least one number and its designator must be present.

The designator "PT" must always be present.

NOTE: The value of the duration may be restricted by the binary encoding that is used when broadcasting an EPG. See TS 102 371 [22] for more information.

E.g.

```
    PT2H
    PT2H30M
    PT15M
```

5.2.5 ensembleIDType

```
<!-- ###################################################################### -->
<!-- Definition of ensembleIDType -->
<!-- ###################################################################### -->
<xs:simpleType name="ensembleIDType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="([0-9a-fA-F]{2}\.[0-9a-fA-F]{4})|([0-9a-fA-F]{6})"/>
    </xs:restriction>
</xs:simpleType>
```

For DAB, this is a string in the form <ECC>.<EId> in hex representing the Extended Country Code and Ensemble Identifier as defined in Eureka-147 [18].

E.g.

```
    e1.ce15
```

For DRM, this is a string in the form <SId> in hex representing the Service Identifier of one of the services in this DRM channel.

E.g.

```
    e1c238
```

5.2.6 mimeType

```
<!-- ###################################################################### -->
<!-- Definition of mimeType (Multipurpose Internet Mail Extension -->
<!-- ###################################################################### -->
<xs:simpleType name="mimeType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="([!-\.0-~]{1,}/[!-\.0-~]{1,})+"/>
    </xs:restriction>
</xs:simpleType>
```

This indicates the MIME type (RFC 2045 [6]) of some data and must be used where it is applicable. The registered list of MIME types is available from the IANA list of Mime Types (RFC 2046 [14] and RFC 2048 [15]). However, an application is permitted to use values not in this list as long as they conform to the requirements set out in RFC 2046 [14]. For example, an application may use the private MIME type "application/x-myapplication", even if this is not a registered MIME type.

E.g

```
    text/shtml
```

5.2.7 originatorType

```
<!-- ##################################################################### -->
<!-- Definition of originatorType -->
<!-- ##################################################################### -->
<xs:simpleType name="originatorType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>
```

This is used to indicate the originator of an EPG schedule.

5.2.8 recommendationType

```
<!-- ##################################################################### -->
<!-- Definition of recommendationType-->
<!-- ##################################################################### -->
<xs:simpleType name="recommendationType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
    </xs:restriction>
</xs:simpleType>
```

This is used by the broadcaster to indicate a recommended programme or programme event.

5.2.9 serviceProviderType

```
<!-- ##################################################################### -->
<!-- Definition of serviceProviderType -->
<!-- ##################################################################### -->
<xs:simpleType name="serviceProviderType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>
```

This is used to indicate the service provider of the services contained in an EPG schedule.

5.2.10 shortCRIDType

```
    <!-- ########################################## -->
    <!-- Definition of shortCRIDType -->
    <!-- ########################################## -->
    <xs:simpleType name="shortCRIDType">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="16777215"/>
        </xs:restriction>
    </xs:simpleType>
```

An identifier for a programme, programme event or programme group. Unlike a full Content Reference ID (CRID - see clause 5.2.2) this is designed to be more appropriate for limited bandwidth data channels and for basic EPG receivers. The short CRID (sCRID) is a 24-bit integer, expressed as a decimal value, with a range of 0 to 16,777,215 inclusive. The following rules should be applied whenever short CRIDs are used in an EPG service.

- The sCRID shall only be unique within a single EPG Service (see note), therefore a receiver must process it in some way on decoding to ensure that it is globally unique.

- The sCRID must not be re-used within that EPG service for a minimum of six months.

NOTE: An "EPG service" is defined as EPG data for one or more services, broadcast in a single (EPG) data channel.

5.2.11 systemType

```
<!-- ##################################################################### -->
<!-- Definition of systemType-->
<!-- ##################################################################### -->
<xs:simpleType name="systemType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="DAB"/>
        <xs:enumeration value="DRM"/>
    </xs:restriction>
</xs:simpleType>
```

This specifies the broadcast system that this EPG data supports. This may be extended in the future to include other systems in addition to DAB and DRM, e.g. Satellite Digital Audio Radios (SDARs).

5.2.12 timePointType

```
    <!-- ########################################## -->
    <!--  Definition of timePointType  -->
    <!-- ########################################## -->
    <!-- restrictions: no leading minus sign, no fractional seconds -->
    <xs:simpleType name="timePointType">
        <xs:restriction base="xs:dateTime">
            <xs:pattern value="[^\-].+T[^\.]+"/>
        </xs:restriction>
    </xs:simpleType>
```

This is a time field in local time. It is based on the ISO 8601 [2] extended format: *YYYY-MM-DDThh:mm:ss* where "*YYYY*" is the year, "*MM*" the month and "*DD*" the date. The letter "*T*" is the date/time separator and "*hh*", "*mm*" and "*ss*" represent the hour, minute and second respectively. To indicate the time zone, i.e. the difference between the local time and UTC, the difference immediately follows the time and consists of a sign, + or -, followed by hh:mm. If this is not present then the difference between local time and UTC is 0.

NOTE 1: For future compatibility the representation may be immediately followed by a "Z" to indicate Co-ordinated Universal Time (UTC).

NOTE 2: The valid values of date may be restricted by the binary encoding that is used when broadcasting an EPG. See TS 102 371 [22] for more information.

E.g.

```
    2001-06-07T15:05:00+01:00
    2001-02-03T23:00:00
```

5.2.13 triggerType

```
<!-- ##################################################################### -->
<!-- Definition of triggerType -->
<!-- ##################################################################### -->
<xs:simpleType name="triggerType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="[0-9a-fA-F]{8}"/>
    </xs:restriction>
</xs:simpleType>
```

This represents a trigger found in the broadcast stream that indicates when a programme is being broadcast. For the purposes of the present document this is the two SId and two PNum bytes from the DAB FIG0/16 Programme Number. The triggerType requires a complete set of 8 hexadecimal characters. Hence, in some cases leading zeros will be required as place-holders.

E.g.

```
    c2213ac1
    0b74ff10
```

5.2.14 urlType

```
<!-- ##################################################################### -->
<!-- Definition of urlType-->
<!-- ##################################################################### -->
<xs:simpleType name="urlType">
    <xs:restriction base="xs:anyURI">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="
        ((crid|CRID|tel|mailto|postal|http|dab|drm):(//|\+|SMS=)?)?([a-zA-Z0-9]|\.|@|%|\-
        |/|_|\+|\?|=|;){1,} "/>
    </xs:restriction>
</xs:simpleType>
```

This is a string describing the address and protocol of a resource, in the URL format defined in RFC 2396 [3]. Where URL schemes have previously been defined these should be used. The following schemes are supported in this version:

- **Telephone and fax:** Defined in RFC 2806 [9].
- **SMS:** Address defined using RFC 3191 [10] combined with the protocol defined in RFC 2368 [11].
- **Postal addresses:** Defined in annex A.
- **Email:** Defined in RFC 2368 [11].
- **WWW and WAP:** Defined in RFC 2396 [3] and WAP Forum [4].
- **DAB and DRM:** Defined in annex B.
- **Programme:** A CRID as defined in clause 5.2.2, or a short CRID as defined in clause 5.2.11 but prefixed with "crid://".

NOTE: The "CRID" prefix does not support mixed case, i.e. it may be either all lower-case or all upper-case.

E.g.

```
tel:+44-1737-839500
mailto:SMS=+44-7788-123456?body=more%20info
postal:Kingswood%20Warren/Tadworth/Surrey/KT20%206NP/United%20Kingdom/
mailto:gilles.peterson@bbc.co.uk
http://www.bbc.co.uk/
data/logo.png
crid://www.bbc.co.uk/BC81123456
crid://25336
```

5.3 Schema complex types

5.3.1 CAType

```
<!-- ##################################################################### -->
<!-- Definition of CAType -->
<!-- ##################################################################### -->
<xs:complexType name="CAType">
    <xs:attribute name="type" default="none">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="unspecified"/>
                <xs:enumeration value="none"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
```

This defines Conditional Access (CA) information. The "type" attribute indicates which CA system is in use (see note); "unspecified" indicates that an unspecified or proprietary CA system is in use and "none" explicitly states that no CA system is in use.

NOTE: There will be additions to this list when CA systems are defined and further information may also be added.

E.g.

```
<CA type="unspecified"/>
```

5.3.2 epgLanguageType

```
<!-- ###################################################################### -->
<!-- Definition of element epgLanguageType-->
<!-- ###################################################################### -->
<xs:complexType name="epgLanguageType">
    <xs:attribute ref="xml:lang" default="en"/>
</xs:complexType>
```

This indicates the language for an element. This is in the form of an xml:lang attribute [5] and RFC 3066 [7].

E.g.

```
xml:lang="en"
```

5.3.3 genreType

```
<!-- ###################################################################### -->
<!-- Definition of genreType -->
<!-- ###################################################################### -->
<xs:complexType name="genreType">
    <xs:sequence>
        <xs:element name="name" minOccurs="0">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="messageType">
                        <xs:attribute name="preferred" type="xs:boolean" use="optional"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="definition" type="messageType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="href" type="termReferenceType" use="required"/>
    <xs:attribute name="type" use="optional" default="main">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="main"/>
                <xs:enumeration value="secondary"/>
                <xs:enumeration value="other"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
```

This indicates the genre of a programme, group or service (audio or data). The genre scheme is based on that used by TV-Anytime [21] and Appendix A of this reference should be referred to for details of the Classification Schemes (CS). The supported classification schemes are:

- IntentionCS.
- FormatCS.
- ContentCS.
- OriginationCS.
- IntendedAudienceCS.
- ContentAlertCS.

- MediaTypeCS.
- AtmosphereCS.

The href is the only required element and this specifies the genre, the Classification Scheme (CS) and the genre scheme used. The name element, if used, should contain the name of the genre. The definition element, if used, should contain a description of the genre. Both of these elements are intended to make the element more readable for humans. The type attribute indicates the type of the genre. The types of genres are defined as follows:

- **main:** The specified genre is the main, or primary. This is the default value.
- **secondary:** The specified genre is a secondary genre, such as a subgenre.
- **other:** The specified genre is an alternative genre, such as one defined or used by 3rd parties.

E.g.

```
<epg:genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.9">
    <epg:name><![CDATA[World/Traditional/Ethnic/Folk music]]></epg:name>
</epg:genre>
```

5.3.4 keywordsType

```
<!-- ###################################################################### -->
<!-- Definition of keywordsType -->
<!-- ###################################################################### -->
<xs:complexType name="keywordsType">
    <xs:simpleContent>
        <xs:extension base="messageType">
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

This contains a comma-separated list of keywords. The language attribute indicates the language of the keyword list and is in the form of an xml:lang attribute [5] and RFC 3066 [7]. The keywords must be separated by commas. The comma-separated list may have leading and trailing spaces, but these are not considered to contain information.

E.g.

```
    <keywords xml:lang="en">music, dance, hip-hop, jazz, soul</keywords>
```

5.3.5 linkType

```
<!-- ###################################################################### -->
<!-- Definition of linkType -->
<!-- ###################################################################### -->
<xs:complexType name="linkType">
    <xs:attribute name="url" type="epg:urlType" use="required"/>
    <xs:attribute name="mimeValue" type="mimeType"/>
    <xs:attribute ref="xml:lang" default="en"/>
    <xs:attribute name="description">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="180"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="expiryTime" type="timePointType"/>
</xs:complexType>
```

This is used to link to additional information or content. The "url" attribute gives the protocol and address of the link. The "mimeValue" attribute indicates the MIME type (RFC 2045 [6]) of any data linked to and must be used where it is applicable. The language attribute indicates the language of the descriptive information and is in the form of an xml:lang attribute [5] and RFC 3066 [7]. The "description" attribute is used to describe the link. The "expiryTime" attribute indicates when a link will expire.

E.g.

```
    <link url="http://www.bbc.co.uk/radio1/urban/peterson_tracklistings_archive.shtml"
        mimeValue="text/shtml"
        xml:lang="en"
        description="Track listing"
        expiryTime="2001-07-09T23:59:59+01:00"
    />
```

5.3.6 locationType

```
    <!-- #################################################################### -->
    <!-- Definition of locationType -->
    <!-- #################################################################### -->
    <xs:complexType name="locationType">
        <xs:sequence>
            <xs:choice>
                <xs:element name="time" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="time" type="timePointType" use="required"/>
                        <xs:attribute name="duration" type="durationType" use="required"/>
                        <xs:attribute name="actualTime" type="timePointType"/>
                        <xs:attribute name="actualDuration" type="durationType"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="relativeTime" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="time" type="durationType" use="required"/>
                        <xs:attribute name="duration" type="durationType" use="required"/>
                        <xs:attribute name="actualTime" type="durationType"/>
                        <xs:attribute name="actualDuration" type="durationType"/>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
            <xs:element name="bearer" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="id" type="contentIDType" use="required"/>
                    <xs:attribute name="trigger" type="triggerType"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
```

This describes the time information and the location in the DAB or DRM channel of a programme. There may be:

- One time element and one bearer element.
- One time element and multiple bearer elements.
- One bearer element and multiple time elements.

"Multiple time elements and multiple bearer elements" is ambiguous and must not be used. If the "bearer" element for a "programmeEvent" is not present, the "bearer" element from the parent "programme" defines the bearer for that "programmeEvent". If the EPG data service is associated with an audio service (i.e. PAD within DAB) and the "bearer" element is not present for the "programme", then the programme information relates to that associated audio service.

The "time" element describes the time information for a programme. The "time" attribute is the billed start time of the programme (i.e. the time advertised to the public as the programme start time) and the "duration" attribute is the billed duration. The "actualTime" attribute is the actual start time of the programme and the "actualDuration" is the actual duration of the programme. For example, a programme may be billed to start at 18:00 and last 30 minutes but is actually scheduled to start at 18:03 after a 3 minute news bulletin, and will therefore last only 27 minutes.

The "relativeTime" element should be used where a start time is relative to another element. This element shall only be used for "programmeEvent" elements that are occurring within programmes. The "time" attribute uses the durationType to represent the time since the start of the "parent" element (i.e. an event starting at the beginning of the programme would have a relativeTime of 0).

NOTE 1: A programme starting at 18:00:00 with a duration of 30 minutes will finish at 18:30:00. The next programme in a contiguous sequence starts at 18:30:00.

NOTE 2: If a programme has more than one "time" element associated with it then the first "time" element chronologically is the first broadcast of the programme. Future "time" elements are repeats of the same programme. If the next programme in, for example, a series has exactly the same programme details, then a separate programme entry should be generated, and the programmes linked together into the same Group.

The "bearer" element describes the location of a programme in the DAB or DRM channel.

Example 1 (DAB)

```
    <location>
        <time time="2001-03-01T00:00:00"
            duration="PT2H"
            actualTime="2001-03-01T00:00:03"
            actualDuration="PT1H55M"/>
        <bearer id="e1.ce15.c221.0" trigger="c2213ac1"/>
    </location>
```

Example 2 (DRM)

```
    <location>
        <relativeTime time="PT15M"
            duration="PT1H"
            actualTime="PT15M15S"
            actualDuration="PT55M"/>
        <bearer id="e1c238"/>
    </location>
```

5.3.7 longDescriptionType

```
<!-- ##################################################################### -->
<!-- Definition of longDescriptionType -->
<!-- ##################################################################### -->
<xs:complexType name="longDescriptionType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="1200"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
```

This element is a string that represents a long description.

5.3.8 longNameType

```
<!-- ##################################################################### -->
<!-- Definition of longNameType -->
<!-- ##################################################################### -->
<xs:complexType name="longNameType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="128"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
```

This element is a string that represents a long name.

5.3.9 mediaDescriptionType

```
<!-- ##################################################################### -->
<!-- Definition of mediaDescriptionType -->
<!-- ##################################################################### -->
<xs:complexType name="mediaDescriptionType">
    <xs:choice>
        <xs:group ref="descriptionGroup"/>
        <xs:element name="multimedia">
            <xs:complexType>
                <xs:attribute name="mimeValue" type="mimeType" use="optional"/>
                <xs:attribute ref="xml:lang" default="en"/>
                <xs:attribute name="url" type="urlType"/>
                <xs:attribute name="type" use="optional">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="logo_unrestricted"/>
                            <xs:enumeration value="logo_mono_square"/>
                            <xs:enumeration value="logo_colour_square"/>
                            <xs:enumeration value="logo_mono_rectangle"/>
                            <xs:enumeration value="logo_colour_rectangle"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="width" type="xs:nonNegativeInteger"/>
                <xs:attribute name="height" type="xs:nonNegativeInteger"/>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:complexType>
```

This element represents an aggregation of all other descriptive elements (text and multimedia). The "multimedia" element links to a multimedia resource for this element. The optional "mimeValue" attribute indicates the MIME type of the resource. The language attribute indicates the language of the source information and is in the form of an xml:lang attribute [5] and RFC 3066 [7]. The "url" attribute points to the multimedia resource. The optional "type" attribute indicates the type of multimedia resource and is intended to support the correct presentation of the resource by the EPG decoder. The enumerated types allowed are defined here:

- **logo_unrestricted:** An ensemble/service/programme/programme group/programme event logo. This image format is unrestricted and must be signalled with the "mimeValue", "width" and "height" attributes.

- **logo_mono_square:** An ensemble/service/programme/programme group/programme event logo. This image must be in Wireless Bitmap (WBMP) format and must be 32 × 32 pixels. The "mimeValue", "width" and "height" attributes should not be used. The intended use of this type of logo is as part of a list of available services on a receiver with limited display capabilities.

- **logo_colour_square:** An ensemble/service/programme/programme group/programme event logo. This image must be in PNG v1.1 format and must be 32 × 32 pixels at a colour depth of 256. The "mimeValue" , "width" and "height" attributes should not be used. The intended use of this type of logo is as part of a list of available services.

- **logo_mono_rectangle:** An ensemble/service/programme/programme group/programme event logo. This image must be in Wireless Bitmap (WBMP) format and must be 32 pixels high and between 33 pixels and 112 pixels (inclusive) wide. The "mimeValue", "width" and "height" attributes should not be used. The intended use of this type of logo is to display at the top of a page for a particular service listing the programmes and can be used in place of the textual name for this service on a receiver with limited display capabilities.

- **logo_colour_rectangle:** An ensemble/service/programme/programme group/programme event logo. This image must be in PNG v1.1 format and must be must be 32 pixels high and between 33 pixels and 112 pixels (inclusive) wide. The "mimeValue", "width" and "height" attributes should not be used. The intended use of this type of logo is to display at the top of a page for a particular service listing the programmes and can be used in place of the textual name for this service on a receiver with limited display capabilities.

NOTE: The context of the logo (i.e. is it related to an ensemble or a service or a programme etc.) is defined by the parent elements of the multimedia element.

The optional "width" and "height" attributes define the size of the logo in pixels.

It is recommended that receivers with graphical capabilities should support one or more of the following formats:

- Wireless Bitmap (WBMP): This format is suitable for small, monochrome images for display on receivers with limited displays. The minimum image specification supported by WBMP-compatible receivers should be 32 × 32 pixels [12].

- PNG v1.1: This format is suitable for colour images for display on more capable receivers. The minimum image specification supported by PNG-compatible receivers should be 32 × 32 pixels at a colour depth of 256 [8].

It is recommended that receivers with the capability to play audio files should support the following formats:

- The native "system" audio format. For DAB the native audio formats are MPEG-1 audio layer II (ISO/IEC 11172-3 [16]), MPEG-2 audio layer II (ISO/IEC 13818-3 [17]) and HE AAC v2 [24]. For DRM the native audio formats are specified in ES 201 980 [23].

- MPEG-1 layer 3 audio playback.

E.g.

```
    <mediaDescription>
        <shortDescription xml:lang="en">Rock and pop music from the BBC.</shortDescription>
        <multimedia type="logo_colour_square" url="http://www.bbc.co.uk/radio1/r1logo.png"/>
    </mediaDescription>
```

5.3.10 mediumNameType

```
<!-- ##################################################################### -->
<!-- Definition of mediumNameType -->
<!-- ##################################################################### -->
<xs:complexType name="mediumNameType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="16"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
```

This element is a string that represents a medium name.

5.3.11 memberOfType

```
<!-- ##################################################################### -->
<!-- Definition of memberOfType -->
<!-- ##################################################################### -->
<xs:complexType name="memberOfType">
    <xs:attribute name="shortId" type="shortCRIDType" use="required"/>
    <xs:attribute name="id" type="CRIDType" use="optional"/>
    <xs:attribute name="index" type="xs:positiveInteger" use="optional"/>
</xs:complexType>
```

This indicates which group this element belongs to. The "id" attribute refers to the id of a group element (see clause 5.2.2). The "index" attribute is an index for the item within the specified group. This would be used, for example, to specify an episode number for a programme in a series

E.g.

```
    <memberOf shortId="123456" id="crid://www.bbc.co.uk/G123456" index="206"/>
```

5.3.12 messageType

```
<!-- ##################################################################### -->
<!-- Definition of messageType -->
<!-- ##################################################################### -->
<xs:complexType name="messageType" abstract="true">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="xml:lang" use="optional" default="en"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

This is an abstract element for textual elements that have an optional language attribute.

5.3.13 programmeType

```
    <!-- ##################################################################### -->
    <!-- Definition of programmeType -->
    <!-- ##################################################################### -->
    <xs:complexType name="programmeType">
        <xs:sequence>
            <xs:group ref="scheduleNameGroup" maxOccurs="unbounded"/>
            <xs:element name="location" type="locationType" maxOccurs="unbounded"/>
            <xs:element name="mediaDescription" type="mediaDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="genre" type="genreType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="CA" type="CAType" minOccurs="0"/>
            <xs:element name="keywords" type="keywordsType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="memberOf" type="memberOfType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="link" type="linkType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="programmeEvent" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:group ref="scheduleNameGroup" maxOccurs="unbounded"/>
                        <xs:element name="location" type="locationType" maxOccurs="unbounded"/>
                        <xs:element name="mediaDescription" type="mediaDescriptionType"
minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="genre" type="genreType" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="CA" type="CAType" minOccurs="0"/>
                        <xs:element name="keywords" type="keywordsType" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="memberOf" type="memberOfType" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="link" type="linkType" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="shortId" type="ShortCRIDType" use="required"/>
                    <xs:attribute name="id" type="CRIDType" use="optional"/>
                    <xs:attribute name="version" type="xs:integer"/>
                    <xs:attribute name="recommendation" type="recommendationType" default="no"/>
                    <xs:attribute name="broadcast" type="broadcastType" default="on-air"/>
                    <xs:attribute ref="xml:lang" use="optional" default="en"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="shortId" type="ShortCRIDType" use="required"/>
        <xs:attribute name="id" type="CRIDType" use="optional"/>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="recommendation" type="recommendationType" default="no"/>
        <xs:attribute name="broadcast" type="broadcastType" default="on-air"/>
        <xs:attribute name="bitrate" type="xs:nonNegativeInteger"/>
        <xs:attribute ref="xml:lang" use="optional" default="en"/>
    </xs:complexType>
```

This is used to describe and locate a programme. Where individual programmes vary from the default service bitrate, the highest bitrate that the programme broadcasts at can be indicated in the "bitrate" attribute. The "broadcast" flag can be set to "off-air" to generate "dummy" programmes for when a service is not being broadcast, by default this flag is "on-air" and should not be set. The "xml:lang" indicates the language of the programme, this should normally be used if this is different to the parent service language. The "CA" defines the Conditional Access for the programme, this should normally be used if this is different to the parent service CA.

The "programmeEvent" element describes an event within a programme, this can be used to break a programme into sections or to highlight particular sections of the programme.

NOTE: The recommended practice for describing repeated programmes is to have a programme element with more than one location element, each location element specifies one of the programme's instances in time (and/or service).

E.g.

```
    <programme shortId="23456" id="crid://www.bbc.co.uk/81123456"
        version="1"
        recommendation="yes"
        bitrate="160">
        …
    </programme>
```

The "version" attribute shall be incremented by one, for every new version of the programme.

5.3.14 shortDescriptionType

```
<!-- ##################################################################### -->
<!-- Definition of shortDescriptionType -->
<!-- ##################################################################### -->
<xs:complexType name="shortDescriptionType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="180"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
```

This element is a string that represents a short description.

5.3.15 shortNameType

```
<!-- ##################################################################### -->
<!-- Definition of shortNameType -->
<!-- ##################################################################### -->
<xs:complexType name="shortNameType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="8"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
```

This element is a string that represents a short name.

5.3.16 simulcastType

```
<!-- ##################################################################### -->
<!-- Definition of simulcastType -->
<!-- ##################################################################### -->
<xs:complexType name="simulcastType">
    <xs:attribute name="system" type="systemType" use="required"/>
    <xs:attribute name="id" type="contentIDType" use="required"/>
</xs:complexType>
```

This is used to indicate simulcast services on other broadcast systems. The "system" attribute gives the other broadcast system and the "id" attribute the service reference.

E.g.

```
    <simulcast system="DAB"
        id="e1.ce15.c238.0"
    />

    <simulcast system="DRM"
        id="e1c238"
    />
```

5.4 Schema groups

5.4.1 descriptionGroup

```
<!-- ##################################################################### -->
<!-- Definition of descriptionGroup -->
<!-- ##################################################################### -->
<xs:group name="descriptionGroup">
    <xs:sequence>
        <xs:element name="shortDescription" type="shortDescriptionType" minOccurs="0"
        maxOccurs="unbounded"/>
        <xs:element name="longDescription" type="longDescriptionType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
```

```
</xs:group>
```

The represents shortDescription and/or longDescription elements.

5.4.2 scheduleNameGroup

```
<!-- ##################################################################### -->
<!-- Definition of scheduleNameGroup -->
<!-- ##################################################################### -->
<xs:group name="scheduleNameGroup">
   <xs:sequence>
      <xs:element name="shortName" type="shortNameType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="mediumName" type="mediumNameType" maxOccurs="unbounded"/>
      <xs:element name="longName" type="longNameType" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>
```

This represents shortName, mediumName, and longName elements where mediumName must occur at least once.

5.4.3 serviceNameGroup

```
<!-- ##################################################################### -->
<!-- Definition of serviceNameGroup -->
<!-- ##################################################################### -->
<xs:group name="serviceNameGroup">
   <xs:sequence>
      <xs:element name="shortName" type="shortNameType" maxOccurs="unbounded"/>
      <xs:element name="mediumName" type="mediumNameType" maxOccurs="unbounded"/>
      <xs:element name="longName" type="longNameType" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:group>
```

This represents shortName, mediumName, and longName elements where both shortName and mediumName must occur at least once.

6 Schedules

6.1 Schema types

6.1.1 programmeGroupTypeType

```
<!-- ##################################################################### -->
<!-- Declaration of type programmeGroupTypeType -->
<!-- ##################################################################### -->
<xs:simpleType name="programmeGroupTypeType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="series"/>
        <xs:enumeration value="show"/>
        <xs:enumeration value="programConcept"/>
        <xs:enumeration value="magazine"/>
        <xs:enumeration value="programCompilation"/>
        <xs:enumeration value="otherCollection"/>
        <xs:enumeration value="otherChoice"/>
        <xs:enumeration value="topic"/>
    </xs:restriction>
</xs:simpleType>
```

This is used to indicate the type of the grouping.

- **series:** an ordered or unordered collection of programmes that is shown in a sequence (e.g. "The News Quiz" season 1).
- **show:** a programme theme that is typically associated with a collection of series (e.g. all episodes of "The News Quiz").
- **programConcept:** the editorial concept for a programme from which specific programme versions have been derived (e.g. the concept of "Blood Runner" as opposed to "Blood Runner - The Director's Cut" as a specific version of that concept).
- **magazine:** a collection of individual programmes that are shown as a group because they are editorially coherent (e.g. a general sports programme with individual sub-programmes covering different events).
- **topic:** a collection of programmes on a particular topic or theme.
- **programCompilation:** a collection of programmes that is used to allow segments from multiple programmes to be combined in segment groups.
- **otherCollection:** can be used for any group not defined in the preceding list where all members of the group should be acquired if the group is selected. For example, a group of channel highlights or recommendations.
- **otherChoice:** can be used for any group not defined in the list above where only one member of the group should be acquired if the group is selected.

6.1.2 programmeGroupType

```
<!-- ##################################################################### -->
<!-- Declaration of type programmeGroupType -->
<!-- ##################################################################### -->
<xs:complexType name="programmeGroupType">
    <xs:sequence>
        <xs:group ref="epg:scheduleNameGroup" maxOccurs="unbounded"/>
        <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="genre" type="epg:genreType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="keywords" type="epg:keywordsType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="memberOf" type="epg:memberOfType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="shortId" type="epg:shortCRIDType" use="required"/>
    <xs:attribute name="id" type="epg:CRIDType" use="optional"/>
    <xs:attribute name="version" type="xs:integer"/>
    <xs:attribute name="type" type="programmeGroupTypeType"/>
    <xs:attribute name="numOfItems" type="xs:positiveInteger"/>
</xs:complexType>
```

This is used to describe a grouping of programmes, programme events or other groups. The "numOfItems" is a positive integer field that indicates the total number of items in this group. The "type" attribute indicates the type of the grouping.

6.1.3 programmeGroupsType

```
<!-- ##################################################################### -->
<!-- Declaration of type programmeGroupsType -->
<!-- ##################################################################### -->
<xs:complexType name="programmeGroupsType">
    <xs:sequence>
        <xs:element name="programmeGroup" type="programmeGroupType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:integer"/>
    <xs:attribute name="creationTime" type="epg:timePointType"/>
    <xs:attribute name="originator" type="epg:originatorType"/>
</xs:complexType>
```

This is used as a container for group elements. The "creationTime" attribute indicates the time at which this group was generated and the "originator" attribute is used to indicate the originator of the group.

6.1.4 scheduleType

```
<!-- ##################################################################### -->
<!-- Declaration of type scheduleType -->
<!-- ##################################################################### -->
<xs:complexType name="scheduleType">
    <xs:sequence>
        <xs:element name="scope" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="serviceScope" minOccurs="0" maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:attribute name="id" type="epg:contentIDType" use="required"/>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="startTime" type="epg:timePointType" use="required"/>
                <xs:attribute name="stopTime" type="epg:timePointType" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:element name="programme" type="epg:programmeType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:integer"/>
    <xs:attribute name="creationTime" type="epg:timePointType"/>
    <xs:attribute name="originator" type="epg:originatorType"/>
</xs:complexType>
```

This allows programmes to be identified within a given time period. The "scope" element is used to indicate the time period covered by this schedule, from the billed start time of the first programme to the billed end time of the last programme. The "serviceScope" element is used to indicate the services covered by this schedule. The "creationTime" attribute indicates the time at which this schedule was generated and the "originator" attribute is used to indicate the originator of the schedule.

NOTE: For contiguous schedules the stop time of a schedule should be equal to the start time of the next schedule.

6.1.5 alternateSourceType

```
<!-- ##################################################################### -->
<!-- Declaration of type alternateSourceType -->
<!-- ##################################################################### -->
<xs:complexType name="alternateSourceType">
    <xs:attribute name="protocol" default="URL">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="DAB"/>
                <xs:enumeration value="DRM"/>
                <xs:enumeration value="URL"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="type" default="identical">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="more"/>
                <xs:enumeration value="less"/>
                <xs:enumeration value="similar"/>
                <xs:enumeration value="identical"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="url" type="epg:urlType" use="required"/>
</xs:complexType>
```

This is used to indicate whether this or related EPG data is available elsewhere. The "url" attribute identifies the location of this data and the "type" attribute indicates if there is more, less, equivalent or identical schedule information at the alternate location.

6.2 epg

```
<!-- ##################################################################### -->
<!-- Declaration of element epg -->
<!-- ##################################################################### -->
<xs:element name="epg">
    <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="programmeGroups" type="programmeGroupsType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="schedule" type="scheduleType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="alternateSource" type="alternateSourceType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute ref="xml:lang" use="required"/>
        <xs:attribute name="system" type="epg:systemType" default="DAB"/>
    </xs:complexType>
</xs:element>
```

The "epg" element is the root element of an EPG schedule. It may contain schedule, alternateSource, programme or group information elements in any order.

Programme, programmeEvent and group elements all point "upwards" to their parent group/s using the memberOf element. The "numOfItems" attribute of a group element can be used by a client-side application to determine when all the items in a group have been found.

E.g.

```
<epg system=DAB">
    <programmeGroups version="1" creationTime="2001-02-28T00:00:00" originator="BBC">
        <programmeGroup shortId="100" id="crid://www.bbc.co.uk/WomansHour"
            version="1"
            type="magazine"
            numOfItems="5">
            …
        </programmeGroup>
    </programmeGroups>

    <schedule version="1" creationTime="2001-02-28T00:00:00" originator="BBC">
        <scope startTime="2001-03-01T00:00:00" stopTime="2001-03-02T00:00:00">
            <serviceScope id="e1.ce15.c221.0"/>
        </scope>
        …
    </schedule>

    <alternateSource type="more" url="http://www.bbc.co.uk/whatson/radio4/xml/"/>
    <xml:lang = "en"/>
</epg>
```

7 Service Information

7.1 Schema types

7.1.1 frequencyType

```
<!-- ##################################################################### -->
<!-- Declaration of type frequencyType -->
<!-- ##################################################################### -->
<xs:complexType name="frequencyType">
    <xs:attribute name="type" default="primary">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="primary"/>
                <xs:enumeration value="alternative"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="kHz" type="xs:nonNegativeInteger"/>
</xs:complexType>
```

This is used to indicate primary and alternate frequencies for an ensemble or channel. When used, there must be a maximum of 1 primary frequency element per ensemble/channel. The "kHz" attribute gives the frequency in kHz.

7.1.2 formatType

```
<!-- ###################################################################### -->
<!-- Declaration of type formatType -->
<!-- ###################################################################### -->
<xs:simpleType name="formatType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="audio"/>
        <xs:enumeration value="data"/>
    </xs:restriction>
</xs:simpleType>
```

This indicates whether the service is audio or data.

NOTE: In previous versions of the present document, the enumeration could take additional values for non-audio services. This function is now provided by the extFormat element.

7.1.3 extFormatType

```
<!-- ###################################################################### -->
<!-- Declaration of type extFormatType -->
<!-- ###################################################################### -->
<xs:simpleType name="extFormatType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="(([0-3]{1}\.[0-9a-fA-F]{2}\.[0-9a-fA-F]{3})|([0-7]{1}\.[0-9a-fA-
F]{4}))((\.([0-9a-fA-F]{2})+)?)"/>
    </xs:restriction>
</xs:simpleType>
```

This gives additional information for data services. For DAB, the information consists of the TMId, the DSCTy, the User Application Type (UATy) and the User Application Data, all represented in hexadecimal form. For DRM the information consists of the Application Domain, the User Application Type and the User Application Data, all represented in hexadecimal form.

7.1.4 serviceIDType

```
<!-- ###################################################################### -->
<!-- Declaration of type serviceIDType -->
<!-- ###################################################################### -->
<xs:complexType name="serviceIDType">
    <xs:attribute name="id" type="epg:contentIDType" use="required"/>
    <xs:attribute name="type" default="primary">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="primary"/>
                <xs:enumeration value="secondary"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
```

This indicates the Service Identifier and whether it is a primary or secondary service component (DAB).

7.1.5 serviceType

```
<!-- ###################################################################### -->
<!-- Declaration of type serviceType -->
<!-- ###################################################################### -->
<xs:complexType name="serviceType">
    <xs:sequence>
        <xs:element name="serviceID" type="serviceIDType" maxOccurs="unbounded"/>
        <xs:element name="simulcast" type="epg:simulcastType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded"/>
        <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
```

```
        <xs:element name="genre" type="epg:genreType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="epgLanguage" type="epg:epgLanguageType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="CA" type="epg:CAType" minOccurs="0"/>
        <xs:element name="keywords" type="epg:keywordsType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:integer"/>
    <xs:attribute name="format" type="formatType" default="audio"/>
    <xs:attribute name="bitrate" type="xs:nonNegativeInteger"/>
    <xs:attribute name="extFormat" type="extFormatType" use="optional"/>
</xs:complexType>
```

This defines a service. The "simulcast" element may be used to provide the equivalent service reference on the same or another broadcast system. The "epgLanguage" element may be used to describe the language used by the service. The "bitrate" is only an indication of the bitrate of the service and the actual value may differ from the one here. For services that broadcast at a variety of bitrates, the highest bitrate that they broadcast at should be indicated here.

7.1.6 ensembleType

```
<!-- ##################################################################### -->
<!-- Declaration of type ensembleType -->
<!-- ##################################################################### -->
<xs:complexType name="ensembleType">
    <xs:sequence>
        <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded"/>
        <xs:element name="frequency" type="frequencyType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="CA" type="epg:CAType" minOccurs="0"/>
        <xs:element name="keywords" type="epg:keywordsType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="service" type="serviceType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="epg:ensembleIDType" use="required"/>
    <xs:attribute name="version" type="xs:integer"/>
</xs:complexType>
```

This is used to describe and locate a DAB ensemble or a DRM channel.

7.2 serviceInformation

```
<!-- ##################################################################### -->
<!-- Declaration of element serviceInformation -->
<!-- ##################################################################### -->
<xs:element name="serviceInformation">
    <xs:annotation>
        <xs:documentation xml:lang="en">Service information includes the structure of and
information about the multiplex and its associated services</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ensemble" type="ensembleType" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="creationTime" type="epg:timePointType"/>
        <xs:attribute name="originator" type="epg:originatorType"/>
        <xs:attribute name="serviceProvider" type="epg:serviceProviderType"/>
        <xs:attribute name="system" type="epg:systemType" default="DAB"/>
        <xs:attribute ref="xml:lang" use="required"/>
    </xs:complexType>
</xs:element>
```

The "serviceInformation" element is used as a container for ensemble elements. The "creationTime" attribute indicates the time at which this schedule was generated and the "originator" attribute is used to indicate the originator of the schedule. The "serviceProvider" attribute is used to indicate the service provider of the services contained in the EPG.

E.g.

```
<serviceInformation version="1"
    creationTime="2001-02-28T00:00:00"
    originator="BBC"
    serviceProvider="BBC"
    system="DAB">

    <ensemble id="e1.ce15" version="1">
        <frequency type="primary" kHz="225648"/>
        <service version="1"
            format="audio"
            bitrate="64">
            <serviceID id="e1.ce15.c238.0"
                type="primary"/>
            <simulcast system="DRM" id="e1c238"/>
            …
        </service>
        <service version="1"
            format="data"
            bitrate="8"
            extFormat="3.3C.007">
            <serviceID id="e1.ce15.e1c12345.0"
                type="primary"/>
            …
        </service>
    </ensemble>

</serviceInformation>
```

Annex A (normative): URL for postal addresses

This clause defines a URL scheme for defining postal addresses. The format is a URI-compliant and case insensitive string in the form:

```
addressuri  = "postal:" addressdata
addressdata = segment *("/" segment)
segment     = *urlchar
urlchar     = unreserved | escaped
```

Either the most generalized part OR the most localized part of the address should come first (depending on the postal scheme practices of the target country), separating each main fragment with a slash ("/"), through the hierarchy until the most localized/generalized resource is reached. `unreserved` and `escaped` are defined in RFC 2396 [3].

NOTE: Where "/" is needed as a character in the address (e.g. "20/22 High St") it should be encoded as the hex equivalent (i.e. "%2F").

E.g.

```
postal:BBC%20Research%20and%20Development/Kingswood%20Warren/Tadworth/Surrey/KT20%206NP/United%2
0Kingdom/
```

Annex B (normative): URL for DAB and DRM addressing

URLs for other content carried within a DAB or DRM channel shall be restricted to other objects in the same MOT carousel as the current EPG service. The URL shall consist of the value of the MOT ContentName parameter, excluding the initial 8 bits (i.e. the character data and not the character set indicator). Receivers shall ignore a URL if it does not match a filename within the EPG's MOT carousel.

NOTE: This section may, in the future, be replaced by a reference to a universal DAB and/or DRM URL specification if and when such a specification is released.

Annex C (informative): Filename conventions

When using file-based schedules the following filename conventions must be used (note that all filenames are case-insensitive).

C.1 Schedule files

C.1.1 DAB

One file per service per day named:

```
YYYYMMDD_<ECC>_<EId>_<SId>_<SCIdS>_PI.xml
```

Where YYYY represents the year, MM represents the month, DD the day and <ECC>,<EId>,<SId> and <SCIdS> uniquely identify the service using the form used in the serviceID XML element. This should contain a single <schedule> element and should contain <programme> elements, ordered by start time, for all programmes carried on this service that are billed to start at or between 00:00:00 and 23:59:59 on the date indicated in the filename.

NOTE: The scope element in a schedule indicates the time period covered by the schedule, from the billed start time of the first programme to the billed end time of the last programme.

E.g.

```
"20020407_e1_ce15_c221_0_PI.xml"
```

C.1.2 DRM

One file per service per day named:

```
YYYYMMDD_<SId>_PI.xml
```

Where YYYY represents the year, MM represents the month, DD the day and <SId> uniquely identify the service using the form used in the serviceID XML element. This should contain a single <schedule> element and should contain <programme> elements, ordered by start time, for all programmes carried on this service that are billed to start at or between 00:00:00 and 23:59:59 on the date indicated in the filename.

NOTE: The scope element in a schedule indicates the time period covered by the schedule, from the billed start time of the first programme to the billed end time of the last programme.

E.g.

```
"20020407_e1c238_PI.xml"
```

C.2 Service information files

One file per ensemble/channel named:

```
YYYYMMDD_xxxxxxxx_SI.xml
```

Where YYYY represents the year , MM represents the month, DD the day and xxxxxxxx is a string of up to 8 characters identifying the channel. There should be one file per ensemble/channel and this should contain a single <ensemble> element which contains ensemble information and <service> elements for all services in the ensemble/channel described by this EPG service. The information should be valid from the date indicated in the filename.

E.g.

```
"20020101_BBC_SI.xml"
```

C.3 Group information files

One file per ensemble/channel named:

```
YYYYMMDD_xxxxxxxx_GI.xml
```

Where YYYY represents the year, MM represents the month, DD the day and xxxxxxxx is a string of up to 8 characters identifying the channel. This should contain a single <groups> element.

E.g.

```
"20020407_BBC_GI.xml"
```

Annex D (normative): epgDataTypes_14.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.worlddab.org/schemas/epgDataTypes/14"
targetNamespace="http://www.worlddab.org/schemas/epgDataTypes/14" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of CAType -->
    <!-- ##################################################################### -->
    <xs:complexType name="CAType">
        <xs:attribute name="type" default="none">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="unspecified"/>
                    <xs:enumeration value="none"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of keywordsType -->
    <!-- ##################################################################### -->
    <xs:complexType name="keywordsType">
        <xs:simpleContent>
            <xs:restriction base="messageType"/>
        </xs:simpleContent>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of mediaDescriptionType -->
    <!-- ##################################################################### -->
    <xs:complexType name="mediaDescriptionType">
        <xs:choice>
            <xs:group ref="descriptionGroup"/>
            <xs:element name="multimedia">
                <xs:complexType>
                    <xs:attribute name="mimeValue" type="mimeType" use="optional"/>
                    <xs:attribute ref="xml:lang" default="en"/>
                    <xs:attribute name="url" type="urlType"/>
                    <xs:attribute name="type" use="optional">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="logo_unrestricted"/>
                                <xs:enumeration value="logo_mono_square"/>
                                <xs:enumeration value="logo_colour_square"/>
                                <xs:enumeration value="logo_mono_rectangle"/>
                                <xs:enumeration value="logo_colour_rectangle"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="width" type="xs:nonNegativeInteger"/>
                    <xs:attribute name="height" type="xs:nonNegativeInteger"/>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of locationType -->
    <!-- ##################################################################### -->
    <xs:complexType name="locationType">
        <xs:sequence>
            <xs:choice>
                <xs:element name="time" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="time" type="timePointType" use="required"/>
                        <xs:attribute name="duration" type="durationType" use="required"/>
                        <xs:attribute name="actualTime" type="timePointType"/>
                        <xs:attribute name="actualDuration" type="durationType"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="relativeTime" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="time" type="durationType" use="required"/>
                        <xs:attribute name="duration" type="durationType" use="required"/>
                        <xs:attribute name="actualTime" type="durationType"/>
                        <xs:attribute name="actualDuration" type="durationType"/>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
            <xs:element name="bearer" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="id" type="contentIDType" use="required"/>
                    <xs:attribute name="trigger" type="triggerType"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of memberOfType -->
    <!-- ##################################################################### -->
    <xs:complexType name="memberOfType">
        <xs:attribute name="shortId" type="shortCRIDType" use="required"/>
        <xs:attribute name="id" type="CRIDType" use="optional"/>
        <xs:attribute name="index" type="xs:positiveInteger" use="optional"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of element epgLanguageType-->
    <!-- ##################################################################### -->
    <xs:complexType name="epgLanguageType">
        <xs:attribute ref="xml:lang" default="en"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of linkType -->
    <!-- ##################################################################### -->
    <xs:complexType name="linkType">
        <xs:attribute name="url" type="urlType" use="required"/>
        <xs:attribute name="mimeValue" type="mimeType"/>
        <xs:attribute ref="xml:lang" default="en"/>
        <xs:attribute name="description">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="180"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="expiryTime" type="timePointType"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Definition of programmeType -->
    <!-- ##################################################################### -->
    <xs:complexType name="programmeType">
        <xs:sequence>
            <xs:group ref="scheduleNameGroup" maxOccurs="unbounded"/>
            <xs:element name="location" type="locationType" maxOccurs="unbounded"/>
            <xs:element name="mediaDescription" type="mediaDescriptionType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="genre" type="genreType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="CA" type="CAType" minOccurs="0"/>
            <xs:element name="keywords" type="keywordsType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="memberOf" type="memberOfType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="link" type="linkType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="programmeEvent" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:group ref="scheduleNameGroup" maxOccurs="unbounded"/>
                        <xs:element name="location" type="locationType" maxOccurs="unbounded"/>
                        <xs:element name="mediaDescription" type="mediaDescriptionType"
                        minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="genre" type="genreType" minOccurs="0"
                        maxOccurs="unbounded"/>
                        <xs:element name="CA" type="CAType" minOccurs="0"/>
                        <xs:element name="keywords" type="keywordsType" minOccurs="0"
                        maxOccurs="unbounded"/>
                        <xs:element name="memberOf" type="memberOfType" minOccurs="0"
                        maxOccurs="unbounded"/>
                        <xs:element name="link" type="linkType" minOccurs="0"
                        maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="shortId" type="shortCRIDType" use="required"/>
                    <xs:attribute name="id" type="CRIDType" use="optional"/>
                    <xs:attribute name="version" type="xs:integer"/>
                    <xs:attribute name="recommendation" type="recommendationType" default="no"/>
                    <xs:attribute name="broadcast" type="broadcastType" default="on-air"/>
                    <xs:attribute ref="xml:lang" use="optional" default="en"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="shortId" type="shortCRIDType" use="required"/>
        <xs:attribute name="id" type="CRIDType" use="optional"/>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="recommendation" type="recommendationType" default="no"/>
        <xs:attribute name="broadcast" type="broadcastType" default="on-air"/>
        <xs:attribute name="bitrate" type="xs:nonNegativeInteger"/>
        <xs:attribute ref="xml:lang" use="optional" default="en"/>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Definition of messageType -->
    <!-- ###################################################################### -->
    <xs:complexType name="messageType" abstract="true">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute ref="xml:lang" use="optional" default="en"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Definition of shortNameType -->
    <!-- ###################################################################### -->
    <xs:complexType name="shortNameType">
        <xs:simpleContent>
            <xs:restriction base="messageType">
                <xs:maxLength value="8"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Definition of mediumNameType -->
    <!-- ###################################################################### -->
    <xs:complexType name="mediumNameType">
        <xs:simpleContent>
            <xs:restriction base="messageType">
                <xs:maxLength value="16"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Definition of longNameType -->
    <!-- ###################################################################### -->
    <xs:complexType name="longNameType">
        <xs:simpleContent>
            <xs:restriction base="messageType">
                <xs:maxLength value="128"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Definition of shortDescriptionType -->
    <!-- ###################################################################### -->
    <xs:complexType name="shortDescriptionType">
        <xs:simpleContent>
            <xs:restriction base="messageType">
                <xs:maxLength value="180"/>
            </xs:restriction>
        </xs:simpleContent>
```

```
</xs:complexType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of longDescriptionType -->
<!-- ##################################################################### -->
<xs:complexType name="longDescriptionType">
    <xs:simpleContent>
        <xs:restriction base="messageType">
            <xs:maxLength value="1200"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of originatorType -->
<!-- ##################################################################### -->
<xs:simpleType name="originatorType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of serviceProviderType -->
<!-- ##################################################################### -->
<xs:simpleType name="serviceProviderType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of genreType -->
<!-- ##################################################################### -->
<xs:complexType name="genreType">
    <xs:sequence>
        <xs:element name="name" minOccurs="0">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="messageType">
                        <xs:attribute name="preferred" type="xs:boolean" use="optional"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="definition" type="messageType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="href" type="termReferenceType" use="required"/>
    <xs:attribute name="type" use="optional" default="main">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="main"/>
                <xs:enumeration value="secondary"/>
                <xs:enumeration value="other"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of termReferenceType -->
<!-- ##################################################################### -->
<xs:simpleType name="termReferenceType">
    <xs:union>
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:whiteSpace value="collapse"/>
                <xs:pattern value=":[^:]+:[^:]+"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
            <xs:restriction base="xs:anyURI"/>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>
<!-- -->
<!-- ########################################## -->
<!-- Definition of CRIDType -->
```

```
<!-- ########################################## -->
<xs:simpleType name="CRIDType">
    <xs:restriction base="xs:anyURI">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="(c|C)(r|R)(i|I)(d|D)://.*/.*"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ########################################## -->
<!-- Definition of shortCRIDType -->
<!-- ########################################## -->
<xs:simpleType name="shortCRIDType">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="16777215"/>
    </xs:restriction>
</xs:simpleType>
<!-- ########################################## -->
<!--  Definition of timePointType  -->
<!-- ########################################## -->
<!-- restrictions: no leading minus sign, no fractional seconds -->
<xs:simpleType name="timePointType">
    <xs:restriction base="xs:dateTime">
        <xs:pattern value="[^\-].+T[^\.]+"/>
    </xs:restriction>
</xs:simpleType>
<!-- ######################################### -->
<!--  Definition of durationType  -->
<!-- ######################################### -->
<!-- restrictions: no leading minus sign, no year/month/day, no fractional seconds -->
<!-- Note: maximum of 18 hours is not enforced -->
<xs:simpleType name="durationType">
    <xs:restriction base="xs:duration">
        <xs:pattern value="PT[^\.]+"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of contentIDType-->
<!-- ##################################################################### -->
<xs:simpleType name="contentIDType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="(([0-9a-fA-F]{2}\.[0-9a-fA-F]{4}\.)?[0-9a-fA-F]{4,8}\.[0-9a-fA-
         F]{1}(\.[0-9a-fA-F]{2})?)|([0-9a-fA-F]{6})"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of urlType-->
<!-- ##################################################################### -->
<xs:simpleType name="urlType">
    <xs:restriction base="xs:anyURI">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="((crid|CRID|tel|mailto|postal|http|dab|drm):(//|\+|SMS=)?)?([a-zA-Z0-
         9]|\.|@|%|\-|/|_|\+|\?|=|;){1,}"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of broadcastType-->
<!-- ##################################################################### -->
<xs:simpleType name="broadcastType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="on-air"/>
        <xs:enumeration value="off-air"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ##################################################################### -->
<!-- Definition of recommendationType-->
<!-- ##################################################################### -->
<xs:simpleType name="recommendationType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
    </xs:restriction>
</xs:simpleType>
```

```
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of systemType-->
<!-- ###################################################################### -->
<xs:simpleType name="systemType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="DAB"/>
        <xs:enumeration value="DRM"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of simulcastType -->
<!-- ###################################################################### -->
<xs:complexType name="simulcastType">
    <xs:attribute name="system" type="systemType" use="required"/>
    <xs:attribute name="id" type="contentIDType"/>
</xs:complexType>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of mimeType (Multipurpose Internet Mail Extension -->
<!-- ###################################################################### -->
<xs:simpleType name="mimeType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="([!-\.0-~]{1,}/[!-\.0-~]{1,})+"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of triggerType -->
<!-- ###################################################################### -->
<xs:simpleType name="triggerType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="[0-9a-fA-F]{8}"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of ensembleIDType -->
<!-- ###################################################################### -->
<xs:simpleType name="ensembleIDType">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse"/>
        <xs:pattern value="([0-9a-fA-F]{2}\.[0-9a-fA-F]{4})|([0-9a-fA-F]{6})"/>
    </xs:restriction>
</xs:simpleType>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of scheduleNameGroup -->
<!-- ###################################################################### -->
<xs:group name="scheduleNameGroup">
    <xs:sequence>
        <xs:element name="shortName" type="shortNameType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="mediumName" type="mediumNameType" maxOccurs="unbounded"/>
        <xs:element name="longName" type="longNameType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:group>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of serviceNameGroup -->
<!-- ###################################################################### -->
<xs:group name="serviceNameGroup">
    <xs:sequence>
        <xs:element name="shortName" type="shortNameType" maxOccurs="unbounded"/>
        <xs:element name="mediumName" type="mediumNameType" maxOccurs="unbounded"/>
        <xs:element name="longName" type="longNameType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:group>
<!-- -->
<!-- ###################################################################### -->
<!-- Definition of descriptionGroup -->
<!-- ###################################################################### -->
<xs:group name="descriptionGroup">
    <xs:sequence>
        <xs:element name="shortDescription" type="shortDescriptionType" minOccurs="0"
        maxOccurs="unbounded"/>
```

```
            <xs:element name="longDescription" type="longDescriptionType" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:group>
    <!-- -->
</xs:schema>
```

Annex E (normative): epgSchedule_14.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.worlddab.org/schemas/epgSchedule/14"
targetNamespace="http://www.worlddab.org/schemas/epgSchedule/14" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xs:import namespace="http://www.worlddab.org/schemas/epgDataTypes/14"
    schemaLocation="epgDataTypes_14.xsd"/>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of type programmeGroupTypeType -->
    <!-- ##################################################################### -->
    <xs:simpleType name="programmeGroupTypeType">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="series"/>
            <xs:enumeration value="show"/>
            <xs:enumeration value="programConcept"/>
            <xs:enumeration value="magazine"/>
            <xs:enumeration value="programCompilation"/>
            <xs:enumeration value="otherCollection"/>
            <xs:enumeration value="otherChoice"/>
            <xs:enumeration value="topic"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of type programmeGroupType -->
    <!-- ##################################################################### -->
    <xs:complexType name="programmeGroupType">
        <xs:sequence>
            <xs:group ref="epg:scheduleNameGroup" maxOccurs="unbounded"/>
            <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
            maxOccurs="unbounded"/>
            <xs:element name="genre" type="epg:genreType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="keywords" type="epg:keywordsType" minOccurs="0"
            maxOccurs="unbounded"/>
            <xs:element name="memberOf" type="epg:memberOfType" minOccurs="0"
            maxOccurs="unbounded"/>
            <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="shortId" type="epg:shortCRIDType" use="required"/>
        <xs:attribute name="id" type="epg:CRIDType" use="optional"/>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="type" type="programmeGroupTypeType"/>
        <xs:attribute name="numOfItems" type="xs:positiveInteger"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of type programmeGroupsType -->
    <!-- ##################################################################### -->
    <xs:complexType name="programmeGroupsType">
        <xs:sequence>
            <xs:element name="programmeGroup" type="programmeGroupType" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="creationTime" type="epg:timePointType"/>
        <xs:attribute name="originator" type="epg:originatorType"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of type scheduleType -->
    <!-- ##################################################################### -->
    <xs:complexType name="scheduleType">
        <xs:sequence>
            <xs:element name="scope" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="serviceScope" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="id" type="epg:contentIDType" use="required"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="startTime" type="epg:timePointType" use="required"/>
                    <xs:attribute name="stopTime" type="epg:timePointType" use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="programme" type="epg:programmeType" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="creationTime" type="epg:timePointType"/>
        <xs:attribute name="originator" type="epg:originatorType"/>
    </xs:complexType>
    <!-- -->
    <!-- #################################################################### -->
    <!-- Declaration of type alternateSourceType -->
    <!-- #################################################################### -->
    <xs:complexType name="alternateSourceType">
        <xs:attribute name="protocol" default="URL">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="DAB"/>
                    <xs:enumeration value="DRM"/>
                    <xs:enumeration value="URL"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="type" default="identical">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="more"/>
                    <xs:enumeration value="less"/>
                    <xs:enumeration value="similar"/>
                    <xs:enumeration value="identical"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="url" type="epg:urlType" use="required"/>
    </xs:complexType>
    <!-- -->
    <!-- #################################################################### -->
    <!-- Declaration of element epg -->
    <!-- #################################################################### -->
    <xs:element name="epg">
        <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="programmeGroups" type="programmeGroupsType" minOccurs="0"
                 maxOccurs="unbounded"/>
                <xs:element name="schedule" type="scheduleType" minOccurs="0"
                 maxOccurs="unbounded"/>
                <xs:element name="alternateSource" type="alternateSourceType" minOccurs="0"
                 maxOccurs="unbounded"/>
            </xs:choice>
            <xs:attribute ref="xml:lang" use="required"/>
            <xs:attribute name="system" type="epg:systemType" default="DAB"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Annex F (normative): epgSI_14.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.worlddab.org/schemas/epgSI/14"
targetNamespace="http://www.worlddab.org/schemas/epgSI/14" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xs:import namespace="http://www.worlddab.org/schemas/epgDataTypes/14"
    schemaLocation="epgDataTypes_14.xsd"/>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Declaration of type frequencyType -->
    <!-- ###################################################################### -->
    <xs:complexType name="frequencyType">
        <xs:attribute name="type" default="primary">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="primary"/>
                    <xs:enumeration value="alternative"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="kHz" type="xs:nonNegativeInteger"/>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Declaration of type formatType -->
    <!-- ###################################################################### -->
    <xs:simpleType name="formatType">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="audio"/>
            <xs:enumeration value="data"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Declaration of type extFormatType -->
    <!-- ###################################################################### -->
    <xs:simpleType name="extFormatType">
        <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse"/>
            <xs:pattern value="(([0-3]{1}\.[0-9a-fA-F]{2}\.[0-9a-fA-F]{3})|([0-7]{1}\.[0-9a-fA-
             F]{4}))((\.([0-9a-fA-F]{2})+)?)"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Declaration of type serviceIDType -->
    <!-- ###################################################################### -->
    <xs:complexType name="serviceIDType">
        <xs:attribute name="id" type="epg:contentIDType" use="required"/>
        <xs:attribute name="type" default="primary">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="primary"/>
                    <xs:enumeration value="secondary"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <!-- -->
    <!-- ###################################################################### -->
    <!-- Declaration of type serviceType -->
    <!-- ###################################################################### -->
    <xs:complexType name="serviceType">
        <xs:sequence>
            <xs:element name="serviceID" type="serviceIDType" maxOccurs="unbounded"/>
            <xs:element name="simulcast" type="epg:simulcastType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded"/>
            <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="genre" type="epg:genreType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="epgLanguage" type="epg:epgLanguageType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="CA" type="epg:CAType" minOccurs="0"/>
            <xs:element name="keywords" type="epg:keywordsType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:integer"/>
        <xs:attribute name="format" type="formatType" default="audio"/>
        <xs:attribute name="bitrate" type="xs:nonNegativeInteger"/>
        <xs:attribute name="extFormat" type="extFormatType" use="optional"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of type ensembleType -->
    <!-- ##################################################################### -->
    <xs:complexType name="ensembleType">
        <xs:sequence>
            <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded"/>
            <xs:element name="frequency" type="frequencyType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="mediaDescription" type="epg:mediaDescriptionType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="CA" type="epg:CAType" minOccurs="0"/>
            <xs:element name="keywords" type="epg:keywordsType" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="service" type="serviceType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="epg:ensembleIDType" use="required"/>
        <xs:attribute name="version" type="xs:integer"/>
    </xs:complexType>
    <!-- -->
    <!-- ##################################################################### -->
    <!-- Declaration of element serviceInformation -->
    <!-- ##################################################################### -->
    <xs:element name="serviceInformation">
        <xs:annotation>
            <xs:documentation xml:lang="en">Service information includes the structure of and
             information about the multiplex and its associated services</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ensemble" type="ensembleType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="version" type="xs:integer"/>
            <xs:attribute name="creationTime" type="epg:timePointType"/>
            <xs:attribute name="originator" type="epg:originatorType"/>
            <xs:attribute name="serviceProvider" type="epg:serviceProviderType"/>
            <xs:attribute name="system" type="epg:systemType" default="DAB"/>
            <xs:attribute ref="xml:lang" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Annex G (informative): Future extensions of the schema

Schema can be changed and not break well-written parsers as long as certain guidelines are followed:

1) Elements cannot be removed.

2) Attributes cannot be removed.

3) Attributes cannot be changed from "implied" to "required".

4) Default values should not be modified (generally).

5) A "value" cannot be removed from an attribute "value" list.

6) The required structure of a document cannot be changed. For example, ? cannot become + and a new element cannot be required to appear inside an existing element. Only ? and * can be used when changing the document structure.

Annex H (informative): Converting DAB and DRM PTy to TV-Anytime genres

This is a very simple mapping from DAB and DRM programme type (PTy) codes to TV-Anytime genres, as used in the present document. Note that there may be more than one TV-Anytime genre suggested for each Pty code.

PTy code	PTy name	TV-Anytime genre equivalents
0	Undefined	<none>
1	News	3.1.1 (Content.Non-fiction.News)
2	CurrentAffairs	3.1.1.16 (Content.Non-fiction.News.Current Affairs)
3	Information	1.2 (Intention.Information)
4	Sport	3.2 (Content.Sport)
5	Education	1.3 (Intention.Education) 3.1.3.6 (Content.Non-fiction.General Non-fiction.Education)
6	Drama	3.4 (Content.Fiction)
7	Culture	3.1.4 (Content.Non-fiction.Arts & Media)
8	Science	3.1.6 (Content.Non-fiction.Sciences)
9	Varied	3.1 (Content.Non-fiction)
10	PopMusic	3.6.4.1 (Content.Music and Dance.Pop-rock.Pop)
11	RockMusic	3.6.4 (Content.Music and Dance.Pop-rock)
12	EasyListening	3.6.3.2 (Content.Music and Dance.Background Music.Easy Listening)
13	LightClassical	3.6.1.5 (Content.Music and Dance.Classical.Light Classical)
14	SeriousClassical	3.6.1.2 (Content.Music and Dance.Classical.Classical)
15	OtherMusic	3.6 (Content.Music and Dance)
16	Weather	1.2.2 (Intention.Information.Pure Information) 3.1.1.13 (Content.Non-fiction.News/Weather forecasts)
17	Finance	1.2.2 (Intention.Information.Pure Information) 3.1.3.5 (Content.Non-fiction.General Non-fiction.Finance)
18	ChildrensProgrammes	4.2.1 (Intended Audience.Age Groups.Children)
19	SocialAffairs	3.1.3.2 (Content.Non-fiction.General Non-fiction.Social)
20	Religion	3.1.2.1 (Content.Non-fiction.Philospphies of Life.Religious Philosophies)
21	PhoneIn	2.1.8 (Format.Structured.Phone-in)
22	Travel	3.3.5(Content.Leisure/Hobby.Travel/Tourism)
23	Leisure	3.3 (Content.Leisure/Hobby)
24	JazzMusic	3.6.2 (Content.Music and Dance.Jazz)
25	CountryMusic	3.6.6 (Content.Music and Dance.Country and Western)
26	NationalMusic	3.6.9 (Content.Music and Dance.World/Traditonal/Ethnic/Folk Music)
27	OldiesMusic	3.6.3.5 (Content.Music and Dance.Background Music.Oldies)
28	FolkMusic	3.6.9 (Content.Music and Dance.World/Traditonal/Ethnic/Folk Music)
29	Documentary	3.1 (Content.Non-fiction) 2.1.4 (Format.Structured.Documentary)

History

Document history		
V1.1.1	December 2002	Publication
V1.2.1	January 2005	Publication
V1.3.1	February 2006	Publication
V1.4.1	June 2008	Publication