



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
17.10.2001 Bulletin 2001/42

(51) Int Cl.7: **H04H 1/00**

(21) Application number: **00107694.2**

(22) Date of filing: **10.04.2000**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **Sony International (Europe) GmbH**
10785 Berlin (DE)

(72) Inventors:
• **Schäfer, Ralf,**
c/o Home Network Comp. Europe R&D
70327 Stuttgart (DE)

• **Müller, Gil,**
c/o Home Network Company Europe R&D
70327 Stuttgart (DE)

(74) Representative: **MÜLLER & HOFFMANN**
Patentanwälte
Innere Wiener Strasse 17
81667 München (DE)

(54) **Method to transmit an information service in a broadcast transmission system**

(57) A method to transmit an information service in a broadcast transmission system, wherein said information service is logically fragmented into data fragments which build together with corresponding signalling information respective broadcast objects which can be transmitted independently from each other comprises the fol-

lowing steps: - mapping of an ID of a broadcast object which is included within said signalling information to a header according to a broadcast file transmission protocol; and - mapping of the data fragment included within said broadcast object to an object body according to said broadcast file transmission protocol which corresponds to said header.

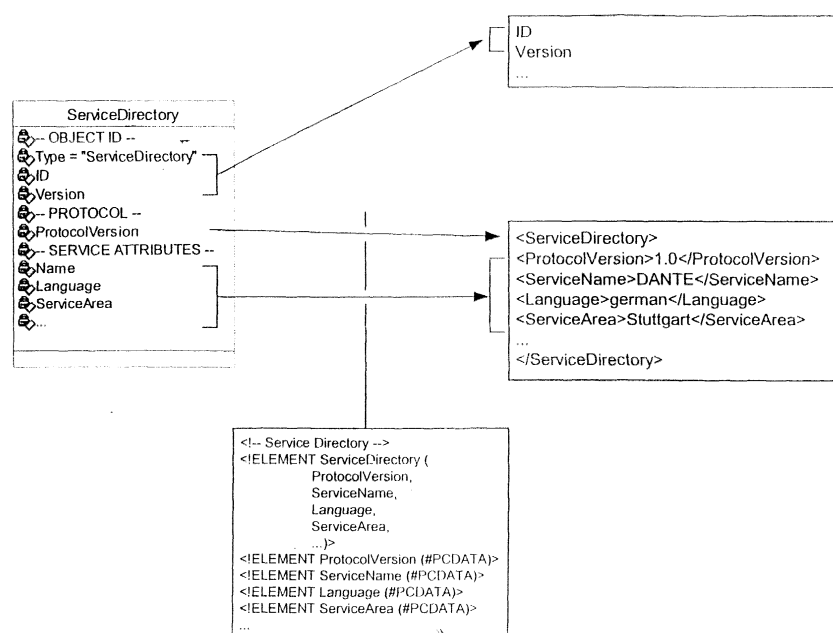


Figure 4: Mapping of Service Directory to XML encoded broadcast object

Description

[0001] The present invention relates to the field of information service provision. More particularly, the present invention relates to the information service provision for a broadcast medium.

[0002] The content of the European Patent Application "Method to transmit an information service in a broadcast transmission system" which is directed to transmission protocols and filed by the applicant of this application on the same filing day as this application is incorporated by reference into this application. This parallel application describes a broadcast transmission protocol for an information service. It defines a method, which divides the whole service data in chunks of data better suited for transmission over a broadcast medium. Additionally, the defined broadcast objects are amended by signalling information and protocol rules are defined in order to guarantee consistent re-assembly in the receiving terminal in a consistent manner.

[0003] XML (= Extensible Markup Language) is an upcoming standard for document exchange in the Internet, standardized by World Wide Web consortium (W3C). XML uses so-called DTDs (=Document Type Definition) to define the structure of documents in terms of elements, tags, attributes and entities. An XML Parser uses the DTD as input and checks compliance of incoming data against the DTD. Several tools are already available on the market for processing of XML data.

[0004] The DAB System is the upcoming standard for digital audio broadcasting in the mobile environment. A so-called DAB Ensemble allows to broadcast several audio channels and data channels simultaneously. For the broadcasting of data services a general broadcast file transfer protocol MOT is standardised. The MOT protocol basically provides a container, which takes any type of data and broadcast the data to any connected broadcast receiver. In order to identify broadcast objects among others an additional header is provided, which can be used to transmit additional information about the broadcast data.

[0005] The present invention is not limited to the DAB system. It basically requires the existence of a broadcast file transfer protocol comparable to the MOT protocol. Fig. 3 depicts the general structure of such a broadcast object container. It consists of a header and a body. The header provides at least two attributes ID and Version. The ID attribute is used to identify a broadcast object uniquely among all other broadcast objects in the same broadcast channel. The Version attribute is used to indicate updates of the broadcast data in the body part. The body carries the service data to be transmitted from the server side to the receivers. Header and body are not necessarily transmitted as one data chunk, but the broadcast file transfer protocol defines rules that guarantee that header and respective body can be put together on receiving side.

[0006] Two main approaches are known today. One approach is to transmit header and body as one data chunk, another is to have a kind of directory which informs about all following bodies until a new directory is sent. But also other forms are known, e.g. a full directory at the beginning of a broadcast cycle and repetitions in between which specify only partly the bodies of a broadcast cycle. For the present invention it is assumed that these details are hidden by the broadcast file transfer protocol and are therefore not relevant.

[0007] Therewith, it is an object underlying the present invention to provide a method to transmit an information service in a broadcast transmission system, in particular a method to transmit broadcast objects which can be transmitted independently from each other.

[0008] This object is solved by a method to transmit an information service in a broadcast transmission system according to claim 1. Preferred embodiments are defined in the dependent claims 2 to 7 and 11 and 12.

[0009] The method to transmit an information service in a broadcast transmission system, wherein said information service is logically fragmented into data fragments which build together with corresponding signalling information respective broadcast objects which can be transmitted independently from each other according to the present invention comprises the following steps:

- mapping of an ID of a broadcast object which is included within said signalling information to a header according to a broadcast file transmission protocol; and
- mapping of the data fragment included within said broadcast object to an object body according to said broadcast file transmission protocol which corresponds to said header.

[0010] Preferably, a type of a broadcast object which is included within said signalling information gets mapped to the header.

[0011] Further preferably, a type of a broadcast object which is included within said signalling information gets mapped to the object body.

[0012] Still further preferably, said mapping of the data fragment is performed according to a document type definition which corresponds to the type of the data fragment.

[0013] Still further preferably, said document type definitions are defined according to the XML standard.

[0014] Therewith, the present invention provides a transmission protocol for an information service, preferably using

XML, based on a broadcast file transfer protocol. Preferably, the abstract protocol mechanisms described in the parallel European Patent Application "Method to transmit an information service in a broadcast transmission system" which is directed to transmission protocols and filed by the applicant of this application on the same filing day as this application is combined with the upcoming standard for document exchange XML in the internet and the means provided by a general broadcast file transfer protocol.

[0015] Thereby the advantages of a broadcast medium, mainly unlimited scalability with respect to the number of users and portability provided by usage of XML are combined to provide a reliable transmission and adequate access times for an information service.

[0016] Further, it is an object underlying the present invention to provide a method to receive an information service in a broadcast transmission system and a receiver therefore.

[0017] This object is solved by a method to receive an information service in a broadcast transmission system according to claim 8. Preferred embodiments are defined in the dependent claim 9 which refers back to claims 2 to 7 and in claims 11 and 12. A receiver according to the present invention is defined in claim 10.

[0018] The method to receive an information service transmitted in a broadcast transmission system according to the present invention comprises the following step: parsing the header and corresponding object body by a parser which determines the type of received data, checks the validity and decodes the received data by the use of tags.

[0019] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and, together with a general description of the invention given above, and the detailed description of the embodiment given below, serve to explain the principles of the invention, wherein:

- Figure 1** shows the generic information service structure according to an embodiment of the present invention;
- Figure 2** depicts a system for information service provision according to the embodiment shown in Fig. 1;
- Figure 3** illustrates the structure of a broadcast object container;
- Figure 4** shows the mapping of a service directory according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 5** shows the mapping of a category directory according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 6** shows the mapping of an item directory according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 7** shows the mapping of an item dynamic data list according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 8** shows the mapping of an item main data list according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 9** shows the mapping of referenced attributes according to the embodiment shown in Fig. 1 to a broadcast object;
- Figure 10** shows the mapping of an item subset directory according to the embodiment shown in Fig. 1 to a XML encoded broadcast object;
- Figure 11** shows the mapping of an item subset according to the embodiment shown in Fig. 1 to a broadcast object; and
- Figure 12** shows the basic object structure in the DAB system.

[0020] In the following a preferred embodiment of the invention is described by use of the accompanying figures. However, the invention is not limited to this specific embodiment which is an advantageous realization and shows in particular the rules for the transmission protocol, i.e. mapping or coding of information to generate broadcast objects to be transmitted. Of course, the reception, i.e. demapping or decoding, needs to be performed according to rules corresponding to the rules for mapping to correctly rebuild the transmitted information service.

[0021] As mentioned before, the figures are mainly used to illustrate how broadcast objects specified as UML models are mapped to broadcast files mostly by use of XML (Figs. 4 - 11). UML (Unified Modelling Language) is a standard for the design of object-oriented systems. Every object defines an entity, which consists of a set of attributes. For better readability some comments are inserted. The comments are surrounded by "--" signs. Additional illustrations depict the assumed information service structure, information system structure and the structure of broadcast files.

[0022] Fig. 1 depicts an example of the generic structure of an information service to be broadcast using the method of the present invention. It consists basically of three types of service objects, which are Service, Category and Item. Every service object may have several attributes with several types and cardinalities. The relationship between Service, Category and Item is that the information service (Service) consists of one to many information categories (Category) and an information category has one to many items.

[0023] The parallel European Patent Application "Method to transmit an information service in a broadcast transmission system" which is directed to transmission protocols and filed by the applicant of this application on the same filing

day as this application provides a detailed description of how to broadcast this generic structure to a large number of terminals. Fig. 2 shows the system structure assumed for the present invention. It consists basically of several content providers 1, a broadcast server 2 and an arbitrary number of terminals 3. Content Providers 1 deliver content of different categories to the broadcast server, e.g. News, Traffic Information and POI, i.e. point of interest. It is beyond the scope of the present invention how this is done. The broadcast server 2 takes the data via an interface 2a and builds the information service comprising of service objects. The service objects are used to build broadcast objects. The broadcast objects are transmitted as a broadcast file sequence by use of a broadcast file transfer protocol.

[0024] A terminal 3 receives broadcast files and feeds the files to the processing of broadcast objects. Then broadcast objects are used to create service objects on the fly and to provide access to the data by any application program.

[0025] Preferably, the present invention combines XML and the broadcast transmission protocol according to the parallel European Patent Application "Method to transmit an information service in a broadcast transmission system" which is directed to transmission protocols and filed by the applicant of this application on the same filing day as this application to provide an information service over a broadcast medium on the basis of a broadcast file transfer protocol. With XML so-called DTDs (Document Type Definitions) are used to define the structure of a valid document for a certain application domain. In the following it is described how broadcast object information is mapped to broadcast files and XML documents.

MAPPING OF OBJECT ID

[0026] All broadcast objects to be sent in one broadcast channel must be identifiable in order to identify a certain object among others (see Fig. 3). The required Object ID consists of a type information (Type), an identifier (ID) and a version information (Version). The type information specifies the structure of the broadcast object in order to allow for appropriate processing of the object.

[0027] The assumed underlying broadcast file transfer protocol provides at least two header parameter, an ID and a Version. The ID identifies uniquely an object among others. The Version parameter indicates updates of the broadcast file. No equivalent header parameter for the type information of a broadcast object can be assumed. This can be solved by two different ways: First, the type information (Type) is encoded together with the identifier (ID) in the ID parameter of the broadcast file header. It depends on the context how this encoding could look like in detail, e.g. in case that the ID is kind of a filename a new string can be built which consists of the original filename and a substring which determines the type of the broadcast object. A second solution is to store the type information in the body of the broadcast file, e.g. as a first parameter. This requires to start processing of the body in order to determine the type of the broadcast object, but can be an appropriate solution when the first solution can not be applied. Whatever solution is chosen, it is used for the whole service. This means all broadcast objects are identified by the same means.

[0028] In the following description of how to map broadcast objects it is assumed that one of the above mentioned solutions is chosen and always applied in the same way. The mapping of Object IDs is therefore not discussed with each broadcast object.

MAPPING OF BROADCAST OBJECTS

[0029] Fig. 4 illustrates the steps to map the Service Directory broadcast object to a XML encoded broadcast file. To the left the Service Directory object together with its attributes is depicted.

[0030] The Service Directory object comprises besides the information service specific attributes Name, Language, ServiceArea and so on from the Service object, and it provides the following signalling information:

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as a Service Directory.
- Protocol: The ProtocolVersion attribute is used by the receiving terminal to check protocol compatibility between the broadcast service and the processing unit in the terminal.

[0031] The attributes of the OBJECT ID are mapped as described above. The figure shows only the first solution. The remaining attributes of the broadcast object are mapped according to the Service Directory DTD and encoded in the broadcast file body.

[0032] The Service Directory DTD has basically the following structure:

```

<!-- Service Directory -->
<!ELEMENT ServiceDirectory (
  ProtocolVersion,
  ServiceName,
  Language,
  ServiceArea,
  ...)>

```

```

<!ELEMENT ProtocolVersion (#PCDATA)>
<!ELEMENT ServiceName (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT ServiceArea (#PCDATA)>
...

```

[0033] The DTD specifies the valid structure for a Service Directory. All received data in a terminal is parsed by a XML Parser, which determines if received data is Service Directory data and checks validity. All information is encoded by the use of tags.

[0034] The first definition is `<!ELEMENT Service Directory (ProtocolVersion, ...)>`. This specifies that the Service Directory tag is a valid entity, which consists of the mentioned tags in brackets (ProtocolVersion and so on). The nested tags are attributes of the Service Directory broadcast object and are defined separately. The ProtocolVersion tag is defined by `<!ELEMENT ProtocolVersion (#PCDATA)>`. This specifies that ProtocolVersion is a valid entity, which provides data of type (#PCDATA). #PCData is an abbreviation for parsed character data means basically that it is text.

[0035] The actual content of the broadcast file body (Fig. 4 right side) carries the attributes of the Service Directory encoded in accordance to the DTD (Fig. 4 in the middle). Every attribute value is surrounded by a begin tag and an end tag, e.g. Protocol Version = 1.0 is encoded as `<ProtocolVersion>1.0</ProtocolVersion>`. The begin tag and the end tag differ only in the slash put first of the end tag.

[0036] Fig. 5 illustrates the steps to map the Category Directory broadcast object to a XML encoded broadcast file. To the left the Category Directory object together with its attributes is depicted.

[0037] The Category Directory object consists of the Object ID, the NoOfCategories attribute and the Category Data.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as a Category Directory.
- NoOfCategories: The NoOfCategories attribute indicates the number of categories the service consists of and how many Category Data attribute sets are delivered with the Category Directory.
- Category Data: Every category is described by the attributes of Category Data. Besides information service specific attributes Name, Icon and so on from the Category object, it provides a Category ID. The Category ID consists of an ID attribute, which uniquely identifies a category among other categories, and a Version attribute, which is used to indicate that a category is updated. Additionally the Category ID is used to link items together with their respective category.

[0038] The Category Directory consists of a Category Directory Header and an associated Category Data object. The attributes of the OBJECT ID are mapped as described above. The figure shows only the first solution. The remaining attributes of the broadcast object are mapped according to the Category Directory DTD and encoded in the broadcast file body.

[0039] The Category Directory DTD has basically the following structure:

```

5      <!-- Category Directory -->
      <!ELEMENT CategoryDirectory (NoOfCategories, CategoryData*)>
      <!ELEMENT NoOfCategories (#PCDATA)>

10     <!-- Category Data -->
      <!ELEMENT CategoryData (
          CategoryName,
          CategoryIconId,
15         ...
      )>
      <!ATTLIST CategoryData CategoryId CDATA #REQUIRED>
      <!ATTLIST CategoryData CategoryVersion CDATA #REQUIRED>
20     <!ELEMENT CategoryName (#PCDATA)>
      <!ELEMENT CategoryIconId (#PCDATA)>
      ...
25

```

[0040] The DTD specifies the valid structure for a Category Directory. All received data in a terminal is parsed by a XML Parser, which determines if received data is Category Directory data and checks validity. All information is encoded by the use of tags.

[0041] The first definition is `<!ELEMENT CategoryDirectory (NoOfCategories, Category*)>`. This specifies that the Category Directory tag is a valid entity, which consists of two further tags NoOfCategories and CategoryData. NoOfCategories is defined in the line below as #PCDATA. The star following CategoryData specifies that one CategoryDirectory entity consists of zero to many CategoryData tags. The CategoryData tag consists of two further tags CategoryName and CategoryIconId and two so-called XML attributes CategoryID and CategoryVersion. The two tags map to the exemplary category attributes Name and Icon. Thereby it is assumed that the icon data is not stored itself in the broadcast file body, but an ID (CategoryIconId) is used for referencing to the icon data. It is not important for the present invention how this is realised.

[0042] The CategoryID attributes ID and Version of CategoryData are mapped to XML attributes. This is defined by `<!ATTLIST CategoryData CategoryId CDATA #REQUIRED>`. An XML attribute is defined as an attribute belonging to a certain tag (here: CategoryData). It provides additional information for the processing and presentation of the tag. CDATA is a specification of the attribute type. It means that the attribute value is character data, which is the most general type for an attribute. #REQUIRED enforces that a value for the attribute is specified. The attributes ID and Version are used to distinguish between several categories and to indicate updates of categories. This means they are used mainly for processing of data and are not dedicated for direct use by the user.

[0043] The actual content of the broadcast file body carries the attributes of the Category Directory encoded in accordance to the DTD.

[0044] The mapping of the Item Directory, Item Dynamic Data List, Item Main Data List and Item Subset Directory follows the same principles as described above, but adapted to the specific structure of the respective broadcast object.

[0045] Fig. 6 depicts on the left hand side the structure of the Item Directory object, i.e. of the header and the data. It consists of the Object ID, the category linking information, the vertical fragmentation information, the NoOfItems attribute and the Item Core Data.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Item Directory.
- Category ID: The category linking information specifies the category to which the provided items belong.
- Vertical Fragmentation: Two attributes are provided which specify the number of subsets used to transmit the complete set of items of the respective category. The NoOfSubsetsMainData attribute indicates the number of subsets used for the Main Attributes group. This means that as many ItemMainDataList broadcast objects are

transmitted as indicated by NoOfSubsetsMainData. The NoOfSubsetsDynamicData attribute indicates the number of subsets used for the Dynamic Attributes group. This means that as many ItemDynamicDataList broadcast objects are transmitted as indicated by NoOfSubsetsDynamicData.

- NoOfItems: The NoOfItems attribute indicates the number of items the respective category consists of and how many attribute sets Item Core Data are delivered with the Item Directory.
- Item Core Data: Every item is described by the attributes of Item Core Data. Besides information service specific attributes like Name and so on from the Item object, it provides an Item ID. The Item ID consists of an ID attribute, which uniquely identifies an item among other items of the respective category, and three Version attributes, which are used to indicate that an item is updated. The CoreDataVersion attribute indicates changes of attributes in the Core Attribute group. All core attributes are delivered with the Item Directory. Additionally, the MainDataVersion and the DynamicDataVersion attributes are delivered. The MainDataVersion attribute indicates changes of attributes in the Main Attribute group. The DynamicDataVersion attribute indicates changes of attributes in the Dynamic Attribute group. All main attributes are delivered with ItemMainDataList objects and all dynamic attributes are delivered with ItemDynamicDataList objects.

[0046] The specification of the Item Directory DTD (Fig. 6 in the middle):

```

<!-- Item Directory -->
<!ELEMENT ItemDirectory (NoOfItems, Item*)>
<!ATTLIST ItemDirectory CategoryId CDATA #REQUIRED>
<!ELEMENT NoOfSubsetsMainData (#PCDATA)>
<!ELEMENT NoOfSubsetsDynamicData (#PCDATA)>
<!ELEMENT NoOfItems (#PCDATA)>

<!-- Item Core Data-->
<!ELEMENT ItemCoreData (
    ItemName
    ...
)>
<!ATTLIST ItemCoreData ItemId CDATA #REQUIRED>
<!ATTLIST ItemCoreData CoreDataVersion CDATA #REQUIRED>
<!ATTLIST ItemCoreData MainDataVersion CDATA #REQUIRED>
<!ATTLIST ItemCoreData DynamicDataVersion CDATA #REQUIRED>
<!ELEMENT ItemName (#PCDATA)>

```

[0047] Fig. 7 depicts the structure of the Item Dynamic Data List object i.e. of the header and the data. It consists of the Object ID, the category linking information, the vertical fragmentation information, the attribute NoOfItems and the Item Dynamic Data.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Item Dynamic Data List.
- Category ID: The category linking information specifies the category to which the provided items belong.
- Vertical Fragmentation: The SubsetNo attribute indicates the number of the subset of items provided with current Item Dynamic Data List object. The Item Directory object of the respective category contains the NoOfSubsetsDynamicData attribute, which indicates the total number of subsets.
- NoOfItems: The attribute NoOfItems indicates the number of items the current subset of the respective category consists of and how many attribute sets Item Dynamic Data are delivered with current Item Dynamic Data List

object.

- Item Dynamic Data: Every item is described by the attributes of Item Dynamic Data. Besides information service specific attributes like NoOfRoomsAvailable and so on from the Item object, it provides an Item ID. The Item ID consists of an ID attribute, which uniquely identifies an item among other items of the respective category, and a Version attribute. The DynamicDataVersion attribute indicates that attributes in the Dynamic Attributes group of an item are updated.

[0048] The specification of the Item Dynamic Data List DTD (Fig. 7 in the middle):

```

<!-- Item Dynamic Data List -->
<!ELEMENT ItemDynamicDataList (
    SubsetNo,
    NoOfItems,
    ItemDynamicData*
)>
<!ATTLIST ItemDynamicDataList CategoryID CDATA #REQUIRED>
<!ELEMENT SubsetNo (#PCDATA)>
<!ELEMENT NoOfItems (#PCDATA)>

<!-- Item Dynamic Data -->
<!ELEMENT ItemDynamicData (
    NoOfRoomsAvailable
    ...
)>
<!ATTLIST ItemDynamicData ItemID CDATA #REQUIRED>
<!ATTLIST ItemDynamicData DynamicDataVersion CDATA #RE-
QUIRED>
<!ELEMENT NoOfRoomsAvailable (#PCDATA)>
...
```

[0049] Fig. 8 depicts the structure of the Item Main Data List object, i.e. header and data. It consists of the Object ID, the category linking information, the vertical fragmentation information, the NoOfItems attribute and the Item Main Data.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Item Main Data List.
- Category ID: The category linking information specifies the category to which the provided items belong.
- Vertical Fragmentation: The SubsetNo attribute indicates the number of the subset of items provided with current Item Main Data List object. The Item Directory object of the respective category contains the NoOfSubsetsMainData attribute, which indicates the total number of subsets.
- NoOfItems: The attribute NoOfItems indicates the number of items the current subset of the respective category consists of and how many attribute sets Item Main Data are delivered with current Item Main Data List object.
- Item Main Data: Every item is described by the attributes of Item Main Data. Besides information service specific attributes like Address, NoOfRooms, and so on from the Item object, it provides an Item ID and Referenced Attribute Picture. The Item ID consists of an ID attribute, which uniquely identifies an item among other items of the respective category, and a Version attribute. The MainDataVersion attribute indicates that attributes in the Main Attributes

group of an item are updated. The Referenced Attribute Picture is supported by a reference to another broadcast object. The reference consists of two attributes PictureID and PictureVersion. The PictureID corresponds to the ID attribute of the broadcast object (ReferencedAttribute) carrying the attribute value (picture data). The PictureVersion attribute identifies the latest version of the picture and corresponds to the Version attribute of the broadcast object.

[0050] The specification of the Item Main Data List DTD (Fig. 8 in the middle):

```

<!-- Item Main Data List -->
<!ELEMENT ItemMainDataList (
    SubsetNo,
    NoOfItems,
    ItemMainData*
)>
<!ATTLIST ItemMainDataList CategoryID CDATA #REQUIRED>
<!ELEMENT SubsetNo (#PCDATA)>
<!ELEMENT NoOfItems (#PCDATA)>

<!-- Item Main Data -->
<!ELEMENT ItemMainData (
    Address,
    NoOfRooms,
    PictureID,
    PictureVersion,
    ...
)>
<!ATTLIST ItemMainData ItemID CDATA #REQUIRED>
<!ATTLIST ItemMainData MainDataVersion CDATA #REQUIRED>

<!ELEMENT Address (#PCDATA)>
<!ELEMENT NoOfRooms (#PCDATA)>
<!ELEMENT PictureID (#PCDATA)>
<!ELEMENT PictureVersion (#PCDATA)>
...
```

[0051] Fig. 10 depicts the structure of the Item Subset Directory object, i.e. header and data. It is the equivalent of the Item Directory object. It consists of the Object ID, the category linking information, the NoOfItems attribute, the vertical fragmentation information, and the Item Subset Data.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Item Subset Directory.
- Category ID: The category linking information specifies the category to which the provided items belong.

- NoOfItems: The NoOfItems attribute indicates the total number of items the respective category consists.
- Vertical Fragmentation: The NoOfSubsets attribute indicates the number of subsets used for delivery of the complete set of items and how many attribute sets Item Subset Data are delivered with current Item Subset Directory object. Additionally, this means that as many ItemSubset broadcast objects are transmitted as indicated by NoOfSubsets.
- Item Subset Data: Every subset is described by the attributes of Item Subset Data. It provides two attributes SubsetID and SubsetVersion. The SubsetID corresponds to the ID attribute of the broadcast object (Item Subset) carrying the subset data. The SubsetVersion attribute identifies the latest version of the subset data and corresponds to the Version attribute of the broadcast object.

[0052] The specification of the Item Subset Directory DTD (Fig. 10 in the middle):

```

<!-- Item Subset Directory -->
<!ELEMENT ItemSubsetDirectory (
    NoOfItems,
    NoOfSubsets,
    ItemSubsetData*
)>

<!ATTLIST ItemSubsetDirectory CategoryID CDATA #REQUIRED>
<!ELEMENT NoOfItems (#PCDATA)>
<!ELEMENT SubsetNo (#PCDATA)>

<!-- Item Subset Data -->
<!ELEMENT ItemSubsetData (
    SubsetID,
    SubsetVersion
)>
<!ELEMENT SubsetID (#PCDATA)>
<!ELEMENT SubsetVersion (#PCDATA)>

```

MAPPING OF REFERENCED ATTRIBUTE BROADCAST OBJECT

[0053] The mapping of the Referenced Attribute broadcast object works differently due to the nature of this broadcast object. A Referenced Attribute is always used when an attribute value requires too much space (bandwidth or storage), is updated more or less frequently or it has a predefined format, which does not fit to the format of the remaining attributes. An example is a picture, which is encoded in JPEG format while remaining attributes are encoded with XML.

[0054] Fig. 9 shows the mapping of a Referenced Attribute.

[0055] The left hand side shows the structure of the Referenced Attribute object. It consists of the Object ID and the referenced attribute.

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Referenced Attribute.
- Referenced Attribute: This is the referenced attribute itself, e.g. the picture data in case of a referenced picture.

[0056] The attributes of the OBJECT ID are mapped as described above. The figure shows only the first solution.

The referenced attribute is carried in the broadcast file body in its original, predefined format.

MAPPING OF ITEM SUBSET BROADCAST OBJECT

[0057] The mapping of the Item Subset broadcast object works differently due to the nature of this broadcast object.
[0058] The ItemSubset object shows the following structure. It consists of the Object ID, the category linking information, the vertical fragmentation information, the NoOfItems attribute and the Item Data:

- Object ID: The Object ID consists of the above mentioned attributes Type, ID and Version. The Type attribute identifies the broadcast object as an Item Subset.
- Category ID: The category linking information specifies the category to which the provided items belong.
- Vertical Fragmentation: The SubsetNo attribute indicates the number of the subset of items provided with current Item Subset object. The Item Subset Directory object of the respective category contains the NoOfSubsets attribute, which indicates the total number of subsets.
- NoOfItems: The attribute NoOfItems indicates the number of items the current subset of the respective category consists of and how many items are delivered with current Item Subset object.
- Item Data: Every item is provided in a predefined format, which might differ from the format used for the broadcast transmission protocol described in the present invention and which is not relevant for the protocol. The protocol provides only a container, which carries this kind of data. It is assumed that every item has an ID attribute, which uniquely identifies an item among other items of the same category and a Version attribute that indicates changes to an item. Additional information provided for an item is not relevant for the present invention.

[0059] An Item Subset is always used when an information category is part of the service that is provided in a predefined format that is different from the one used for the remaining service and that should not be adapted. In this case only a vertical fragmentation scheme is applied and a so-called Item Subset Directory determines which subsets (vertical fragments) exist. Each subset contains several items (ItemData) and every item is provided with its whole set of item attributes (no horizontal fragmentation). But the structure of the subsets is not specified by the transmission protocol. Instead it is assumed that each item has an ID and a Version information and an additional processing unit is responsible for accessing the Item attributes. Additionally the attributes CategoryID, SubsetNo and NoOfItems are also carried together with the item data in the broadcast file body, but their format is also not specified. It depends on the format of the item data, which is an appropriate format.

APPLICATION ON DAB/MOT

[0060] In the DAB system the MOT protocol is standardised for the transmission of broadcast files. Each data object is transmitted by use of a basic structure, which is illustrated in Fig. 12, i.e. a 7 bytes header followed by a header extension of variable length and an object body of variable length. The header contains elementary information for the handling in the receiver. The header extension contains additional information, which is necessary for the handling of certain types of data objects. The object body carries the object to be broadcast. The MOT protocol provides two parameters ContentName and VersionNumber, which are part of the header extension. The ContentName is a character field that contains a unique name or identifier for the object with a variable length. Different character sets are supported by a character set indicator field. The VersionNumber is an unsigned binary number starting at 0 and being incremented by 1 each time the version changes. A change of the VersionNumber indicates that the content of the object body has changed. The VersionNumber is encoded with 8 bit and enables to distinguish 256 states.

[0061] For the application of the present invention on DAB/MOT the ID attribute of each broadcast object is mapped to the ContentName parameter. For the type information of each broadcast object still both above described solutions are possible. With the first solution the Type attribute is encoded together with the ID attribute and mapped to the ContentName parameter. As an example assume that the ID is "HotelDirectory" and Type is "ItemDirectory", then a combined string can be built "HotelDirectory_ItemDirectory.xml". The underscore separates ID and Type. Additionally the MOT protocol works with file extensions. Here "xml" is an appropriate file extension. With the second solution only the ID attribute is mapped to the ContentName and the Type information is part of the object body, e.g. as a binary encoded parameter at the beginning of the object body. The Version attribute of each broadcast object is mapped to the VersionNumber parameter.

[0062] As a last step the two parameters ContentType and ContentSubtype, which are part of the header must be set. Both parameters together provide a 2-step classification of the broadcast file. For the XML encoded broadcast objects the 6-bit ContentType parameter is set to "text" (=000001). The ContentSubtype parameter is not already defined for XML, but the next remaining entry should be used for this. This is probably "XML" (=000000011). In case of other broadcast object types like Referenced Attributes or Item Subsets the setting depends on the specific content

and must be set in accordance to the protocol rules of the MOT protocol, e.g. image/BMP for bitmap images.

[0063] Although the present invention has been described by way of an information service to be broadcast consisting of three types of service objects, an information service to be broadcast according to the present invention may comprise more or less types of service objects which also might be build according to other rules. It is self evident that in such a

[0064] Further, it is described that XML and DTDs are used to define how broadcast object information is mapped to broadcast files and documents. However, other similar working concepts could also be used.

[0065] As already mentioned above, the present invention can of course be applied to other transmission protocols than DAB/MOT.

Claims

1. Method to transmit an information service in a broadcast transmission system, wherein said information service is logically fragmented into data fragments which build together with corresponding signalling information respective broadcast objects which can be transmitted independently from each other, **characterized by** the following steps:

- mapping of an ID of a broadcast object which is included within said signalling information to a header according to a broadcast file transmission protocol; and
- mapping of the data fragment included within said broadcast object to an object body according to said broadcast file transmission protocol which corresponds to said header.

2. Method according to claim 1, **characterized by** mapping of a type of a broadcast object which is included within said signalling information to the header.

3. Method according to claim 1, **characterized by** mapping of a type of a broadcast object which is included within said signalling information to the object body.

4. Method according to anyone of the preceding claims, **characterized in that** said mapping of the data fragment is performed according to a document type definition which corresponds to the type of the data fragment.

5. Method according to claim 4, **characterized in that** said document type definitions are defined according to the XML standard.

6. Method according to anyone of the preceding claims, **characterized in that** for said mapping of the data fragment said object body carries said data fragment in its original predefined format.

7. Method according to anyone of the preceding claims, **characterized by** mapping of an information to defragment said information service which is included within said signalling information to the object body.

8. Method to receive an information service transmitted in a broadcast transmission system according to anyone of the preceding claims, **characterized by** parsing the header and corresponding object body by a parser which determines the type of received data, checks the validity and decodes the received data by the use of tags.

9. Method according to claim 8, **characterized in that** said decoding is performed corresponding to the coding defined in anyone of claims 2 to 7.

10. Receiver to perform the method steps according to claim 8 or 9.

11. Method according to anyone of the preceding claims 1 to 9, **characterized in that** said broadcast transmission system is DAB.

12. Method according to anyone of the preceding claims 1 to 9 and 11, **characterized in that** said broadcast file transmission protocol is MOT.

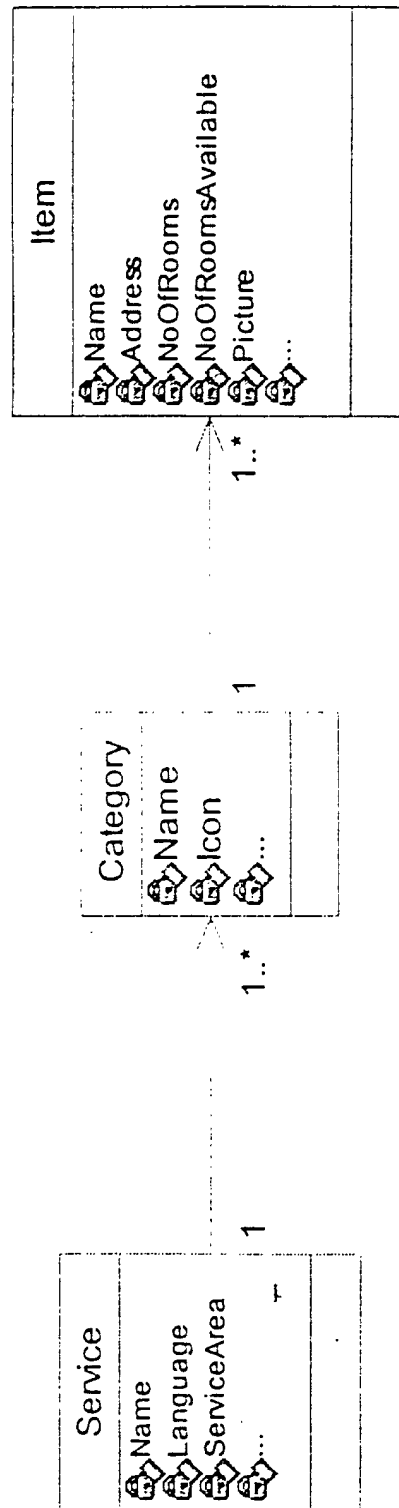


Figure 1: Generic information service structure

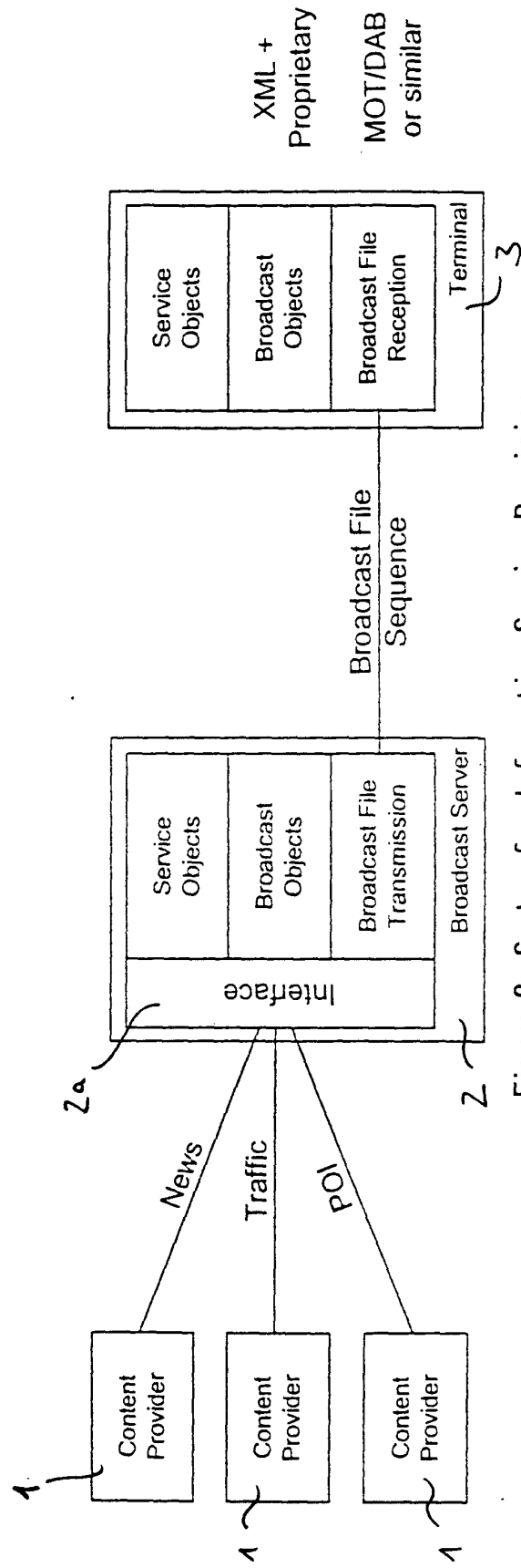


Figure 2: System for Information Service Provision

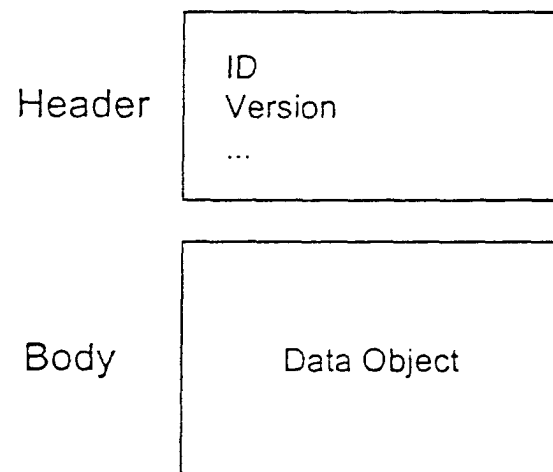


Figure 3: Structure of broadcast object container

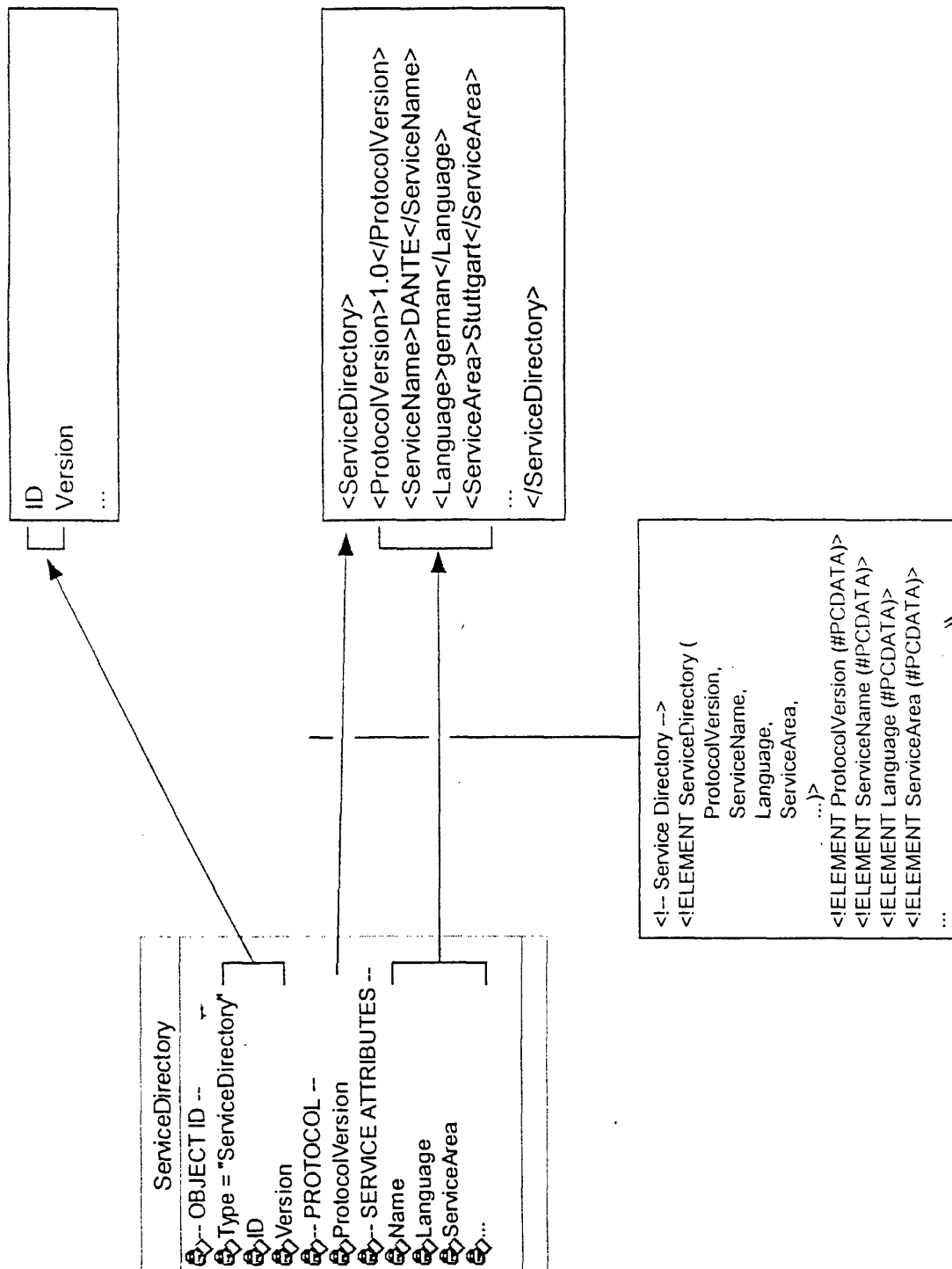


Figure 4: Mapping of Service Directory to XML encoded broadcast object

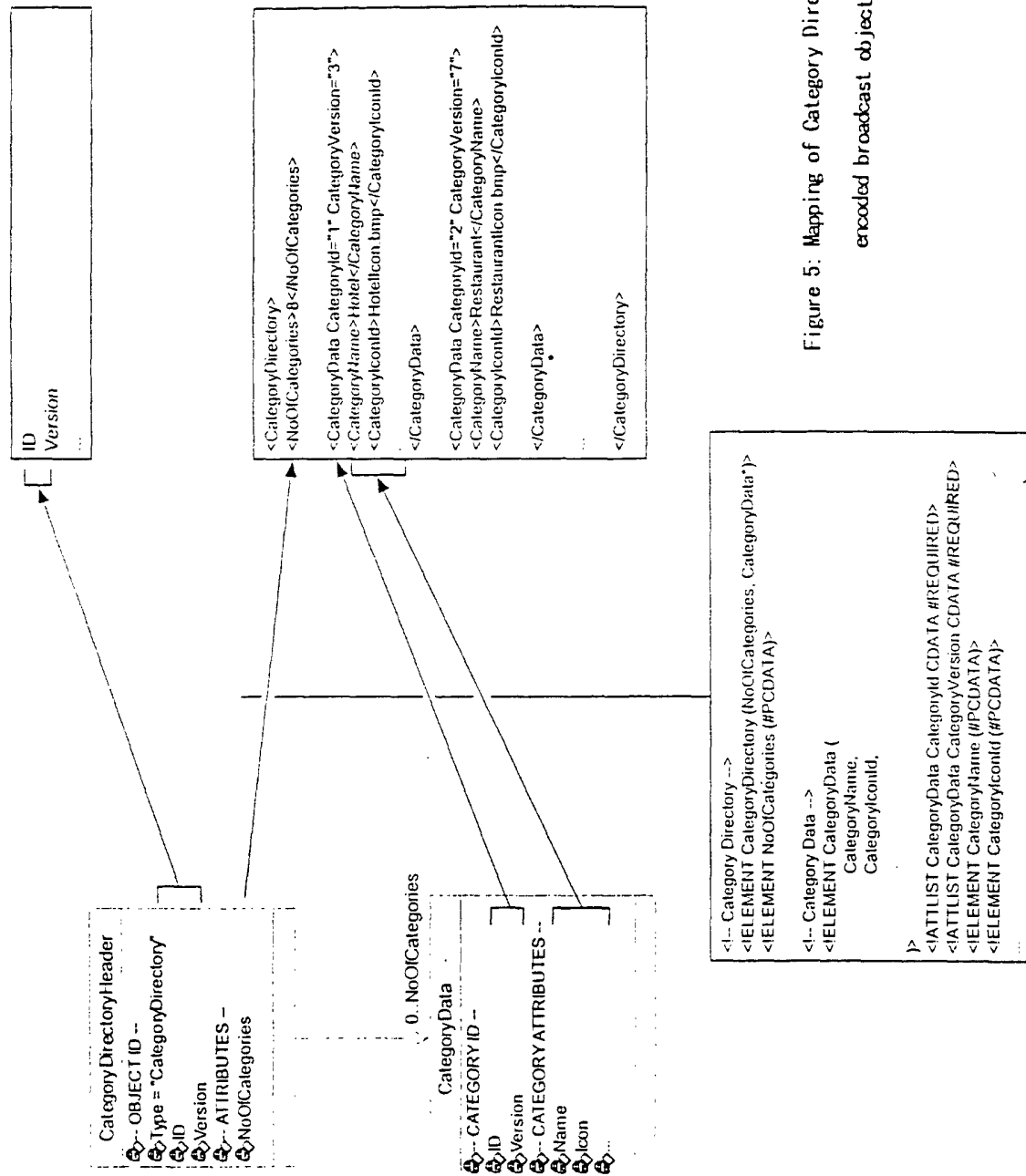


Figure 5: Mapping of Category Directory to XML encoded broadcast object

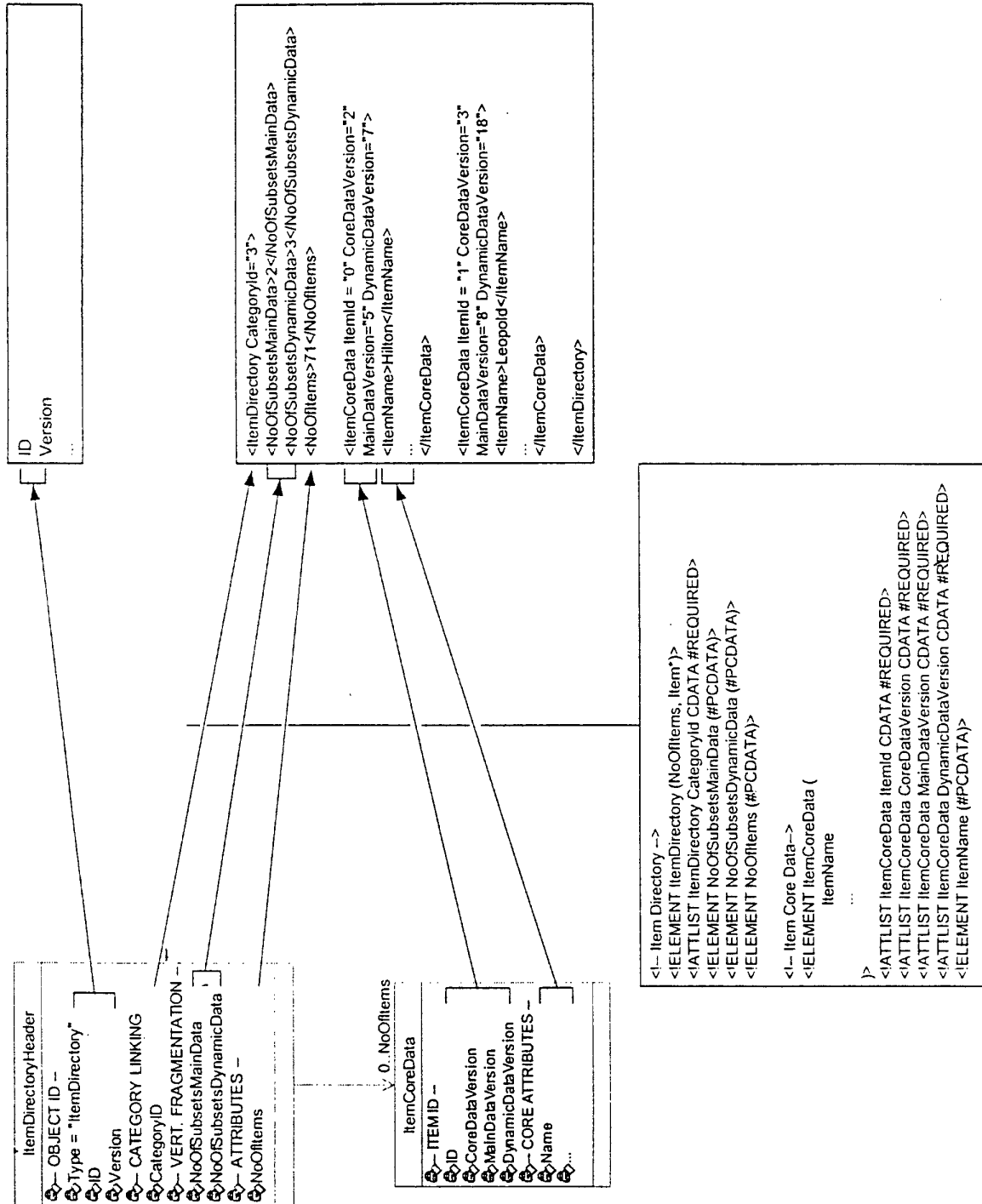


Figure 6: Mapping of Item Directory to XML encoded broadcast object

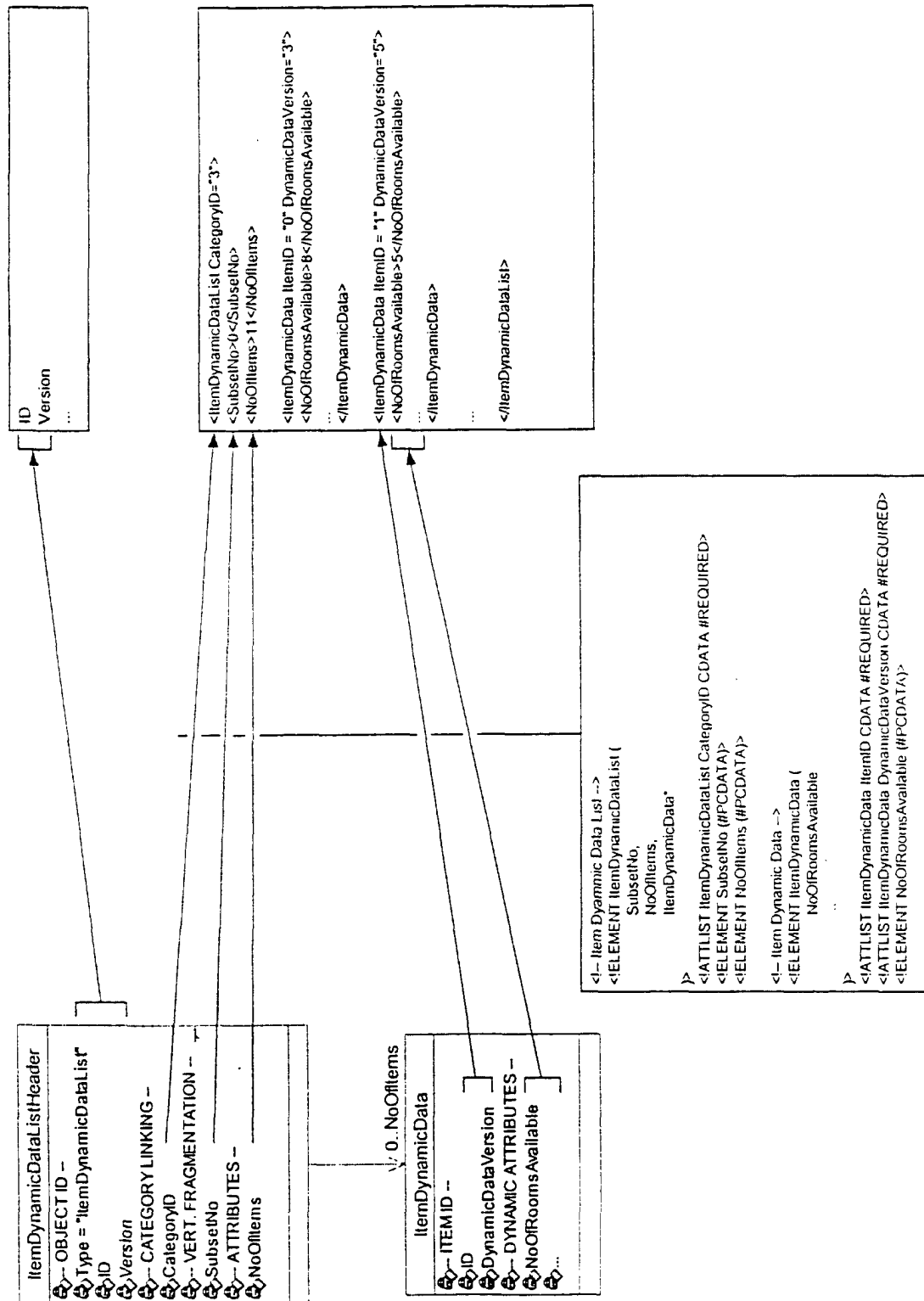


Figure 7: Mapping of Item Dynamic Data List to XML encoded broadcast object

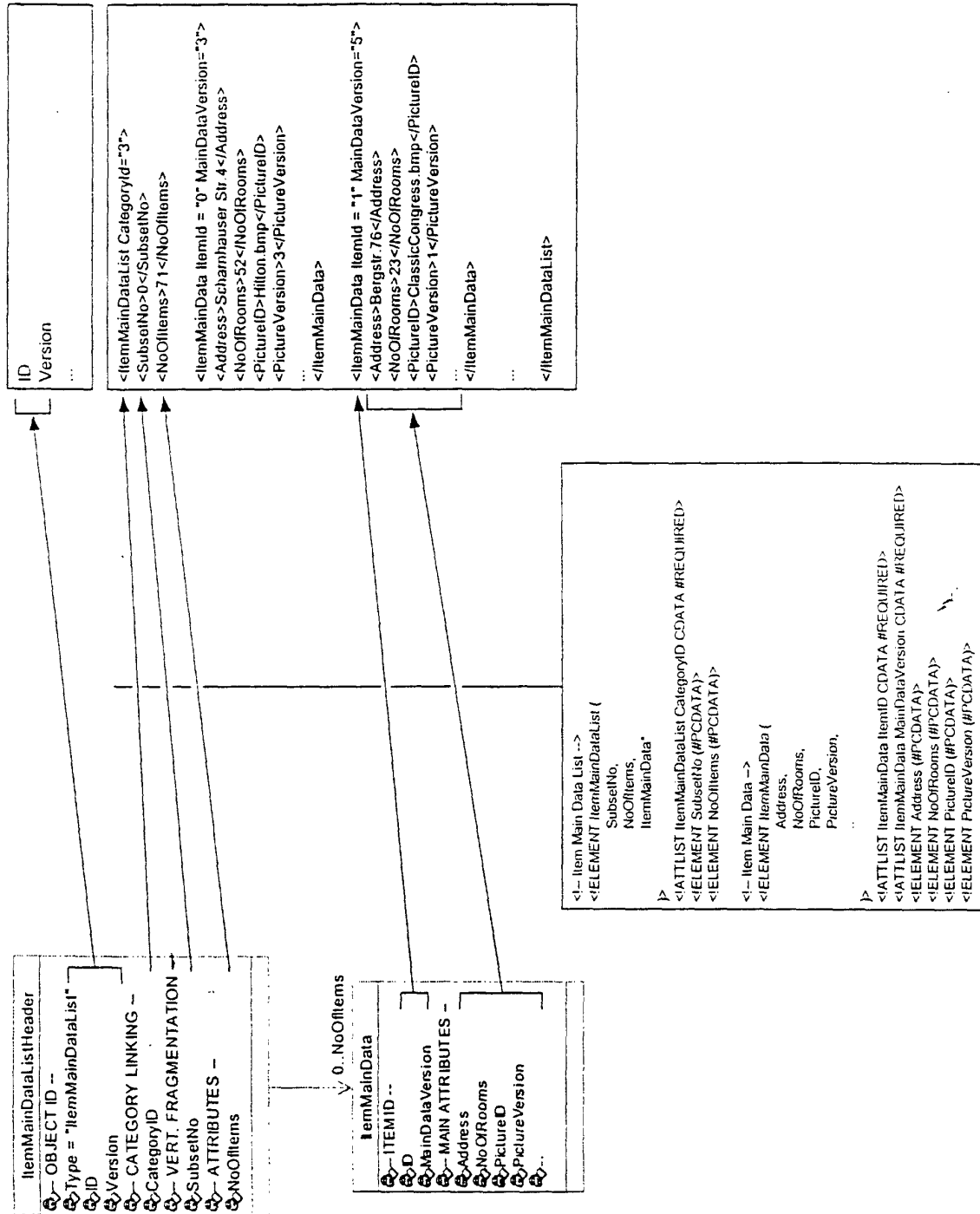


Figure 8: Mapping of Item Main Data List to XML encoded broadcast object

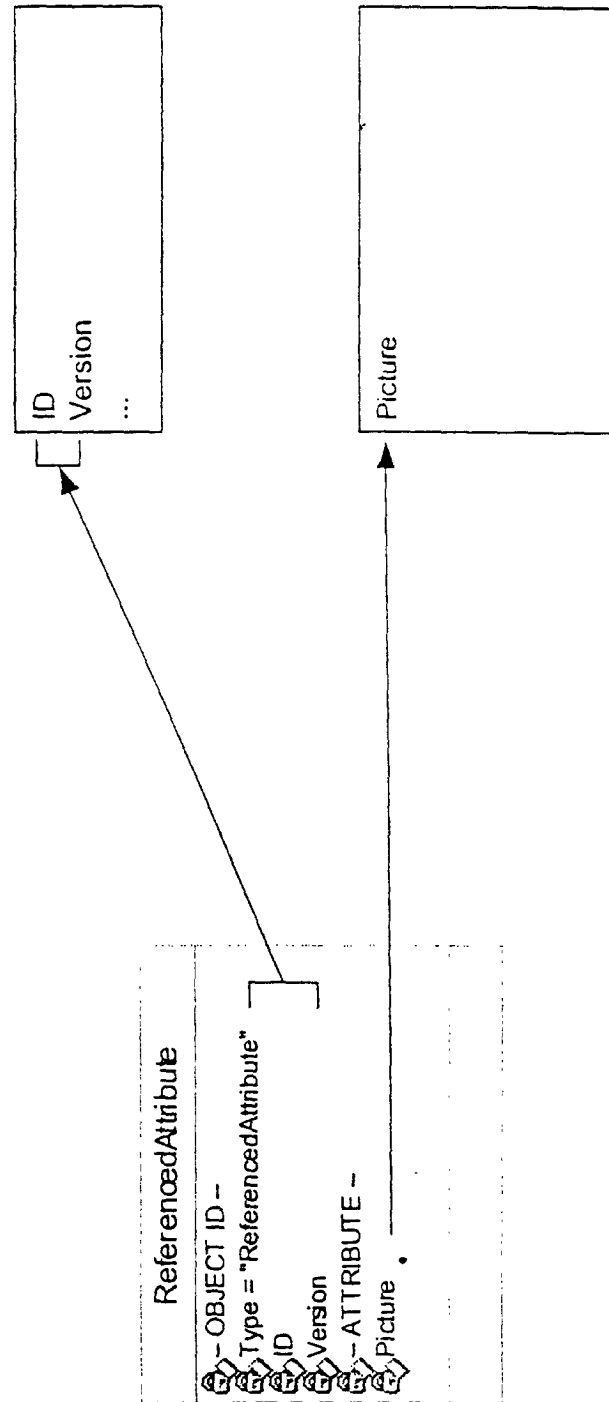


Figure 9: Mapping of Referenced Attribute to broadcast object

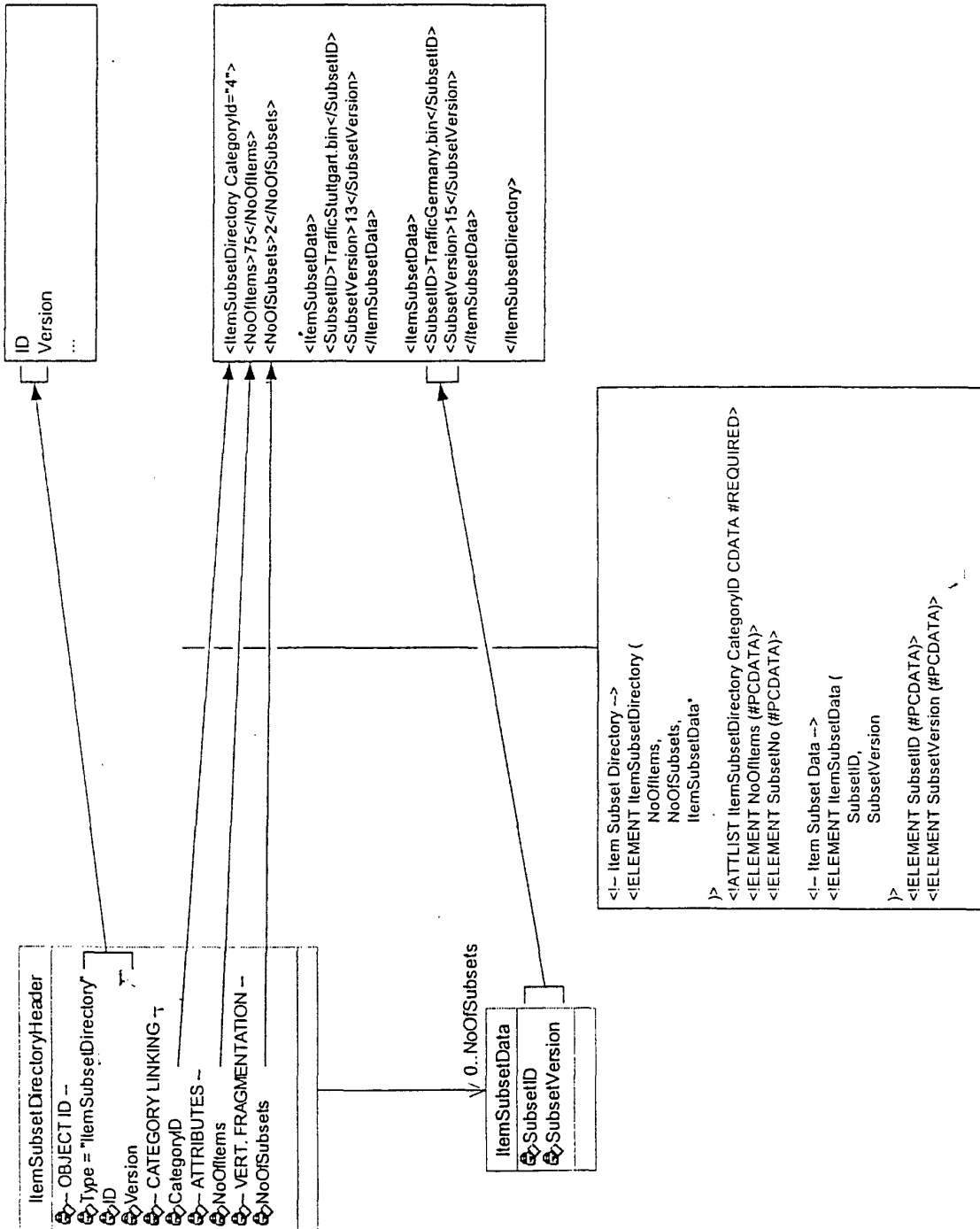


Figure 10: Mapping of Item Subset Directory to XML encoded broadcast object

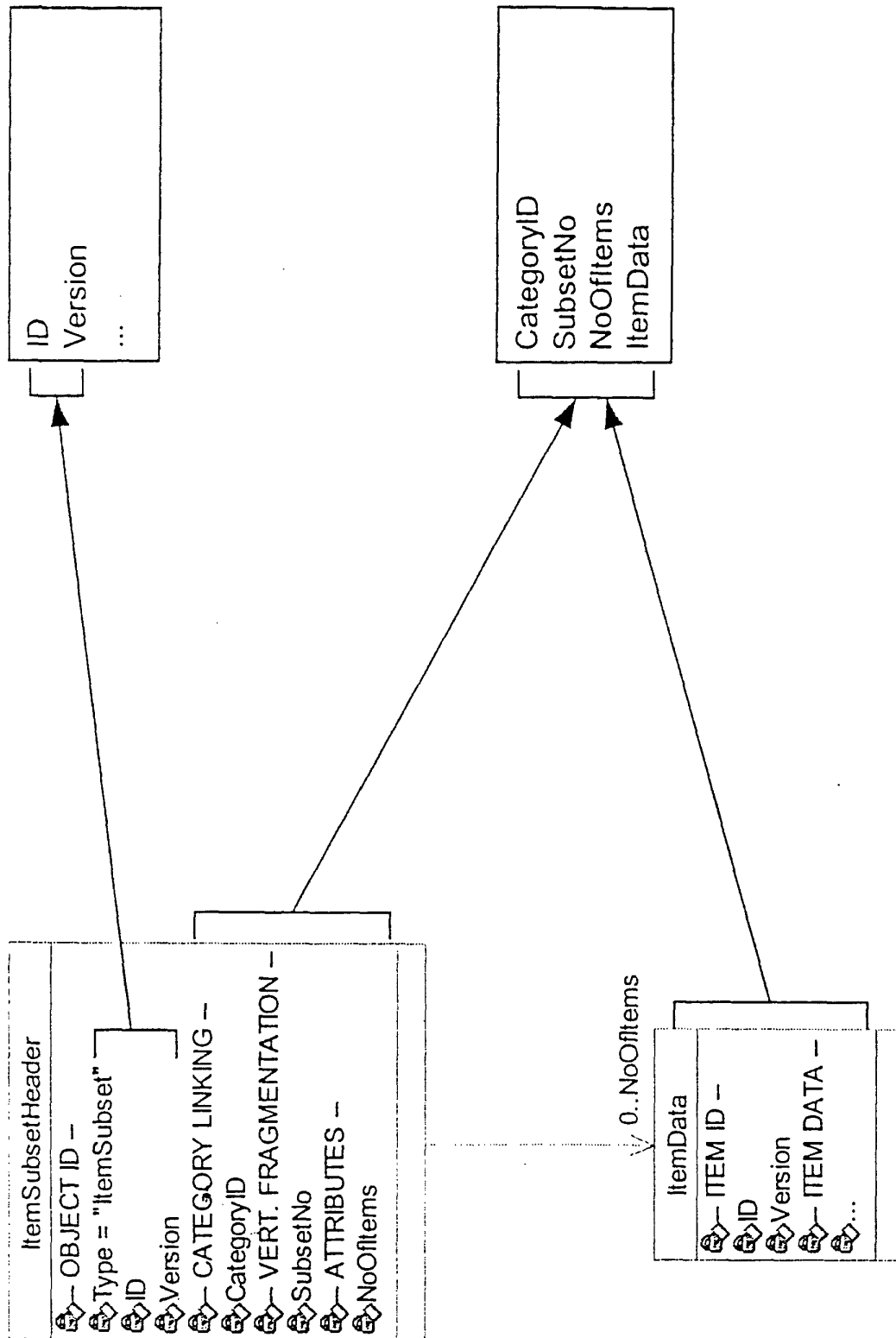


Figure 11: Mapping of Item Subset to broadcast object

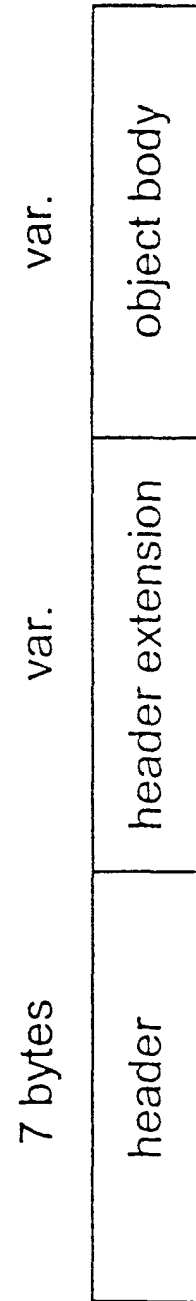


Figure 12: Basic object structure in the DAB system



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 10 7694

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	ETSI: "Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol" EN 301 234 - V1.2.1, [Online] February 1999 (1999-02), pages 1-31, XP002146079 Retrieved from the Internet: <URL:http://www.etsi.org> [retrieved on 2000-08-23] * paragraph [03.1] * * paragraph [04.2] * * paragraph [0005] *	1,2,11, 12	H04H1/00
X	--- EUREKA 147 MOT TASK GROUP: "Digital Audio Broadcasting System - Rules of Operation for the Multimedia Object Transfer Protocol (RO MOT)" DRAFT ETSI TR 101 497 - V1.1.1, [Online] March 2000 (2000-03), pages 1-47, XP002146080 Retrieved from the Internet: <URL:http://www.eurekadab.org/etsi/tr101497web.pdf> [retrieved on 2000-08-24] * paragraph [0006] * * paragraph [0007] *	1,2,11, 12	
Y	---	4-6,8-10	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04N H04H G06F
Y	LUBELL J: "STRUCTURED MARKUP ON THE WEB" MARKUP LANGUAGES,US,MIT PRESS, CAMBRIDGE, MA, vol. 1, no. 3, 1999, pages 7-22, XP000863188 ISSN: 1099-6621 * page 18 - page 20; figure 5 * -----	4-6,8-10	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 15 September 2000	Examiner Beaudet, J
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

... 03 82 (P04C01)