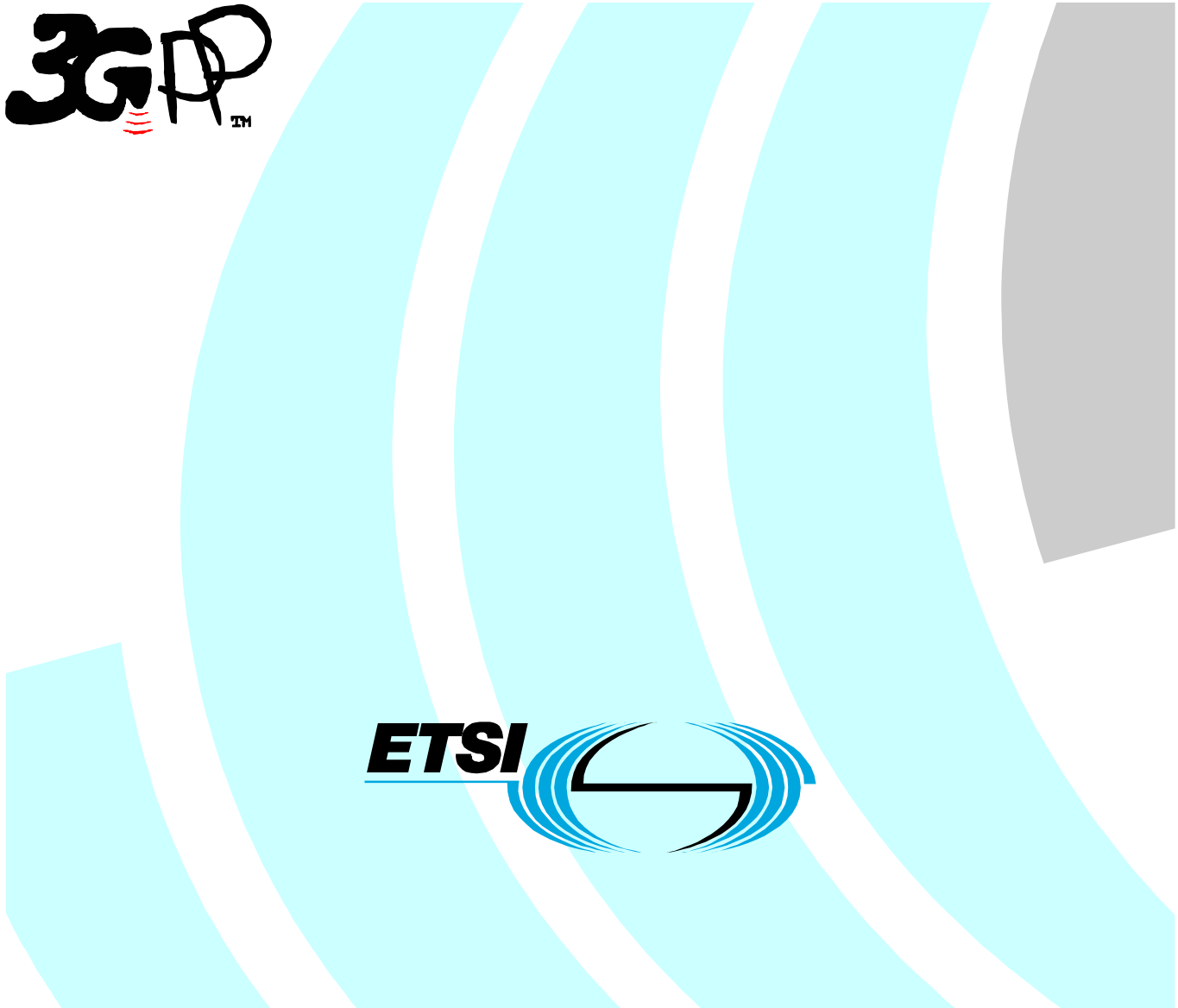
ETSI TR 123 941 V6.0.0 (2004-12)

Technical Report

Universal Mobile Telecommunications System (UMTS);
3GPP Generic User Profile (GUP);
Stage 2;
Data Description Method (DDM)
(3GPP TR 23.941 version 6.0.0 Release 6)



Reference
DTR/TSGN-0423941v600

Keywords
UMTS

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

Intellectual Property Rights

Foreword

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp .

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document introduces a Data Description Method (DDM) and Datatype Definition Method (DtDM) to be used for Generic User Profile (GUP) data.

The DDM is a set of common rules on how to specify the data components. These rules meet the requirements specified in 3GPP TS 22.240 [10] Stage 1 Generic User Profile. The DDM defines a method for describing 3GPP GUP data in both the 3GPP network and the User Equipment.

The Datatype Definition Method is a set of rules for defining new datatypes. The DtDM forms part of the Data Description Method. The DtDM can also be applied to datatypes outside of GUP. The built-in datatypes and rules are defined in [5] the W3C XML Schema Part 2: Datatypes specification.

The present document for a DDM and DtDM will capture features that will allow:

1) A method to describe the data and structure in a User Profile in a consistent manner.
2) Efficient usage and/or replication of data.
3) Coexistence with existing data description methods such as OMA UAProf and SyncML Device Management.
4) Effective support for management and maintenance of data.
5) Extensibility for future needs and the easy addition of new features.

1 Scope

The present document is the Stage Two description for the Data Description Method (DDM) and the Datatype Definition Method (DtDM) of the 3GPP Generic User Profile.

The present document specifies the method for describing the logical structure of the data description, Generic User Profile, Profile Components, methods for describing Datatypes and other constructs for use in 3GPP. The GUP data description method is designed to coexist with other data description technologies.

The document includes:

- DDM and DtDM features and definition;
- Process and procedures for administration of GUP/DDM and 3GPP TS 24.241 [12].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] W3C Recommendation: "Extensible Mark-up Language (XML) 1.0 (Second Edition)". http://www.w3.org/TR/REC-xml/

[2] W3C Recommendation: "Namespaces in XML", January 1999. http://www.w3.org/TR/REC-xml-names/

[3] W3C Recommendation: "XML Schema Part 0: Primer", May 2001. http://www.w3.org/TR/xmlschema-0/

[4] W3C Recommendation: "XML Schema Part 1: Structures", May 2001. http://www.w3.org/TR/xmlschema-1/

[5] W3C Recommendation: "XML Schema Part 2: Datatypes", May 2001. http://www.w3.org/TR/xmlschema-2/

[6] W3C Recommendation: "XML Path Language (XPath) Version 1.0", 16 November 1999. http://www.w3.org/TR/xpath

[7] W3C Candidate Recommendation: "XML Pointer Language (XPointer) Version 1.0", 16 August 2002. http://www.w3.org/TR/xptr/

[8] W3C Recommendation: "XSL Transformations (XSLT) Version 1.0", 16 November 1999. http://www.w3.org/TR/xslt

[9] 3GPP TS 23.078 (V5.5.1): "Customized Applications for Mobile network Enhanced Logic (CAMEL); Stage 2 (Release 5)".

[10] 3GPP TS 22.240: "Service requirements for 3GPP Generic User Profile (GUP); Stage 1".

[11] 3GPP TS 23.240: "3GPP Generic User Profile (GUP) requirements; Architecture; Stage 2".

[12] 3GPP TS 24.241: "3GPP Generic User Profile (GUP) Common Objects; Stage 3".

[13] 3GPP TS 21.101: "Technical Specifications and Technical Reports for a UTRAN-based 3GPP system".

[14] 3GPP TS 29.240: "3GPP Generic User Profile (GUP) Stage 3, Network".

[15] IETF (Internet Engineering Task Force) RFC2396: "Uniform Resource Identifiers (URI): General Syntax".

3 Definitions and abbreviations

3.1 Definitions

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

The terms "GUP" and "Profile" are synonymous within the present document.

3GPP Generic User Profile (GUP): collection of user related data which affects the way in which an individual user experiences services and which may be accessed in a standardized manner as described in this specification. The Generic User Profile is defined using the W3C XML recommendation [1].

Profile Instance: physical representation of a Profile, and is a collection of Profile Component Instances. For every user there is exactly one Profile Instance which is regarded as the Master. Additional copies containing the same data are allowed.

Profile Component (logical): A Profile Component is logically a part of the Generic User Profile. A Profile Component is instantiated and thus has an associated Profile Component Instance.

Profile Component Instance (physical): physical representation of a Profile Component. To one Profile Component (logical) corresponds one Profile Component Instance which is regarded as the Master and one or more component instance copies, i.e. physical copies. Component instances may be located in the Home Network, in the Value Added Service Provider Environment and/or the User Equipment.

Profile Data Element: indivisible unit of Generic User Profile information.

GUP Information Model: method describing how to define data structure, the way data elements are defined and the relationship to each other. The Information Model is describing the concept of Generic User Profile.

Data Description Method (DDM): method describing how to define the data contained in the Generic User Profile. The description is defined using the W3 XML Schemas recommendations [5] and [6].

Master Instance: among the instances (physical) associated with a Profile or Profile Component (logical), one of them is tagged with the role of "master instance". The master instance is responsible for the correct value of the corresponding Profile component.

Identity: permanent Identifier used to identify one Instance of a data entity in 3GPP TR 23.941. An Identity exists at the level of a Profile and a Profile Component. An Identity includes a representation of the User Identity within its structure.

User Identity: is the means to uniquely identify a User (such as an IMSI).

Datatype Definition Method (DtDM): method describing how to define the new datatypes contained in the Generic User Profile, including an initial set of built-in datatypes.

Data Payload: is the useful data in Profile and Profile Components. It consists of a number of Attributes carrying the data values.

3.2 Abbreviations

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

DDM	Data Description Method
DE	Data Element
DtDM	Datatype Definition Method
GUP	Generic User Profile
PC	Profile Component

4 General Description

As the data contained in the 3GPP Generic User Profile is going to be handled by different applications and entities for different purposes, there is a risk that various description methods might lead to duplications and/or inconsistencies. Therefore, a standard description method to describe the data should be used in the 3GPP mobile systems specifications, i.e. the Data Description Method (DDM) and Datatype Definition Method (DtDM).

Clause 5 defines the concepts used to model the GUP data and their relationships. It also defines characteristics of all Profile Component types and is used in the design and implementation of generic functions for using, maintaining and managing GUP data. Properties specific to a Profile Component type are described using the DDM and DtDM.

Clause 6 describes the DDM, which defines how to describe GUP data in a standardized manner. The description of the users' services configuration and personalization data using the DDM results in manipulating and accessing these data in a structured and standardized way. The DDM will help to overcome some of the challenges associated with the introduction of sophisticated user terminals and services with widely varying capabilities, hybrid combinations of mobile network domains, the advent of downloadable applications, and the desire of users to customize potentially complex services to individual preferences and needs.

The Data Description Method can be viewed as templates for constructing the data description. The templates (set of rules) enable the standardization of the data description such that they and the described data can be shared (used) by many applications. The data descriptions are abstract in the sense that the data are described independently of data formats specific to transport protocols or application technologies. Using standardized and abstract data descriptions simplifies the mapping between different data formats, and facilitates future extensions.

Clause 7 describes the Datatype Definition Method, DtDM. It is a method describing how to define the new datatypes contained in the Generic User Profile. The DDM and the DtDM data description defines:

1) the syntax;

2) the semantics; and

3) one XML-representation of the data.

The XML-schema, a W3C specification, is used to define the XML-representation. To describe the semantics of the data normal text and XML Schema code is used. Rules in the DDM describe how the XML-representation is interpreted to get the syntax. The syntax description can manually be mapped and or automatically translated to other syntax and format descriptions. Using XML-schema makes mapping and translation to other formats easier and facilitates concurrent support of different data formats.

Clause 8 describes the rules and procedures which are required to administer GUP and DDM in 3GPP. This clause also includes rules and procedures for the administration of 3GPP TS 24.241 [12] and Common Objects.

It is not intended to substitute data description methods that already exist. The 3GPP DDM and DtDM should be used, where appropriate, when Generic User Profiles are specified for new applications. See figure 4.1 for an explanation of the mapping principles.

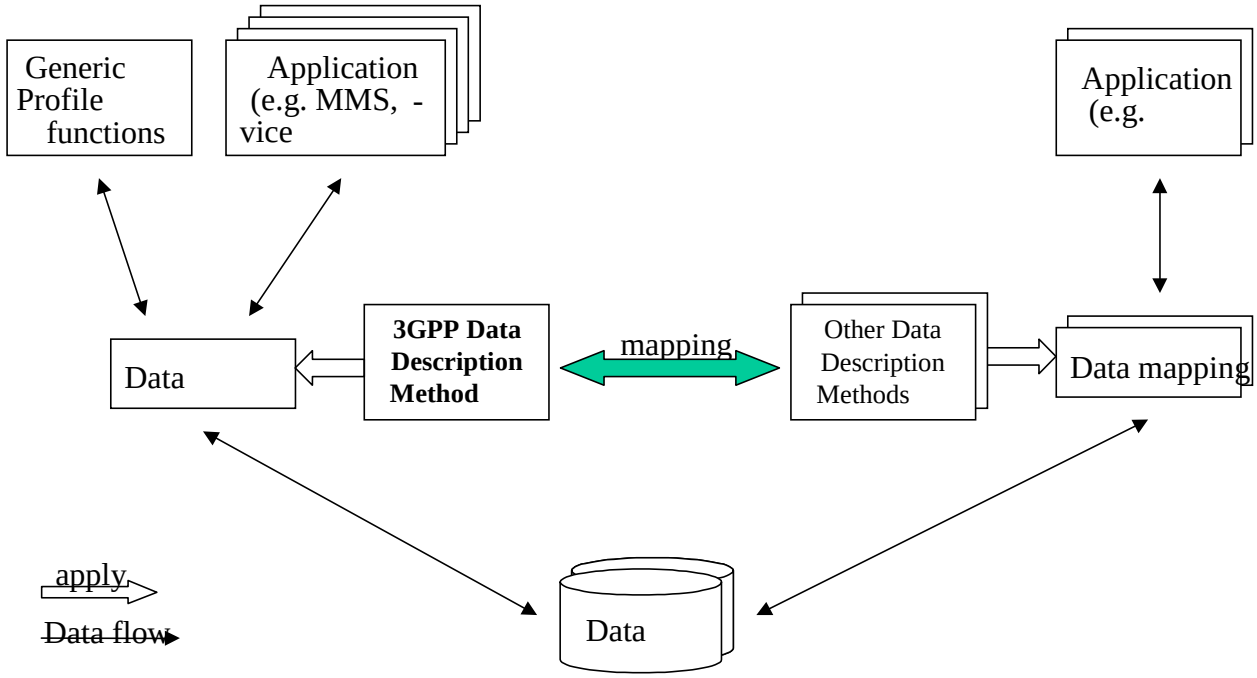


Figure 4.1: Data Description Method coexistence - the mapping principles

The present document is part of a series of documents specifying Generic User Profile functionality in UMTS network with application services. The functional description of the GUP specified for stage 1 in 3GPP TS 22.240 [10] and stage 2 in 3GPP TS 23.240 [11] is taken as the basis and reference for this work. The DDM and DtDM part in the present document may be used outside 3GPP. 3GPP TS 24.241 [12] is the Stage 3 Specification, and it contains the Common Objects resulting from the data defined using the DDM and DtDM. 3GPP TS29.240 [14] is the Stage 3 Network Specification .

5 Structure of Profiles

5.1 Introduction

This clause further defines parts of the Information Model (see 3GPP TS 23.240 [11]) which are relevant to Profiles (or Generic User Profiles) and Profile Components.

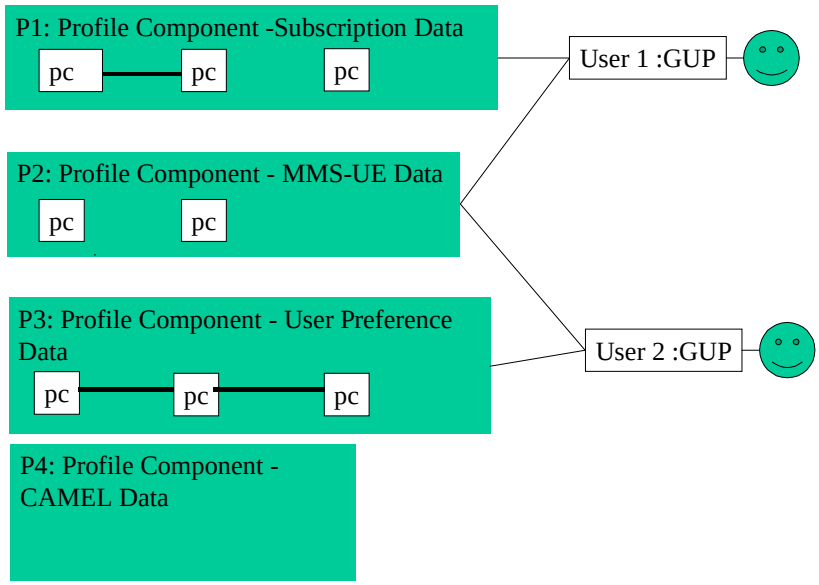
It describes the concepts used to model the GUP data and the relationships between those concepts and defines characteristics of all Profile Component types. It is used in the design and implementation of generic functions using, maintaining and managing GUP data. Properties specific to a Profile Component type are described using the DDM and DtDM.

Important issues described as illustrations and examples only, include different types of Generic User Profile usage and storage distribution.

5.1.1 Profile

A Profile or Generic User Profile is the collection of Profile Components. At least one Profile Component should be mandatory in a Profile. A Profile is instantiated and thus has an associated Profile Instance. The Profile implements the functional definition of Profile according to GUP Stage 1 (3GPP TS 22.240 [10]. But there may be Profiles or Profile Components that are not related to a certain user, e.g. see P4 in figure 5.1.1-1.

Figure 5.1.1-1 shows an example of Generic User Profile as a collection of Profile Components.



NOTE: pc: Profile Component. Connection lines indicate examples of hierarchical relationships.

Figure 5.1.1-1: An example of Generic User Profile as a collection of Profile Components

Figure 5.1.1-2 shows the structure of the Profile and the relationship between the main parts. The main parts in this Figure are described in the clauses immediately following. Figure 5.1.1-2 shows the generic structure, and it is not essential that all elements be present in all implementations of the structure.

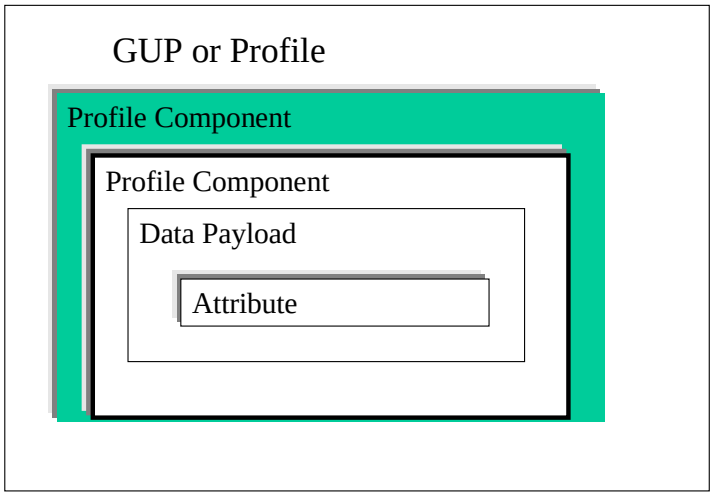


Figure 5.1.1-2: Structural relation between main parts

5.1.2 Profile Instance

A Profile is instantiated and thus a Profile has an associated Profile Instance. A Profile Instance keeps together logically related Profile Component Instances (data), which are possibly distributed in several storage nodes.

A **Profile** Instance contains one or more **Profile Component** Instances each containing the **Data Payload**.

5.2 Structure of Profile Component and Profile Component Instance

5.2.1 Profile Component

A Profile Component is the collection of Data Elements (Data Payloads). A Profile Component is instantiated and thus has an associated Profile Component Instance. The Profile Component Instance may be the independent unit for creation, deletion, storage, and access control. However, deletion should be permissible only according to the rules in Section 8.

An individual service may make use of a number of Profile Components (i.e. a subset) from the Profile.

A Profile Component will contain one or more other Profile Components. Profile Components may be nested in a hierarchical manner. Profile Components may contain Data Element Groups which may contain other Data Element Groups, and which contain Data Elements.

A Profile Component is defined at a Logical level, thus it has a Logical relationship to a Profile Component defined within it. A Profile Component Instance has a relationship to the Profile Component from which it is instantiated.

5.2.2 Profile Component Instance

A Profile Component Instance is coupled to the Profile Instance or another Profile Component Instance. Several Profile Component Instances are grouped into a Profile Instance or another Profile Component Instance.

5.2.2.1 Data Payload

A Data Payload is the Profile data contained in a Profile Component Instance. A Data Payload carries the data values, e.g. MMS terminal capability information regarding supported media types and media formats. A Data Payload contains one or several Attributes.

5.2.2.2 Attribute

An Attribute carries the individual Profile value (content). One or more Attributes are carried in a Data Payload.

5.3 Structure of Profile Description and Profile Component Description

5.3.1 Profile Description

Every Profile has an associated Profile Description. A Profile Description describes the Profile type, semantic, payloads, and properties.

A Profile is described in one Profile Description. A Profile Description describes zero or more Profiles.

A Profile Description consists of Profile Component Descriptions which consist of other Profile Component Descriptions, in an analogous manner to a Profile consisting of Profile Component and a Profile Component consisting of other Profile Components.

A **Profile Description** is the definition of a **Profile Type**, which is the class or type of Profile Instances. The common properties of a number of Profile Instances are described by a Profile Type. This relationship is shown in figure 5.3.1-1.

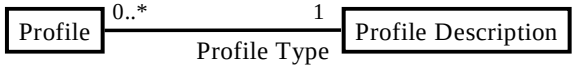


Figure 5.3.1-1: Profile Description

The purpose of a Profile is to keep together logically related data, which are potentially distributed in several storage nodes.

The definition of a Profile Type is kept together in a Profile Description.

A Profile Description contains:

- **Semantics:** defines the meaning of the Profile.
- **Profile Type:** defines the Type of the Profile.
- **Payload Datatype:** is a reference to a Composite Datatype that describes the content of the Profile's Data Payload.
- **Profile Property reference:** is referencing a Common Properties containing data controlling the usage and handling of the Profile.

The Property contains information that defines the rules, which control the usage and handling of Profiles.

Examples of property information are:

- Dynamics, change rate of:
  - Component creation/deletion.
  - Data value.
- Ownership.
- Access rights for different users:
  - No access, read, write access.
  - Right to create, delete.

Figures 5.3.1-2 and 5.3.1-3 show this relationship in UML diagram form. Profile Component Identities and Profile Identities are described in the next clause.

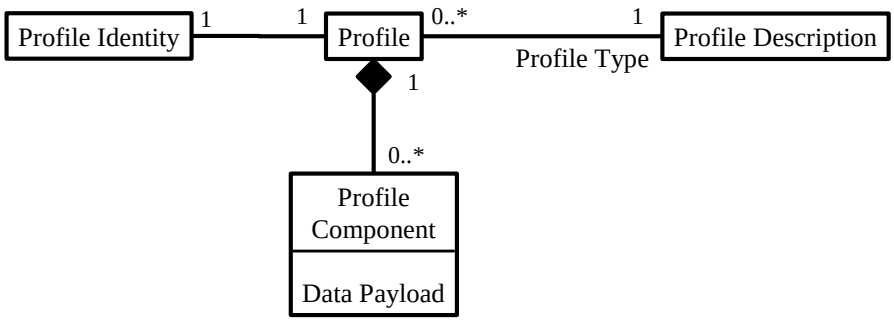


Figure 5.3.1-2: UML-diagram, Profile Description

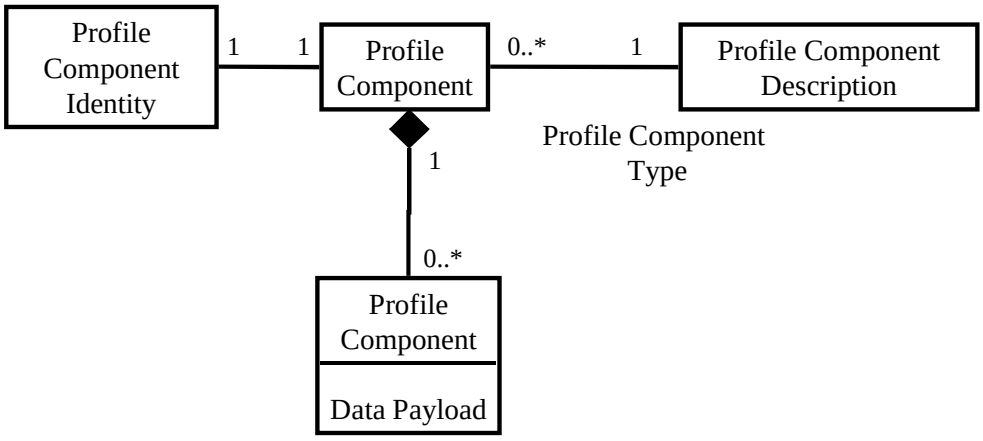


Figure 5.3.1-3: UML-diagram, Profile Component Description

5.3.2 Profile Component Description

Every Profile Component has an associated Profile Component Description. A Profile Component Description describes the Profile Component type, semantic, payloads, and properties.

A Profile Component Description contains:

- **Semantics:** defines the meaning of the Profile Component.
- **Profile Component Type:** defines the Type of the Profile Component.
- **Payload Datatype:** is a reference to a Composite Datatype that describes the content of the Profile Component's Data Payload.
- **Component Property reference:** is referencing a Common Properties containing data controlling the usage and handling of the Profile Component.

The Component Property contains information that defines the rules, which control the usage and handling of Profile Components.

Examples of property information are:

- Dynamics, change rate of:
  - Component creation/deletion.
  - Data value.
- Ownership.
- Access rights for different users:
  - No access, read, write access.
  - Right to create, delete.

Figure 5.3.2-1 describes the Relationship between a Profile Component and its associated Profile Component Description.

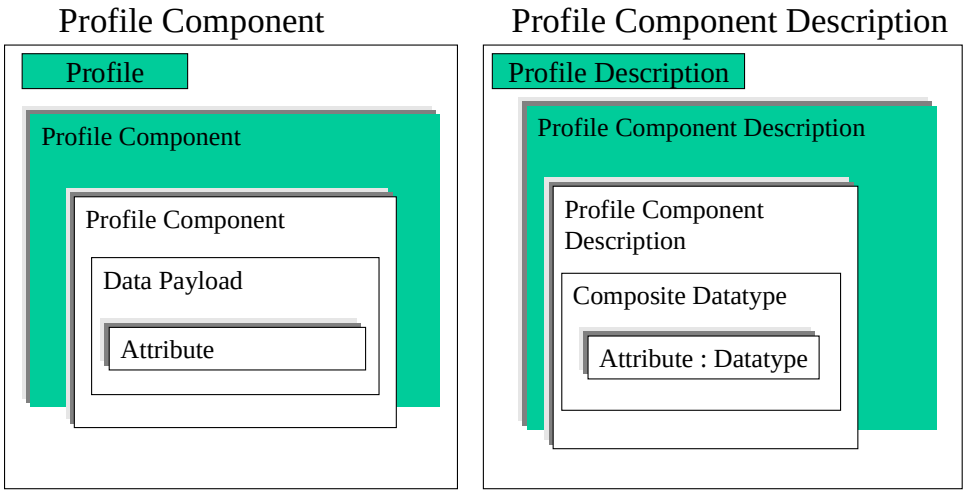


Figure 5.3.2-1: Relationship between Profile Component and Profile Component Description

5.4 Structure of Identities

5.4.1 Profile Identity

A Profile Identity is a permanent Identifier used to identify one Profile Instance.

A **Profile** Instance associates some data with a **Profile Identity**. The data in the **Data Payload** is the data that the Profile Instance associates with the Profile Identity.

A **Profile** Instance is always associated with one **Profile Identity** and the **Profile Identity** is associated with one **Profile** Instance, i.e. they have a one-to-one relationship.

There can be many ways besides the Profile Identity to navigate to a Profile Instance.

5.4.2 Profile Component Identity

In an analogous manner to the Profile Identity and its structure and properties described above, a Profile Component Identity is a permanent Identifier used to identify one Profile Component Instance.

A **Profile Component** Instance associates some data with a **Profile Component Identity**. The data in the **Data Payload** is the data that the Profile Component Instance associates with the Profile Component Identity.

A **Profile Component** Instance is always associated with one **Profile Component Identity** and the **Profile Component Identity** is associated with one **Profile Component** Instance, i.e. they have a one-to-one relationship. A Profile Component has a unique Identity within the Generic User Profile..

A part of the Profile Component Instance name is the Profile Instance name. The names of all the Profile Component Instances belonging to a Profile Instance contain the Profile Identity. By using the name of the Profile Component Instances all Profile Component Instances belonging to a Profile Instance can be found.

A Naming Convention is used to enforce this association. The Naming Convention is specified in clause 7.

5.5 Structure of User Identities and Run-time binding

5.5.1 Relationship between a User and a Profile Instance

This clause describes a number of possible implementations of the functional relationship between the User and a Profile Instance. These relations are used to find the Profile Component Instances included in the User's Generic User Profile.

The method to identify a User is by a User Identity. A User Identity may contain a Public Identity part and a Private Identity part (e.g. IMSI as a Public Identity and/or IMS Private ID as a Private Identity part).

The following clauses treat a number of possible ways to connect a User Identity and a Profile Component Instance. Three different kinds of relations are described:

1) User Identity is a part of the Profile Identity: when a Profile Instance is created it is associated to a User and the User's Identity is included in the Profile Instance name.

2) Referenced Profile: a Profile is referred from another Profile related to the User.

3) Run-time binding: an already existing Profile component is associated to a User in run-time.

5.5.2 User Identity part of the Profile Identity

The name of a Profile Instance is easily related to a User. The User Identity can be a part of the Profile Instance name. (The User Identity must be known when the Profile is created, because the Profile Identity must be permanent.)

Examples of such Profiles are:

- The Profile Instance(s) created when a new User is registered in the access network.
- The Profile Instance(s) created by a service provider when a new User is created for a specific service.

5.5.3 Referenced Profile

The Data Payload in a Profile may contain a reference to a Profile using a Profile Identity. A User's Profile Instance references another Profile instance and by that the referenced Profile instance is included in the User's Generic User Profile Instance as a Profile Component Instance.

If a Profile Component Instance needs to be shared by more Users, references can be used. A referenced Profile Component Instance can be shared by many Users and can be a part of many Generic User Profiles. The referenced Profile Component Instance must be created independently of the referencing Profiles.

Examples of such Profiles are:

- The Profile describing a set of services, which can be subscribed by many Users.
- Profiles describing the subscription rules used by many Users.

5.5.4 Run-time binding

Normally a lot of Profile Component Instances are built into a phone during manufacturing. The Profile Component Instances must be given a Profile Identity that can be used locally in the phone. The User Identity is first known when the (U)SIM is inserted and cannot be used in the naming of Profile Components. The binding between a User and the Profile Component Instances can first be done in run-time.

To be able to navigate from the User to the Profile Component Instances belonging to the User's Generic User Profile in the phone there need to be some run-time support. A dynamic binding which is a reference which value is defined first at run-time can be used.

Examples of such Profile Component Instances are:

- The Profile Component Instances in a mobile phone that is connected to a user by a removable (U)SIM.

- The Profile Component Instances in an accessory when the accessory is connected to a mobile phone (with (U)SIM-card).
- The Profile Component Instances in a computer connected to the mobile phone.
- The Profile Component Instances stored on a User's (U)SIM. (The (U)SIM is normally manufactured before the (U)SIM is connect to a specific User.)

5.6 Details of Profile Structure

Figure 5.6-1 gives a UML overview of the High-Level Structure of Profiles and their constituent parts described in clause 5.

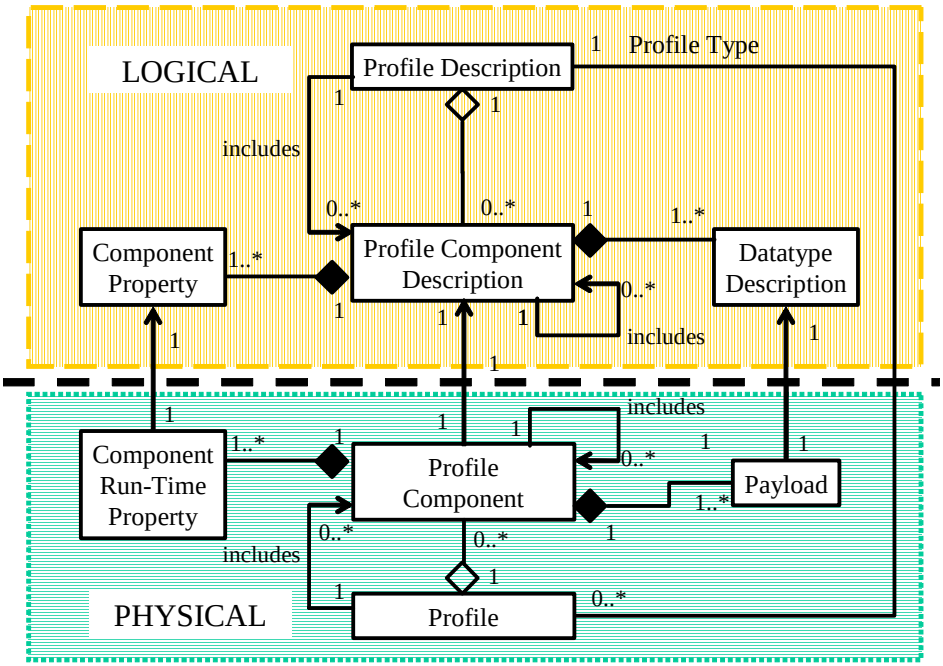


Figure 5.6-1: UML-diagram, details of Profile High Level Structure and constituent parts

Note that figure 5.6-1 only shows an exemplary physical representation of the data structure to indicate the relation between the logical structure which is defined in this document and a possible physical implementation. The physical implementation is beyond the scope of this specification.

5.7 Distributed Profile storage

Multiple copies of a Profile Component Instances are expected to be stored at different storage nodes. In every such instance, one of these is tagged as the Master Profile Component Instance.

Figure 5.7-1 shows an example possible storage distribution of Profile Component Instances.

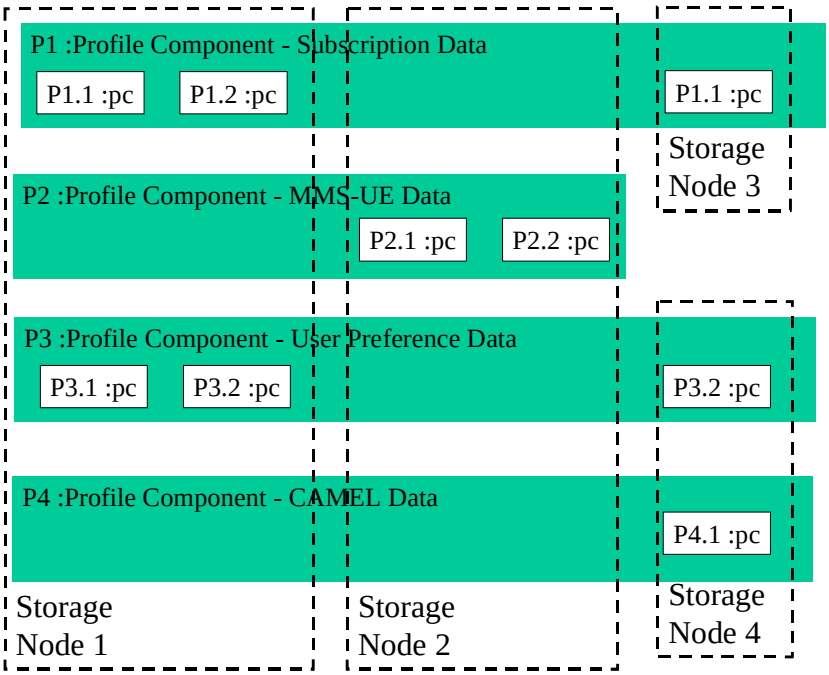


Figure 5.7-1: An example of Profile storage distribution

6 Data Description Method (DDM)

6.1 Introduction

Clause 6 describes the DDM, which defines how to organize GUP data in a standardized manner.

The description of the users' services configuration and personalization data using the DDM may result in manipulating and accessing these data in a structured, and standardized way. The DDM will help to overcome some of the challenges associated with the introduction of sophisticated user terminals and services with widely varying capabilities, hybrid combinations of mobile network domains, the advent of downloadable applications, and the desire of users to customize potentially complex services to individual preferences and needs.

The Data Description Method (DDM) for GUP is based on the XML Schema. XML-Schema is a Schema definition language defined by W3C. The W3C Recommendation consists of three parts: XML Schema Part 0 that is a non-normative document intended to provide an easily readable description of the XML Schema facilities, XML Schema Part 1that describes Structures, and XML Schema Part 2 that describes Datatypes.

The Data Description Method can be viewed as a set of templates for constructing the data description. The templates (sets of rules) enable the standardization of the data description such that it and the described data can be shared (used) by many applications. The data descriptions are abstract in the sense that the data are described independently of data formats specific to data storage, transport protocols or application technologies. Abstraction of data descriptions simplifies the mapping between different data formats, and facilitates future extensions.

The common use of the Data Description Method will avoid incompatibilities and inconsistencies between different Profile Components.

6.2 3GPP XML Schema Namespace definition

This clause defines the XML Schema namespace rules for 3GPP GUP.

All entities not specifically declared under the namespace defined in this clause are by definition defined under the W3C XML Schema namespace (see [3] for further information).

The namespace URI for GUP specific documents is a URL, starting with 'http://www.3GPP.org/GUP-ns/' and followed by a sub-namespace specific string in the syntax according to [15].

A Profile Component should be defined in its own sub-namespace.

The following are the rules and guidelines for namespaces. The definition of multiple namespaces within 3GPP GUP and DDM is considered beneficial, and is allowed. A namespace should defined at the Profile Component level. However, for the case of a deep hierarchical structure composed of Profile Components, it is intended that a namespace should be defined for only the top levels of the hierarchical structure of Profile Components, not for every Profile Component at even the lowest levels. It is not intended that all Profile Components must have their own namespace. It is considered good practice, to avoid the large proliferation of namespaces which could be caused by every Profile Component being assigned its own namespace. This would result in a large administrative overhead of cataloging and checking for uniqueness, assignment of unique prefixes, declaration of a large number of namespaces etc.

The sub-namespaces are defined in GUP specifications and/or implementations. It must be assured that ambiguities are avoided, via searches for name collisions.

The definition of Profile Components directly under the root namespace 'http://www.3GPP.org/GUP-ns/' or any other predefined sub-namespace is not allowed.

The predefined sub-namespace for the GUP Profile is

```
http://www.3GPP.org/GUP-ns/profile
```

The predefined sub-namespace for Profile Components is

```
http://www.3GPP.org/GUP-ns/comp
```

This namespace is divided into Profile Component specific sub-namespaces by adding the name of the defined Profile Compontent in the format

```
http://www.3GPP.org/GUP-ns/comp/<name>
```

e.g. for definition of CAMEL Profile Component it would be

```
http://www.3GPP.org/GUP-ns/comp/CAMEL
```

There is a predefined sub-namespace for the common definitions schema which is described in clause 6.6. This namespace is

```
http://www.3GPP.org/GUP-ns/common
```

The prefix which should be used with this namespace is 'comdef'.

6.3 Profile Description

This clause creates a top-level Schema for the Profile Description called the Profile Root Schema. This Profile Root Schema is to be used as a template for the creation of new Profile Description Schemas.

This Profile Root Schema contains Datatype Definition Schemas containing Datatype definitions.

Figure 6.3-1 shows the UML diagram for the Profile Root Schema. The blocks are defined in clause 5.

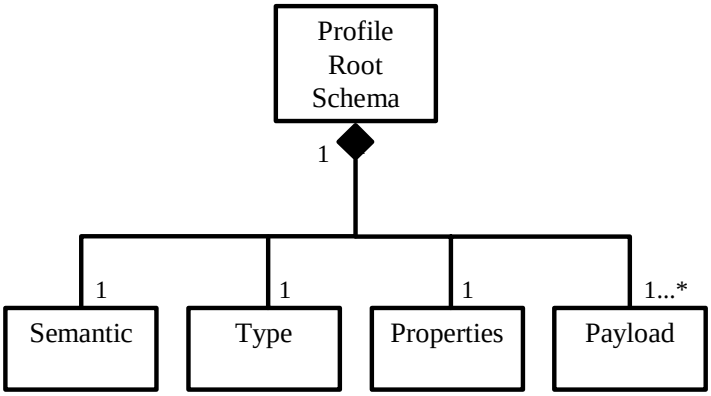


Figure 6.3-1: Profile Root Schema - UML

The Profile Root Schema is defined by the XML Schema, figure 6.3-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3GPP.org/GUP-ns/profile"
    xmlns:GUP="http://www.3GPP.org/GUP-ns/profile"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:comdef="http://www.3GPP.org/GUP-ns/common"
    xmlns:comp1="http://www.3GPP.org/GUP-ns/comp/ProfileComp1"
    xmlns:comp2="http://www.3GPP.org/GUP-ns/comp/ProfileComp2"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://www.3GPP.org/GUP-ns/common"/>
    <xs:import namespace="http://www.3GPP.org/GUP-ns/comp/ProfileComp1"/>
    <xs:import namespace="http://www.3GPP.org/GUP-ns/comp/ProfileComp2"/>
    <!-- import all PCs, which are used (referenced) in this schema
    - the import of two PCs and the properties is shown above as an example -->

    <xs:annotation>
        <xs:documentation xml:lang="en">
            Profile: P name
            Profile consists of following PCs:
                list of PCs
            Version x.y
            History: x.y created on yyyy-mm-dd
            Keywords:
        </xs:documentation>
    </xs:annotation>

    <!-- root element (P) -->
    <xs:element name="P_Name" type="GUP:PNameType"/>
    <!-- no other declarations necessary as all elements are imported from other PCs -->
    <xs:complexType name="PNameType">
        <xs:sequence>
            <xs:element ref="comdef:Properties" minOccurs="0"/>

            <xs:element ref="comp1:PC_1"/>
            <xs:element ref="comp2:PC_2"/>
            <!-- etc. -->
        </xs:sequence>
        <xs:attribute name="identity_P" type="xs:string"/>
    </xs:complexType>
</xs:schema>
```

Figure 6.3-2: Profile Root Schema - XML

Note that all Profile Components are imported to the Profile as the definition of Profile Components in the root namespace is not allowed and such an inclusion can be ruled out.

The Profile Root Schema will be used to create Profile Description schemas. The process is shown in figure 6.3-3.

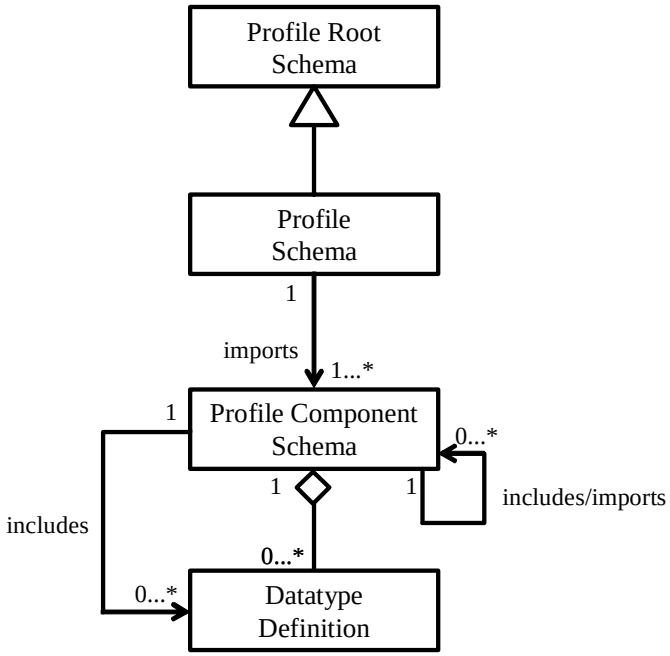


Figure 6.3-3: Creating Profile Description schema using the Profile Root schema - UML

The process of creating Profile Description Schema described in figure 6.3-3 is as follows. First the Profile Description Schema for the use is created based on the XML Schema for the Profile Root Schema. Then the next lower level structures are added as shown in the XML Schema. Then these lower level structures are developed as in the following clauses.

While in the usual case a Profile will be structured by use of the Profile Root Schema template as described above and will have a Profile Schema construct, the case is also permitted where a profile does not necessarily have to be defined, as it can also be a mere collection of profile components and therefore a virtual construct. The collection of profile components can then be validated using the descriptions of the components. (An example of such a virtual Profile may be the case where only a few, e.g. one or two, Profile Components are required. In such a case, a Profile may not be necessary).

6.4 Profile Component Description

This clause creates a top-level Schema for the Profile Component Description called the Profile Component Root Schema. This Profile Component Root Schema is to be used as a template for the creation of new Profile Description Schemas.

This Profile Component Root Schema may contain Datatype definitions and/or include / import one or more Datatype Definition Schemas containing Datatype definitions.

Figure 6.4.1 shows the UML diagram for the Profile Component Root Schema. The blocks are defined in clause 5.

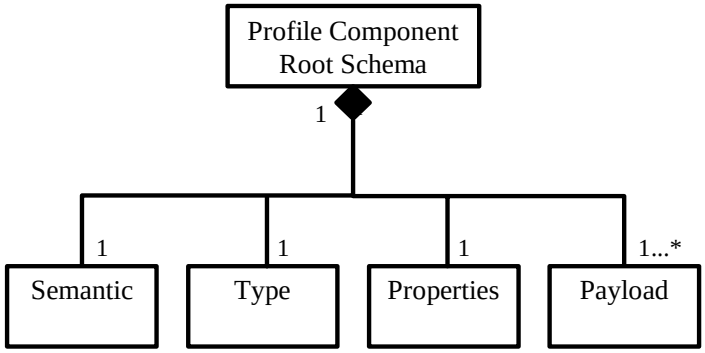


Figure 6.4-1: Profile Component Root Schema - UML

The Profile Component Root Schema is defined by the XML Schema, figure 6.4-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3GPP.org/GUP-ns/comp/ProfileCompX"
    xmlns:compX="http://www.3GPP.org/GUP-ns/comp/ProfileCompX"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:comdef="http://www.3GPP.org/GUP-ns/common"
    xmlns:compY="http://www.3GPP.org/GUP-ns/comp/ProfileCompY"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://www.3GPP.org/GUP-ns/common"/>
    <xs:import namespace="http://www.3GPP.org/GUP-ns/comp/ProfileCompY"/>
    <!-- import all PCs of other sub-namespaces, which are used (referenced) in this schema
    - the import of a PC and the properties is shown above as an example -->

    <xs:annotation>
        <xs:documentation xml:lang="en">
            Profile Component: PC name
            Parents: PC name(s), P name
            PC consists of following PCs, PC fragments and/or DEs:
                list of PCs, PC fragments and/or DEs
            Version x.y
            History: x.y created on yyyy-mm-dd
            Keywords:
        </xs:documentation>
    </xs:annotation>

    <!-- include all PCs, which are used (referenced) in this schema or
     whose parts, e.g.fragments and/or data elements are used (referenced)
    - the include of one PC is shown below as an example -->
    <xs:include schemaLocation="http://www.3GPP.org/GUP-ns/comp/ProfileCompX/PC_1.xsd"/>

    <!-- root element (PC) -->
    <xs:element name="PC_Name" type="compX:PCNameType"/>
    <!-- list of all data elements follows -->
    <xs:element name="Data_Element_1" type="compX:DataType1"/>
    <xs:element name="Data_Element_2" type="compX:DataType2"/>
    <!-- etc. -->
    <!-- type definition of the root element -->
    <xs:complexType name="PCNameType">
        <xs:sequence>
            <xs:element ref="comdef:Properties" minOccurs="0"/>
            <xs:element ref="compX:Data_Element_1"/>
            <xs:element ref="compX:Data_Element_2"/>
            <!-- etc. -->
            <!-- references to included / imported PCs or PC fragements follow -->
            <xs:element ref="compX:PC_1"/>
            <xs:element ref="compY:DE_5ofPC_2"/>
            <!-- etc. -->
        </xs:sequence>
        <xs:attribute name="identity_PC" type="xs:string"/>
    </xs:complexType>

    <!-- definition of data elements follows -->
    <xs:simpleType name="DataType1">
        <xs:restriction/>
        <!-- type definition for Data_Type_1 -->
    </xs:simpleType>

    <xs:complexType name="DataType2">
```

```
    <!-- type definition for Data_Type_2 -->
</xs:complexType>
<!-- etc. -->
</xs:schema>
```

Figure 6.4-2: Profile Component Root Schema - XML

The Profile Component Root Schema will be used to create Profile Component Description schemas. The process is shown in figure 6.4-3.

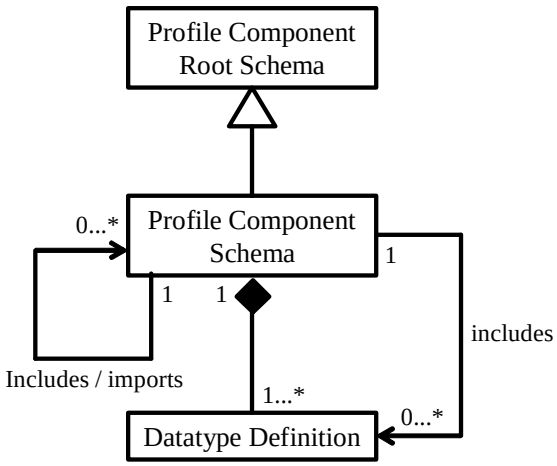


Figure 6.4-3: Creating Profile Component Description Schema using the Profile Component Root Schema - UML

The process of creating Profile Component Description Schema described in figure 6.4-3 is as follows. First the Profile Component Description Schema for the use is created based on the XML Schema for the Profile Component Root Schema. Then the next lower level structures are added as shown in the XML Schema. Then these lower level structures are developed.

An Example of the use of the Profile Component Root Schema to create a Profile Component Description XML Schema for CAMEL data is shown in clause 8.6.2.

An Example of a Data Element Description XML Schema for CAMEL data is shown in clause 8.6.3.

6.5 Description of Versions

The entities described in clause 6 should all be assigned version numbers, in order that unique identification of the latest approved version for use should be possible.

The version number should be a part of the Properties as defined by the XML Schema in this clause.

Version numbers should follow the rules:

- Version x.y:
  - Where x is a ready for use number.
  - Where y indicates a developmental unapproved number.
  - X, y are integers which should be incremented upon version change.

See clause 8 for rules about changing versions and version numbers.

6.6 Properties (Common Definitions) Schema

The purpose of the GUP Properties is to define common XML definitions which can be used by other GUP specific XML schemas.

This schema is out of scope of this specification. It can be found under the name Common Definition Schema in [14].

6.6.1 Typical Parameters in Properties (Common Definitions) Schema

This section shows some typical parameters which would be collected as Properties or Common Definition Schema in [14].

- Private IDs (e.g. Private attributes of Subscriber identification).
- Public IDs (e.g. Subscriber identification).
- Other addresses.
- Service identifications (e.g. Application identification).
- Generic privacy control data (e.g. Privacy policy).
- Generic error data (e.g. Invalid operation, Invalid parameter, Unauthorized operation, Data unavailable, Unexpected error).
- Date and time.
- Service state (e.g. Redirection indications).
- Subscription data (e.g. Credentials, Authorization assertion).
- Terminal Management.
- Subscription Check (by 3rd party).
- Service Personalization data.
- Capabilities data of terminals.
- Preferred access.
- User-level blacklist.
- Security Policy - Support for Security functions. Authorization control policies and rules data is described here. This clause will be aligned with SA3 Security specs when available.
- Authorization and Access Control Policies - Support for Security functions. Authorization control policies and rules is described here. This clause will be aligned with SA3 Security specs when available.
- Privacy Policies - Privacy control policies and rules are described here. This clause will be aligned with SA3 Security specs when available.

7 Datatype Definition Method (DtDM)

7.1 Introduction

Clause 7 describes the Datatype Definition Method, DtDM, a method describing how to define the new datatypes contained in the Generic User Profile. It is used to describe datatypes, which define the possible values a data item can have. This clause also describes how to use the DtDM.

The DtDM includes many types of data entities described in the present document. The term datatype could be understood to be the XML Schema simpleType or complexType, but the use of the term datatype in this clause is intended to include all entities which are discussed within this clause.

7.2 Datatype Definition guidelines

The following guidelines are defined:

1) Each data element should be defined as an XML element of a suitable type.

2) XML attributes should be used only to qualify the data element defined as XML elements and not contain the actual data values.

3) An XML element either contains other XML elements or actual data value. An XML element should not have both a value and other XML elements as subelements.

4) The type definitions provided by the XML schema should be used.

5) All elements should be defined as global elements. When elements with complex type are defined references to global elements are used. The reason is to make it possible to validate parts of the XML data with the schema and that local elements cannot be imported to other namespaces.

7.2.1 Identification of Datatypes

1) A Naming Convention should be followed which should inherently make visible the hierarchical structure of the lower entities.

2) The names should be meaningful, but as short as possible.

3) Different criteria for selection of names should apply for:

   a) the datatypes used to generate the Profile Components;

   and for

   b) the Profile Components themselves.

   For a), easily understood generic names should be used for datatypes themselves whenever feasible, in order to maximize the re-use of already-defined datatypes subsequent to the first definition and use, so that experts working on subsequent potential uses should be able to easily understand the datatype from the name.

   As an example, in the examples shown in clause 8.7.3 for CAMEL, the Profile Component with the name O-CSI in CAMEL should be defined by the complexType datatype with the name OrigCamelSubscrInfo.

4) If a name consists of more than one word or abbreviated word, capitalization should be used to keep the long names readable. Each new word after the first word in a name should start with a capital letter. Also the first word should start with a capital letter for names of XML elements and types (e.g. ElementName, TypeName).

5) The attribute names should start with a lower case letter (e.g. attributeName).

6) When abbreviations which take the first letter of each word are used, the whole abbreviation is capitalized (e.g. GSMPhone). When the abbreviation is few characters from a word like addr for address, the abbreviation is handled like a word, i.e. sometime the first letter should be capitalized and sometimes not (e.g. HomeAddr for element containing home address and addrType for attribute containing the address type information qualifying the data element).

7.2.2 Naming Convention

The Naming Convention should be of the type

3GPP/GUP/Profile_name/Profile_Component_name/…/Profile_Component_name/Data_Element_name/attribute_name

Naming Convention = 3GPP/GUP/P/PC/…/PC/DE/AT

Example of use of the naming convention:

3GPP/GUP/P_GUP/PC_Subscription/PC_CAMEL/PC_O_CSI/DE_TDP_List_originating

7.3 XML Schema Usage for Datatype Definitions

This clause describes the XML Schema to be used for creating new Datatype Definitions. It defines how different syntax constructions such as atomic types and composite datatypes are represented using XML Schema.

The XML Schema to be used is defined in [1] to [8]

8 Process for administration of GUP/DDM

8.1 Introduction

This clause specifies the Process for Administration of GUP/DDM Common Objects data in 3GPP in TS24.241. Certain GUP/DDM Common Objects data are stored in 3GPP TS 24.241 [12] as may be proposed by various Working Groups of 3GPP.

8.2 Responsibility for administration of GUP/DDM Common Objects

A Working Group within 3GPP will be assigned for the task of administration of GUP/DDM Common Objects in TS24.241. This Working Group is referred to in this clause as the Responsible WG. The Responsible WG should also be the Owner and Editor of 3GPP TS 24.241 [12] which should be used for the purpose described in this section.

8.3 Formal Recognition of GUP/DDM Common Objects

Formal recognition of the new Common Objects or other constructs should be through the use of a CR to 3GPP TS 24.241 [12] requesting the addition to the specification of the Common Objects or other constructs to be recognized.

Any 3GPP Working Group may submit approved Change Requests for creation, storage, modification, deletion, etc. of those of its GUP/DDM Common Objects which it would like to propose to be stored in TS24.241, to the Responsible WG in accordance with the rules set out in this clause. These Change Requests should use their own namespaces as defined in clause 6.6.

3GPP Working Groups should conduct a thorough search and evaluation of the existing Common Objects using the methods described in this clause prior to deciding that creation of a new Common Object is necessary. The WGs should also perform a check that the new Common Object terms being proposed by them are not already in use, or in conflict with, the existing Common Objects. If a new sub-namespace is defined by a Working Group it should check that it is not already in use.

The Responsible WG should administer the GUP/DDM Common Objects in 3GPP TS 24.241 [12] and bring proposed changes to 3GPP TS 24.241 [12] as Change Requests for approval as per normal 3GPP process (see 3GPP TS 21.101 [13]).

8.4 Creation, Modification, Deletion of GUP/DDM Common Objects

The Responsible WG should follow special rules for modification or deletion/ removal of Common Objects.

Common Objects are likely to be re-used by more than the originating WG. Therefore, in order to avoid possible problems in other areas due to planned changes by one of the using WGs, the Responsible WG should define and follow a procedure called 'Prior Notification of planned modification / deletion of a Common Object'. The procedure should require wide circulation of planned modifications/deletions and should as a minimum include all using WGs shown in the Common Object information.

The Responsible WG should follow 3GPP processes for creation of GUP/DDM Common Objects data based on requests received from 3GPP WG's, in accordance with the procedures set out in this clause.

8.5 Versioning rules

This clause describes how version numbers should be administered.

The Version number is specified in clause 6 as:

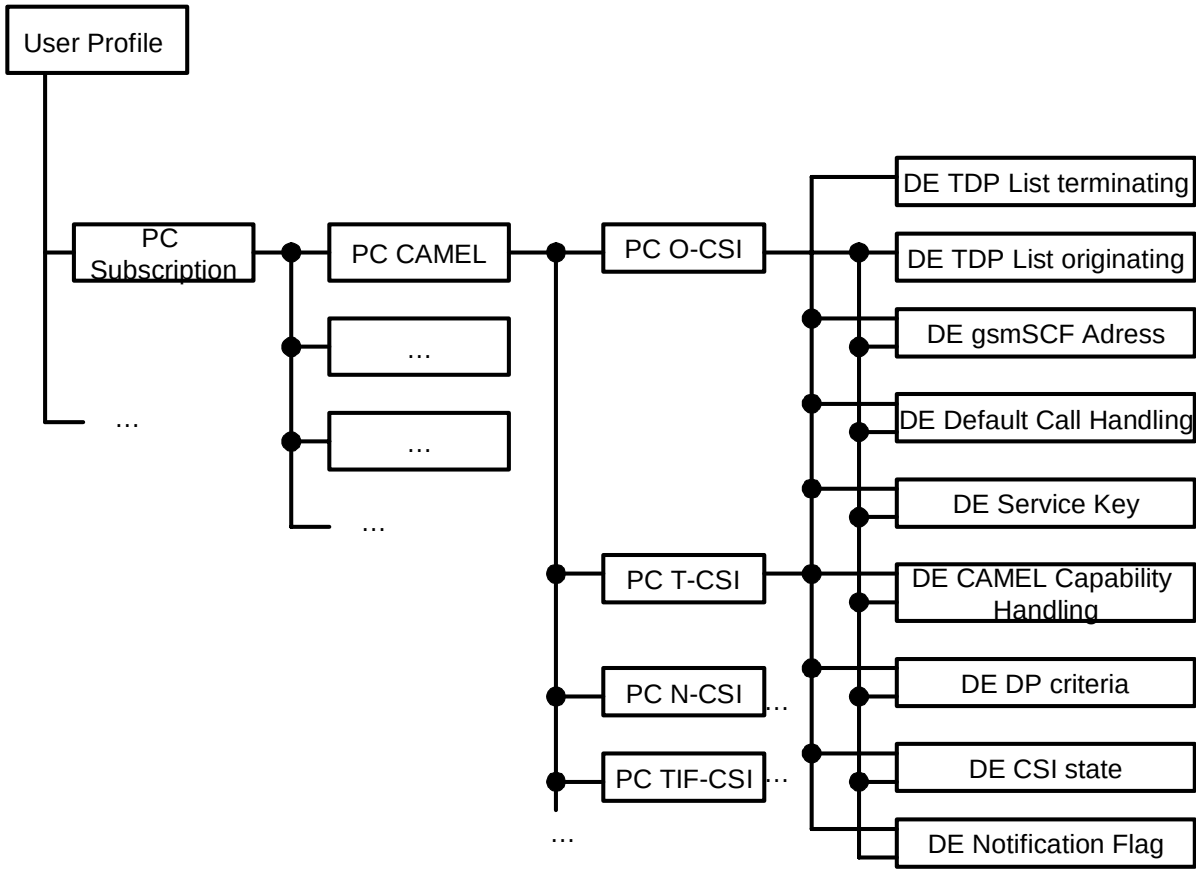
- Version x, y
  - y can be incremented by the developing WG or the Responsible WG in coordination.
  - x can be incremented by the Responsible WG only.

8.6 Common Objects storage Syntax

8.6.1 Introduction

Common Objects to be stored in 3GPP TS 24.241 [12] should be described using a Storage Syntax which provides essential and pertinent information to the reader about that Common Object. The structure of the Storage Syntax is specified in the subsequent clauses, including examples with XML schema. The examples are based on CAMEL subscription data (3GPP TS 23.078 [9]) which are also partly shown in figure 8.6.1-1.

The following is an informative example only showing a small part of the CAMEL data, the full definition of CAMEL data will be given in 3GPP TS 24.241 [12].



PC Profile Component.
DE Data Element

Figure 8.6.1-1: Example structure based on CAMEL subscription data

8.6.2 Profile Component Common Object storage Syntax

This clause specifies the Storage Syntax for the Profile Component Common Object.

Two examples have been added which represent the nested structure of PCs. These examples are the Profile Component CAMEL parameter, which includes a number of other Profile Components and the Profile Component O-CSI, which is included in the CAMEL parameter.

Table 8.6.2-1: Syntax of a Profile Component Common Object in 3GPP TS 24.241 [12]

Defining Group	< group >
Date Created (original)	< date >
Defining Application	< application >
Type of item	PROFILE COMPONENT
Name of Item (being defined)	< name >
Datatype	< type name >
Namespace	< namespace >
Parent tree of original use (definition)	< names of PC, P >
Attribute list	< attribute 1 > < type 1 > < attribute 2 > < type 2 > … < attribute n > < type n >
Search Criteria	< keywords >
Modifications/enhancements for original use	< date >, < modifying WG >, < summary of modification >
List of Re-uses	< date 1 >, < using WG 1 >, < Using Application 1 >, <Using Parent tree 1> < date 2 >, < using WG 2 >, < Using Application 2 >, <Using Parent tree 2> ...
Common Object consists of (INCLUDES / IMPORTS)	< item 1 > < item 2 > ... < item n >

Table 8.6.2-2: Example Syntax of a Profile Component for CAMEL parameter in 3GPP TS 24.241 [12]

Defining Group	CN2
Date Created (original)	2003-10-23
Defining Application	CAMEL
Type of item	PROFILE COMPONENT
Name of Item (being defined)	CAMEL
Datatype	CamelType
Namespace	http://www.3GPP.org/GUP-ns/comp/Subscription/Camel
Parent tree	Subscription, User Profile
Attribute list	< attribute 1 > < type 1 > < attribute 2 > < type 2 > … < attribute n > < type n >
Search Criteria	CAMEL, HLR data, CAMEL Subscription
Modifications/enhancements for original use	-
List of Re-uses	-
Common Object consists of (INCLUDES / IMPORTS)	PC O-CSI PC T-CSI PC N-CSI PC TIF-CSI ... < item n >

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel"
    xmlns:CAM="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel"
    xmlns:xs="http://www.w3.org/2001/XMLSchema""
    xmlns:comdef="http://www.3GPP.org/GUP-ns/common"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://www.3GPP.org/GUP-ns/common"/>

    <xs:annotation>
        <xs:documentation xml:lang="en">
            Profile Component: CAMEL
            Parents: PC Subscription, P User
            PC consists of following PCs:
                O-CSI
                T-CSI
                VT-CSI
                GPRS-CSI
                U-CSI
                ...
            Version 1.0
            History: 1.0 created on 2003-10-23
            Keywords: CAMEL, HLR data, CAMEL Subscription
        </xs:documentation>
    </xs:annotation>

    <xs:include schemaLocation="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel/O-CSI.xsd"/>
    <xs:include schemaLocation="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel/T-CSI.xsd"/>

    <!-- root element (PC) -->
    <xs:element name="CAMEL" type="CAM:CamelType"/>
    <xs:complexType name="CamelType">
        <xs:sequence>
            <xs:element ref="comdef:Properties" minOccurs="0"/>
            <xs:element ref="CAM:O-CSI"/>
            <xs:element ref="CAM:T-CSI"/>
            <!--xs:element ref="CAM:x-CSI-->
            <!-- ... -->
            <!-- All elements except Properties must occur exactly once
             (default values of minOccur and maxOccur="1") and are
             referenced to global elements.-->
        </xs:sequence>
        <xs:attribute name="identityCAMEL" type="xs:string"/>
    </xs:complexType>
</xs:schema>
```

Figure 8.6.2-1: Example XML Schema of a Profile Component for CAMEL parameter in 3GPP TS 24.241 [12]

Table 8.6.2-3: Example Syntax of a Profile Component Common Object for O-CSI in 3GPP TS 24.241 [12]

Defining Group	CN2
Date Created (original)	2003-10-23
Defining Application	CAMEL
Type of item	PROFILE COMPONENT
Name of Item (being defined)	O-CSI
Datatype	OrigCamelSubscrInfo
Namespace	http://www.3GPP.org/GUP-ns/comp/Subscription/Camel
Parent tree of original use (definition)	CAMEL, Subscription, User Profile
Attribute list	< attribute 1 > < type 1 > < attribute 2 > < type 2 > … < attribute n > < type n >
Search Criteria	CAMEL, HLR data, Subscription, O-CSI, CSI, Subscription Information
Modifications/enhancements for original use	-
List of Re-uses	-
Common Object consists of (INCLUDES / IMPORTS) - Data Element list	TDP List originating gsmSCF Adress Default Call Handling Service Key CAMEL Capability Handling DP criteria CSI state Notification Flag

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel"
    xmlns:CAM="http://www.3GPP.org/GUP-ns/comp/Subscription/Camel"
    xmlns:comdef="http://www.3GPP.org/GUP-ns/common"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://www.3GPP.org/GUP-ns/common"/>

    <xs:annotation>
        <xs:documentation xml:lang="en">
            Profile Component: O-CSI
            Parents: PC CAMEL, PC Subscription, P User
            PC consists of following DEs:
                TDP List originating
                gsmSCF Adress
                Default Call Handling
                Service Key
                CAMEL Capability Handling
                DP criteria
                CSI state
                Notification Flag
            Version 1.0
            History: 1.0 created on 2003-10-23
            Keywords: CAMEL, HLR data, Subscription, Subscription Information, O-CSI, CSI
        </xs:documentation>
    </xs:annotation>


    <!-- root element (PC) -->
    <xs:element name="O-CSI" type="CAM:OrigCamelSubscrInfo"/>

    <!-- list of all referenced data elements follows -->
    <xs:element name="TDP_List_originating" type="CAM:OrigTDPList"/>
    <xs:element name="gsmSCF_Adress" type="CAM:E164Number"/>
    <!-- etc. -->
    <!-- all other elements are declared here, i.e. Default Call Handling, Service Key,
    CAMEL Capability Handling, DP criteria, Notification Flag -->

    <xs:complexType name="OrigCamelSubscrInfo">
        <xs:sequence>
            <xs:element ref="comdef:Properties" minOccurs="0"/>
            <xs:element ref="CAM:TDP_List_originating"/>
            <xs:element ref="CAM:gsmSCF_Adress"/>
```

```
            <xs:element ref="CAM:Default_Call_Handling"/>
            <xs:element ref="CAM:Service_Key"/>
            <xs:element ref="CAM:CAMEL_Capability_Handling"/>
            <xs:element ref="CAM:DP_criteria"/>
            <xs:element ref="CAM:Notification_Flag"/>
            <!-- All elements except Properties must occur exactly once (default values of
                minOccur and maxOccur="1") and are referenced to global elements.-->
        </xs:sequence>
        <xs:attribute name="identityOCSI" type="xs:string"/>
    </xs:complexType>

    <xs:simpleType name="OrigTDPList">
        <xs:restriction base="xs:string">
            <xs:enumeration value="DP Collected_Info"/>
            <xs:enumeration value="DP Route_Select_Failure"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="E164Number">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{15}"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- type definitions of other used types follows -->
    <!-- etc. -->
</xs:schema>
```

Figure 8.6.2-2: Example XML Schema of a Profile Component Common Object for O-CSI in 3GPP TS 24.241 [12]

8.6.3 Data element Common Object storage Syntax

This clause specifies the Storage Syntax for the Data Element Common Object.

Table 8.6.3-1: Syntax of a Datatype Common Object in 3GPP TS 24.241 [12]

Defining Group	< group >
Date Created (original)	< date >
Defining Application	< application >
Type of item	DATA ELEMENT
Name of Item (being defined)	< name >
Datatype	< simple or complex datatype >
Parent tree	< names of PC(s), P>
Common Object consists of Attribute list	< attribute 1 > < type 1 > < attribute 2 > < type 2 > … < attribute n > < type n >
Search Criteria	< keywords >
Modifications/enhancements for original use	< date > <Modifying WG> < summary of modification>
List of Re-uses	< date 1 > < using WG 1 > < Using Application 1 > <Using Parent tree 1 > < date 2 > < using WG 2 > < Using Application 2 > <Using Parent tree 2 > ...

Table 8.6.3-2: Example Syntax of a Datatype Common Cbject in 3GPP TS 24.241 [12]

Defining Group	CN2
Date Created (original)	2003-10-23
Defining Application	CAMEL
Type of item	DATA ELEMENT
Name of Item (being defined)	TDP List originating
Datatype	OrigTPDList
Parent tree	O-CSI, CAMEL, Subscription, User Profile
Common Object consists of Attribute list	< attribute 1 > < type 1 > < attribute 2 > < type 2 > … < attribute n > < type n >
Search Criteria	CAMEL, Subscription, HLR data, O-CSI, CSI, TDP, TDP List, originating
Modifications/enhancements for original use	-
List of Re-uses	-

The datatype is defined once by an XML schema.

```
<xs:simpleType name="OrigTDPList">
        <xs:restriction base="xs:string">
            <xs:enumeration value="DP Collected_Info"/>
            <xs:enumeration value="DP Route_Select_Failure"/>
        </xs:restriction>
    </xs:simpleType>
```

Figure 8.6.3-1: Example datatype definition for CAMEL TDP List in 3GPP TS 24.241 [12]

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2004-12	CN#26	NP-040539			TS 23.241 v6.1.0 has changed to TR 23.941		6.0.0

History

Document history		
V6.0.0	December 2004	Publication