

3GPP TS 23.241 V1.0.0 (2003-12)

Technical Specification

**3rd Generation Partnership Project;
Technical Specification Group Terminals;
3GPP Generic User Profile (GUP);
Stage 2;
Data Description Method
(Release 6)**



The present document has been developed within the 3rd Generation Partnership Project (3GPP™) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification. Specifications and reports for implementation of the 3GPP™ system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

<http://www.3gpp.org>

Copyright Notification

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

© 2003, 3GPP Organizational Partners (ARIB, CWTS, ETSI, T1, TTA, TTC).
All rights reserved.

Contents

Foreword.....	5
Introduction.....	5
1 Scope.....	6
2 References.....	6
3 Definitions and abbreviations.....	7
3.1 Definitions.....	7
3.2 Abbreviations.....	8
4 General description.....	8
5 Structure of profiles.....	9
5.1 Introduction.....	9
5.1.1 Profile.....	9
5.1.2 Profile instance.....	10
5.2 Structure of Profile Component group, Profile Component, Profile Component group instance and Profile Component instance.....	11
5.2.1 Profile Component group and Profile Component.....	11
5.2.2 Profile Component group Instance and Profile Component instance.....	11
5.2.2.1 Data payload.....	11
5.2.2.2 Attribute.....	11
5.3 Structure of Profile description, Profile Component group description and Profile Component description.....	11
5.3.1 Profile description and Profile Component group description.....	11
5.3.2 Profile Component description.....	14
5.4 Structure of identities.....	15
5.4.1 Primary profile identity.....	15
5.4.2 Profile Component group identity and Profile Component identity.....	15
5.5 Structure of user identities and run-time binding.....	15
5.5.1 Relationship between a user and a profile instance.....	15
5.5.2 User identity part of the primary profile identity.....	16
5.5.3 Referenced profile.....	16
5.5.4 Run-time binding.....	16
5.6 Details of profile structure.....	16
5.6.2 Profile Component payload.....	17
5.6.3 Profile Component run-time properties.....	17
5.6.4 Profile Component payload internal identification.....	17
5.7 Distributed profile storage.....	17
5.7.1 Introduction.....	17
5.7.2 Storage principles.....	18
6 Data Description Method (DDM).....	19
6.1 Introduction.....	19
6.2 3GPP XML schema namespace definition.....	20
6.3 Profile description.....	20
6.4 Profile Component group description.....	22
6.5 Profile Component description.....	23
6.6 Description of versions.....	25
6.7 Properties.....	25
6.7.1 Introduction.....	25
6.7.1.1 Overview.....	25
6.7.1.2 Security policy.....	26
6.7.1.3 Authorization and access control policies.....	26
6.7.1.4 Privacy policies.....	26
6.7.2 Description.....	26
7 Datatype Definition Method (DtDM).....	27

7.1	Introduction.....	27
7.2	Datatype definition guidelines	27
7.2.1	Identification of datatypes	28
7.2.2	Naming convention.....	28
7.2.3	Semantics and comments	28
7.2.4	Extensions of datatypes	28
7.3	XML schema usage for datatype definitions.....	29
7.3.1	Introduction	29
7.3.2	Atomic datatypes	29
7.3.2.1	Introduction.....	29
7.3.2.2	Predefined atomic datatypes.....	29
7.3.2.3	Derived atomic datatypes.....	29
7.3.2.4	Atomic datatypes derived by restriction.....	30
7.3.2.5	Constraining facets	30
7.3.2.6	Union datatype	30
7.3.3	Composite datatypes	30
7.3.3.1	Introduction.....	30
7.3.3.2	Field and data element names.....	30
7.3.3.3	Record datatype.....	30
7.3.3.4	Selection datatype.....	31
7.3.3.5	Field.....	31
8	Process for administration of GUP/DDM	31
8.1	Introduction.....	31
8.2	Responsibility for administration of GUP/DDM common objects.....	31
8.3	Formal recognition of GUP/DDM common objects.....	31
8.4	Creation, modification, deletion of GUP/DDM common objects.....	32
8.5	3GPP XML schema namespace span	32
8.6	Versioning rules	32
8.7	Common objects storage syntax.....	33
8.7.1	Introduction	33
8.7.2	Profile Component group common object storage syntax	33
8.7.3	Profile Component common object storage syntax.....	35
8.7.4	Data element common object storage syntax.....	37
Annex A (informative):	etc.	39
Annex B (informative):	Examples of datatype definitions.....	40
Annex C (informative):	Change history	42

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 built-in datatypes and 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 a subset of the built-in datatypes and rules 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".

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 and Profile Component Group 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 Group: pre-defined set of Profile Components and/or other Profile Component Groups closely related to each other

One or more Profile Component Groups can be collected in a Profile Instance.

Profile Component Group Instance: physical representation of a Profile Component Group

To one Profile Component Group (logical) corresponds one Profile Component Group Instance which is regarded as the Master, and one or more instance copies, i.e. physical copies.

Profile Component (logical): logically an individual part of the Generic User Profile

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, Profile Component Group 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 TS 23.241

An Identity exists at the level of a Profile, a Profile Component Group, 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 or IMS Private ID)

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, Profile Component Groups, 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
PCG	Profile Component Group

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 describes those parts of the GUP Information Model (see 3GPP TS 23.240 [11]) which are relevant to Profiles (or Generic User Profiles), Profile Component Groups and Profile Components. This Information Model defines the concepts used to model the GUP data and the relationships between those concepts. This Information Model defines properties applicable to all Profile Component (Group) types and is used in the design and implementation of generic functions using, maintaining and managing GUP data. Properties specific to a Profile Component (Group) 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 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 can be viewed as a template for constructing the data description. The template (set of rules) enables 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 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, including an initial set of built-in datatypes. The DDM and the DtDM data description defines:

- 1) the syntax;
- 2) the semantics; and
- 3) one XML-representation of the data.

A subset of the XML-schema, a W3C specification, is used to define the XML-representation. To describe the semantics of the data normal text 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 a subset of XML-schema make mapping and translation to other format 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 shall be used, where appropriate, when Generic User Profiles are specified for new applications. See figure 4.1 for an explanation of the mapping principles.

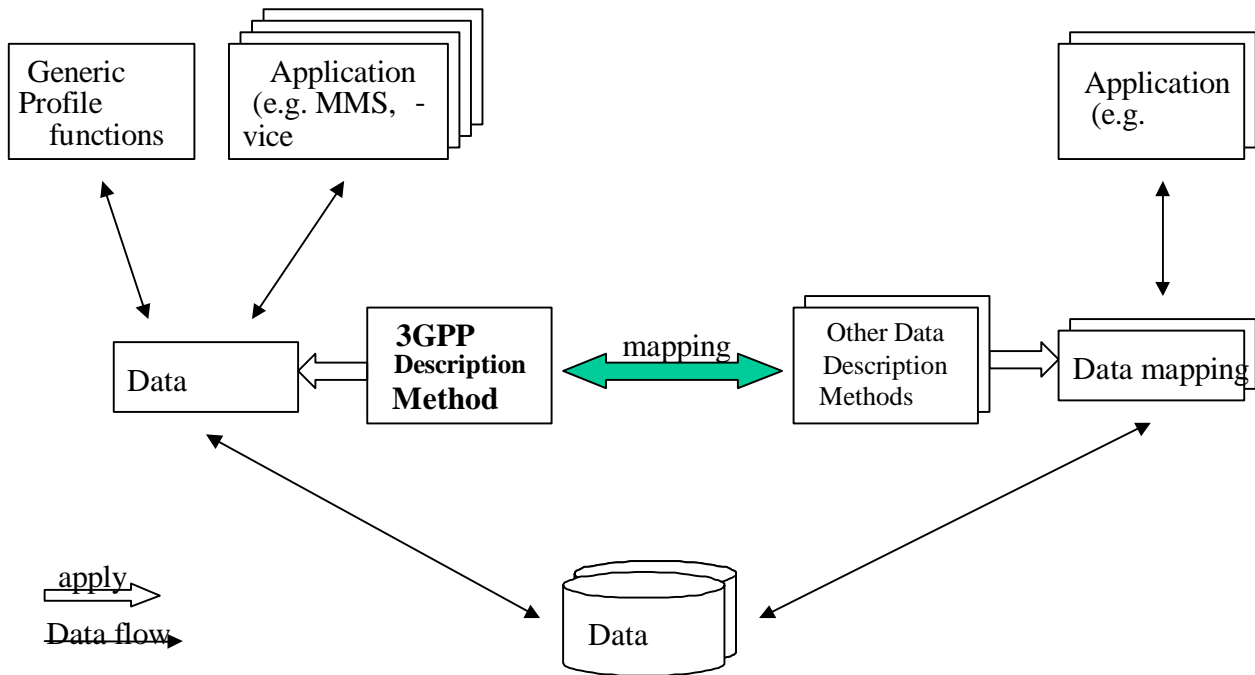


Figure 4.1: Data Description Method coexistence - the mapping principles

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.

5 Structure of Profiles

5.1 Introduction

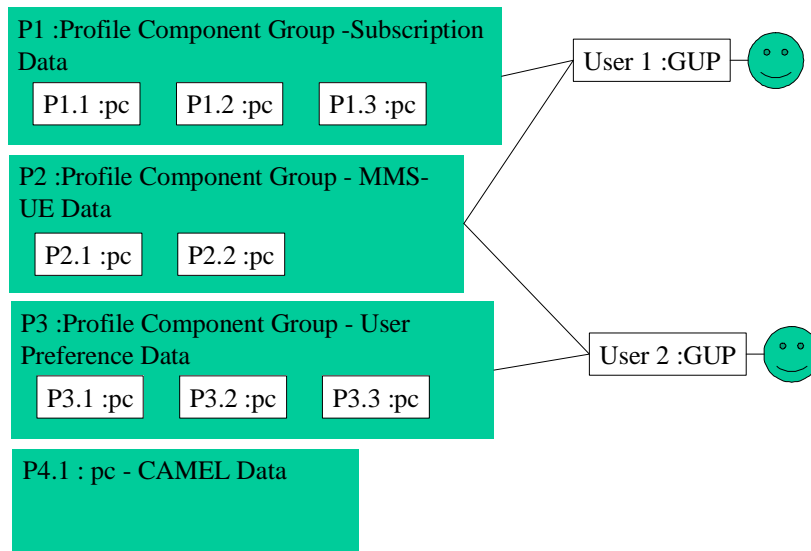
This clause defines those parts of the Information Model (see 3GPP TS 23.240 [11]) which are relevant to Profiles (or Generic User Profiles), Profile Component Groups and Profile Components.

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

Important issues described 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 and where applicable, Profile Component Groups. At least one Profile Component shall be mandatory in a Profile, however, a Profile Component Group is optional. 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 Component Groups that are not related to a certain user, e.g. see P4 in figure 5.1.1-1.



NOTE: PC: Profile Component.

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

Figure 5.1.1-2 shows the structure of the Profile and 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.

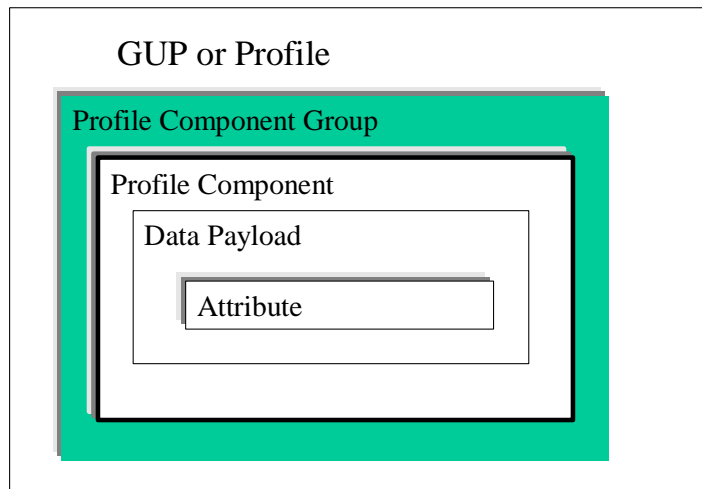


Figure 5.1.1-2: Structural relation between main parts

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 Group Instances or Profile Component Instances (data), which are possibly distributed in several storage nodes.

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

5.2 Structure of Profile Component Group, Profile Component, Profile Component Group Instance and Profile Component Instance

5.2.1 Profile Component Group and 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.

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

A Profile Component may be a part of a Profile Component Group. A Profile Component Group will contain one or more Profile Components. Profile Component Groups may be nested in a hierarchical manner, thus a Profile Component Group may consist of one or more of the following - Profile Component Group or Profile Component. A Profile Component Group is instantiated and thus has an associated Profile Component Group Instance. When there is a Profile Component Group Instance it may be the independent unit for creation, deletion, storage, and access control. Thus, the Profile Component Group and Profile Component may both be independent units for creation, deletion, storage, and access control. However, if these are declared as Common Objects, deletion shall be permissible only at the Common Object level.

A Profile Component Group is defined at a Logical level, thus it has a Logical relationship to a Profile Component defined within it. There is no direct relationship between a Profile Component Group and a Profile Component Instance - a Profile Component Instance has a relation to the Profile Component from which it is instantiated.

5.2.2 Profile Component Group Instance and Profile Component Instance

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

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

5.2.2.1 Data Payload

A Data Payload is the Profile data contained in a Profile Component Group Instance and 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, Profile Component Group Description and Profile Component Description

5.3.1 Profile Description and Profile Component Group 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 Group Descriptions and a Profile Component Group Description consists of Profile Component Descriptions, in an analogous manner to a Profile consisting of Profile Component Groups and a Profile Component Group consisting of 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.

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

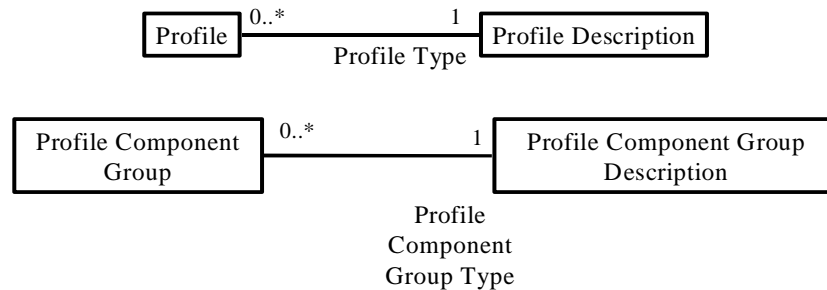


Figure 5.3.1-1: Profile Description and Profile Component Group Description

The purpose of a Profile and a Profile Component Group 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.

The definition of a Profile Component Group Type is kept together in a Profile Component Group 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.

A Profile Component Group Description contains:

- **Semantics:** defines the meaning of the Profile Component Group.
- **Profile Component Group Type:** defines the Type of the Profile Component Group.
- **Payload Datatype:** is a reference to a composite Datatype that describes the content of the Profile Component Group's Data Payload.

- **Profile Component Group Property reference:** is referencing a Common Properties containing data controlling the usage and handling of the Profile Component Group.

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

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 the Primary Profile Component Group Identity and Primary Profile Identities are described in the next clause.

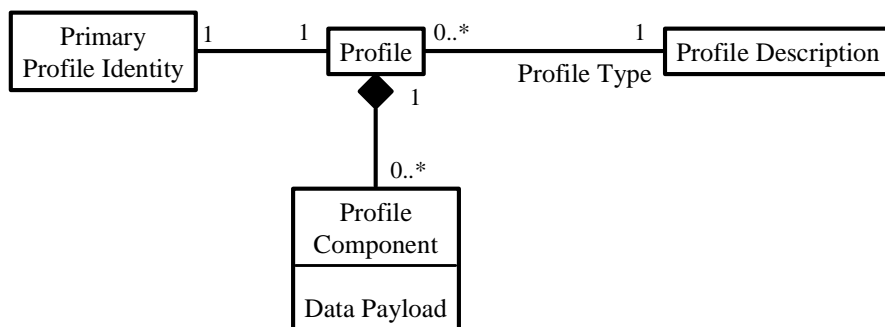


Figure 5.3.1-2: UML-diagram, Profile

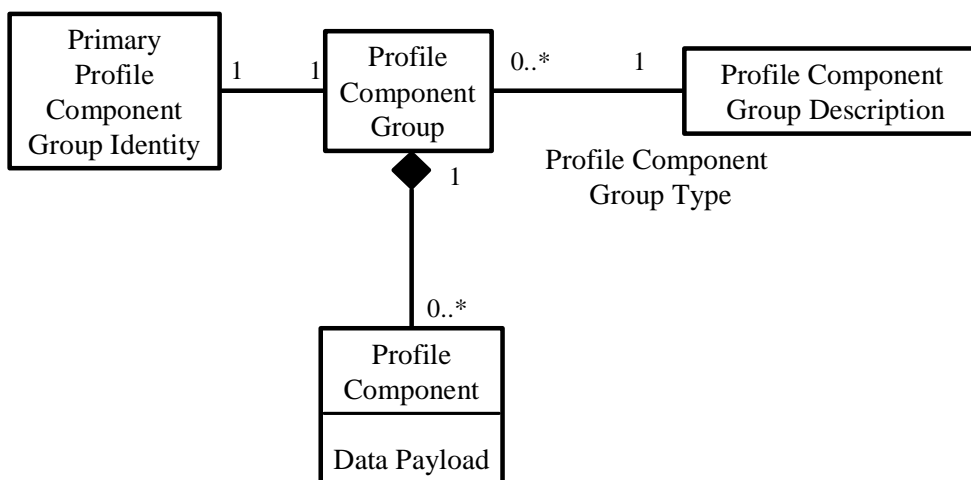


Figure 5.3.1-3: UML-diagram, Profile Component Group Description

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.

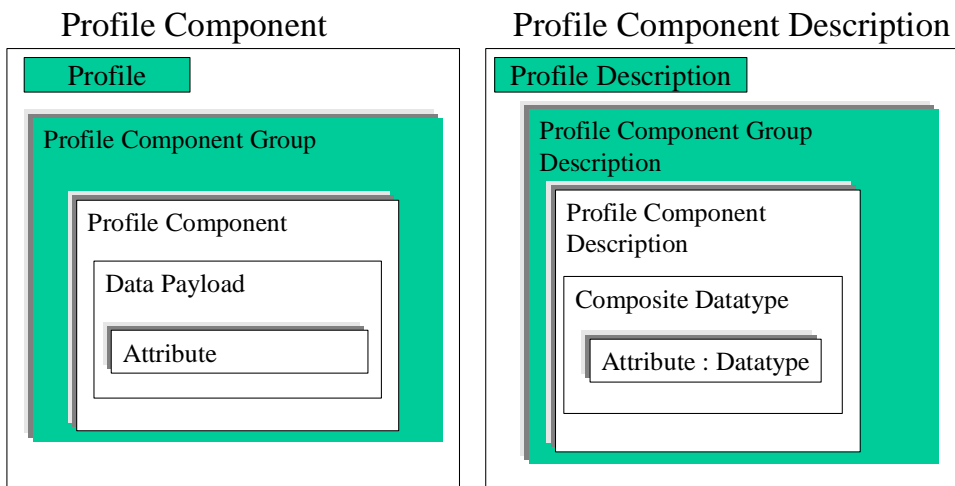


Figure 5.3.2-1: Relationship between Profile Component and Profile Component Description

5.4 Structure of Identities

5.4.1 Primary Profile Identity

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

A **Profile** Instance associates some data with a **Primary Profile Identity**. The type of a Profile, **Profile Type**, is defined by one **Profile Description**.

A **Profile** Instance is always associated with one **Primary Profile Identity**. The **Primary Profile Identity** is associated with one **Profile** Instance.

The data in the **Data Payload** is the data that the Profile Instance associates with the Primary Profile Identity.

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

5.4.2 Profile Component Group Identity and Profile Component Identity

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

A **Profile Component Group** Instance associates some data with a **Primary Profile Component Group Identity**. The type of a Profile Component Group, **Profile Component Group Type**, is defined by one **Profile Component Group Description**.

A **Profile Component Group** Instance is always associated with one **Primary Profile Component Group Identity**. The **Primary Profile Component Group Identity** is associated with one **Profile Component Group** Instance.

The data in the **Data Payload** is the data that the Profile Component Group Instance associates with the Primary Profile Component Group Identity.

Profile Component (Group) is meant to be read herein as Profile Component and as Profile Component Group.

A Profile Component (Group) has a unique Identity within the Generic User Profile.

A part of the Profile Component (Group) Instance name is the Profile Instance name. By using the name of the Profile Component (Group) Instances all Profile Component (Group) 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.

The names of all the Profile Component (Group) Instances belonging to a Profile Instance contain the Primary Profile Identity.

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 (Group) 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 and/or IMS Private ID).

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

- 1) User Identity is a part of the Primary 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 Primary 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 Id must be known when the Profile is created, because the Primary 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 Primary 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 (Group) Instance.

If a Profile Component (Group) Instance needs to be shared by more Users, references can be used. A referenced Profile Component (Group) Instance can be shared by many Users and can be a part of many Generic User Profiles. The referenced Profile Component (Group) 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 (Group) Instances are built into a phone during manufacturing. The Profile Component (Group) Instances must be given a Primary 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 (Group) 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 (Group) Instances are:

- The Profile Component (Group) Instances in a mobile phone that is connected to a user by a removable (U)SIM.
- The Profile Component (Group) Instances in an accessory when the accessory is connected to a mobile phone (with (U)SIM-card).
- The Profile Component (Group) Instances in a computer connected to the mobile phone.
- The Profile Component (Group) 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.

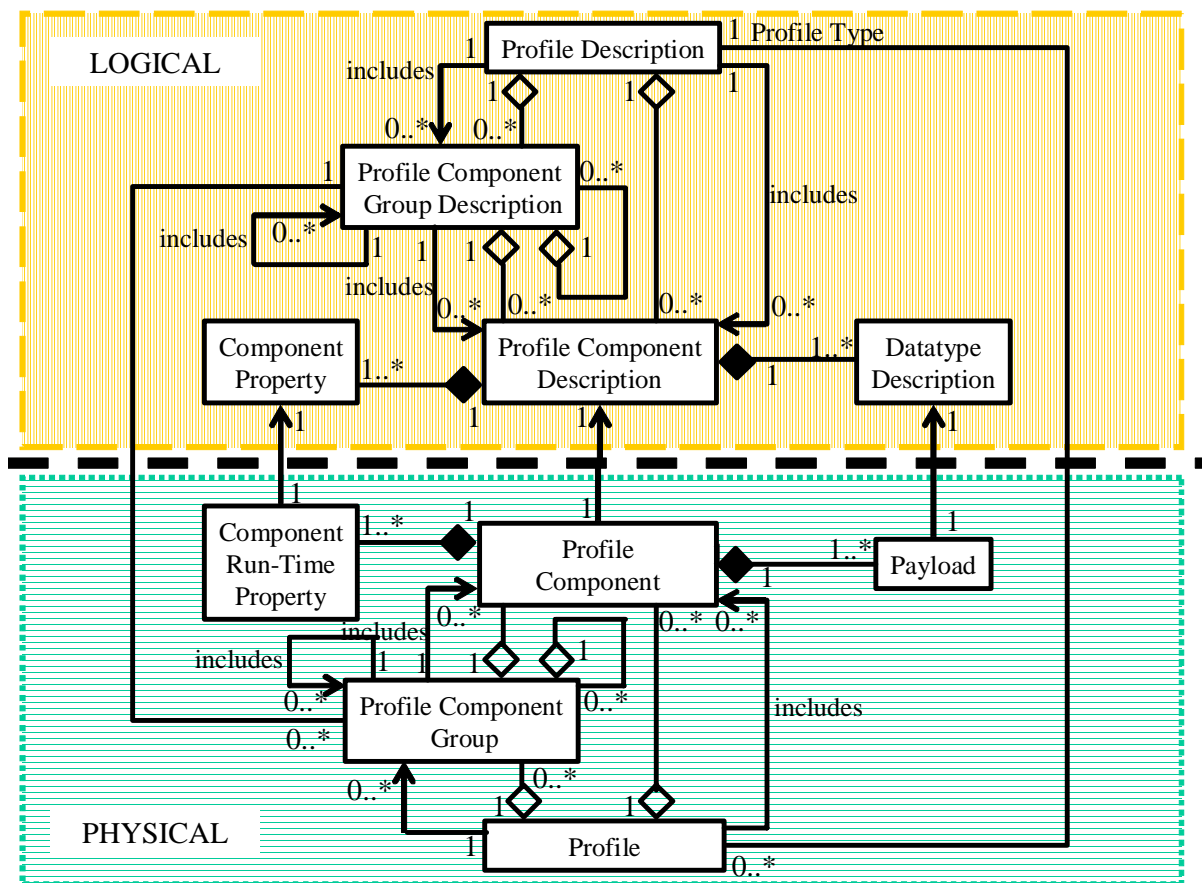


Figure 5.6-1: UML-diagram, details of Profile High Level Structure and constituent parts

5.6.2 Profile Component Payload

5.6.3 Profile Component Run-Time Properties

Ed. Note: add master copy indicator, synchronized copy indicator, working copy indicator.

5.6.4 Profile Component Payload internal identification

None.

5.7 Distributed Profile storage

5.7.1 Introduction

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 5.7.1-1 shows an example possible storage distribution of Profile Component Instances.

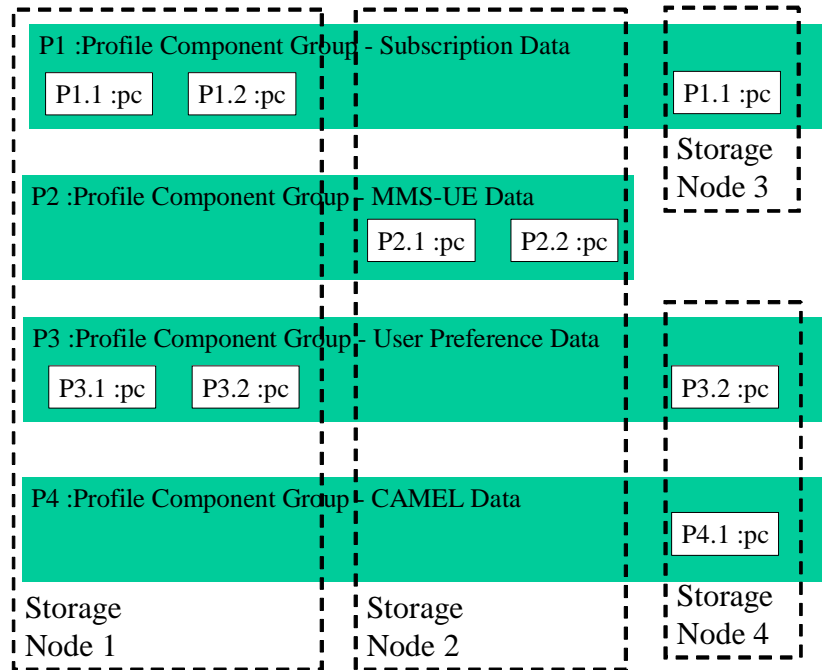


Figure 5.7.1-1: An example of Profile storage distribution

5.7.2 Storage principles

Figure 5.7.2-1 shows a data storage model.

There is a need to distinguish between two types of storage (in a run-time system): Data Store and Description Store. A system contains normally many Data Stores and Description Stores.

Profile Components and Run-Time Component Properties are stored in Data Stores. The Profile Components belonging to a Profile can be stored in different Data Stores and different locations. A Profile Component and the referenced Run-Time Component Property is stored in the same Data Store.

Profile Descriptions are stored in Description Stores. Profiles and the corresponding Profile Description can be stored in different locations.

A Profile Component is associated with one Primary Profile Id and one Component Description and it must know its:

- Profile Instance Identity.
- Profile Component Identity.
- Profile Description Identity.

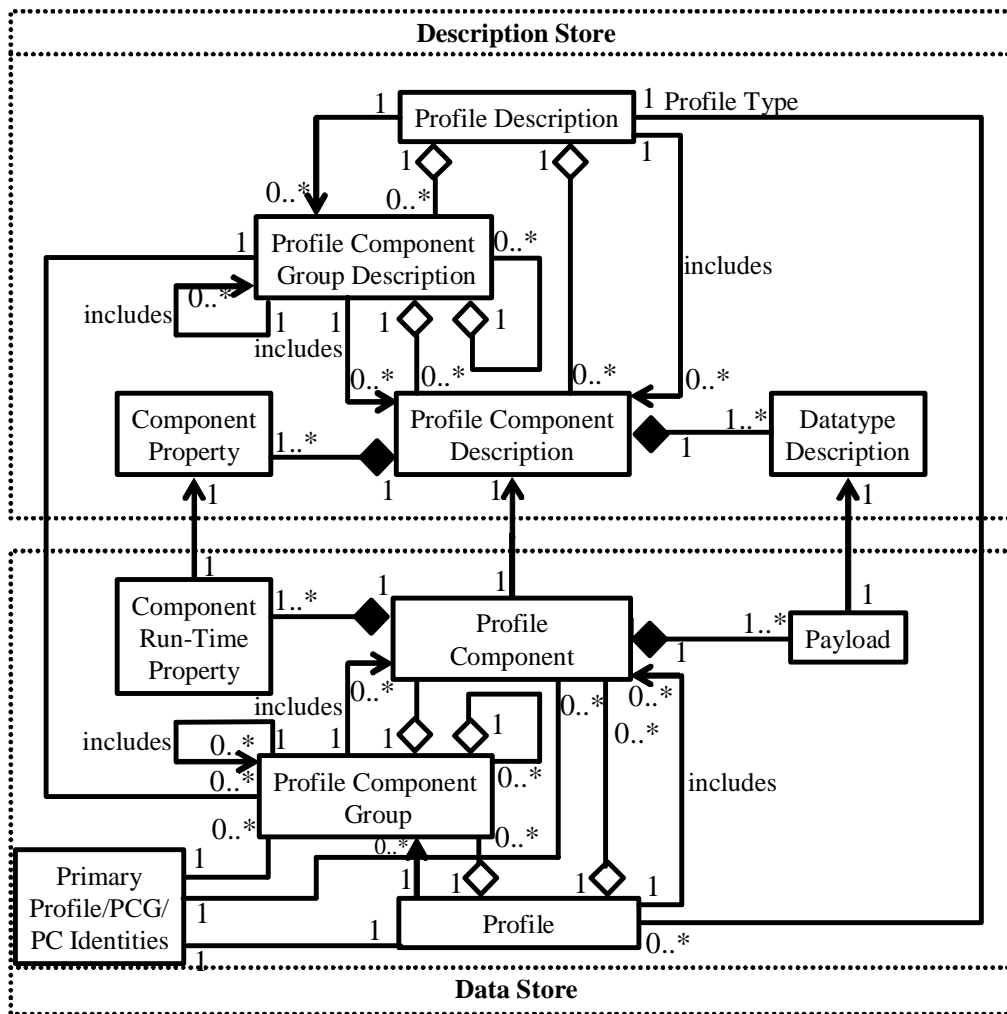


Figure 5.7.2-1: UML-diagram, distributed storage

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 a restricted subset of 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 1 that describes Structures, and XML Schema Part 2 that describes Datatypes. The functionality of the XML Schema as defined by W3C is more than what is needed by DDM. The needed restriction is done through derivation of complex types, by guidelines, and templates.

The Data Description Method can be viewed as a template for constructing the data description. The template (set of rules) enables 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.

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).

Only one namespace shall be defined for all of 3GPP GUP and DDM. This namespace shall also include GUP/DDM entities within designated 3GPP-associated servers and designated 3rd party (value added) servers. The designation of such servers shall be maintained in clause 8.

3GPP namespace declaration:

```
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.3GPP.org/GUP/"
  xmlns:GUP="http://www.3GPP.org/GUP/" >
```

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 may contain Datatype definitions and or include one or more 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.

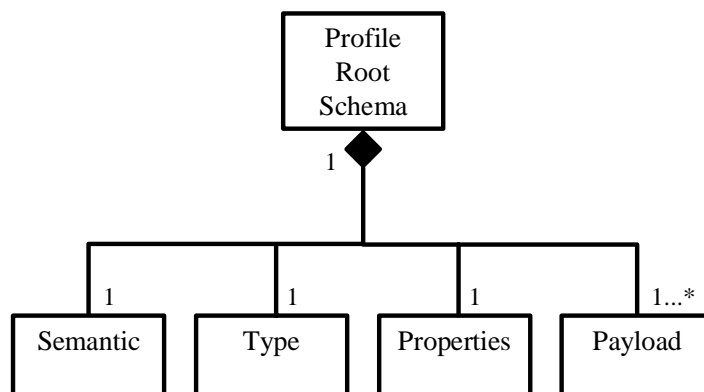


Figure 6.3-1: Profile Root Schema - UML

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" xmlns:GUP="http://www.3GPP.org/GUP"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Profile: P name
      Profile consists of following PCGs and PCs:
        list of PC(G)(s)
      Version x.y
      History: x.y created on yyyy-mm-dd
      Keywords:
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="http://www.3GPP.org/GUP/Properties.xsd"/>
  <!-- include all PC(G)s, which are used (referenced) in this schema or
  whose data elements are used (referenced)
  - the include of one PC_Name is shown below as an example -->
  <xs:include schemaLocation="http://www.3GPP.org/GUP/PC_Name.xsd"/>
  <!-- root element (P) -->
  <xs:element name="P_Name" type="GUP:PNameType"/>
  <!-- no other declarations necessary as all elements are included from other PCGs and PCs -->
  <xs:complexType name="PNameType">
    <xs:sequence>
      <xs:element ref="GUP:Properties" minOccurs="0"/>
      <xs:element ref="GUP:PCG_1"/>
      <xs:element ref="GUP:PCG_2"/>
      <xs:element ref="GUP:PC_1"/>
      <xs:element ref="GUP: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

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

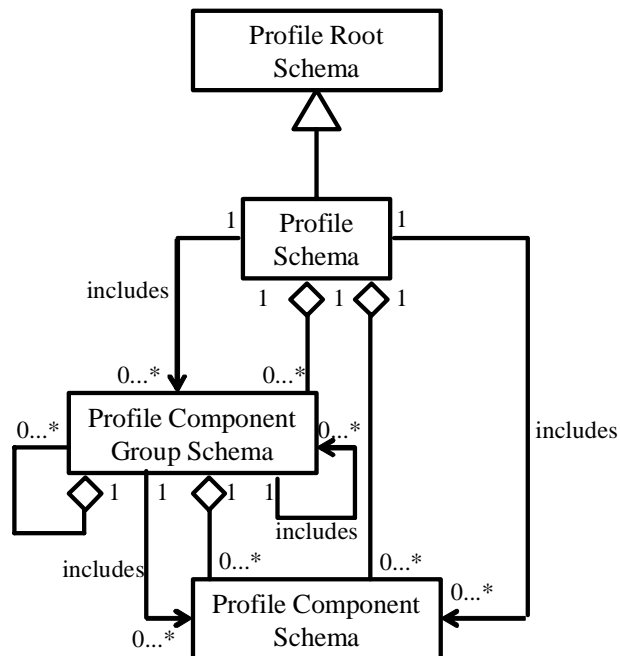


Figure 6.3-3: Creating Profile Description schema using the Profile Root schema - UML

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.

6.4 Profile Component Group Description

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

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

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

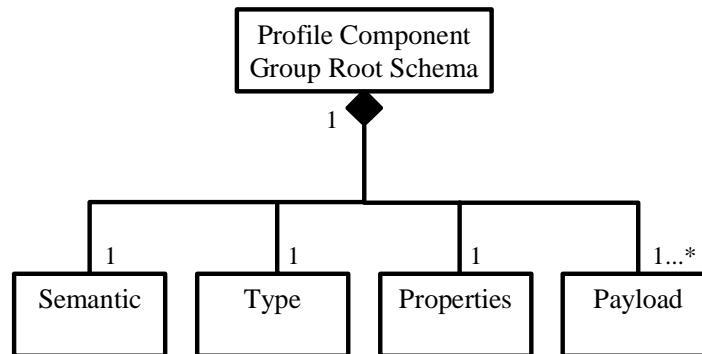


Figure 6.4-1: Profile Component Group Root Schema - UML

The Profile Component Group 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" xmlns:GUP="http://www.3GPP.org/GUP"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Profile ComponentGroup: PCG name
      Parents: PCG name(s), P name
      PCG consists of following PCGs and PCs:
        list of PC(G)(s)
      Version x.y
      History: x.y created on yyyy-mm-dd
      Keywords:
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="http://www.3GPP.org/GUP/Properties.xsd"/>
  <!-- include all PC(G)s, which are used (referenced) in this schema or
  whose data elements are used (referenced)
  - the include of one PC_Name is shown below as an example -->
  <xs:include schemaLocation="http://www.3GPP.org/GUP/PC_Name.xsd"/>
  <!-- root element (PCG) -->
  <xs:element name="PCG_Name" type="GUP:PCGNameType"/>
  <!-- no other declarations necessary as all elements are included from
  other PCGs and PCs -->
  <xs:complexType name="PCGNameType">
    <xs:sequence>
      <xs:element ref="GUP:Properties" minOccurs="0"/>
      <xs:element ref="GUP:PCG_1"/>
      <xs:element ref="GUP:PCG_2"/>
      <xs:element ref="GUP:PC_1"/>
      <xs:element ref="GUP:PC_2"/>
      <!-- etc. -->
    </xs:sequence>
    <xs:attribute name="identity_PCG" type="xs:string"/>
  </xs:complexType>
</xs:schema>

```

Figure 6.4-2: Profile Component Group Root Schema - XML

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

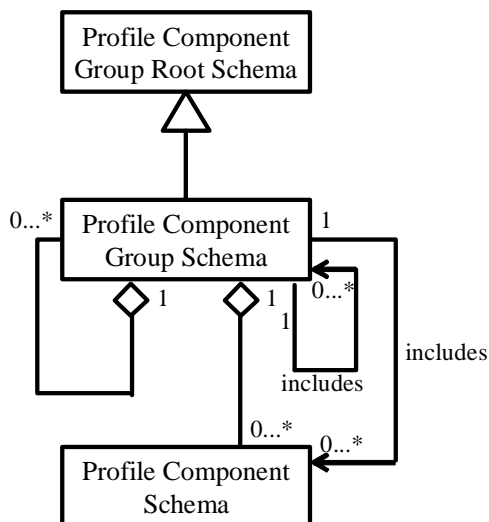


Figure 6.4-3: Creating Profile Component Group Description Schema using the Profile Component Group Root Schema - UML

The process of creating Profile Component Group Description Schema described in figure 6.4-3 is as follows. First the Profile Component Group Description Schema for the use is created based on the XML Schema for the Profile Component Group 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.

An example of the use of the Profile Component Group Root Schema to create a Profile Component Group Description XML Schema for CAMEL data is shown in clause 8.7.2.

6.5 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 one or more Datatype Definition Schemas containing Datatype definitions.

Figure 6.5.1 shows the UML diagram for the Profile Component Root Schema. The blocks are defined in clause 5.

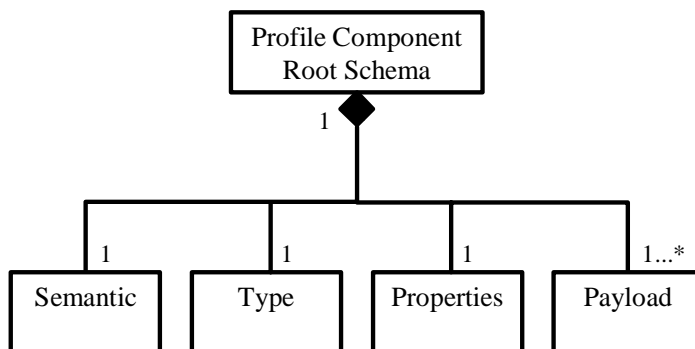


Figure 6.5-1: Profile Component Root Schema - UML

The Profile Component Root Schema is defined by the XML Schema, figure 6.5-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3GPP.org/GUP" xmlns:GUP="http://www.3GPP.org/GUP"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Profile Component: PC name
      Parents: PCG name(s), P name
      PC consists of following DEs:
        list of DEs
      Version x.y
      History: x.y created on yyyy-mm-dd
      Keywords:
    </xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="http://www.3GPP.org/GUP/Properties.xsd"/>
  <!-- include all PC(G)s, which are used (referenced) in this schema or
  whose data elements are used (referenced)
  - the include of one PC_Name is shown below as an example -->
  <xs:include schemaLocation="http://www.3GPP.org/GUP/PC_Name.xsd"/>
  <!-- root element (PC) -->
  <xs:element name="PC_Name" type="GUP:PCNameType"/>
  <!-- list of all referenced data elements follows -->
  <xs:element name="Data_Element_1" type="GUP:DataType1"/>
  <xs:element name="Data_Element_2" type="GUP:DataType2"/>
  <!-- etc. -->
  <!-- all other elements which are referenced in the type definition of
  the PC and are not included from other schemas are declared above -->
  <xs:complexType name="PCNameType">
    <xs:sequence>
      <xs:element ref="GUP:Properties" minOccurs="0"/>
      <xs:element ref="GUP:Data_Element_1"/>
      <xs:element ref="GUP:Data_Element_2"/>
      <!-- etc. -->
    </xs:sequence>
    <xs:attribute name="identity_PC" type="xs:string"/>
  </xs:complexType>
  <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. -->
  <!-- all other datatypes which are declared are defined above -->
</xs:schema>
```

Figure 6.5-2: 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 6.5-3.

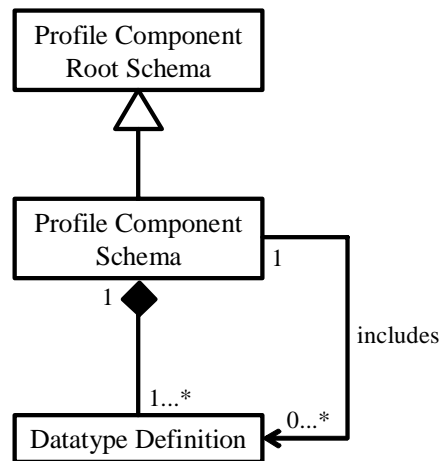


Figure 6.5-3: Creating Profile Component Description Schema using the Profile Component Root Schema - UML

The process of creating Profile Component Description Schema described in figure 6.5-3 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 8.7.3.

An Example of a Data Element Description XML Schema for CAMEL data is shown in clause 8.7.4.

6.6 Description of Versions

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

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

Version numbers shall 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 shall be incremented upon version change.

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

6.7 Properties

6.7.1 Introduction

6.7.1.1 Overview

The common data shall contain datatypes 3GPP TS 22.240 [10] and 3GPP TS 23.240 [11] for at least:

- Private IDs (e.g. Subscriber identification).
- Public IDs.
- 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).

The common data may contain datatypes 3GPP TS 22.240 [10] and 3GPP TS 23.240 [11] for at least:

- Subscription data (e.g. Credentials, Authorization assertion).
- Terminal Management.
- Subscription Check (by 3rd party).
- Service Personalization data.
- Capabilities data of terminals.

Also, the common data may contain datatypes for at least:

- Preferred access.
- User-level blacklist.

6.7.1.2 Security Policy

Editor's Note: Support for Security functions. Authorization control policies and rules data described here. Align with SA3 Security specs

6.7.1.3 Authorization and Access Control Policies

Editor's Note: Support for Security functions. Authorization control policies and rules described here. Align with SA3 Security specs.

6.7.1.4 Privacy Policies

Editor's Note: Privacy control policies and rules described here.

6.7.2 Description

Figure 6.7.2-1 shows the XML Schema for Properties. Note that this is not a Root Schema like in the previous clauses which shall be used to derive other schemas. This is the schema which must be included to the highest hierarchy level in a Schema. E.g. if a PCG Schema is derived from the PCG Root Schema it must include the Properties Schema. All included PCGs and PCs may include the Properties Schema if necessary.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.3gpp.org/GUP" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:GUP="http://www.3gpp.org/GUP">
  <xs:annotation>
    <xs:documentation xml:lang="en">

      Properties Schema
      Version 1.0
      Created 2003-10-23

      Properties gives information about the owner of the GUP and the
      handling of the data.
      Properties are included in every schema (P, PCG, PC). At least one
      complete properties element must be present at the highest level of
      the instance document.

      Data included:
      Private Identities - IMSI and IMPI
      Public Identities - e.g. MSISDN, IMPUs, names etc.
      Default access rights
      Default privacy control data
      Error data
      date and time
      service state
      Master copy indicator
      Synchronized copy indicator
      Working copy indicator

    </xs:documentation>
  </xs:annotation>
  <!-- list of all referenced data elements follows -->
  <xs:element name="Properties" type="GUP:PropertiesType"/>
  <xs:element name="Private_Id" type="xs:string"/>
  <!-- etc. -->
  <xs:complexType name="PropertiesType">
    <xs:sequence>
      <xs:element ref="GUP:Private_Id"/>
      <!-- references to all necessary properties must be included,
      FFS -->
      <!-- etc. -->
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Figure 6.7.2-1: Properties XML Schema

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, including an initial set of built-in data types. 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 shall be defined as an XML element of a suitable type.
- 2) XML attributes shall 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 shall not have both a value and other XML elements as subelements.
- 4) The type definitions provided by the XML schema shall be used.

7.2.1 Identification of Datatypes

- 1) A Naming Convention shall be followed which shall 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 shall apply for:
 - a) the datatypes used to generate the Profile Components (and Profile Component Groups); and for
 - b) the Profile Components (and Profile Component Groups) themselves.

For a), easily understood generic names shall 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 shall 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 shall be defined by the complexType datatype with the name OrigCamelSubscrInfo.

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

3GPP/GUP/Profile_name/Profile_Component_Group_name/.../Profile_Component_name/Data_Element_name/attribute_name/)

Naming Convention = 3GPP/GUP/P/PCG/PCG/.../PC/DE/AT

Example of use of the naming convention:

3GPP/GUP/P_GUP/PCG_Subscription/PCG_CAMEL/PC_O_CSI/DE_TDP_List_originating

7.2.3 Semantics and Comments

None.

7.2.4 Extensions of Datatypes

None.

7.3 XML Schema Usage for Datatype Definitions

7.3.1 Introduction

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.

7.3.2 Atomic Datatypes

7.3.2.1 Introduction

This clause describes list of built-in datatypes and derivation rules for new datatypes.

Atomic datatypes are those having values, which are regarded by as being indivisible or not further decomposable.

There are predefined atomic datatypes and derived atomic datatypes. Derived datatypes can be defined by restricting a predefined atomic datatype or by defining a union datatype.

7.3.2.2 Predefined Atomic Datatypes

Predefined atomic datatypes can only be added by revisions to the present document.

The XML-schema primitive datatypes are: string, boolean, decimal, float, double, duration, dateTime, time, date, gYearMonth, gYear, gMonthDay, gDay, gMonth, hexBinary, base64Binary, anyURI, QName, and NOTATION.

The XML-schema primitive derived datatypes are: normalizedString, token, language, NMTOKEN, NMTOKENS, Name, NCName, ID, IDREF, IDREFS, ENTITY, ENTITIES, integer, nonPositiveInteger, negativeInteger, long, int, short, byte, nonNegativeInteger, unsignedLong, unsignedInt, unsignedShort, unsignedByte, positiveInteger.

The predefined atomic datatypes are a subset of the XML-schema primitive datatypes. The datatypes from the XML-schema which are excluded are below in comments "{* ... *}".

```
{predefinedAtomicDatatype} ::=
string | boolean
    { * | decimal | float | double * }
| duration | dateTime | time | date
    { * | gYearMonth | gYear | gMonthDay | gDay | gMonth * }
    { * | hexBinary | base64Binary * }
| anyURI
    { * | QName | NOTATION * }
| normalizedString
    { * | token * }
| language
    { * | NMTOKEN | NMTOKENS | Name | NCName * }
| ID | IDREF
    { * | IDREFS | ENTITY | ENTITIES * }
    { * | integer | nonPositiveInteger | negativeInteger | long * }
| int | short | byte
    { * | nonNegativeInteger | unsignedLong * }
| unsignedInt | unsignedShort | unsignedByte
    { * | positiveInteger * }
```

7.3.2.3 Derived Atomic Datatypes

Derived atomic datatypes can be defined by restricting a predefined atomic datatype or by defining a union datatype.

7.3.2.4 Atomic Datatypes Derived by Restriction

A datatype is said to be derived by restriction from another datatype when values for zero or more constraining facets are specified that serve to constrain its value space and/or its lexical space to a subset of those of its base type. A constraining facet is an optional property that can be applied to a datatype to constrain its value space.

```
<xs:simpleType name="{datatypeName}">
  <xs:restriction base="{datatypeName}">
    <xs:{facet} value="{value}"/>*
  </xs:restriction>
</xs:simpleType>
```

7.3.2.5 Constraining Facets

Constraining Facets in XML-schema are: length, minLength, maxLength, pattern, enumeration, whiteSpace, maxInclusive, maxExclusive, minExclusive, minInclusive, totalDigits, and fractionDigits.

```
{facet} ::= minExclusive | minInclusive | maxExclusive|maxInclusive
          | totalDigits | fractionDigits | length | minLength
          | maxLength | enumeration | pattern
```

7.3.2.6 Union Datatype

A union type enables an attribute value to be one instance of one type draw from the union of multiple atomic.

Union datatypes are those whose value spaces and lexical spaces are the union of the value spaces and lexical spaces of one or more other datatypes.

The datatypes that participate in the definition of a union datatype are called member types of that union datatype.

```
<xs:simpleType name="{datatypeName}">
  <xs:union memberTypes="{simpleDatatypeName}*" />
</xs:simpleType>
```

7.3.3 Composite Datatypes

7.3.3.1 Introduction

A Composite Datatype contains a number of name items each with a defined Datatype. A field can be simple or a vector. A simple field can contain one element and a vector field a number of elements of the specified Datatype.

7.3.3.2 Field and Data Element names

A Composite Datatype has a number of fields each with a local name {fieldName}, which must be unique within the datatype.

An instance of a Composite Datatype is a composite data element containing a number of sub elements corresponding to the fields. The local name of a sub element is a {elementName}. The {elementPath} is used to identify a sub element or a sub-sub element and so on.

7.3.3.3 Record Datatype

A record datatype contains a number of named items called fields each with a defined datatype. The field names must be unique with a record datatype. The datatype of a field can be any atomic data type or composite datatype.

```
<xs:complexType name="{datatypeName}">
  <xs:sequence>
    {field}*
  </xs:sequence>
</xs:complexType>
```

7.3.3.4 Selection Datatype

A selection datatype defines a number of named items called fields each with a defined datatype. For a selection datatype only one of its fields can be stored at the same time.

The field names must be unique within a selection datatype. The datatype of a field can be any atomic data type or composite datatype.

```
<xs:complexType name="{datatypeName}">
  <xs:choice>
    {field}*
  </xs:choice>
</xs:complexType>
```

7.3.3.5 Field

```
{field} ::= {fieldSimple} | {fieldSimpleOptional} | {fieldVector}
Simple field:
  {fieldSimple} ::=
    <xs:element name="{elementName}" type="{datatypeName}"
    />*
Simple optional field:
  {fieldSimpleOptional} ::=
    <xs:element name="{elementName}" type="{datatypeName}"
    minOccurs="0" maxOccurs="1"
    />*
Vector field:
  {fieldVector} ::=
    <xs:element name="{elementName}" type="{datatypeName}"
    minOccurs="{min}" maxOccurs="{max}"
    />*
```

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. GUP/DDM Common Objects data are stored in 3GPP TS 24.241 [12]. This clause also specifies the span of the XML Schema namespace defined in clause 6.2.

8.2 Responsibility for administration of GUP/DDM Common Objects

3GPP TSG-SA will assign a Working Group within 3GPP for this task. This Working Group is referred to in this clause as the Responsible WG. The Responsible WG shall also be the Owner and Editor of 3GPP TS 24.241 [12] which shall be used for this purpose.

8.3 Formal Recognition of GUP/DDM Common Objects

Formal recognition of the new Common Objects or other constructs shall 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.

All 3GPP Working Groups shall submit approved Change Requests for creation, storage, modification, deletion, etc. of their GUP/DDM Common Objects to the Responsible WG in accordance with the rules set out in this clause. These Change Requests shall use their own namespaces as interim (shall not use the 3GPP namespace in clause 6.2), until they have been approved by the Responsible WG. As part of this approval, the interim namespace shall be updated to the 3GPP Namespace defined in clause 6.2 by the Responsible WG.

All 3GPP Working Groups shall 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 shall 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.

The Responsible WG shall administer the GUP/DDM Common Objects in 3GPP TS 24.241 [12] for all 3GPP 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 shall 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 shall define and follow a procedure called "Prior Notification of planned modification / deletion of a Common Object". The procedure shall require wide circulation of planned modifications/deletions and shall as a minimum include all using WGs shown in the Common Object information.

The Responsible WG shall 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.

Editor's Note: Text will be included to describe an example of such a procedure.

8.5 3GPP XML Schema namespace span

This clause specifies the entities which are included within the 3GPP XML Schema namespace declared in clause 6.2.

Entities included in the clause 6.2 namespace:

1. 3GPP GUP:
 - a. All entities specified under GUP in 3GPP Specifications.
 - b. All entities in 3GPP TS 24.241 [12].
2. OAM&P servers:
 - a. [td].
 - b. [td].
3. 3rd Party servers:
 - a. [td].
 - b. [td].

8.6 Versioning rules

This clause describes how version numbers shall be administered.

The Version number is specified in clause 6 as:

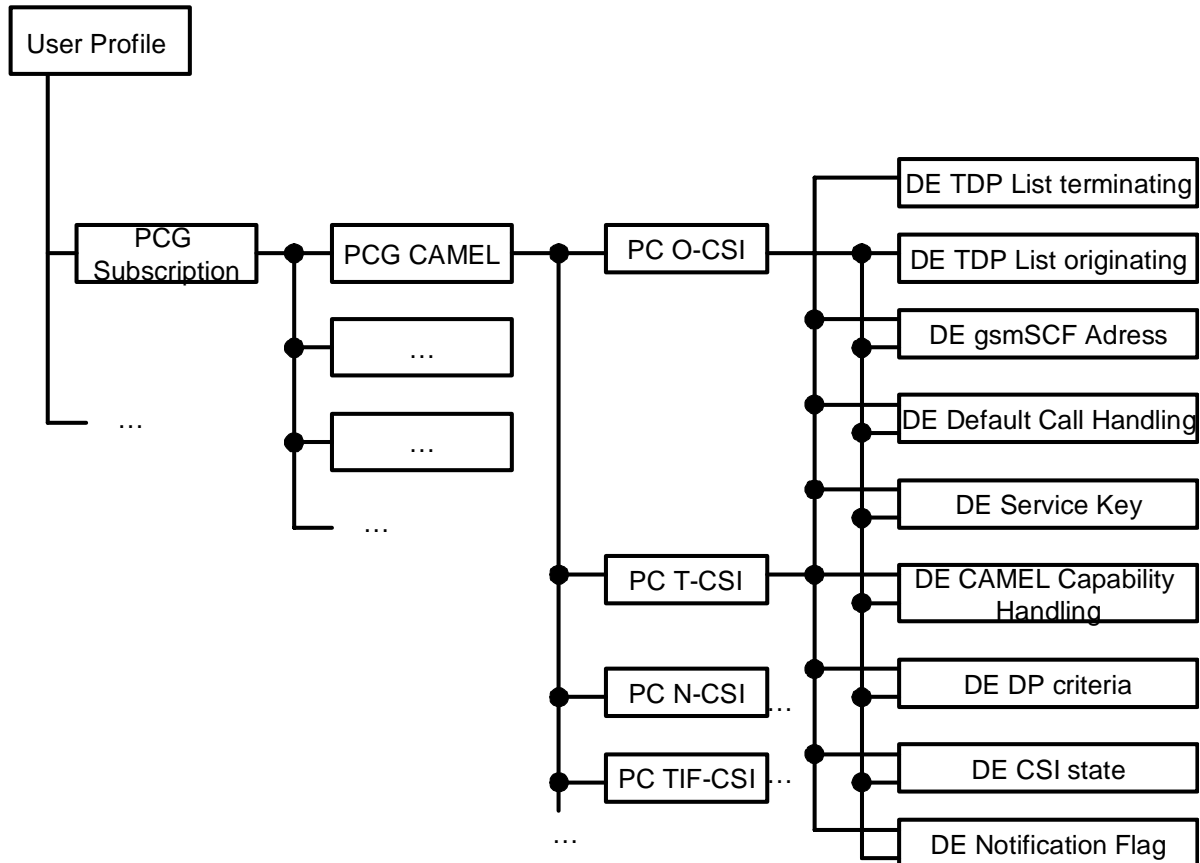
- Version x, y
 - y can be incremented by the developing WG or the Responsible WG in coordination.
 - x can be incremented by the Responsible WG only.

8.7 Common Objects storage Syntax

8.7.1 Introduction

Common Objects to be stored in 3GPP TS 24.241 [12] shall 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 8.7.1-1.

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].



PCG Profile Component Group.
 PC Profile Component.
 DE Data Element

Figure 8.7.1-1: Example structure based on CAMEL subscription data

8.7.2 Profile Component Group Common Object storage Syntax

Editor's Note: Text will be included to describe this example.

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

Table 8.7.2-1: Syntax of a Profile Component Group Common Object in 3GPP TS 24.241 [12]

Defining Group	< group >
Date Created (original)	< date >
Defining Application	< application >
Type of item	PROFILE COMPONENT GROUP
Name of Item (being defined)	< name >
Datatype	< type name >
Parent tree of original use (definition)	< names of PCG, 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)	< item 1 > < item 2 > ... < item n >

NOTE: The following are definitions of terms in table 8.7.2-1.

Table 8.7.2-2: Example Syntax of a Profile Component group 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 GROUP
Name of Item (being defined)	CAMEL
Datatype	CamelType
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)	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" xmlns:GUP="http://www.3gpp.org/GUP"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Profile Component Group: CAMEL
      Parents: PCG Subscription, P User
      PCG 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/Properties.xsd"/>
  <xs:include schemaLocation="http://www.3gpp.org/GUP/O-CSI.xsd"/>
  <xs:include schemaLocation="http://www.3gpp.org/GUP/T-CSI.xsd"/>
  <!-- root element (PCG) -->
  <xs:element name="CAMEL" type="GUP:CamelType"/>
  <xs:complexType name="CamelType">
    <xs:sequence>
      <xs:element ref="GUP:Properties" minOccurs="0"/>
      <xs:element ref="GUP:O-CSI"/>
      <xs:element ref="GUP:T-CSI"/>
      <!--xs:element ref="GUP:x-CSI"-->
      <!-- ... -->
      <!-- All elements except Properties must occur exactly once
      (default values of minOccurs and maxOccur="1") and are
      referenced to global elements.-->
    </xs:sequence>
    <xs:attribute name="identityCAMEL" type="xs:string"/>
  </xs:complexType>
</xs:schema>

```

Figure 8.7.2-1: Example XML Schema of a Profile Component Group for CAMEL parameter in 3GPP TS 24.241 [12]

8.7.3 Profile Component Common Object storage Syntax

Editor's Note: Text will be included to describe this example.

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

Table 8.7.3-1: 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 >
Parent tree of original use (definition)	< names of PCG(s), 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) - Data Element list	< DE 1 > < DE 2 > ... < DE n >

Table 8.7.3-2: 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
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) - 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" xmlns:GUP="http://www.3GPP.org/GUP"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Profile Component: O-CSI
      Parents: PCG CAMEL, PCG 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>
  <xs:include schemaLocation="http://www.3GPP.org/GUP/Properties.xsd"/>
  <!-- root element (PC) -->
  <xs:element name="O-CSI" type="GUP:OrigCamelSubscrInfo"/>
  <!-- list of all referenced data elements follows -->
  <xs:element name="TDP_List_originating" type="GUP:OrigTDPList"/>
  <xs:element name="gsmSCF_Adress" type="GUP: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="GUP:Properties" minOccurs="0"/>
      <xs:element ref="GUP:TDP_List_originating"/>
      <xs:element ref="GUP:gsmSCF_Adress"/>
      <xs:element ref="GUP:Default_Call_Handling"/>
      <xs:element ref="GUP:Service_Key"/>
      <xs:element ref="GUP:CAMEL_Capability_Handling"/>
      <xs:element ref="GUP:DP_criteria"/>
      <xs:element ref="GUP:Notification_Flag"/>
      <!-- All elements except Properties must occur exactly once (default values of
      minOccurs and maxOccurs="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 8.7.3-1: Example XML Schema of a Profile Component Common Object for O-CSI in 3GPP TS 24.241 [12]

8.7.4 Data element Common Object storage Syntax

Editor's Note: Text will be included to describe this example.

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

Table 8.7.4-1: 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, PCG(s), P>
Common Object consists of (INCLUDES 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 8.7.4-2: Example Syntax of a Datatype Common Object 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 (INCLUDES 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 8.7.4-1: Example datatype definition for CAMEL TDP List in 3GPP TS 24.241 [12]

Annex A (informative): etc.

Editor's Note: Annexes A, etc., will contain Examples of structure of DDM for subsets of some use cases.

Annex B (informative): Examples of Datatype Definitions

Following examples will illustrate the description of a number of datatype definitions using DtDM.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xs:simpleType name="CoordinateX">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="60"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="CoordinateY">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="30"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="Point">
    <xs:sequence>
      <xs:element name="x" type="CoordinateX">
      </xs:element>
      <xs:element name="y" type="CoordinateY">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Line">
    <xs:sequence>
      <xs:element name="startPoint" type="Point">
      </xs:element>
      <xs:element name="endPoint" type="Point">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Triangle">
    <xs:sequence>
      <xs:element name="point" type="Point" minOccurs="3" maxOccurs="3">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="LinePathOpen">
    <xs:sequence>
      <xs:element name="point" type="Point" minOccurs="2" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="LinePathClosed">
    <xs:sequence>
      <xs:element name="point" type="Point" minOccurs="3" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="Radius">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="70"/>
    </xs:restriction>
  </xs:simpleType>
```



```
<xs:complexType name="Circle">
  <xs:sequence>
    <xs:element name="midPoint" type="Point">
    </xs:element>
    <xs:element name="radius" type="Radius">
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="Angle">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
    <xs:maxExclusive value="360"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="PointPolar">
  <xs:sequence>
    <xs:element name="radius" type="xs:int">
    </xs:element>
    <xs:element name="angle" type="xs:int">
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PointCartOrPolar">
  <xs:choice>
    <xs:element name="cartesianCoordinate" type="Point">
    </xs:element>
    <xs:element name="polarCoordinate" type="PointPolar">
    </xs:element>
  </xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```

Annex C (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
01-11-13		UP-010104			First draft of the specification from UP-010066		
01-11-14		UP-010106			Second Draft of the specification		
01-12-05		UP-010139			V0.2.1 After UP#07 Editorial changes based on UP-010118: clause 11.1 moved after clause 5, becoming chapter 6		
02-02-07		T2GUP-020004 T2GUP-020012			V0.3.0 after T2GUP#1 Annex A: New XML-schema files added. New Annex G added, describing data description related files and tools. New Annex H added, containing examples of GUP profiles		
2002-05		UP-020026 T2-020442			V0.3.1 after T2 SWG2 #17 Editorial and content changes based on T2-020443. Annex B: replaced with new annex.	0.3.0	0.3.1
2002-11		T2-020685 T2-020686 T2-020705			V0.3.2 after T2 SWG2 #18 Update of Annex A. Update of Annex G. Enhancement of Introduction, Scope, and Background.	0.3.1	0.3.2
2003-01		T2-030107 T2-030108 T2-030109			Restructuring of the document as proposed in T2-030107. New Intro and Scope text to reflect new structure. General Description section added.	0.3.2	0.3.3
2003-01		T2-030112 T2-030113 T2-030114 T2-030115			Updates to clauses 6.2, 7.2 and 7.3.	0.3.3	0.3.4
2003-04		T2C-030035,36,37			Widespread Updates to clauses 2 to 7	0.3.4	0.3.5
2003-05		T2-030258 T2-030264 T2-030266			Restructuring and Updates to clause 5 Updates to clause 7.4	0.3.5	0.4.0
2003-08		T2-030424 T2-030426			Introduction of Profile Component Group construct - clauses 3.1 and 5 Editorial Changes in clauses 3.1 and 5	0.4.0	0.4.1
2003-08		T2-030428			Redefinition of Profile Instance construct, changes according to this and Profile Component Group construct in clause 5 Editorial changes in clauses 1 to 5	0.4.1	0.4.2
2003-08		T2-030519 T2-030523			Changes in clause 5 regarding Profile Component Group construct, deletion of clauses 5.5.1 to 5.5.5 Restructuring clause 6	0.4.2	0.4.3
2003-10		T2C-030068,72,73,75-79			Widespread updates to clauses 1 to 7, introduction of clause 8 (Process of Administration of GUP/DDM), content of clause 7.2 moved to annex B, old annex B deleted.	0.4.3	0.5.0
2003-10		T2C-030083			Updates to clauses 5, 6 and 7	0.5.0	0.5.1
2003-11		T2-030576,588,651,652			Updates clause 6: XML diagrams, introduction of Properties XML Schema (clause 6.7); Updates to clause 8.7, XML schema examples; General updates for better readability and updates of references and abbreviations	0.5.1	0.5.2
2003-12					Editorial cleanup by MCC	0.5.2	1.0.0

Presentation of Specification to TSG T

Presentation to: TSG T Meeting #22
Document for presentation: TS 23.241, Version 1.0.0
Presented for: Information

Abstract of document:

This document is a technical specification titled 'Stage 2 Data Description Method' for the Release 6 Work Item "3GPP Generic User Profile Data Description Method".

This document specifies the templates for DDM to be used for generation of GUP data entities by 3GPP WGs and other bodies. It also includes procedures and rules for the administration of GUP Common Objects for TS24.241.

Changes since last presentation to TSG T:

This is the first presentation of this document to TSG T

Outstanding Issues:

Further development of the DDM - clarification of section 5 text to better align with sections 6 and 7. Miscellaneous improvements, changes to Hierarchical structure.

Contentious Issues:

None.

References: