3GPP TSG SA WG3 Security — S3#22 **S3-020020**
25 - 28 February 2002
Bristol, UK

3GPP Joint Ad-Hoc on Generic User Profile 7 ***UP-010128***
Cancun, Mexico
3-5 December 2001

Title:	Status of the Generic User Profile Work
Source:	3GPP Joint ad-hoc on Generic User Profile (GUP)
To:	SA1, SA2, SA3, SA4, SA5, T2, T3, CN1, CN4, CN5, SA1 GUP ad hoc, T2 GUP ad hoc
Cc:	none
Response to:	none

Contact Person:

Name:	**Rob Lockhart**
Tel. Number:	+1.561.739.2650
E-mail Address:	rob.lockhart@motorola.com

Attachments: **<In S3-020021>**: UP-010129 [LS to SA1 in response to S1-011176 on 22.240, v0.3.0 Stage 1]
UP-010141 [23.240, v0.3.0 SA2 Stage 2]
UP-010142 [23.241, v0.2.1 T2 Stage 2]
UP-010143 [24.241, v0.2.0 T2 Stage 3]

1. Overall Description:

The 3GPP Joint ad-hoc on Generic User Profile (GUP Joint ad hoc) is defining the concept of the Generic User Profile as defined in the Work Item Description SP-010548. The objective of this is to provide a *conceptual description* to enable shared usage of the information content by the different entities. The current status of work on this task is as follows:

- Draft Stage 1 has been advanced to v0.3.0 and is given to SA1 for further refinements. The associated file UP-010129 [LS to SA1 in response to S1-011176 on 22.240, v0.3.0 Stage 1] includes 22.240 plus suggested updates to this version to aid SA1 in their refinement of the Stage 1 work in progress.
- Draft Stage 2 (SA2), 23.240, has been advanced to v0.3.0. SA2 is invited to continue the development of this document, and the Joint ad hoc would be delighted to continue in assisting SA2 in this work.
- Draft Stage 2 (T2), 23.241, has been advanced to v0.2.1. The GUP Joint ad hoc will, in association with T2's newly-formed T2 GUP ad hoc, continue to refine this Stage 2 work in progress but responsibility for the specification will transfer to the T2 GUP ad hoc.
- Draft Stage 3 (T2), 24.241, has been advanced to v0.2.0. The GUP Joint ad hoc will, in association with T2's newly-formed T2 GUP ad hoc, continue to refine this Stage 3 work in progress but responsibility for the specification will transfer to the T2 GUP ad hoc.

The GUP Joint ad hoc offers to provide, where appropriate and possible, a GUP Joint ad hoc member to those groups wishing to evaluate these works in progress as such groups schedule reviews of these documents and activities.

2. Actions:

To the SA1 - SA5, T2, T3, CN1, CN4, CN5, and the T2 GUP ad hoc groups.

ACTION: The GUP Joint ad hoc asks these groups to review the listed documents in progress and to provide feedback, where appropriate, to this Joint ad hoc. To assist in this effort, the GUP Joint ad hoc offers to provide, where appropriate and possible, a GUP Joint ad hoc member to assist those groups wishing to evaluate these works in progress and their associated activities as such reviews may be scheduled.

To the SA1, SA1 GUP ad hoc, and the T2 GUP ad hoc groups.

ACTION: With this liaison, the GUP Joint ad hoc formally relinquishes control of the TS 22.240 specification to SA1 and the TS 23.241 and TS 24.241 specifications to T2.

To the T2 GUP ad hoc.

ACTION: Please inform the listed groups in this liaison statement when your T2 GUP ad hoc reflector is up and running so that those interested may subscribe to it.

3. Date of Next GUP Joint ad hoc Meetings:

S1 GUP (for reference)	15-18 Jan 2002	Phoenix, AZ, USA
T2 GUP #1 (for reference)	05-07 Feb 2002	Sophia Antipolis
GUP Joint #8	07-08 Feb 2002	Sophia Antipolis

3GPP Joint ad-hoc on Generic User Profile **UP-010141**
Cancun, Mexico
3-5th of Dec., 2001

3GPP TS 23.240 V0.3.0 (2001-12)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP Generic User Profile - Architecture; Stage 2 (Release 5)

Remove GSM logo from the cover page for pure 3rd Generation documents.

Select keywords from list provided in specs database.

Keywords

<keyword[, keyword]>

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

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

This clause is optional. If it exists, it is always the second unnumbered clause.

1 Scope

This clause shall start on a new page.

The present document …

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*.

[<seq>] <doctype> <#>[([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

[1] 3GPP TR 41.001: "GSM Release specifications".

[2] 3GPP TR 21 912 (V3.1.0): "Example 2, using fixed text".

[xx] 3GPP TS 22.240

3 Definitions, symbols and abbreviations

Delete from the above heading those words which are not applicable.

Subclause numbering depends on applicability and should be renumbered accordingly.

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

Definition format

***<defined term>**: <definition>.*

example: text used to clarify abstract rules by applying them literally.

3.2 Symbols

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

Symbol format

<symbol>	<Explanation>

3.3 Abbreviations

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

Abbreviation format

<ACRONYM> <Explanation>

4 GUP concept

The objective of specifying the Generic User Profile concept is to provide a *conceptual description* to enable shared usage of the information content by the different entities. The Generic User Profile concept aims at:

- Defining the GUP components
- Clarifying the mutual influence of the different components
- Specify the concept of active sub-profiles
  [Editor's note: Align with 22.240]
- Specify the storage and physical distribution of GUP data
- Define the ownership of the GUP data
- Define a Data Description Framework for GUP data definitions
- Develop datatypes and components for 3GPP common usage (see the use cases)
- Other Generic User Profile related datatypes and components
- Specify access mechanisms
- Specify a synchronization mechanism
- Specify the protocol for transfer of GUP data between core network elements
- Specify the protocol for transfer of GUP data between the UE and the core network
- Addressing the GUP Policy (e.g. Privacy)

The advantage of specifying a generic concept is that the user profile can easily be extended. All new data that will become part of the Generic User Profile, and all new services that will be developed can use the already existing mechanism defined for the 3GPP Generic User Profile concept. The GUP concept focuses on how to define the GUP, not mandating what must be in it. Only some parts of the User related data are part of the 3GPP Generic User Profile.

The Data Description Framework within the GUP concept provides co-existence with presently available industry standards as well as provides a migration path for these by defining a mapping and a default transport format.

4.1 What is the Generic User Profile?

The Generic User Profile is defined as "the collection of data which is stored and managed by different entities such as the UE, the Home Environment, the Serving Network and Value Added Service Provider, which affects the way in which an individual user experiences services". See 22.240 [xx]. This implies that the GUP data is user related in the sense that it allows personalization, handles variations and controls the behaviour of services or applications.

4.2 Use case examples and related services

Use cases include but are not limited to:

- UE configuration support
- Subscription management
- Content Negotiation (Pull)
- Content Negotiation (Push)

The GUP concept may be utilized by several features, such as Presence, Push, MExE, MMS, OSA, VHE and more.

Standards related to the Data Description Framework and Data Descriptions include RDF and (CC/PP), developed by the W3C, and the WAP UAProf, developed by the WAP Forum.

4.3 GUP Security and Privacy

Access to Generic User Profile data shall only be permitted in an authorized and secure manner. The secure mechanisms to be applied shall be appropriate to the level of confidentiality of the data.

The security mechanisms shall ensure that the entity storing the GUP data applies the appropriate level of security for the access and transfer of the GUP data as required by the owner.

4.4 The Generic User Profile Data Definition

[Editor's note: Align with GUP stage 1]

The following assumptions are made for the Generic User Profile Data:

- The Generic User Profile (GUP) is composed of a number of Generic User Profile Components.
- Each Generic User Profile Component (GUPC) has
  - Identification
  - Semantic (i.e. meaning of the GUPC)
  - Syntax (i.e. vocabulary, GUPC data type)
- A Generic User Profile Description consists of
  - List of Generic User Profile Components
The list contains the identification and type of each UPC
  - Set of schemas that defines the syntax and semantic of all Generic User Profile Component types
- Use XML where suitable

5 GUP Architecture

Editors note: further elaboration of this section required.

5.1 Classification for supporting of selection of the Profile Components

The purpose for this classification of 3GPP Generic User Profile data is to:

- Support grouping of data in Profile Components.
- Select the storage node or nodes.

- Select synchronisation principle.

In this Stage 2 specification the classification principles are elaborated and guidance how to group data in components is included.

This section contains a non-exhaustive list of classification criteria. For each criterion there are examples, see TS 22.240[xx].

[Editor's note: a, b, and c need further elaboration]

(a) Information Characteristics
(b) Storage Location
A general feature of the 3GPP Generic User Profile is that the different nodes are consumers for a certain subset of the data and are sources for another part. As a result, the parts of the 3GPP Generic User Profile are stored in different places. The same information can also be stored in many places.

- **Core Network**
  - Home network
  - Serving Network
- **User Equipment (UE)**
  - ME (MT and TE)
  - UICC
- **Application Valued Added Service Provider Equipment**

[Editor's note(from stage1): Application Service Provider Equipment – need to clarify definition. For both ownership and storage location]

(c) Ownership

(d) Data usage Role:
The applications accessing the 3GPP Generic User Profile data are classified based on the purpose of the access.

- **3GPP Generic User Profile Data: Consumer**
  These are consumers (such as applications or network entities), whose behaviour is controlled or influenced by the 3GPP Generic User Profile content. The consumer understands the meaning of the data and can use the information contained in the data. This is the most common usage role

- **3GPP Generic User Profile: Data Management**
  These are functions supporting the management of the content of 3GPP Generic User Profile, which include the creation, modification and deletion of components. These functions may be utilized by, e.g., the operator, a user or group of users, or a third party service provider

  - Personalization of User data:
    These applications support the User in updating the 3GPP Generic User Profile content. These are applications controlled by the individual User, possibly interactively

  - UE Management:
    xx

  - Subscription Management:
    (Includes service provisioning data management)

  - Service Customization
    The GUP related data that originates in a service provided by any service provider is controlled by the SP

- 3GP**P Generic User Profile: Data Maintenance**
  Application supporting the integrity and consistency of the distributed 3GPP Generic User Profile. The

applications have knowledge of the syntax of the data, but not the semantics of the data. The application shall maintain the data but not use the information represented by the data.

- Synchronization
- Backup Restore

(e) Value Change Frequency
The values of an attribute stored in the 3GPP Generic User Profile have different frequencies of change.

- **Constant**
  the value is the same in a certain context. It can, for example, be the same for all devices of a certain type.
- **Static**
  the value is constant and shall not change. Needs to be clarified
- **Semi-static**
  A semi-static value is a value that is seldom changed and can in most cases be regarded as static for a long time.
- **Dynamic**
  The value can be changed frequently.

(f) Data Object Life Time:
The lifetime of and data object can be classified as follows:

[Editor's note: needs to be elaborated, try to find different naming than the ones used in the previous classification]

- **Static**
  The data object exists as long as the profile exists.
- **Dynamic creation**
  The data object can be created .
- **Dynamic deletion**
  The data object can be deleted.

6 Motivation for the 3GPP Data Description Framework

The Data Description Framework defines the method to describe the data in a Generic User Profile. It defines syntax and semantics of data descriptions, a default representation (or transport format) of data descriptions, and the data in a Generic User Profile.

It is not intended to substitute Data Description Frameworks that already exist. The 3GPP Data Description Framework shall be used, where appropriate, when Generic User Profiles are specified for new applications.

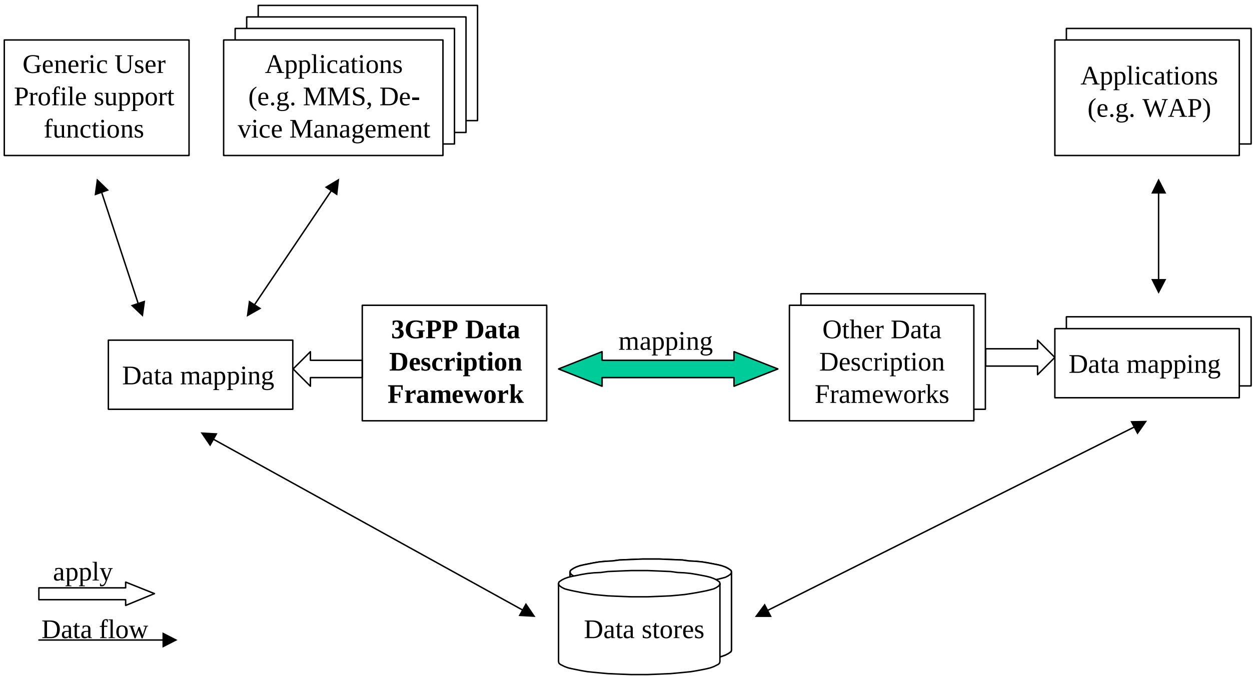

Generic User Profile support functions
Applications (e.g. MMS, Device Management
Applications (e.g. WAP)
Data mapping
3GPP Data Description Framework
mapping
Other Data Description Frameworks
Data mapping
apply
Data flow
Data stores

Figure xx: Data Description Framework Coexistence

Motivation to specify a new Data Description Framework:

- **Applicable to many 3GPP applications**
  The establishment of the Data Description Framework is supposed to avoid a growing number of different description methods by providing a favourite alternative for user profiles of emerging applications. This means that the 3GPP Data Description Framework is specified once and may be used for various applications afterwards.

- **Tailored to the needs of 3GPP**
  The Data Description Framework is defined specifically for the requirements of 3GPP. This comprises common characteristics of mobile networks as well as coexistence with existing frameworks like WAP UAProf and SyncML Device Management.

- **Ease of data handling**
  The general handling of Generic User Profile data including access and synchronisation is easier because applications can benefit from the wide deployment of the Data Description Framework. The number of necessary data exchange protocols can be minimised.

- **Efficient support mechanisms**
  Generic User Profile management and maintenance applications for devices are cost-efficiently supported.

- **Reduced data processing**
  The 3GPP Data Description Framework reduces the effort required to resolve and process user profile data and descriptions in networks, user equipment, and application servers. One example is the specification of a clear XML-based data description.

- **Ease of extensibility**
  The Data Description Framework provides the means to allow uncomplicated changes and extensions of data descriptions and data exchange formats. The introduction of new attributes and components is thereby simplified.

- **Simplified formats through generic mapping**
  Generic mapping to other formats and representations achieves flexibility and minimises the possibility of errors.

No existing technology adequately covers all of the above points, thus a new solution, based on the best of the old, must be defined to achieve these goals.

7 Requirements on Data Description

[Editor's note: Align with chapter 12 from stage 1, Tdoc 44, 60,]

This is an initial list of requirements on the description of the User Profile. It is important that this list includes all the requirements on the data description coming from the different usage of the data, in order to promote the common data description.

7.1 Fulfil Functional Requirements from use cases

The requirement from the Use Cases describing functions related to User profile: read/write/delete/add elements in the user profile…

7.2 Security and accessibility

It is also important that the data description fulfil the requirements related to security such as access rights.

7.3 Re-usable description components

It must be possible to divide the description in parts called description components. A description component can be used in many User Profile descriptions to be re-usable.

Motivations:

- Identical parts of the User Profile are described once
- The responsibility of defining description components can be distributed between different organisations, standardisation bodies or technical groups in 3GPP.
- Some description components are standardised and some components are late defined and/or just published (manufacturer specific components).

7.4 Data syntax

The data description must have a good way to describe structure, ranges, default values of the data elements.

Motivations:

- Decrease the errors when handling the data.
- It will help the development of management tools.
- Automatic validity checks of values possible.

7.5 Data semantic

The data description must have a good way to describe the meaning (semantic) of the data elements.

Motivations:

- Decrease the errors when handling the data.
- It will help the development of management tools.

7.6 Computer parse-able Data Description

The data description must be computer parse-able and interpretable by human beings.

Motivations:

- There will be a significant amount of data description and it will be very costly if manual translation is needed.
- An automatic translation to other description formats can be implemented.
- It will help the development of management tools.

7.7 Define a Default Transport Format

Define a default transport format for the Data Description and for the Data

7.8 Support Backward compatibly

The data description must support co-existence with other already existing description method.

Motivation:

- Avoid to redo all existing data descriptions

7.9 Extendible

The data description shall support the addition of new data/description components. That gives the possibility, as well, to start with a small set of features and then adds support for more things.

Motivations:

- The introduction of the data description must be done stepwise.
  Potential first application areas are Terminal Configuration and Capability Information Exchange.
- In the future more data will be part of the user profile, it should be easy to add it in the selected data description
- Add easily Manufacturer specific data/description components

7.10 Data Oriented

The data description should describe the data and not an interface to the data.

Motivations:

- The User Profile is a set of data and not an interface to a program used to access the data. It is probably more cost efficient to use a schema definition language, such as XML schema, then an interface oriented approach.
- It is easier to map an interface to the data on the data description, than the reverse.

7.11 Expressiveness Balance

There must be a balance between the expressiveness to describe data and the needed complexity in the implementations using data descriptions.

-

8 Data Description Logical Levels

[Editor’s note: Align with T2 stage 2]

The data description can be split in the followinglogical levels:

- **Data**
  Data stored and / or accessed in a User Profile
- **Data Description**
  describes the data contained in the User Profile. (This is also called the Schema level.)
- **Data Description Framework**
  Defines how to create the data description. (This is also called the Schema-Schema level i.e. the Schema describing the Schema, which describes the data.)

Editor’s note: Revise to clarify the relation to the GUP definition; fix the colours.

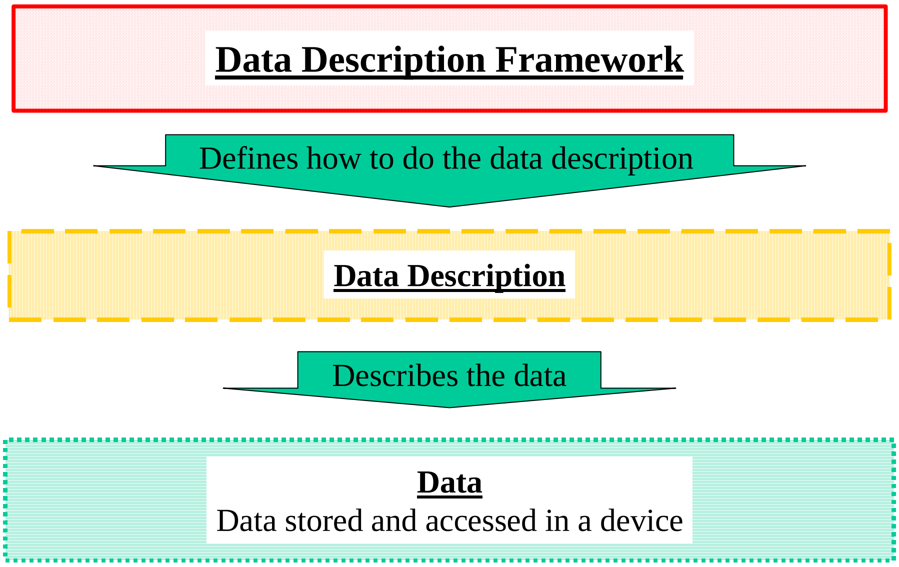

Data Description Framework
Defines how to do the data description
Data Description
Describes the data
Data
Data stored and accessed in a device

Figure yy: Data Description Framework logical levels

8.1 Data Description Framework

The Data Description Framework defines the method to describe the data in a User Profile. It defines the structure of the data description.

It is standardised and used for all User Profiles.

The Data Description Framework also defines a default representation (or transport format) of Data Descriptions and the data in a User Profile.

8.2 Data Description

A specific User Profile will be described, according to the Data Description Framework, resulting in a Data Description.

8.3 Data

The structure and semantic of the data in the User Profile is described in the Data Description. The Data Description Framework also defines a default representation of the data in the User Profile.

8.4 A use case example: UE Configuration

This is an example showing how the Data Description Framework, a Data Description and the related data can be used. A User Equipment (UE) Configuration Support System is used as an example and the data described in a Data Description is the data used to configure or personalise a UE.

Figure X illustrates how the Data Description Framework, Data Description, and Data interrelate.

The **Data Description Framework** defines the syntax and semantics of the Data Description. The **Data Description** is describing the **Data**, i.e., device configuration, which can be accessed by the UE Configuration Support. The Data Description describes the structure or syntax of the configuration data. The semantics or meaning of data are also given using normal language.

There is only **one** Data Description Framework. As a consequence, this framework is common to all device types. For each device type, there is one Data Description; thus, **several hundred** Data Descriptions will exist. The device configuration data is specific to each unique device; thus, **several million** device configurations will exist.

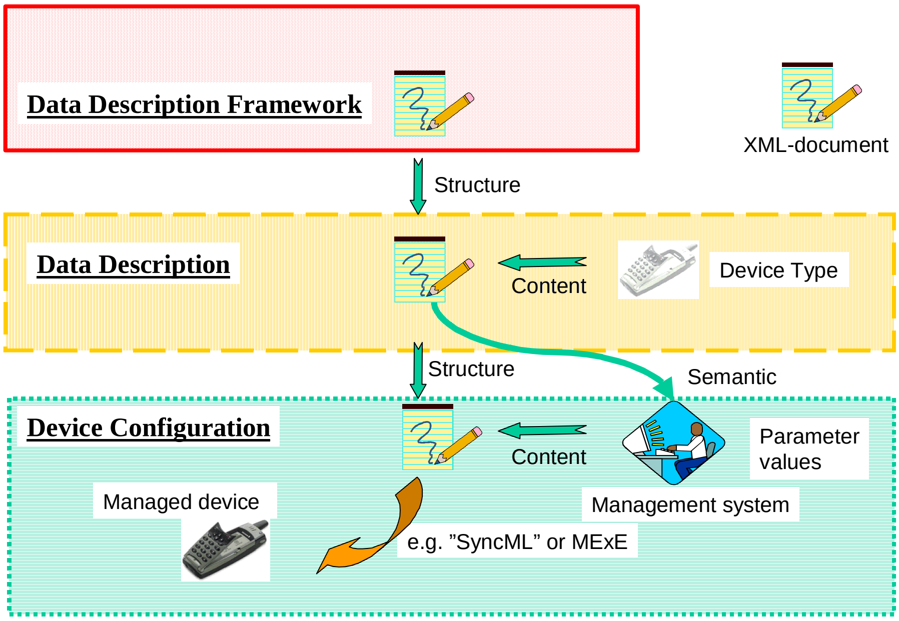


Figure X: Usage of Data Description Domains in User Equipment configuration support

In the figure, a number of XML-documents may be found. The next few paragraphs provide short descriptions of the purpose of the different XML-documents.

The Data Description Framework contains a number of XML-schemas (XML-documents) describing the allowed structure of a Data Description. Tools used to create Data Descriptions can be controlled by these XML-schemas.

The Data Description comprises a number of XML-documents describing the data used to configure a certain type of device. The main parts of the Data Description are the descriptions of the syntax and of the semantics. The syntax defines the allowed structure and values in the device configuration and may be used to check the validity of the data. The syntax is used by the Management System. The semantics guide the understanding and selection of values and are used by persons checking or changing the configuration data. The Management System presents semantics-based information to Management System user. The Management System is made **device-type-independent** through the use of the device configuration data descriptions.

The configuration data, in this example, is represented by an XML-document. The data can be transferred using SyncML device management protocol, a MExE application, or some other means. The following outlines an example of the device configuration data transmitted to a device:

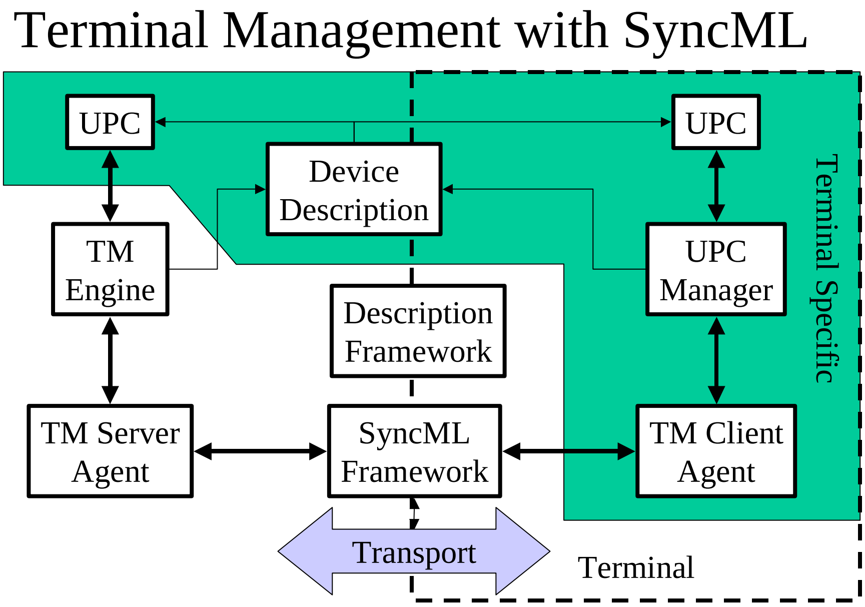
```
<Settings>
  <AudioSettings>
  <RingVolume>2</RingVolume>
  <IncreasingRing>ON</IncreasingRing>
  <VibratingAlert>ON</VibratingAlert>
   ...
  <PersonalRing>
    <Contact>Anders</Contact>
    <Melody>13</Melody>
  </PersonalRing>
  <PersonalRing>...</PersonalRing>
  ....
</AudioSettings>
<DisplaySettings/>
<VoiceControlSettings/>
<HandsfreeSettings/>
</Settings>
```

Annex A (informative): Terminal Management and Data Description

One usage of some of the data included in the User Profile is in the Terminal Management framework.

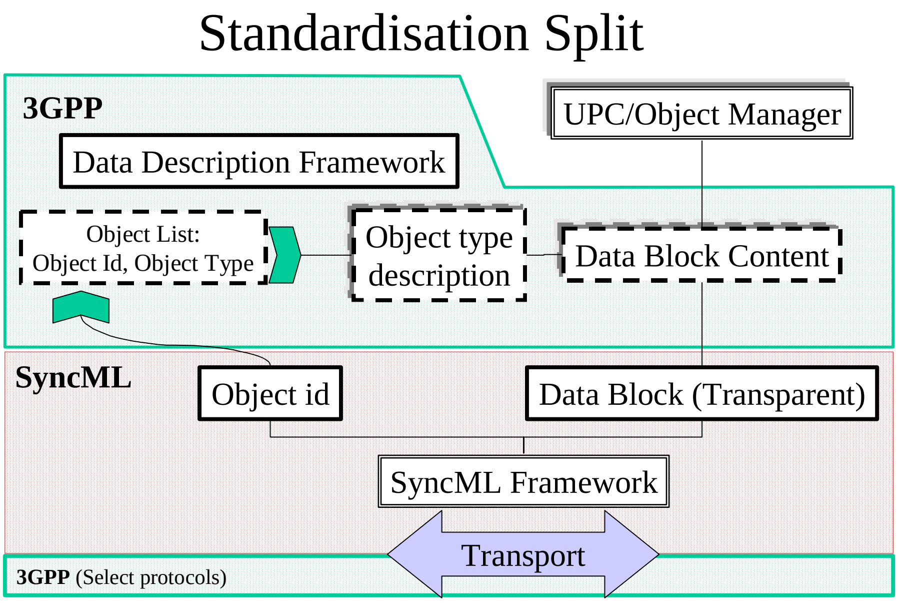
In the SyncML initiative there is some work going on to specify a device management protocol based on SyncML.

The following picture shows where the Data Description architecture can be used together with the SynML Device Management approach.



The effort on standardisation to have a common way to describe data implies a separation on the work, in order not to define it in different fora. The following picture shows a proposed way to split the work between 3GPP and SyncML Device Management.

Editor's note: Colours to be aligned with above. Entire section: thorough revision.



As a good way to achieve the objective, the data description framework goes into 3GPP, leaving to SyncML the protocol and the Object Id to identify the components from the User Profile that the device management will handle.

Annex B (informative): Change history

It is usual to include an annex (usually the final annex of the document) for specifications under TSG change control which details the change history of the specification using a table as follows:

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
01-11-13		UP-010103			First version of the draft specification from UP-010065		
01-11-14		UP-010109			After UP#6 V0.2.0		
01-12-05		UP-010136			After UP#7 V0.3.0 Added changes from UP-010116 , UP-010134 and UP-010135 Reference added to GUP stage 1 TS 22.240 Chapter 8 moved under chapter 4 Chapter 10 moved to Annex A Editorial changes		

3GPP Joint ad-hoc on Generic User Profile **UP-010142**
Cancún, México
3-5th of December, 2001

3GPP TS 23.241 V0.2.1 (2001-12)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Terminals; 3GPP Generic User Profile - Data Description Framework; Stage 2 (Release 5)

Remove GSM logo from the cover page for pure 3rd Generation documents.

Select keywords from list provided in specs database.

Keywords

<keyword[, keyword]>

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

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

This clause is optional. If it exists, it is always the second unnumbered clause.

1 Scope

This clause shall start on a new page.

The present document …

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 Markup Language (XML) 1.0 (Second Edition)". http://www.w3.org/TR/2000/REC-xml-20001006

[2] W3C Recommendation: "Namespaces in XML", 2 May 2001. http://www.w3.org/TR/1999/REC-xml-names-19990114/

[3] W3C Recommendation: "XML Schema Part 0: Primer", 2 May 2001. http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

[4] W3C Recommendation: "XML Schema Part 1: Structures", 2 May 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[5] W3C Recommendation: "XML Schema Part 2: Datatypes", 2 May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[6] W3C Recommendation: "XML Path Language (XPath) Version 1.0", 16 November 1999. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[7] W3C Candidate Recommendation: "XML Pointer Language (XPointer) Version 1.0", 11 September 2001. http://www.w3.org/TR/1999/REC-xpath-19991116

[8] ISO (International Organization for Standardization): "ISO 11404, Language-independent Datatypes.

3 Definitions, symbols and abbreviations

Delete from the above heading those words which are not applicable.

Subclause numbering depends on applicability and should be renumbered accordingly.

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

Definition format

***<defined term>**: <definition>.*

example: text used to clarify abstract rules by applying them literally.

3.2 Symbols

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

Symbol format

<symbol>	<Explanation>

3.3 Abbreviations

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

Abbreviation format

<ACRONYM>	<Explanation>

4 Data Description Framework

Editor's note: Align with S2 stage 2 on GUP and DDF, 23.xyz.

The data description "matter" can be split in the following domains:

- **Data**
  Data stored and or access in a User Profile
- **Data Description**
  describes the data contained in the User Profile. (This also called the Schema level.)
- **Data Description Framework**
  Defines how to create the data description. (This also called the Schema-Schema level i.e. the Schema describing the Schema, which describes the data.)

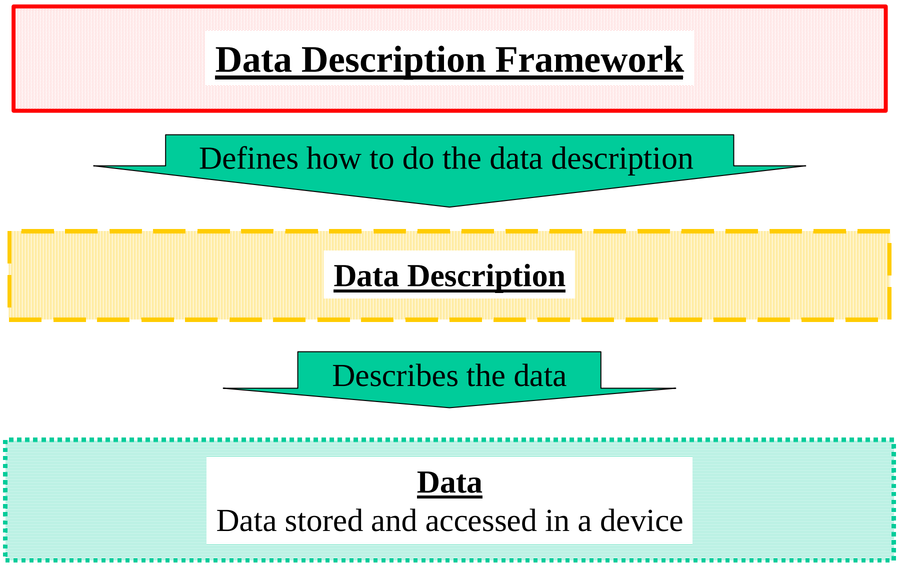

Data Description Framework
Defines how to do the data description
Data Description
Describes the data
Data
Data stored and accessed in a device

Figure 1: xxx

The Data Description Framework defines the method to describe the data in a Generic User Profile. It defines the structure of the Data Description as based on XML-schema. The Data Description Framework also defines a default representation (or transport format) of Data Descriptions and the data in a Generic User Profiles.

A specific Generic User Profile will be described, according to the Data Description Framework, resulting in a Data Description.

The structure and semantic of the data in a Generic User Profile is described in the Data Description. The Data Description Framework also defines a default representation of the data in the User Profile.

4.1 Basic Structure of the Generic User Profile

The Data Description Framework is used to describe Generic User Profiles with the basic structure shown in the picture below.

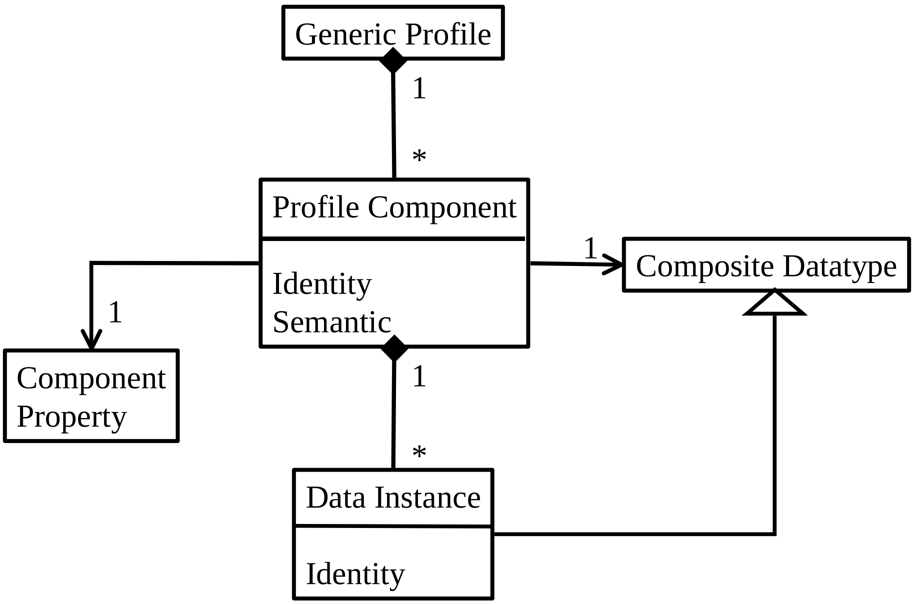


Figure 2: xxx

4.2 Types of Data Description Documents

A Data Description consists of a number of XML-documents.

The picture “Relationship between Data Description Documents” indicates the types of documents and the relation between those documents.

Note: An arrow means a reference between documents.

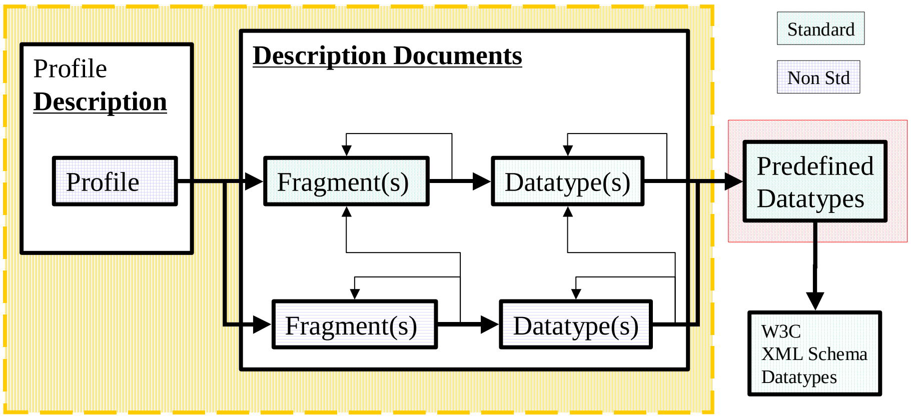


Figure 3: Relationship between Data Description Documents

There are three types of Data Description documents:

- Profile
  A Profile document is the main document in the description of the data in a Generic User Profile. It mainly contains references to Generic User Profile parts defined in Fragment documents. It is specific for a class of Generic User Profiles.

- Fragment
  Generic User Profiles parts containing Profile Components are defined in Fragment documents. Profile Components are declared mainly by connecting a Profile Component identity to a Datatype. Fragment documents can be shared between Data Descriptions.

- Datatype
  In this kind of document Datatypes are defined. Datatypes are defined using in the Data Description Framework predefined Datatypes and **user defined** Datatypes. Datatype documents can be shared between Data Descriptions.

4.3 Data Description Framework Parts

The Data Description Framework consists of:

- XML-schema files (in the appendix of this document)

- Description rules

- Default XML-based transport format

5 Logical Structure of Data Descriptions

Main elements used in the Data Descriptions are:

Profile	Declares a Generic User Profile.
Fragment	Defines a reusable Generic User Profile part.
Component	Declares a Profile Component.
Property	Declares a set of properties, which can be associated with some Profile Components.
Semantic	Defines the meaning of things. Is used to understand the content of the Generic User Profile.
Comment	Provides information intended for (the remainder of)? the Data Description.
Datatype	Defines a Datatype.

Each of the elements is briefly described in the following section and in more detail in a separate chapter (NS: what chapter?).

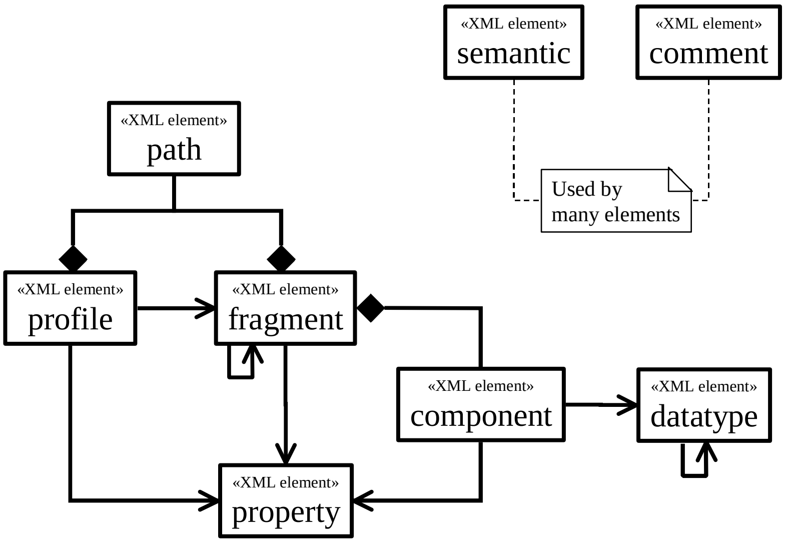


Figure 4: xxx

5.1 Profile

The Profile element is used to declare a Generic User Profile. It mainly contains the references to a number of Profile Fragments. The results of merging the referenced Profile Fragments are:

- A hierarchical name structure containing a number of Profile Components.
- A description of each Profile Component
    - o Abstract syntax
    - o Semantic

5.2 Fragment

In a Fragment element a number of Profile Components are declared. A Profile Fragment is normally used in many Generic User Profiles.

A Profile Fragment can include other Profile Fragments by referencing them.

5.3 Component

The Component element is used to declare the Profile Component.

The following are defined for a Profile Component :

- Identity
  a hierarchical namespace, similar to that in a file system, is used. The identities are selected in a way that they can be used as Universal Resource Locators, URLs.
- Semantic
  gives the meaning of the Profile Component.
- For the Data Instance
  - Possible number of Data Instances
  - Datatype
    The abstract syntax of the Data Instance is defined by referencing a Composite Datatype.
- Component Properties

5.4 Property

The property element declares a Component Property, which is used to control the usage and handling of Profile Components.

A Profile Component is directly or indirectly referencing one Component Property.

5.5 Semantic

The Semantic element is used in many elements to define the meaning. Examples are defining the meaning of a Profile Component, a Datatype, or a specific value in enumerations. The semantic is given using normal language. It is possible to give it in several different languages.

The information in the semantic elements is used by the interpretation and usage of the values in a Generic User Profile

5.6 Comment

Comment elements are used to add(?) comments to the Data Description.

5.7 Datatype

A group of elements is used to describe Datatypes. In the Data Description Framework there are a number of predefined Datatypes. Simple user defined Datatypes can be defined based on the predefined simple Datatypes. Composite Datatypes are defined using simple and other composite Datatypes.

The abstract syntax of a Component Data Instance is defined by referencing a Composite Datatype.

6 Notation used in this Specification

6.1 Rules

In this document an informal "Extended Backus-Naur Form (EBNF)" like notation is used.

EBNF rules are used in the document. The syntax of a rule is:

```
{symbol} ::= expression.
```

The rule is describing the {symbol} using an expression (or some text).

The special symbols used in the expression are:

{symbol**}**	{symbol} is explained in a rule with {symbol} as its right side (`{symbol}::=` ...). It can be regarded as a placeholder for the thing described in the rule.
(expression**)**	Expression is treated as a unit when combined as described in the following 5 rows.
A**?**	Zero or one occurrences of A; optional A
A B	A followed by B (Concatenation). This operator has higher precedence than alternation; thus A B | C D is identical to (A B) | (C D).
A **|** B	A or B but not both (Alternation).
A**+**	One or more occurrences of A. Concatenation has higher precedence than alternation; thus A+ | B+ is identical to (A+) | (B+).
A*****	Zero or more occurrences of A. Concatenation has higher precedence than alternation; thus A* | B* is identical to (A*) | (B*).
{* comment ***}**	A comment in the expression.
{xsi:datatype**}**	Data of a simple type defined in "XML Schema Part 2: Datatypes" [5].
elementName	Used in XML-element content models.

6.2 XML-element

The following layout is used in the description of XML-elements:

Synopsis:

```
<tag
  attributeName
  attributeType = {xsi:datatype}
  optionalAttribute?
  enumAttribute = (large | medium | small) : medium
>
Content: expression
</tag>
```

In the start tag there is a list of attribute names (`attributeName`, `attributeType`, `optionalAttribute` and `enumAttribute`).

Optional attributes has a "?" after its name.

Attribute of a simple type defined in "XML Schema Part 2: Datatypes" [5] is indicated with `{xsi:datatype}` as for the `attributeType` above.

Where an attribute is of an enumerated datatype, the possible values are shown separated by vertical bars, as for the `enumAttribute` above; if there is a default value, it is shown following a colon.

The expression following "Content:" is an expression describing the allowed content of the element. Name not surrounded by {}, used in the expression is the name of an element, which may appear as a child element. The optional character following a name or sub-expression, governs whether the element or the sub-expression may occur one or more (+), zero or more (*), or zero or one times (?). The absence of such an operator means that the element or content particle must appear exactly once.

Example:

```
<example
  count = {xsi:integer}
  size? = (large | medium | small) : medium
>
Content: (all | any*)
</example>
```

6.2.1 Headlines used in XML-element descriptions

[editor's note: need a 11.1.2.2]

The following headlines are used in the description of XML-elements:

Synopsis:

Indicating the syntax in the description of datatype.

Example:

Contains an example of a (part) of a datatype description.

Example data in XML-format:

Contains an example of a data in XML-format conforming to the datatype description.

Synopsis as expanded XML-schema:

The synopsis expressed using in XML-schema.

Example as expanded XML-schema:

The example expressed using in XML-schema.

7 Generic User Profile Description

7.1 Element profile

Synopsis:

```
<profile {propertyRef}?
>
  Content: {semantic} (fragmentRef | pathInProfile)*
</profile>
```

7.2 Element path used in profile element

Synopsis pathInProfile:

```
<path pathName = {pathName}>
  Content: {optSemantic} (fragmentRef | pathInProfile)*
</path>
```

8 Profile Fragment Description

8.1 Element fragment

Synopsis:

```
      <fragment
            name = {fragmentName}
            pathName? = {pathName}>
        Content: {optSemantic} (component | fragmentRef | pathInFragment)*
      </fragment>
```

The {fargmentName} is used when the fragment is referenced form a profile element or other fragment elements.

8.2 Element fragmentRef

Synopsis:

```
      <fragmentRef
            fragmentRef = {fragmentRef}
            pathName? = {pathName}
            propertyRef? = {propertyRef}
      >
        Content: {optSemantic}
      </fragmentRef>
```

The fragment referenced by {fragmentRef} will in the resulting profile replace this element. It works like an include statement.

It is regarded an error if both the referring fragmentRef-element and the referenced fragment-elements have a pathName attribute.

8.3 Element path used in fragment element

Synopsis pathInFragment:

```
      <path pathname =”{pathName}”
            propertyRef?="{propertyRef}">
        Content: {optSemantic} (component | fragmentRef | pathInFragment)*
      </path>
```

9 Profile Component Description

9.1 Element component

Synopsis:

```
<component
      pathname? = {pathName}
      datatypeRef = {compositeDatatype}
      propertyRef? = {propertyRef}
      minInstances? = {minOccur}
      maxInstances? = {maxOccur}
>
  Content: {optSemantic}
</component>
```

Synopsis:

```
{minOccur} ::= {xsi:unsignedShort}
{maxOccur} ::= {xsi:unsignedShort} | “unbounded”
```

10 Component Property

The Component Property contains information controlling the usage and handling of a Profile Component. To allow several Profile Components to use the same Component Property, references are used. A Profile Component is directly or indirectly referencing one Component Property. Profile Components sharing Component Properties will be handled in the same way.

Example of property information is:

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

10.1 Element property

Synopsis:

```
      <property
            name = {propertyName}
      >
        Content: [TBD]
   </property>
```

11 Semantic and Comment

Comment elements are used to give comments in English to the Data Description itself.

Semantic is used to define the meaning of the main concepts used in Data Description. Examples are: Profile Component, Datatype, item in Datatype and specific value (in enumerations).

The semantic is given using normal language. It is possible to give it in several different languages.

Synopsis:

```
{semantic} ::= comment? semantic
{optSemantic} ::= comment? semantic?
```

11.1 Element comment

Synopsis:

```
<comment xml:lang="en">
  Content: [TBD]
</comment>
```

Comments elements are used to give comments to the Data Description. It is given in the English language.

11.2 Element semantic

Synopsis:

```
<semantic>
  Content: label+ definition? description?
</semantic>
```

The information in the semantic elements is used by the interpretation and usage of the described content. The semantic can be given in three different levels of detail:

- Label
  A human-readable label.
- Definition
  A statement that describes the essential nature of the element been described.
- Description
  Additional information (optional).

Synopsis:

```
<label xml:lang = {language} >
  Content:
</label>
```

Examples:

```
<label xml:lang="en">Understandable label</label>
<label xml:lang="se">Tolkbar etikett</label>
```

Synopsis:

```
<definition xml:lang = {language} >
  Content:
</definition>
```

Examples:

```
<definition xml:lang="en">
      A short definition</definition>
<definition xml:lang="se">
      En kort definition</definition>
```

Synopsis:

```
<description xml:lang = {language} >
  Content:
</description>
```

Examples:

```
<description xml:lang="en">
  A longer description ...
</description>
<description xml:lang="se">
  En längre beskrivning ...
</description>
```

Synopsis:

```
{languageId} ::= {xsi:language}
```

Language represents natural language identifiers as defined by [RFC 1766].

12 Datatype Description

12.1 Definitions

12.1.1 Datatype

[From XML-schema specification]

In this specification, a datatype is a 3-tuple, consisting of a) a set of distinct values, called its ·value space·, b) a set of lexical representations, called its ·lexical space·, and c) a set of ·facets that characterize properties of the ·value space·, individual values or lexical items.

12.1.2 Atomic datatypes

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

12.1.3 Predefined Atomic datatypes

Predefined Atomic datatypes are atomic datatypes, which are defined in this specification.

12.1.4 Derived Atomic datatypes

Derived Atomic datatypes are Atomic datatypes derived from the Atomic predefined datatypes by constraining them or by defining a union of Atomic datatypes

12.1.5 Composite datatypes

Composite datatypes are defined using atomic and other composite datatypes.

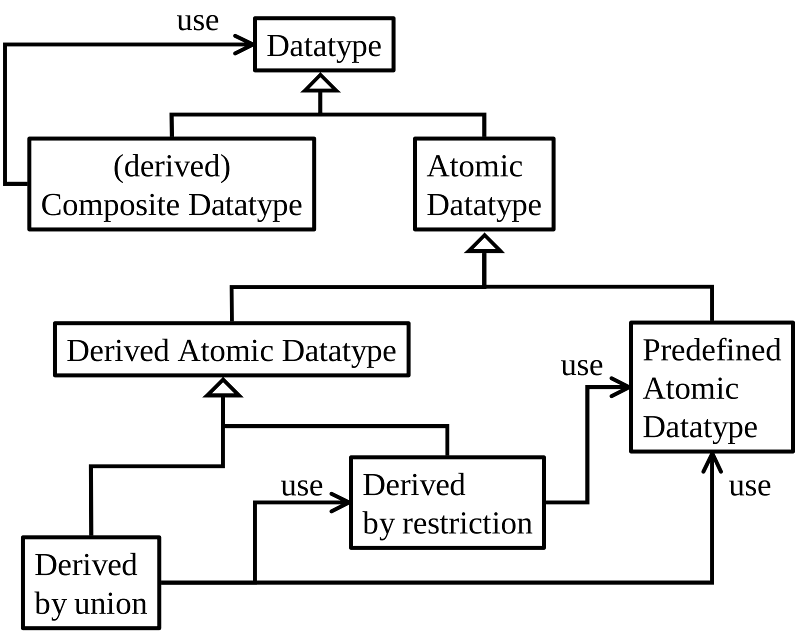


Fig xx Relationship between kinds of datatypes

Synopsis:

```
      {datatypeDefinition} ::= atomicType | compositeType
```

{datatypeName} ::= [TBD]

12.2 Atomic Datatypes

12.2.1 Introduction

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.

12.2.2 Predefined atomic datatypes

Predefined atomic datatypes can only be added by revisions to this specification.

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 in the comments “{* … *}” below are for the moment excluded.

```
      {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*}
```

12.2.3 Derived Atomic Datatypes

Synopsis:

```
      <atomicType name = {datatypeName} >
            Content: {optSemantic} ({restriction} | {union})
      </atomicType>
```

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

Synopsis as expanded XML-schema:

```
      <simpleType
            final=“list”,“restriction”
            id = [TBD]
            name = "{dataTypeName}"
      >
        {optSemantic} ({restriction} | {union})
      </simpleType>
```

Example:

12.2.4 Atomic datatypes derived by restriction

[From XML-schema specification]

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.

Note: Atomic datatypes derived by restriction can only be derived directly from Predefined atomic datatypes and not as restriction on derived atomic datatypes as in XML-Schema.

Synopsis:

```
      <restriction base = {predefinedAtomicDatatype} >
        Content: {optSemantic} ({constrainingFacet})*
      </restriction>
```

Example:

```
      <atomicType  name="more-than-ninety-nine">
                   <restriction base="int">
         <minExclusive value=’99’/>
       <restriction/>
      </atomicType>
```

Example data in XML-format:

```
100
```

Synopsis as expanded XML-schema:

```
<xs:simpleType name="{datatypeName}">
  <xs:restriction base="{predefinedAtomicDatatypeName}">
    {constrainingFacet}
  </xs:restriction>
</xs:simpleType>
```

Example as expanded XML-schema:

```
<xs:simpleType name='more-than-ninety-nine'>
  <xs:restriction base='int'>
    <xs:minExclusive value='99'/>
  </xs:restriction>
</xs:simpleType>
```

12.2.5 Constraining Facets

Constraining Facets in XML–schema are: length minLength maxLength pattern enumeration

whiteSpace maxInclusive maxExclusive minExclusive minInclusive

totalDigits fractionDigits.

[Issue: Which XML-schema Constraining Facets to select to be used]

Synopsis:

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

Synopsis:

```
<{constrainingFacetTag}  value>
      Content: {optSemantic}
</{constrainingFacetTag}>
```

Example:

12.2.6 Union datatype

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

[From XML-schema specification] 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.

Synopsis:

```
<union>
      Content: {optSemantic} member*
</union>

<member  ref = {atomicNonUnionDatatype} >
      Content: {optSemantic}
</member>
```

Example:

```
<atomicType name="booleanOrDate">
  <union >
      <member ref="xsi:boolean"/>
      <member ref="xsi:date"/>
  </union>
</atomicType>
```

Example data in XML-format:

```
<someTag xsi:type="xsi:boolean">false</someTag>
```

or

```
<someTag xsi:type="xsi:date">1948-10-11</someTag>
```

Synopsis as expanded XML-schema:

```
<union id=”ID”>
  <simpleType ref="{atomicNonUnionDatatype}">
      {optSemantic}
  </simpleType>
  <simpleType ref= …>
      …
  </simpleType>
  …
</union>
```

Example as expanded XML-schema:

```
<atomicType name="booleanOrDate" … >
  <union … >
      <simpleType ref="xsi:boolean"/>
      <simpleType ref="xsi:date"/>
  </union>
```

</atomicType>

12.3 Composite Datatypes

12.3.1 Introduction

A composite datatype contains a number of name items each with a defined datatype.

```
{compositeType} ::= recordType
```

12.3.2 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.

Synopsis:

```
<recordType name = {datatypeName} >
      Content: {optSemantic} (field | fieldVector)*
</recordType>
```

Synopsis:

```
{fieldName} ::= [TBD]
```

There are two types of fields: Simple field and vector field. A simple field contain on instance of the field datatype. A vector field contains a vector or a number of instances of the datatype.

Example:

```
<recordType name="screenCoordinate">
  …
</recordType>
```

Synopsis as expanded XML-schema:

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

Example as expanded XML-schema:

```
<xs:complexType name=" screenCoordinate">
  <xs:sequence>
   …
  <xs:sequence>
</xs:complexType>
```

12.3.3 Simple field

Synopsis:

```
<field name="{fieldName}" dataType="{dataTypeName}"/>
```

Example:

```
<recordType name="screenCoordinate">
 <field name="x" datatype="xCoordinate"/>
 <field name="y" datatype="yCoordinate"/>
</recordType>
```

Synopsis as expanded XML-schema:

```
<xs:element name="{fieldName}" type="{datatypeName}"/>
```

Example as expanded XML-schema:

```
<xs:complexType name=" screenCoordinate">
  <xs:sequence>
   <xs:element name="x" type=" xCoordinate"/>
   <xs:element name="y" type=" yCoordinate"/>
  <xs:sequence>
</xs:complexType>
```

Example data in XML-format:

```
<x>12</x>
<y>5</y>
```

12.3.4 Vector field

Synopsis:

```
<fieldVector
      name="{fieldName}"
      dataType="{datatypeName}
      minOccurs="{minOccur}"
      maxOccurs="{ maxOccur}"
```

```
>
  Content: {optSemantic}
        </fieldVector>
```

Example:

```
<fieldVector name="c" dataType="coordinate"
      minOccurs="3" maxOccurs="3"/>
```

Example data in XML-format:

```
<c index="1">10</x>
<c index="2">20</x>
      <c index="3">30</x>
```

Synopsis expanded XML-schema:

```
<xs:element name="{fieldName}" type="{datatypeName}"
      minOccurs="{minOccur}" maxOccurs="{maxOccur}">
      <xs:attribute name="index" type="xs:byte"/>
</xs:element>
```

Example expanded XML-schema:

```
<xs:element name="x" type="coordinate"
      minOccurs="3" maxOccurs="3">
      <xs:attribute name="index" type="xs:byte"/>
</xs:element>
```

Annex A (normative): XML-schema files

This annex is a placeholder for the XML-schema files that are part of the Data Description Framework. These files will be used for the creation and consistency check of the Data Descriptions.

The following files represent XML-schemas. “3GPPsemantic.xsd” is the schema describing how to represent the semantics. “3GPPdatatype.xsd” is the schema describing how to represent the datatypes. “3GPPdatatype.xslt” is the definition of the translation between 3GPP representation of datatypes and the corresponding schema.

3GPPsemantic.xsd 3GPPdatatype.xsd 3GPPdatatype.xslt

Annex B (Informative): XML-schema in brief

XML-schema is a Schema definition language. The functionality is above and beyond what is provided by DTDs.

The W3C Recommendation consists of three parts:

- XML Schema Part 0: http://www.w3.org/TR/xmlschema-0/
  Primer is a non-normative document intended to provide an easily readable description of the XML Schema facilities, and is oriented towards quickly understanding how to create schemas using the XML Schema language. This primer describes the language features through numerous examples, which are complemented by extensive references to the normative texts.

- XML Schema Part 1, Structures: http://www.w3.org/TR/xmlschema-1/
  and

- XML Schema Part 2 ,Datatypes: http://www.w3.org/TR/xmlschema-2/
  provide the complete normative description of the XML Schema language.

B.1 XML-Schema Type System

The XML-schema Part 2 defines a Data Type System.

A Datatype is defined as follows:

- A Datatype is a 3-tuple, consisting of:

  a) a set of distinct values, called its·**value space**,

  b) a set of lexical representations, called its·**lexical space**, and

  c) a set of ·facet·s that characterize properties of the·value space, individual values or **lexical items**.

The definition of Boolean is:

- Boolean has the·value space·required to support the mathematical concept of binary-valued logic: {true, false}.

The lexical space of Boolean is defined:

- An instance of a datatype that is defined as ·boolean· can have the following legal literals {true, false, 1, 0}.

B.2 Examples of user defined types

To create a new type of integer called myInteger whose range of values is between 10000 and 99999 (inclusive) can be done by restricting the built-in simple type integer, whose range of values also includes integers less than 10000 and greater than 99999. To define myInteger, we restrict the range of the integer base type by employing two facets called minInclusive and maxInclusive:

```
<xsd:simpleType name="myInteger">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="10000"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

XML Schema defines fifteen facets. Among these, the enumeration facet is particularly useful and it can be used to constrain the values of almost every simple type, except the boolean type. The enumeration facet limits a simple type to a set of distinct values. For example, we can use the enumeration facet to define a new simple type called USState, derived from string, whose value must be one of the standard US state abbreviations:

```
<xsd:simpleType name="USState">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="AK"/>
  <xsd:enumeration value="AL"/>
  <xsd:enumeration value="AR"/>
  <!-- and so on ... -->
</xsd:restriction>
</xsd:simpleType>
```

New complex types are defined using the complexType element For example, USAddress is defined as a complex type, and within the definition of USAddress we see five element declarations.

```
<xsd:complexType name="USAddress" >
  <xsd:sequence>
   <xsd:element name="name"   type="xsd:string"/>
   <xsd:element name="street" type="xsd:string"/>
   <xsd:element name="city"   type="xsd:string"/>
   <xsd:element name="state"  type="xsd:string"/>
   <xsd:element name="zip"    type="xsd:decimal"/>
  </xsd:sequence>
```

```
                        </xsd:complexType>
```

B.3 DTD and XML-schema in the data Description architecture

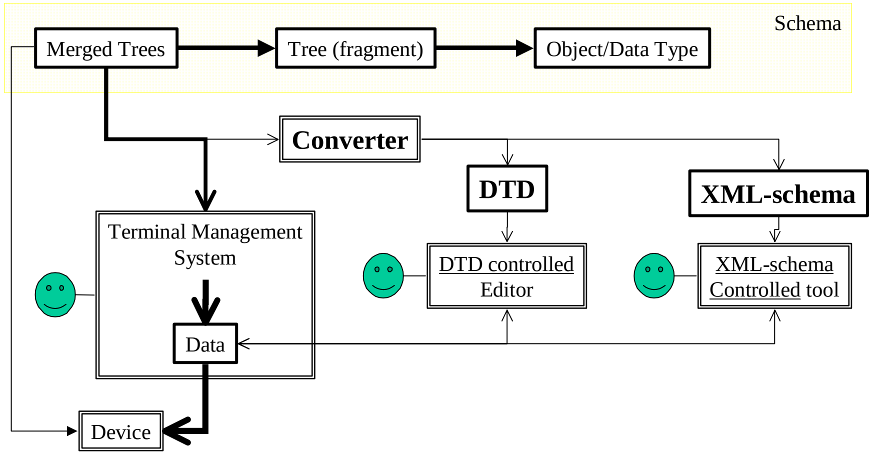
The following picture puts together the data description proposed with the DTD/XML-schema coding.

It also shows some usages of a Data Description:

- Terminal Management System
  The Data Description is defining the syntax and the format of the data sent to the device. The data values can be checked. The text describing the meaning of the parameters is fetched form the Data Description.

- Standard tools
  The Data Description can be translate to a XML DTD and an XML-schema. These can be used by tools, which understand DTDs and XML-schemas.

Data Description Architecture (2)



Annex C (Informative): Examples of Data Modelling Languages

Here follow a number of possible data modelling languages and principles.

Editor’s note: Annex C is kept for reference for the time being.

C.1 ASN.1

ASN.1 defines the abstract syntax of information but does not restrict in any way, how the information is encoded.

There are various ASN.1 encoding rules, which provide transfer syntax (a concrete representation) of the data values, whose abstract syntax is described in ASN.1. The standard ASN.1 encoding rules include:

- BER (Basic Encoding+ Rules),
- CER (Canonical Encoding Rules),
- DER (Distinguished Encoding Rules) and
- PER (Packed Encoding Rules).

C.2Interface Definition Language, IDL

The OMG Interface Definition Language (IDL) is the language used to describe the interfaces that client objects call and object implementations provide. An interface definition written in OMG IDL completely defines the interface and fully specifies each operation’s parameters. An OMG IDL interface provides the information needed to develop clients that use the interface’s operations.

CORBA 2.4.2 OMG IDL Syntax and Semantics chapter http://www.omg.org/cgi-bin/doc?formal/01-02-39

C.3 Unified Modelling Language, UML

The Unified Modelling Language (UML) is a graphical language for visualising, specifying, constructing, and documenting the artifacts of a software-intensive system.

The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.

OMG Modelling Specifications:
http://www.omg.org/technology/documents/formal/omg_modeling_specifications_avai.htm

UML Forum, a virtual community and knowledge portal that provides current information for modellers interested in UML: http://www.celigent.com/uml/

C.4 Document Type Definition, DTD and XML

The XML document type declaration contains or points to markup declarations that provide a grammar for a class of documents. This grammar is known as a document type definition, or DTD.

The DTD is defined in the XML specification.

XML home page: http://www.w3.org/XML/

Main specification: http://www.w3.org/TR/REC-xml

C.5 Resource Description Framework (RDF)

RDF integrates a variety of web-based metadata activities using XML as interchange syntax.

RDF home page: http://www.w3.org/RDF/

Model and Syntax Specification: http://www.w3.org/TR/REC-rdf-syntax/

Schema Specification 1.0: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

C.6 XML Schema

XML Schemas express shared vocabularies. It provides a means for defining the structure, content and semantics.

- XML home page: http://www.w3.org/XML/Schema
- XML Schema Part 0: Primer: http://www.w3.org/TR/xmlschema-0/
- XML Schema Part 1: Structures: http://www.w3.org/TR/xmlschema-1/
- XML Schema Part 2: Datatypes: http://www.w3.org/TR/xmlschema-2/

C.7 Composite Capability/Preference Profiles (CC/PP)

The W3C Metadata Activity addressed the combined needs of several groups for a common framework to express assertions about information on the Web. The primary work in this activity was the Resource Description Framework.

Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation is one of the W3C Metadata Activities.

Here are some links:

http://www.w3.org/Metadata/

- CC/PP Working Group: http://www.w3.org/Mobile/CCPP/
- CC/PP home page: http://www.w3.org/TR/NOTE-CCPP/
- Composite Capabilities/Preference Profiles: Requirements and Architecture: http://www.w3.org/TR/2000/WD-CCPP-ra-20000721/
- Composite Capability/Preference Profiles (CC/PP): Structure: http://www.w3.org/TR/2000/WD-CCPP-struct-20000721/
- CC/PP Attribute Vocabularies: http://www.w3.org/TR/CCPP-vocab/

C.8 Common Information Model (CIM)

The DMTF http://www.dmtf.org/index.html Common Information Model (CIM) is an approach to the management of systems, software, users, networks and more, that applies the basic structuring and conceptualisation techniques of the object-oriented paradigm.

A management model is provided to establish a common conceptual framework for a description of the managed environment. A fundamental taxonomy of objects is defined — both with respect to classification and association, and with respect to a basic set of classes intended to establish a common framework.

The white paper about Common Information Model (CIM) Core Model: http://www.dmtf.org/var/release/Whitepapers/DSP0111.htm gives a good introduction to CIM.

C.9 Language Independent Datatypes, LID

- ISO/IEC 11404:1995, Information technology - Language-Independent Datatypes
http://pueblo.lbl.gov/~olken/mendel/w3c/iso11404.html
- ISO/IEC TR 10182:1994 - Binding Techniques for Programming Languages
- ISO/IEC 13886:1996 - Language Independent Procedure Calling
- ISO/IEC TR 14369:1999 - Guidelines for the Preparation of Language Independent Service Specifications
http://wwwold.dkuug.dk/JTC1/SC22/WG11/docs/n455.rtf
- A taxonomy of datatypes http://www.kcl.ac.uk/kis/support/cit//staff/brian/taxosn.html

C.10 ISO/IEC 11179 - Specification and Standardization of Data Elements

http://pueblo.lbl.gov/~olken/X3L8/drafts/draft.docs.html

International Standard ISO/IEC 11179 parts are:

- Part 1: Framework for the Generation and Standardization of Data Elements;
- Part 2: Classification of Concepts for the Identification of Domains;
- Part 4: Rules and Guidelines for the Formulation of Data Definitions
- Part 5: Naming and Identification Principles for Data Elements; and
- Part 6: Registration of Data Elements.

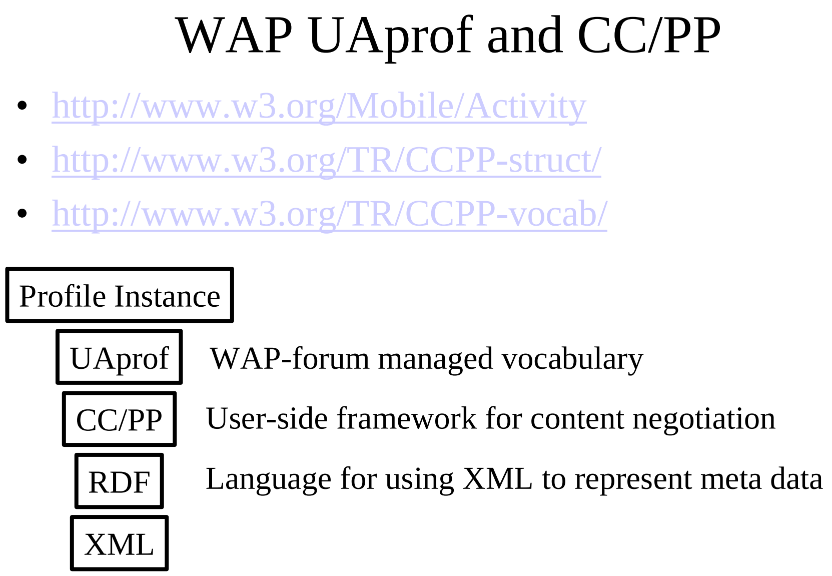
Annex D (informative): Examples of Vocabularies

Here follow examples on Vocabularies.

Editor's note: Annex D is kept for reference for the time being.

D.1 WAP UAProf

A WAP Forum specification. WAP UAProf Wireless Application Group, User Agent Profile Specification: http://www1.wapforum.org/tech/terms.asp?doc=SPEC-UAProf-19991110.pdf



WAP UAprof and CC/PP

- http://www.w3.org/Mobile/Activity
- http://www.w3.org/TR/CCPP-struct/
- http://www.w3.org/TR/CCPP-vocab/

Profile Instance

UAprof WAP-forum managed vocabulary

CC/PP User-side framework for content negotiation

RDF Language for using XML to represent meta data

XML

D.2 SyncML device specific information

The DevInf.DTD is intended to be used to exchange device specific information. Exchange of device specific information such as available memory and item identifiers, supported local databases is a prerequisite to successful data synchronization. http://www.syncml.org/docs/syncml_devinf_v10_20001207.pdf

D.3 A Comparison of Schemas for Video Metadata Representation

This could give some inputs. http://www8.org/w8-papers/3c-hypermedia-video/comparison/comparison.html

D.4 vCard and vCalendar

vCard and vCalendar defines a transport and platform-independent format for exchanging personal information typically found on a traditional business card calendaring and scheduling information.

Personal Data Interchange (PDI) http://www.imc.org/pdi/

vCard in RDF draft-iannella-vcard-rdf-00.txt http://www.oasis-open.org/cover/draft-dawson-vcard-xml-dtd-00.txt

Annex E (informative): Example of Data Type Description

Data type “screenCoordinate”

This XML document describes a composite datatype screenCoordinate. It has two items x and y of type xCoordinate and yCoordinate. XCoordinate can have values 0 to 60.

```
<?xml version="1.0" encoding="UTF-8"?>

<dataTypes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ecsbojn\proj\dataDescription\DDcurrent\3GPPdataType.xsd">


   <compositeType name="screenCoordinate">

      <item name="x" dataType="xCoordinate"/>

      <item name="y" dataType="yCoordinate"/>

   </compositeType>


   <atomicType name="xCoordinate" base="int">

      <minInclusive value="0"/>

      <maxInclusive value="60"/>

   </atomicType>


   <atomicType name="yCoordinate" base="int">

      <minInclusive value="0"/>

      <maxInclusive value="30"/>

   </atomicType>


</dataTypes>
```

XML representation of data

This XML document is an example of how data of datatype screenCoordinate can be represented.

The item names x and y is used as tags. The format of text in the x-tag and y-tag follows the XML-schema specification.

```
<?xml version="1.0" encoding="UTF-8"?>

<screenPositionS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ecsbojn\proj\dataDescription\DDcurrent\dataTypeExample.xsd">

   <screenPosition>

      <x>12</x>

      <y>15</y>

   </screenPosition>

   <screenPosition>
```

```
        <x>5</x>
        <y>13</y>
    </screenPosition>
</screenPositionS>
```

Full XML-schema

This XML document is a XML-schema defining the constrains on a XML-document containing screen Positions. The part in **bold** text can automatically be generated from the data description of screenCoordinate.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0 beta 3.1 build Aug 27 2001 (http://www.xmlspy.com) by Bo Johansson (Ericsson Mobile Communications AB) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="screenPositionS">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="screenPosition" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="screenPosition" type="screenCoordinate">
        <xs:annotation>
            <xs:documentation>Comment describing screenPosition</xs:documentation>
        </xs:annotation>
    </xs:element>

    <xs:complexType name="screenCoordinate">
        <xs:sequence>
            <xs:element name="x" type="xCoordinate"/>
            <xs:element name="y" type="yCoordinate"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="xCoordinate">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="60"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="yCoordinate">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="30"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Annex F Examples of Datatype Definitions

F.1. Introduction

This annex shows some examples of datatypes definition.

F.2. Test Datatype Definitions Examples

The datatypes tested are called:

- "more-than-ninety-nine"
- "booleanOrDate"
- "screenCoordinate"
- "wapId"

F.2.1 Test datatypes definition according to the UP-010089

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0 U (http://www.xmlspy.com) by Bo Johansson (Ericsson Mobile Platforms AB) -->
<?xml-stylesheet type="text/xsl" href="3GPPdatatype.xslt"?>
<datatypes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="3GPPdatatype.xsd">
    <atomicType name="more-than-ninety-nine">
        <restriction base="int">
            <minExclusive value="99"/>
        </restriction>
    </atomicType>
    <atomicType name="booleanOrDate">
        <union>
            <member ref="xs:boolean"/>
            <member ref="xs:date"/>
        </union>
    </atomicType>
    <recordType name="screenCoordinate">
        <field name="x" datatype="coordinateX"/>
        <field name="y" datatype="coordinateY"/>
    </recordType>
    <atomicType name="coordinateX" base="int">
        <restriction base="int">
            <minInclusive value="0"/>
            <maxInclusive value="60"/>
        </restriction>
    </atomicType>
    <atomicType name="coordinateY" base="int">
        <restriction base="int">
            <minInclusive value="0"/>
            <maxInclusive value="30"/>
        </restriction>
    </atomicType>
    <atomicType name="wapId">
        <semantic>
```

```
            <label lang="en">Globally unique wap id</label>
            <definition xml:lang="en">
     Uniqueness MUST be obtained by either using a fully qualified Internet domain name
     (i.e. hostname as defined in section 3.2.2 of {RFC2396])
     or a globally unique IP address (IPv4 {RFC791] in decimal format with dots as delimiters
     or IPv6 {RFC2373], as hexadecimal numbers with colons as delimiters or as a combination of
hexadecimal and decimal numbers with dots and colons as delimiters)
     </definition>
            <description lang="en">
            http://www.ietf.org/rfc/rfc2396.txt
            http://www.ietf.org/rfc/rfc791.txt
            http://www.ietf.org/rfc/rfc2373.txt
     </description>
        </semantic>
        <union>
            <member ref="wapfqIDN"/>
            <member ref="IPv4Add"/>
            <member ref="IPv6Add"/>
        </union>
    </atomicType>
    <atomicType name="wapfqIDN">
        <semantic>
            <label lang="en">Fully qualfied host name</label>
            <description lang="en">
     http://www.ietf.org/rfc/rfc2396.txt
     </description>
        </semantic>
        <restriction base="string">
            <pattern value=".+"/>
        </restriction>
    </atomicType>
    <atomicType name="IPv4Add">
        <semantic>
            <label lang="en">IPv4 address</label>
            <description lang="en">
     http://www.ietf.org/rfc/rfc791.txt
          </description>
        </semantic>
        <restriction base="string">
            <pattern value="\d{1,3}(.\d{1,3}){3}"/>
        </restriction>
    </atomicType>
    <atomicType name="IPv6Add">
        <semantic>
            <label lang="en">IPv6 address</label>
            <description lang="en">
     http://www.ietf.org/rfc/rfc2373.txt
     </description>
        </semantic>
        <restriction base="string">
```

```
            <pattern value="[0-9a-fA-F]{0,4}(:[0-9a-fA-F]{0,4}){7}|[0-9a-fA-F]{0,4}(:[0-9a-fA-F]{0,4}){5}(.\d{1,3}){4}"/>
        </restriction>
    </atomicType>
</datatypes>
```

F.2.2 Test datatypes as expanded XML-schema:

```
  <?xml version="1.0" encoding="UTF-16" ?>
- <xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <xs:simpleType name="more-than-ninety-nine" final="list restriction">
    - <xs:restriction base="xs:int">
        <xs:minExclusive value="99" />
      </xs:restriction>
    </xs:simpleType>
  - <xs:simpleType name="booleanOrDate" final="#all">
      <xs:union memberTypes="xs:boolean xs:date" />
    </xs:simpleType>
  - <xs:complexType name="screenCoordinate">
    - <xs:sequence>
        <xs:element name="x" type="coordinateX" />
        <xs:element name="y" type="coordinateY" />
      </xs:sequence>
    </xs:complexType>
  - <xs:simpleType name="coordinateX" final="list restriction">
    - <xs:restriction base="xs:int">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="60" />
      </xs:restriction>
    </xs:simpleType>
  - <xs:simpleType name="coordinateY" final="list restriction">
    - <xs:restriction base="xs:int">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="30" />
      </xs:restriction>
  </xs:simpleType>
- <xs:simpleType name="wapId">
  - <xs:annotation>
      <xs:documentation>Globally unique wap id</xs:documentation>
      <xs:documentation>Uniqueness MUST be obtained by either using
        a fully qualified Internet domain name (i.e. hostname as
        defined in section 3.2.2 of {RFC2396]) or a globally unique IP
        address (IPv4 {RFC791] in decimal format with dots as
        delimiters or IPv6 {RFC2373], as hexadecimal numbers with
        colons as delimiters or as a combination of hexadecimal and
        decimal numbers with dots and colons as
        delimiters)</xs:documentation>
      <xs:documentation>http://www.ietf.org/rfc/rfc2396.txt
        http://www.ietf.org/rfc/rfc791.txt
        http://www.ietf.org/rfc/rfc2373.txt</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="wapfqIDN IPv4Add IPv6Add" />
  </xs:simpleType>
- <xs:simpleType name="wapfqIDN">
  - <xs:annotation>
      <xs:documentation>Fully qualfied host name</xs:documentation>

        <xs:documentation>http://www.ietf.org/rfc/rfc2396.txt</xs:do
        cumentation>
    </xs:annotation>
  - <xs:restriction base="xs:string">
      <xs:pattern value=".+" />
    </xs:restriction>
  </xs:simpleType>
- <xs:simpleType name="IPv4Add">
  - <xs:annotation>
      <xs:documentation>IPv4 address</xs:documentation>

        <xs:documentation>http://www.ietf.org/rfc/rfc791.txt</xs:doc
        umentation>
    </xs:annotation>
  - <xs:restriction base="xs:string">
      <xs:pattern value="\d{1,3}(.\d{1,3}){3}" />
      </xs:restriction>
    </xs:simpleType>
  - <xs:simpleType name="IPv6Add">
    - <xs:annotation>
        <xs:documentation>IPv6 address</xs:documentation>
          <xs:documentation>http://www.ietf.org/rfc/rfc2373.txt</xs:documentation>
      </xs:annotation>
    - <xs:restriction base="xs:string">
        <xs:pattern value="[0-9a-fA-F]{0,4}(:[0-9a-fA-F]{0,4}){7}|[0-9a-fA-F]{0,4}(:[0-9a-fA-F]{0,4}){5}(.\d{1,3}){4}" />
      </xs:restriction>
    </xs:simpleType>
  - <xs:element name="testDatatype">
    - <xs:complexType>
      - <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="more-than-ninety-nine" type="more-than-ninety-nine" />
          <xs:element name="booleanOrDate" type="booleanOrDate" />
          <xs:element name="screenCoordinate" type="screenCoordinate" />
          <xs:element name="coordinateX" type="coordinateX" />
          <xs:element name="coordinateY" type="coordinateY" />
          <xs:element name="wapId" type="wapId" />
          <xs:element name="wapfqIDN" type="wapfqIDN" />
          <xs:element name="IPv4Add" type="IPv4Add" />
          <xs:element name="IPv6Add" type="IPv6Add" />
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

F.2.3 Generated documentation of the test datatype XML-schema.

test_datatype_html.zip

Annex <X> (informative): Change history

It is usual to include an annex (usually the final annex of the document) for specifications under TSG change control which details the change history of the specification using a table as follows:

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			v.0.2.1 After UP#07 Editorial changes based on UP-010118: chapter 11.1 moved after chapter 5, becoming chapter 6		

3GPP Joint Ad-Hoc on Generic User Profile UP-010143
Cancun Mexico, 3-5 December 2001

3GPP TS 24.241 V0.2.0 (2001-12)

3rd Generation Partnership Project;
Technical Specification Group Terminals;
3GPP Generic User Profile Common Objects;
Stage 3
(Release x)

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 Organisational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organisational 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 Organisational Partners' Publications Offices.

Reference

3TS/TSGT-02xxxxx

Keywords

R4 Specification 3G

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

Contents

Foreword

This Technical Specification has been produced by the 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 this TS, 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 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 specification;

1 Scope

This specification serves as a vessel to manage the process of adding new datatypes, Generic User Profile fragments, and other constructs for use in 3GPP applications within various specifications.

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.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] 3GPP TS 27.103 V4.1.0 (2001-04), 3rd Generation Partnership Project; Technical Specification Group Terminals; Wide Area Network Synchronisation Standard (Release 4)

[2] 3GPP 22.240

[3] 3GPP 23.240

[4] 3GPP 23.241

3 Definitions and abbreviations

3.1 Definitions

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

vObject: ….

data store ..

Datatype

GUP fragment

…

3.2 Abbreviations

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

[tbd]

4 Background

[Editor's note: to be redrafted]

The request for data synchronisation support for the VHE MExE User Profile extensions brought up the long term need to define standards for and manage the process of adding new vObjects and other constructs as data store types for use in data synchronisation activities. Managing this process targets the following areas:

- Definition of new vObject and Other Constructs standardised formats for use in data synchronisation as required by other groups within 3GPP.
- Management of the process of publishing these new standardised formats for use within and external to 3GPP.
- Support of terminal and network interoperability through the use of a standardised approach to the definition of these new formats.
- Extension of the usefulness of the TSG-T2-defined data synchronisation architecture and mechanisms to new data store semantic content.
- Identification of required protocols and development, if needed

The vObjects and other constructs listed in Section 6 should enhance interoperability and be implemented in a way that ensures backwards compatibility, where possible.

Standardised vObject and Other Construct formats must allow users and operators to keep local copies up to date with remotely stored copies of the user's and the operator's mission-critical data stores in a manner that will allow data synchronisation to a wide variety of potentially disparate data stores. These standardised formats must allow rapid expansion of the nature and type of future data store enhancements.

Data synchronisation of vObjects and Other Constructs should standardise charging mechanisms, especially in roaming situations and between different operators. Other charging mechanisms (e.g. air time) may be needed when data synchronization of vObjects and Other Constructs is attempted outside of the operator's domain.

5 Process for Addition of New Datatypes

[tbd]

6 Process for Addition of New Fragments

[tbd]

7 Process for Addition of New Common Objects

[tbd]

8 Process for Addition of New vObjects or Other Constructs

[Editor’s Note: to be redrafted]

8.1 New vObjects or Other Constructs

New vObjects or other constructs shall be defined in a stand-alone 3GPP specification. This specification may be a wholly self-contained definition or it may simply be a reference to an independent SDO’s specification, where such exists.

8.2 Formal Recognition

Formal recognition of the new vObject or other construct shall be through the use of a CR to this specification requesting the addition to Section 6 of the specification of the vObject or other construct to be recognized.

8.3 Approval Process

[tbd]

Annex A (Normative)

Generic User Profile Components

Annex A1 Datatypes

Annex A2 GUP Fragments

Annex A3 Other Constructs

Annex B (Informative)

Parameters for Component Construction

Annex B1 GPRS Parameters

Parameter Requirements	Data Description	Generated Documentation	Generated XML Schema
UP-010091 (GPRS parameters).zip	gprsDatatype.xml	gprsDatatype.htm	gprsDatatype.xsd

[Please note that these files are included here for information only. The final structure (i.e., to include the object files within this specification or to reference them as external specifications) has not been decided.

Please note, also, that these are very early, representative drafts. More work must be done by experts on GPRS prior to these being used by developers.]

Annex B2 Subscription Management

[tbd]

Annex B3 MMS

[tbd]

Annex C (Normative)

Recognised vObjects and Other Constructs

Specification	Title	Comment(s)
(T2-000676)	Bookmark	Bookmark URL standard

Annex D:
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2001-11					Submitted by T2 to TSG-T for preliminary information		0.1.0
2001-12					Added GPRS examples to Annex B	0.1.0	0.2.0