

IEEE P1900.4 WG

Information Model

Date: **2008-03-26; #4, Washington**

Authors:

Name	Company	Address	Phone	email
Mahesh Sooriyabandara, Tim Farnham	TREL			
Oliver Holland, Alireza Attar	KCL			
Makis Stamatelatos, Costas Polychronopoulos	UoA			
Klaus Nolte	Alcatel-Lucent			

Notice: This document has been prepared to assist IEEE P1900. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.

Release: The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE P1900.

Patent Policy and Procedures: The contributor is familiar with the IEEE P1900 Patent Policy and Procedures <<http://iee802.org/guides/bylaws/sb-bylaws.pdf>>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <stephen.berger@ieee.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within IEEE P1900. **If you have questions, contact the IEEE Patent Committee Administrator at** <patcom@ieee.org>.

Part(s) of the P1900.4 D0.01 (approved in Madrid) addressed by the document?

4. System Architecture	
5. Use Cases	
6. General System Requirements	
7. Functional baseline Architecture	
8. Information Model and Representation	✓
9. Procedures	
Other : <i>(please detail)</i>	

Purpose

- ➔ Progress report on Information Modelling activities
- ➔ Refined Manage Objects and associated Meta Data relevant to P1900.4 scenarios
- ➔ New data types as UML classes
- ➔ Refined Key Information Elements & their relationships
 - Class Diagrams

Overview

- ➔ Class Description Tables
- ➔ Meta-Data Managed Object
- ➔ Terminal Managed Object
- ➔ RAN Managed Object
- ➔ Assumed functions for usage examples
- ➔ Examples of scenarios

Class Description Tables

Class <Class name> [(abstract class)]

<Description of the class>

•DERIVED FROM

<List of super-classes>

ATTRIBUTES

<Attribute name>

<Attribute value type>

<Attribute
access qualifier>

Default: <Default
value>

<Description of the attribute>

CONTAINED IN

<List of classes, whose instances may contain an instance of this class. If this class is an abstract class - i.e. it is used for further refinement only and will never be instantiated – then this list is empty>

CONTAINS

<List of classes, whose instances may be contained in an instance of this class. Constraints for the number of instances are marked with [*], [+], [<n>] or [<m>-<n>] >

SUPPORTED EVENTS

<List of event names that are detected by this class and lead potentially to a corresponding event report>

Time: A type used to present a time (e.g. GeneralizedTime from ASN.1)
Name: A type used to assign to an object a name (usually string or number, f.f.s.)
Other self-explaining types
ASN.1 syntax used

Read', 'Write', 'Read-Write', 'Add-Remove' (for list-type attributes), 'Read-Add-Remove', 'None' (for the purpose of report parameters or value characteristics only)

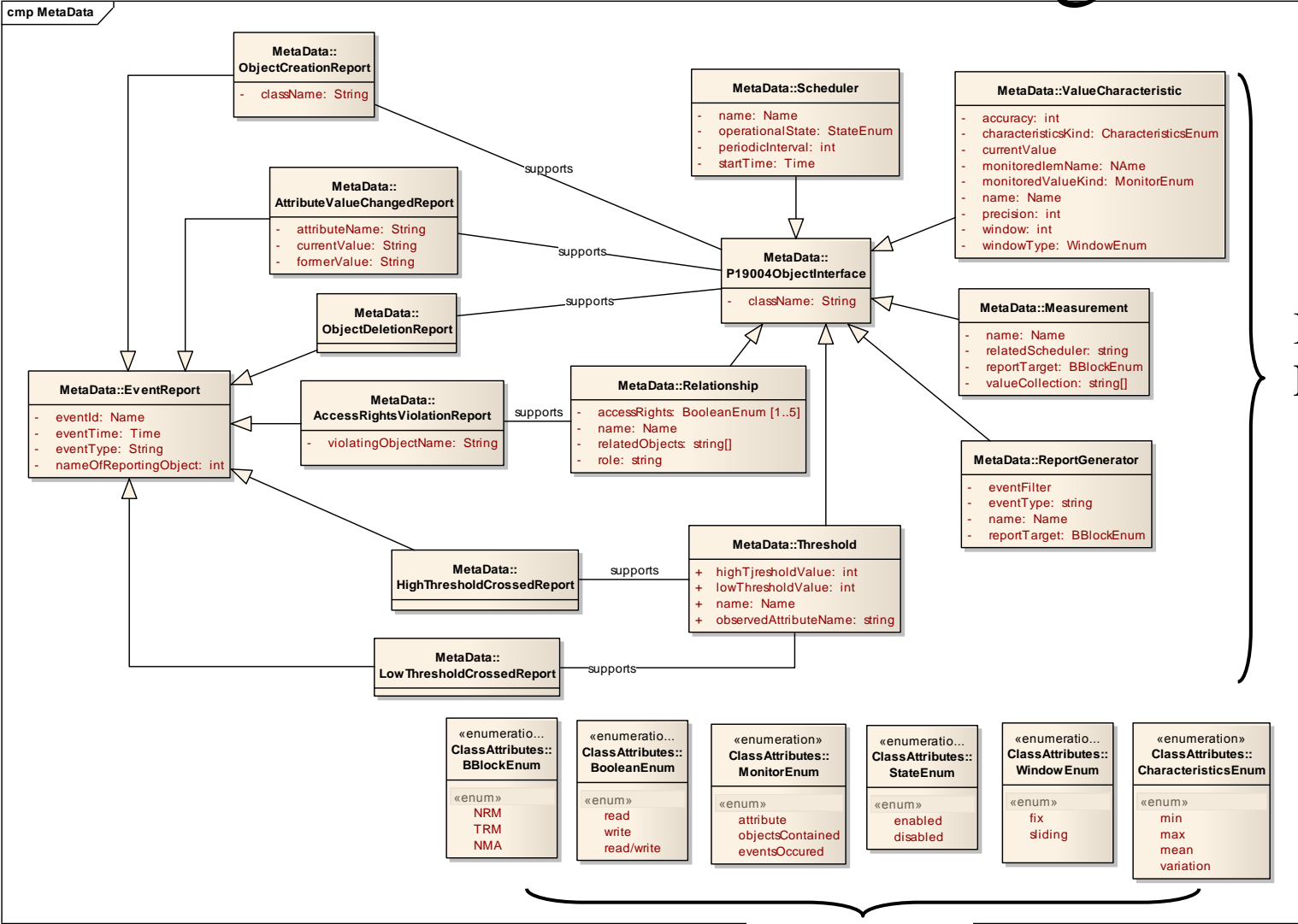
Meta Data Managed Object

- ➔ **Meta Data:** Provides Additional information about the managed Object
- ➔ **Meta Data Classes include:**
 - **P19004ObjectInterface:** Mandatory base interface for all P1900.4 class definitions.
 - **EventReport:** Mandatory base interface for all P1900.4 event reports.
 - **ObjectCreationReport:** Event report generated (on request) after a new managed object has been created.
 - **ObjectDeletionReport:** Event report generated (on request) after a new managed object has been created.
 - **AttributeValueChangedReport:** Event report generated (on request) after the value of an attribute has been changed.

Meta Data Managed Object

- ➔ **Meta Data** Classes include (2):
 - **Threshold**: provides the possibility to define various threshold crossed event reports for an object.
 - **ValueCharacteristic**: allows for defining different statistics to be calculated for attribute values.
 - **Relationship**: provides the possibility to define various relationships to other objects.
 - **Measurement**: provides the possibility to define various measurements for an object.
 - **Scheduler**: providing the possibility to define various schedulers within an object that might be needed to schedule different operations in a common way
 - **ReportGenerator**: providing the possibility to define various reports on reportable events of an object.

Meta Data Class Diagram

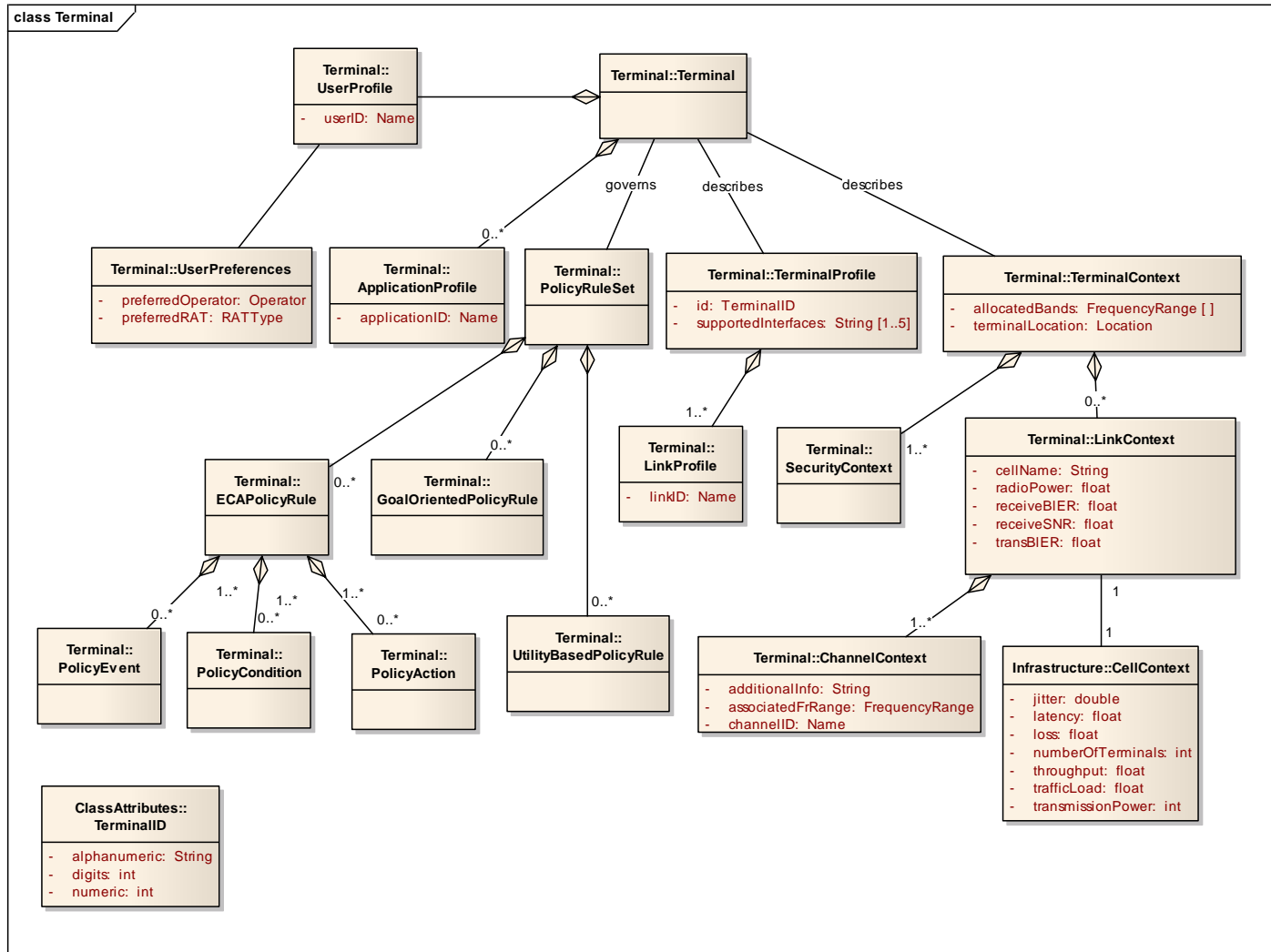


Information Elements

Managed Objects in Terminals

- ➔ Manage Objects includes
 - **User Profile – User Preferences**
 - **Application Profile**
 - **Policy Rule Set:** governs terminal behaviour in terms of selecting link configurations
 - **Terminal Profile:** includes attributes such as TerminalId, SupportedInterfaces
 - **Link Profile:** abstracts the information for a link to a RAN
 - **Link Context:** includes attributes such as associatedChannels, radioPower...
 - **Terminal Context:** includes information about the terminal location and the allocated bands
 - **Channel Context:** abstracts the basic radio resource and associated measurements

Terminal Managed Objects



Information Elements

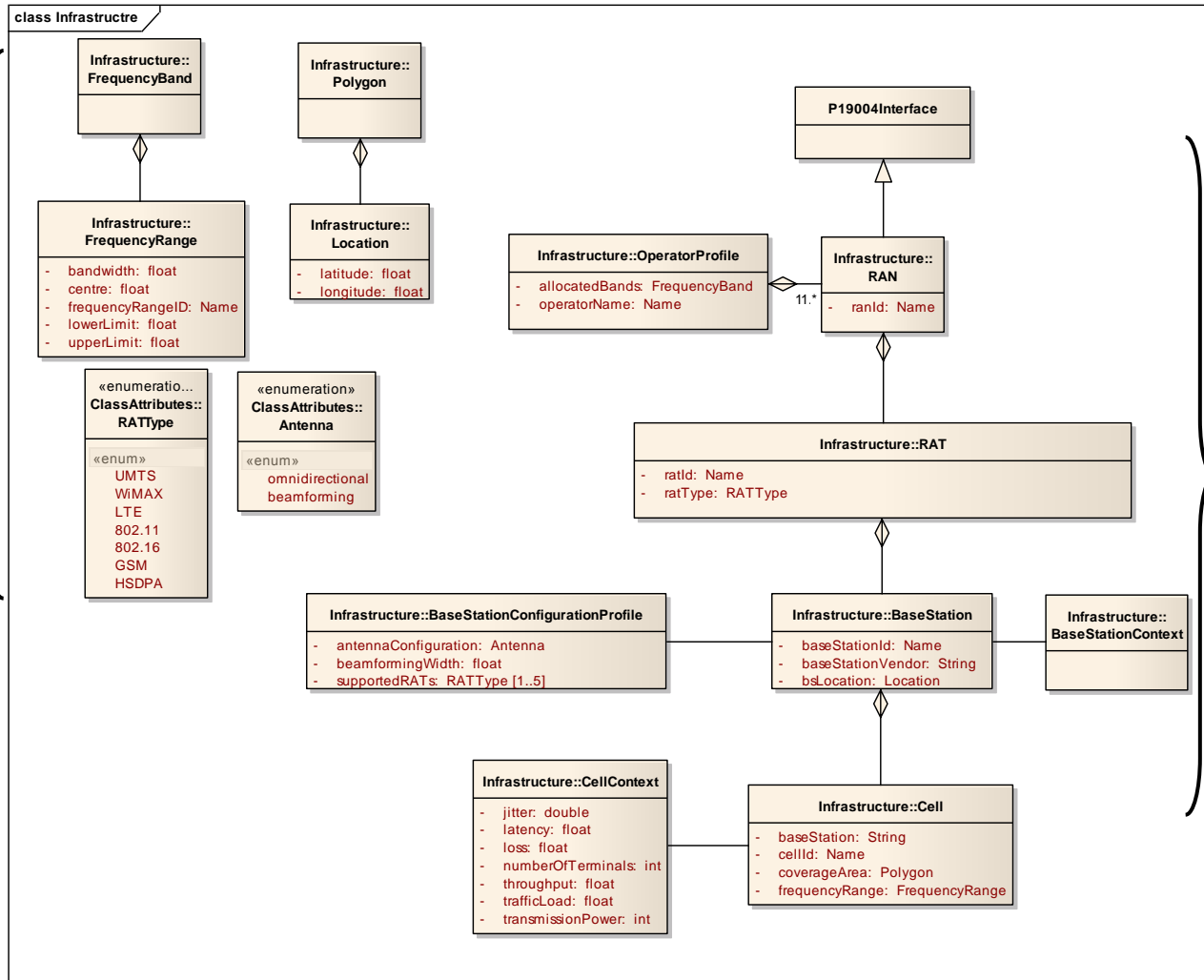
Data types

RAN Managed Objects

- ➔ RAN Managed Object includes
 - RAN – class providing RAN related information (ranID, ...)
 - Operator Profile providing information about the RAN Operator (name, allocated bands)
 - RAT class as being aggregated by RAN class including attributes about the RAT type and identification
 - RATType class is defined as enumerating various RAT types (UMTS, 802.11, WLAN,)
 - BaseStation class providing information about the vendor, location, ...
 - BaseStationConfigurationProfile includes the configuration of the antenna of the base station, the RAT types that are supported by the base station, ...
 - Cell class incorporates the serving base station, the area of coverage, the frequency range, ...
 - CellContext incorporates attributes about the load, transmission power, number of supported RATs, ...

RAN Managed Objects

Data types



Information Elements

Assumed functions for usage examples

- ➔ A number of functions are assumed to be available at the managed system side.
- ➔ Local Functions (not necessarily exposed to an outside interface)
 - ObjectSet (baseObject, level, filter)
 - ObjectCount (baseObject, level, filter)
 - EventCount (eventType, timeInterval, baseObject, level, filter)
 - Ensure (baseObject, level, filter, valueAssertion, timeslot)

Assumed functions for usage examples

- ➔ Functions for the external management interface
 - SetAttributeValue (baseObject, level, filter[, attributeName, value]+)
 - GetAttributeValue (baseObject, level, filter[, attributeName]+)
 - CreateObject (baseObject, className[, attributeName, value]+)
 - DeleteObject (objectName)
 - EventReport(reportType, reportingObject, reportData)

Examples of scenarios

- ➔ Example Scenario within the scope of Distributed Radio Resource Optimization use-case
- ➔ The example scenario includes:
 - Text description for the involved information elements and the corresponding object operations activated during the scenario
 - Message Sequence Chart illustration of the scenario

Policy Definition

- ➔ Policy represents only “management” of objects
 - Chosen Representation - ECA
 - ON <event> IF <condition> DO <action>
 - Policy “distribution” (i.e. filtering and addressing) is not modelled in the Information Model, because there is no need to constrain the information model to a particular choice of policy distribution selection method (our aim is to keep to IM independent of any implementation)
 - Policy Distribution could use many different approaches – broadcast, unicast, multicast etc.
 - Conditions and/or distribution method can be used to narrow scope as preferred by the operator

Policy - Example

```

policy {
    ID: <policyID>
    VALIDITY: <timeConstraint> OPTIONAL
    ON: <event>
    IF: <condition>
    THEN: <action>
}

```

```

event {
    eventType: objectCreation | objectDeletion |
              stateChanged | attributeChanged |
              scheduledTimer | ...
    eventQualifier: depending on eventType
}

```

```

condition {
    logical expression built from elementary logical expressions
}

```

Elementary logical expression: $A <, =, >, != B$ where A and B are figures or the number of objects satisfying given constraints that are expressed by a conceptual function:

ObjectCount (baseObject, scope, filter) where

- baseObject (by ID, eventObject)
- scope (baseObject itself, all directly contained objects, all contained objects)
- filter (attribute value conditions, class conditions)

```

action {
    sequence of attribute value restriction (may force reconfigurations)
}

```

A conceptual function is used to describe those restrictions:

Ensure (baseObject, scope, filter, valueAssertion, timeslot) where

- baseObject, scope, filter same as in conditions
- valueAssertion (set|avoid, attributeName, valueSet)
- timeslot: time for the action to be completed

```

NRM → TRM: AddPolicy (
  ID: "periodicSelection"
  ON:
  eventType: scheduledTimer,
  eventQualifier:
    schedulerStart: 8:00:00:000 +
<TerminalID>%10000,
  interval: 10s,
  schedulerStop: 9:00:00:000
  IF: true
  THEN:
  Ensure (
    terminal,
    all directly contained objects,
    class: link,
    (set, linkRAT, ["LTE", "802.16", "802.11"]), 3 ms
  );
  Ensure (
    terminal,
    all contained objects,
    class: SNR or BER or RadioPowerIndicator,
    (set,
    DynamicMetaData.ValueCharacteristic.Mean.Activity,
    report), 2 ms
  );
  Ensure (
    terminal,
    all contained objects,
    class: SNR or BER or RadioPowerIndicator,
    (set,
    DynamicMetaData.ValueCharacteristic.Mean.Activity,
    reset), 0 ms
  )
)

```

ALU and UoA would like to acknowledge their work in this contribution to E3 project which has received research funding from the Community's Seventh Framework programme. This presentation reflects only the author's views, and the Community is not liable for any use that may be made of the information contained herein.