Project	**IEEE P1900.4 WG** http://www.ieeep1900.org/	
Title	**Contribution describing Common Objects for P1900.4-07-04-11r4**	
DCN	P1900.4-07-06-91	
Date Submitted	**2007-12-06**	
Source(s)	Editor: John Strassner Contributors: Klaus Nolte, Tim Farnham, Mahesh Sooriyabandara, Makis Stamatelatos, Oliver Holland, Vladimir Ivanov, Soodesh Buljore Others present: Kentaro Ishizu, Go Miyamoto, Homare Murakami, …	Sub-group contributors : Motorola , Alcatel-Lucent, Toshiba, UoA, KCL, NICT, Intel Other sub-group members : Hitachi, KKE, Willnet, Tokyo University of Science, BAe Systems, Worldpicom, Pultek, Cosmo Research
Re:	IEEE P1900.4 Berlin meeting 3-7 December 2007	
Abstract	This documents provides a new section to the information model that describes common managed objects and, in particular, common policy managed objects.	
Purpose	Text proposal for the baseline document chapter on the information model	
Notice	This document has been prepared to assist the IEEE P1900.4 Study Group. 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 IEEE P1900.B SG may make this contribution public.	
Patent	The contributor is familiar with IEEE patent policy, as outlined in Section 6.3 of the IEEE-SA	

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

Please tick the section of the Baseline Document D0.01 addressed by this contribution.

4. System Architecture *Overall system architecture with main interfaces (subject to standardization) between the building blocks. Note: in this section generic interfaces should be considered only.*	
5. Use Cases	
5.1 Dynamic Spectrum Allocation	
5.2 Dynamic Spectrum Access	
5.3 Distributed Radio Resource usage optimization	
6. General System Requirements *This section contains a) generic requirements (use-case agnostic or common to all of the use cases). Note: the nature of this section (text only AND/OR high level modeling should be defined during the course of document "development".*	
7. Functional baseline Architecture	
7.1 Dynamic Spectrum Allocation design realization	
7.2 Dynamic Spectrum Access design realization	
7.3 Distributed Radio Resource usage optimization design realization	
8. Information Model and Representation	✓
9. Procedures *This section contains the procedures the TRM should follow in order to "consume" the information (of section 8) conveyed of the radio enabler. This section should also capture the behavior of the TRM with respect to the policies. Note: Working assumption: The protocol aspects are considered to be informative and should be taken into account by the protocol task group which will design the actual protocol.*	

9.1 State Diagram(s)	
10. Annex	
Other : *(please detail)*	

This contribution is based on the baseline document D1.0. It contains an updated proposal for an addition to chapter 9 (section 9.3) of the information model. This contribution describes the concept of "common" managed objects and, in particular, describes common policy managed objects. It does this in two steps. First, this document describes an exemplary scenario involving the main P1900.4 stakeholders: the regulator, the operator, and the user/terminal. Second, this document proposes text that describes, at a high-level, different types of managed policy objects that are required to support this scenario.

As this is a new contribution, ALL text is new.

Text begin

9 Information Model and Representation

9.1 Common Managed Objects

The purpose of this chapter is to describe the concept of “common” managed objects. It does this in two steps. First, an exemplary scenario is described that involves the main P1900.4 stakeholders: the regulator, the operator, and the user/terminal. Second, different types of managed policy objects that are required to support this scenario are described.

9.1.1 Rationale for Common Objects

The dramatic increase in system, business and technical complexity in network technologies, devices, management approaches, and functionality has been well documented. The primary motivation for using an information model is to define a shared, interoperable foundation for representing the capabilities, functionality and behavior of entities in a system.

Current modeling approaches tend to be focused on one specific aspect, such as representing the functionality of a network. This excludes other aspects, such as business requirements and goals. In particular, for the purpose of P1900, this results in the *inability to represent how changing user and business requirements affect the services and resources offered by the network*. Hence, we need a means to represent the characteristics and behavior of system elements. ITU-T recommendations, along with IETF and other fora documents, describe current state data for managed objects, but do not describe how to manage the lifecycle aspects of managed objects.

The purpose of an information model is to describe the characteristics and behavior of managed entities, as well as the relationship between managed entities, in an unambiguous fashion. Managed entities represent concepts of interest to people using the model.

The purpose of having a single information model is to be able to relate different entities to each other. For example, consider policies applied to govern a mobile terminal. If the policies (and their components), the mobile terminal, the services, and the network are all modeled in a single information model, then they can be directly related to each other in a consistent language, with consistent syntax and semantics. If they exist in separate models, then even if each model uses the same underlying language (e.g., MOF for UML), there is the danger that syntax and semantics will not be consistent. This in turn means that the resulting functionality and behavior may not be able to be predicted.

The following subsections define the basic types of common objects that a P1900 system (a) *must* be used, (b) are *recommended* to be used (but are not required), and (c) are completely *optional* (i.e., implementation dependent).

9.1.2 Common Application Scenario

The purpose of this section is to describe a common application scenario that involves the three main stakeholders in a P1900-based system: the *regulator*, the *operator*, and the *user/terminal*. This scenario defines the context for understanding why managed objects are needed for a typical P1900 scenario.

It should be noted that such objects are exemplary and are *not inclusive*.

Consider a user using a mobile terminal. As the user moves, the mobile terminal will be subjected to different effects in the environment which may cause the mobile terminal to opportunistically search for available spectrum. If this happens, then the mobile must obey any regulatory policies that govern this particular operational area. Of course, one would hope that if the mobile decided *not* to observe this etiquette, then it is incumbent on the operator to enforce compliance.

This scenario will be used for all subsequent subsections of subsection 9.1.

9.1.3 Common Policy Objects

Policies are necessary because of the growing complexity of network elements, along with the need to manage network and application resources according to business rules and procedures. Policy is important, because policy is used to map how different applications use network resources, and in the event of network congestion, which applications get priority use of which network resources. Policies define which entities can access which resources and services, and how much of a given resource or service is allocated to a particular client. Policies enable the fine-tuning of resource allocation and service assignment based on the changing environment, such as time-of-day, cost of service, or client identity.

This section describes the concept of common *policy* managed objects.

9.1.3.1 Application Of P1900 Policy Rules to the Example Scenario

There are three main stakeholders: regulator, operator and user/terminal. Each of these stakeholders can, in theory, use one or more P1900-PolicyRules (this term is used to differentiate such policy rules from other uses of the word “policy” and “policy rule” to (1) control or otherwise change their functionality, (2) control or otherwise change the functionality of other entities under their jurisdiction, and (3) control or otherwise change the functionality of entities that a stakeholder owns.

Figure 1 shows that, even if separate languages exist for all three main stakeholders, each language must have a common set of syntax and semantics that enable the regulatory policy rules to establish what can and cannot be performed, so that the operator can ensure that all emissions from its network by its subscribers are compliant with those regulatory rules. Furthermore, since a user/terminal uses the services of an operator, the oval representing the language of the user/terminal is fully contained in the oval representing the language of the operator.

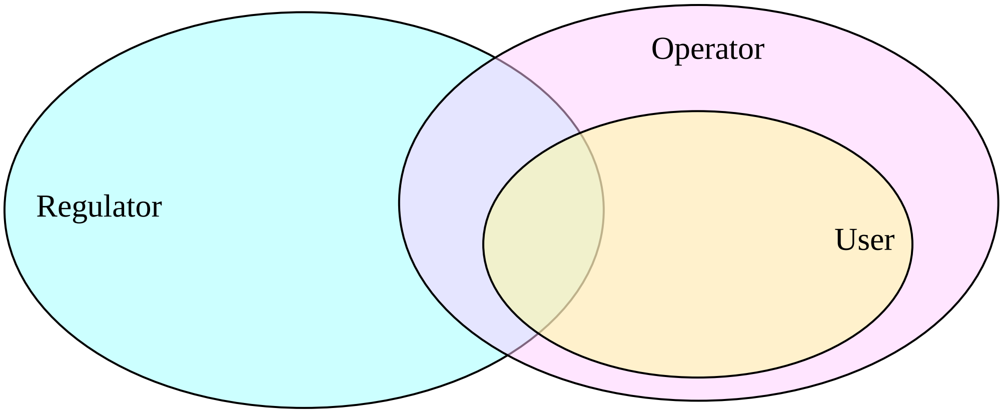

Figure 1. Content Commonality Between P1900-PolicyRules

9.1.3.2 Regulator Policy

The regulator will write P1900-PolicyRules in order to define, in a machine-readable form, the characteristics and behavior of allowed emissions for a given set of constraints (e.g., location and time) for a given set of metrics (e.g., interference and power).

9.1.3.3 Operator Policy

The operator will write P1900-PolicyRules in order to define, in a machine-readable form, the characteristics and behavior of networks that it owns and the mobiles that use services from the networks that it owns. Due to the complexity of operator networks, there are two general sub-types of operator P1900-PolicyRules, as described below.

9.1.3.3.1 Network management policies

Network Management P1900-PolicyRules define, in a machine-readable form, how a network is managed. This includes, but is not limited to, functions such as managing the configuration, security, and access (physical and logical) to groups of network devices and functions; accounting and auditing of network traffic; fault and performance analysis; and other functions that have to do with determining the overall health and performance of the network and its components.

9.1.3.3.2 Equipment management policies

Equipment Management P1900-PolicyRules define, in a machine-readable form, how a network device is managed. This is similar to Network Management P1900-PolicyRules, except that Equipment Management P1900-PolicyRules are used to define functions for individual devices, not for groups and networks of devices.

9.1.3.4 User / terminal Policy

The User, and optionally the Manufacturer of the terminal, may also be able to write P1900-PolicyRules in order to define, in a machine-readable form, the characteristics and behavior of the terminal. The User *asserts* his or her desires onto the terminal in the form of policies that define particular functions or tasks; these often take the form of *Profiles* and *Preferences*. Similarly, the Manufacturer of the terminal may create an “intelligent” terminal that contains one or more programmable functions that can be controlled using P1900-PolicyRules.

9.1.3.4.1 User preferences & profiles

A User Preference defines a goal or function that the User wants to use; it can also be used to mean the desire for the terminal to behave in a particular fashion. A User Profile is a collection of User Preferences that define how the terminal should behave for a set of related conditions (e.g., when a call is detected, which calls should be answered vs. sent immediately to voicemail). P1900-PolicyRules enable User Preferences and Profiles to be programmed, reused, and applied to multiple contexts and scenarios.

9.1.3.4.2 Terminal capabilities & profiles

A Capability is an object-oriented description of functionality and/or behavior as defined in a pre-defined taxonomy in the P1900 Information Model. This is similar to a User Preference, but represents a function provided by the terminal as opposed to a desire of functional operation by the User. A Terminal Profile is a collection of terminal Capabilities and functionality that is orchestrated to meet the needs of a User for one or more contexts. P1900-PolicyRules enable Terminal Capabilities and Profiles to be programmed, reused, and applied to multiple contexts and scenarios.

9.1.4 Policy Objects Within this Standard

The above scenario, and the application of P1900-PolicyRules to the scenario for different stakeholders, is provided so the reader can better understand how P1900-PolicyRules operate in a scenario. From the above, two conclusions can be drawn:

1. The *structure* of a P1900-PolicyRule remains constant for all stakeholders
2. The *content* of a P1900-PolicyRule is application- and context-specific

The first observation is very important. If the structure of a P1900-PolicyRule is allowed to change, then the burden placed on the machines that have to parse, interpret and apply P1900-PolicyRules is dramatically and significantly increased. This is because if the structure varies, the syntax may also vary, but the semantics must vary. This standard advocates a *single, constant structure* for P1900-PolicyRules for each of the three types of policies:

- Event-condition-action policies
- Goal policies
- Utility function policies

That is, *all* P1900-PolicyRules of a given type (i.e., one of the above three) *must* use the same basic structure. This in turn implies that the Information Model *must* supply model elements (i.e., classes, attributes, associations, and constraints) that can be used to represent each of the above three types of policies. Currently, this working group is *only* using event-condition-action policies; goal and utility function policies *may* be used at a later date.

The second point above means that the Information Model *must* be rich and extensible enough to represent many different types of services, resources, users, environments, and other concepts. This in turn implies that the Information Model *must* use software patterns and abstraction mechanisms (e.g., roles).

<< note to readers: I'm stopping now because I'm tired, and because this is a lot to agree to in a short time. ☺ Once this is agreed to, I can start filling in details. –John >>

Text end