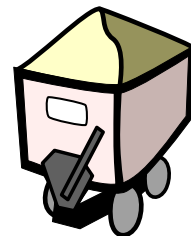


# Database model and data entities

Juan J. García de Soria



ORE  
Open Rating Environment



\$Id: entities.lyx,v 1.3 2004/12/01 14:47:59 skandalfo Exp \$

# Chapter 1

## Database model concepts

The database model for ORE will have to deal with the following kinds of data:

- User/Customer Accounts and related Subscriptions.
- Product/Service/Catalog definitions.
- Parameter Definitions and Values.
- Event Types and Rated Event Records.
- Invoices.
- Business procedures:
  - Event Guidance Procedures.
  - Event Rating Procedures.
  - Subscription Modifier Procedures.
  - Billing Modifier Procedures.

### 1.1 User and Customer Accounts

An Account will be the entity to represent a single customer or user in the service platform. Actually it represents a single billable entity, although it is not a monetary-only concept, since it could refer to several currencies at the same time, and will usually contain associated contact data.

That said, a single account will typically contain data for the following kinds of informations:

- Default, secondary currencies.
- Currency balances.
- Billing period and date information.

- Account status (active/blocked/deleted...).
- Commercial model information determining the Products available for the Account.
- Contact data, like:
  - Personal data (Name, surname...).
  - Address.
  - Phone number(s).
  - E-mail address(es).
- Additional parameter values as needed for the specific deployment.
- Product and associated Service Subscriptions, representing each service that the Account has subscribed to with the operator.

Each independent field or parameter value may be optional depending on specific deployment needs, and may be assigned a default value for the whole platform.

Each Account may have a parent Account. In that case, the child Account inherits every parameter value defined in the parent Account, except for the ones overridden or redefined at the child Account level.

## 1.2 Product, Service and Catalog information

A Service is the definition of a kind of service that may be provided to a customer or user.

A Service definition may define specific Parameters that determine specific characteristics of the Service. For instance, for a basic phone Service, the phone number for the subscriber line would be a Parameter to be defined.

Services may be subclassed. Each Service may be a root Service, so that it has no parent Service, or may inherit from exactly one parent Service. When a Service inherits from a parent Service, it inherits all the Parameter defined for its parent, as well as any Parameter inherited by its parent Service.

Each root Service defines an Event Type (a definition for the data that defines each Service usage) that is inherited and cannot be modified by child Services.

In the same fashion, each root Service defines an event guidance procedure that will be used in order to find to which Account and Service Subscription every incoming event of the matching Event Type will be routed to.

Each Service may assign Values for the Parameters that itself defines, or for any Parameter inherited from its parent Service, if any. If a Service doesn't define a Value for a Parameter, it inherits any Value that may be defined or inherited by its parent Service, if any.

Products are combinations of Services that may be sold as a unit to a customer. Services themselves may not be sold. Each Product will group one or more Services that will be sold together under a commercial name.

Products may define Parameters and give Values to them too, in the same way as Services do.

Products may subclass other products. In this case Parameter and Value inheritance applies in the same way as it does for Services. Additionally, each Product that subclasses another Product inherits its parent Services, and may add new Services to the group of Services inherited. However, a child Product definition cannot remove any one of the Services inherited from its parent Product.

A Catalog is a group of Products that are available to a given class of customers. In each one of these groups, each individual Product may be mandatory or optional. Mandatory Products are automatically subscribed whenever an Account is assigned the containing Catalog. Optional Products may be subscribed or terminated by the customer afterwards in any free combination.

### 1.3 Parameter Definitions and Values

Most of the actual information in ORE related to customers, services and subscriptions is stored in the Values of Parameters defined through the system.

There are three kind of Parameters:

**Account Parameters:** These are Parameters that are related to Accounts. They model attributes, characteristics or features related to Account objects or customers. An example could be the contact phone number for a customer.

**Product Parameters:** These are Parameters that are related to Products or Product Subscriptions. They model attributes, characteristics and features related to the Products that customers may subscribe to, or related to the specific Product Subscriptions they may have done to these Products. An example could be the end date-time when a given subscription has a minimum contract term<sup>1</sup>.

**Service Parameters:** These are Parameters that are related to Services or Service Subscriptions. They model attributes, characteristics and features related to the Services that may be provided to customers, or related to the specific Service Subscriptions that the customers may have as their rights to receive the Service. An example could be the domain, like “my-

---

<sup>1</sup>It would be the date-time to check against when unsubscribing a product that may impose penalties for being unsubscribed before six months elapse since its subscription, assigned at Product Subscription level. The actual period of six months could be another Product Parameter whose Value is assigned at Product level. The first Parameter Value would be calculated by using the second one by a Subscription Modifier Procedure at subscription time.

greatmail.com”, that is applied to every e-mail Service that customers of a given e-mail provider receive or subscribe<sup>2</sup>.

Apart from applying to different kinds of entities and being subject to the inheritance rules defined by those entities, the three kinds of Parameters are defined in the same way and may contain the same kinds of information.

A Parameter definition is done by specifying the following data:

- Parameter ID (auto-generated by the platform), that is a numeric unique identifier for the parameter inside ORE.
- Parameter definition place, a pointer to the entity on which the Parameter is defined<sup>3</sup>.
- Parameter name, a string with a unique, short name used for Procedures and other program entities to identify the Parameter.
- Parameter label, a text label to display in forms where this parameter may be filled in or displayed.
- Parameter description, a text explaining the Parameter purpose and usage, to be shown in tool-tips and help dialogs.
- Parameter type, the kind of data that the Values for this Parameter are. Supported Parameter types in ORE are:
  - String (strings of up to a given maximum number<sup>4</sup> of Unicode characters).
  - Integer (64 bit values).
  - Double (floating point values).
  - Decimal (decimal, arbitrary precision<sup>5</sup> real values).
  - Date-time (millisecond-level 64 bit precision timestamps).
  - ID (numeric identifier) types referring to either:
    - \* an Account ID.
    - \* a Catalog ID.
    - \* a Product ID.
    - \* a Product Subscription ID.

---

<sup>2</sup>The Value would be assigned at Service level, and it would be inherited by every Service Subscription that users get via their Product Subscriptions to Products including the former Service.

<sup>3</sup>This could be a specific Service, a specific Product, or the special case of Parameters defined for every Account.

<sup>4</sup>The maximum length of the String-type fields may be configured at database schema installation time, and is subject to the limits imposed by the specific database management system in use. PostgreSQL databases may be configured in order to impose no restriction (up to 2GB as of PostgreSQL 7.4) on the length of these Values.

<sup>5</sup>As with the String type, the actual precision may be configured at database schema installation time, and is subject to the limits imposed by the specific database management system in use. PostgreSQL databases as of PostgreSQL version 7.4 may be configured in order to impose no restriction precision of these Values.

- \* a Service ID.
- \* a Service Subscription ID.
- \* a Procedure ID.
- List of <a given type> (arbitrary-length, index-based ordered arrays of values of a given type).
- Structure of <set of name,type tuples> (composite type made of fields, each field with its own name and type).
- Whether the Parameter is mandatory or not; entities having a definition but no value for a mandatory Parameter are not allowed.
- Whether Parameter Values should be unique; the same Value for a unique Parameter may not be repeated.
- Whether the Parameter Values are historical; Parameters marked as historical store their Values in a special format that records how these Values change along the time. This format consists of a succession of time intervals with a specific Value assigned to each one of these time intervals. This allows for past Values to be recorded, as well as for future Values to be scheduled.

Values poollo