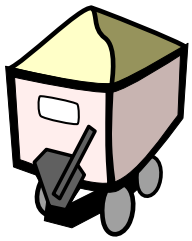
Database model and data entities

Juan J. García de Soria

ORE
Open Rating Environment



$Id: entities.lyx,v 1.9 2004/12/21 12:21:58 skandalfo Exp $

Contents

List of Figures

List of Tables

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.

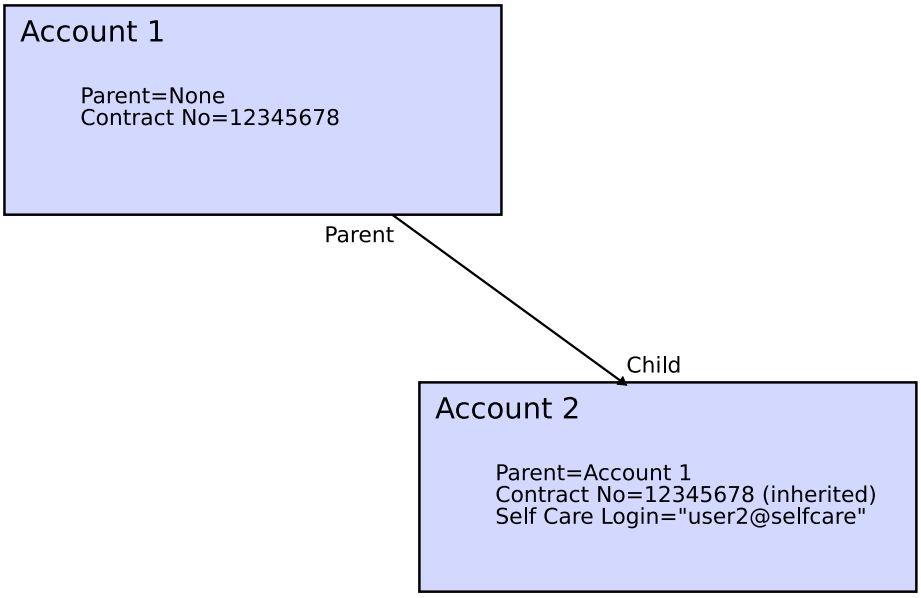


Figure 1.1: Account inheritance.

- Billing period and date information.
- Account status (active/blocked/deleted...).
- Commercial model information determining the Products available for the Account, including the Catalog assigned to a given 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. See figure 1.1 for an example of Account inheritance.

1.2 Product, Service and Catalog information

Services, Products and Catalogs model the services that a company offers to their customers and the way in which they are sold.

1.2.1 Services

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.

1.2.2 Products

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 it's 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.

1.2.3 Catalogs

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.

1.3.1 Parameter Definitions

The Parameters to be used throughout the system have to be configured by means of matching Parameter Definitions.

There are four 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[1].

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 "mygreatmail.com", that is applied to every e-mail Service that customers of a given e-mail provider receive or subscribe[2].

Structure Type Parameters: These are special Parameters used in order to define composite structure data types as explained in section 1.3.3.

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:

[1]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.

[2]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.

- Parameter ID (auto-generated by the platform[3]), 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[4].
- 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[5] of Unicode characters).
  - Integer (64 bit values).
  - Double (floating point values).
  - Decimal (decimal, arbitrary precision[6] real values).
  - Date-time (millisecond-level 64 bit precision UTC timestamps).
  - Boolean (true or false).
  - ID (numeric identifier) types referring to either:
    - an Account ID.
    - a Catalog ID.
    - a Product ID.
    - a Product Subscription ID.
    - 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) as defined by a referenced Structure Type.

[3]Note that Structure Parameters will use a number of different ID's for each one of its component sub-Parameters (fields).

[4]This could be a specific Service, a specific Product, a Structure Type, or the special case of Parameters defined for every Account.

[5]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.

[6]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.

- 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.

ORE may pre-define some internal or general Parameters as needed for special purposes.

1.3.2 Parameter Values

Values store actual values for the Parameters defined throughout the system.

Each value stored in ORE consists of the following data:

- Parameter Value ID (auto-generated by the platform[7]), that is a numeric unique identifier for the Parameter Value inside ORE.
- Parameter ID, the identifier of the Parameter Definition which defines this kind of Values.
- Parent Parameter Value ID, the identifier of the parent Parameter Value for list or structure Values elements.
- Parameter Value assignment place, a pointer to the entity on which the Parameter has been assigned this Value[8].
- As alternatives, either of:
  - Actual value[9], that may be a value of the type of the Parameter, or the special value *delete,* which means that any inherited Value is discarded for this Parameter. Parent entity Values are automatically inherited when no Value is defined at a given entity level.
  - Reference to another Parameter Value of the same type that is used as this Value had the same value, by including the Parameter ID of the Value to reference.

The usage of reference values explained above allows, for instance, to use an Account-related Value for a Parameter in a Service Subscription. References are restricted to certain combinations that make sense, with the allowed combinations shown in table 1.1.

[7] Note that Values for List or Structure Parameters will use a number of different ID's for each one of its component sub-Values.

[8] This could be a specific Service or Service Subscription, a specific Product or Product subscription, a specific Account, or the special cases of Parameter Values pre-defined for every Account.

[9] Filled in directly for basic-type Parameters, filled in in sub-Values for complex (list or structure) Parameters.

	Referenced Parameter type		
Referencing Value assignment place	**Account**	**Product**	**Service**
(all Accounts) or Account	OK	-	-
Product or Product Subscription	OK	OK	-
Service or Service Subscription	OK	OK	OK

Table 1.1: Allowed Value references.

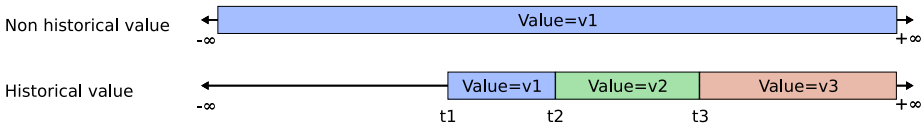


Figure 1.2: Time lines for historical and non-historical Values.

Values for Parameters that are configured to have historical Values are stored in a time line like manner. That means that each Value is stored multiple times, once for each time its Value changes in a new, non-overlapping time segment. So, for Values that are historical, two more fields are stored along with each Value instance along time:

- Effective start date-time (millisecond-level 64 bit precision UTC timestamps), the timestamp for the instant (included in timestamp comparisons) in which the associated Value instance takes effect, with a minimum negative value that may be encoded meaning $-\infty$.
- Effective end date-time (millisecond-level 64 bit precision UTC timestamps), the timestamp for the instant (excluded in timestamp comparisons) in which the associated Value instance stops being in effect, with a maximum positive value that may be encoded meaning $+\infty$.

For historical Values, these segment delimiters may be used as a part of the Parameter Value ID, and inherited by child Values for composite datatypes. For non-historical Values the same convention may be used by always using only one segment in the range $[-\infty, +\infty)$. See figure 1.2 for an illustration.

1.3.3 Structure Types

In order to use a given structure as the type of a Parameter or as the element type of a list Parameter, the structure to be used must be defined first.

For this purpose, ORE holds a repository of Structure Types. For each one of these the following information is stored:

- Structure Type ID (auto-generated by the platform), that is a numeric unique identifier for the Structure Type inside ORE.
- Structure Type name, a string with a unique, short name used for Procedures and other program entities to identify the Structure Type.

Fields for each Structure Type are defined as Parameter Definitions whose definition place points to the specific Structure Type.

1.4 Event Types and Rated Event Records

1.5 Invoices

1.6 Business Procedures