International Telecommunication Union

ITU-T Technical Report

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(dd MMM YYYY)

MSTR-CREG
Customer and Product Registration

Draft, neither submitted, approved or published

01.05.2017 Arve Meisingset

ITU-T

International Telecommunication Union

Summary

This technical report is a tutorial for how to register customer and product data.

This technical report illustrates how customer and product data can be registered.

Keywords

[if any provided]

Change Log

This document contains Version 1 of the ITU-T Technical Report on "*Customer and Product Registration*" approved at the ITU-T Study Group 2 meeting held in Geneva, dd-dd MMMM YYYY.

Editor: Arve Meisingset
Movinpics AS
Norway

Tel: +47 9139 2863
Fax:
Email:
arve.meisingset@movinpics.com

CONTENTS

Page

Contents

List of Tables

Page

Page

List of Figures

Page

Technical Report ITU-T MSTP-NREG

Technical Report ITU-T
Customer and Product Registration

Summary

This technical report is a tutorial for how to register customer and product data.

This technical report illustrates how customer and product data can be registered.

1 Scope

This technical report defines a terminology for registration of customer and product data of a telecommunication operator. The technical report is a tutorial on ITU-T Recommendation M.1402 Formalization of data for service management. Also, there is a chapter on M.1405 on Formalization of orders for service management among operators.

M.1402 is defined for service management among operators. However, the data definitions also apply for use within an operator, which is the focus of this technical report.

The technical report defines data to be used in customer self service, customer relationship management (CRM), billing, charging, product catalogues, pricing and channel management. The data definitions may be used for coordinating product information from several sources.

The technical report defines CRM data to be used to orchestrate information from several sources to several sinks.

The technical report does not cover management of technical resources, such as phone numbers, trails etc. Also, accounting, billing and data flows between organizations and between systems are not properly addressed.

The technical report presents the data as they will be perceived by human users at their human-computer interface. The data are defined as they will be seen in the customer facing channels. The internal representations used for network management and in service platforms are not shown.

This technical report may be used as reference material

- for writing requirements in Requests for Quotations (RfQs)
- design of efficient implementations
- coordinating interworking between several system domains
- teaching of end users on CRM data and related domains

Managers of Business Support Systems (BSS) should read this Technical Report to understand what entities the management is about, and what challenges there are with current registration.

Managed service providers should read the book of the same reason.

2 References

[1] ITU. ITU-T M.1400-M.1999. *Designations and Information Exchange*. http://www.itu.int/ITU-T/recommendations/index.aspx?ser=M.

[2] ITU. ITU-T M.1401. *Formalization of interconnection designations among operators' telecommunication networks*. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=8854. Geneva 07/2006.

[3] ITU. ITU-T M.1402. *Formalization of data for service management*. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11620. Geneva 05/2012.

[4] ITU. ITU-T M.1405. *Formalization of orders for service management among operators*. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=9093. Geneva 08/2007.

[5] ITU. ITU-T M.3190. Shared information and data model (SID). http://www.itu.int/itu-t/recommendations/rec.aspx?rec=9541. Geneva 07/2008.

[6] ITU. ITU-T Z.351. *Data oriented human-machine interface specification technique – Introduction.* http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=3187. Geneva 3/1993.

[7] ITU. ITU-T Z.352. *Data oriented human-machine interface specification technique – Scope, approach and reference model*. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=3188. Geneva 3/1993.

[8] ITU. ITU-T Z.601. *Data architecture of one software system*. http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=9061. Geneva 2/2007.

[9] ITU. List of ITU Carrier Codes (ICCs). https://www.itu.int/oth/T0201.

[10] NECA. Company Codes (OCNs). https://www.neca.org/Code_Administration.aspx.

3 Terms and definitions

3.1 Terms defined elsewhere

This technical report uses the following terms defined elsewhere:

None.

3.2 Terms defined here

This technical report defines the following terms:

Each subsection of sections 8 - 12 defines a term.

4 Abbreviations

BSS	Business Support Systems
CC	Country Code
CRM	Customer Relationship Management
ICC	ITU Carrier Code
ISO	International Organization for Standardization
NECA	National Exchange Carrier Association

5 Background

Rec M.1402 [3] is an extension of M.1400 - Designations for interconnections among operators' networks [1]. Texts on identification of Countries and Operators are derived from M.1400.

M.1402 is using the same methodology as of M.1401 [2], which is a formalization of M.1400.

M.1402 provides a modern documentation, using data structures. The data structures are used to document the syntax and grammatical rules of end user data. These data structures are not used to document data bases or conceptual structures, as of SID from TeleManagement Forum. See M.3190 [5] Appendix V Positioning SID to the ITU-T M.1400-series Recommendations.

This technical report aims at

- help understanding
- help communication
- teach terminology
- teach identification
- teach registration

of customer and product data.

The technical report will not teach the various management tasks and not teach about software tools to manage the data. Rather, we teach the data structures, which will be common for both human users and software applications.

The data structure will define the recommended common terminology to be used for customer self service, customer management, product management, call centers, retailers, charging, billing, orders and touches upon accounting.

The data structure defines a common terminology across multiple customer services.

ITU is publishing ITU Carrier Codes at its web site [9]. This site is integrated with NECA's site on Company Codes [10].

6 Terminology

This technical report aims at explaining the terminology on customers and products as being seen by human users at their human-computer interfaces. The terms may appear both as headings and values at the user interfaces.

This technical report additionally explains the grammar for combing these terms into valid statements. The grammar is expressed through containments and references.

The combination of terms and their grammar make up the definition of the end user language. The combination of terms and grammar is called a data structure. The expression data structure is used for many other purposes, eg. to define the structure of data in data bases. We use the data structure

to define the end user language. This is a different usage, with its own features and notation, even if its resemblance to data base structures is evident.

ITU-T Recommendation M.1401 uses a particular formalism for defining the data. This technical report uses a more colloquial style, and adds illustrations.

The particular data structure that defines the language used at the human-computer interface is defined in an External terminology schema. This is one component of the Data transformation architecture for IT systems, defined in ITU-T Recommendation Z.601 Data architecture of one software system [8]. M.1402 defines the External terminology schema for service management among operators.

An External terminology schema defines classes that act as general prototypes/templates for the data instances about a particular Universe of Discourse. M.1402 documents a schema. A collection of instances is called a population. This technical report will define the classes and show example instances.

Both schemata and populations may be documented in a graphical or alphanumerical notation. In this technical report we will use only the graphical notation. We use the same notation for classes and instances. Hence, the data are not classes or instances in an absolute way; they are only so relative to each other.

Figure 7-1 shows a class diagram of M.1402.

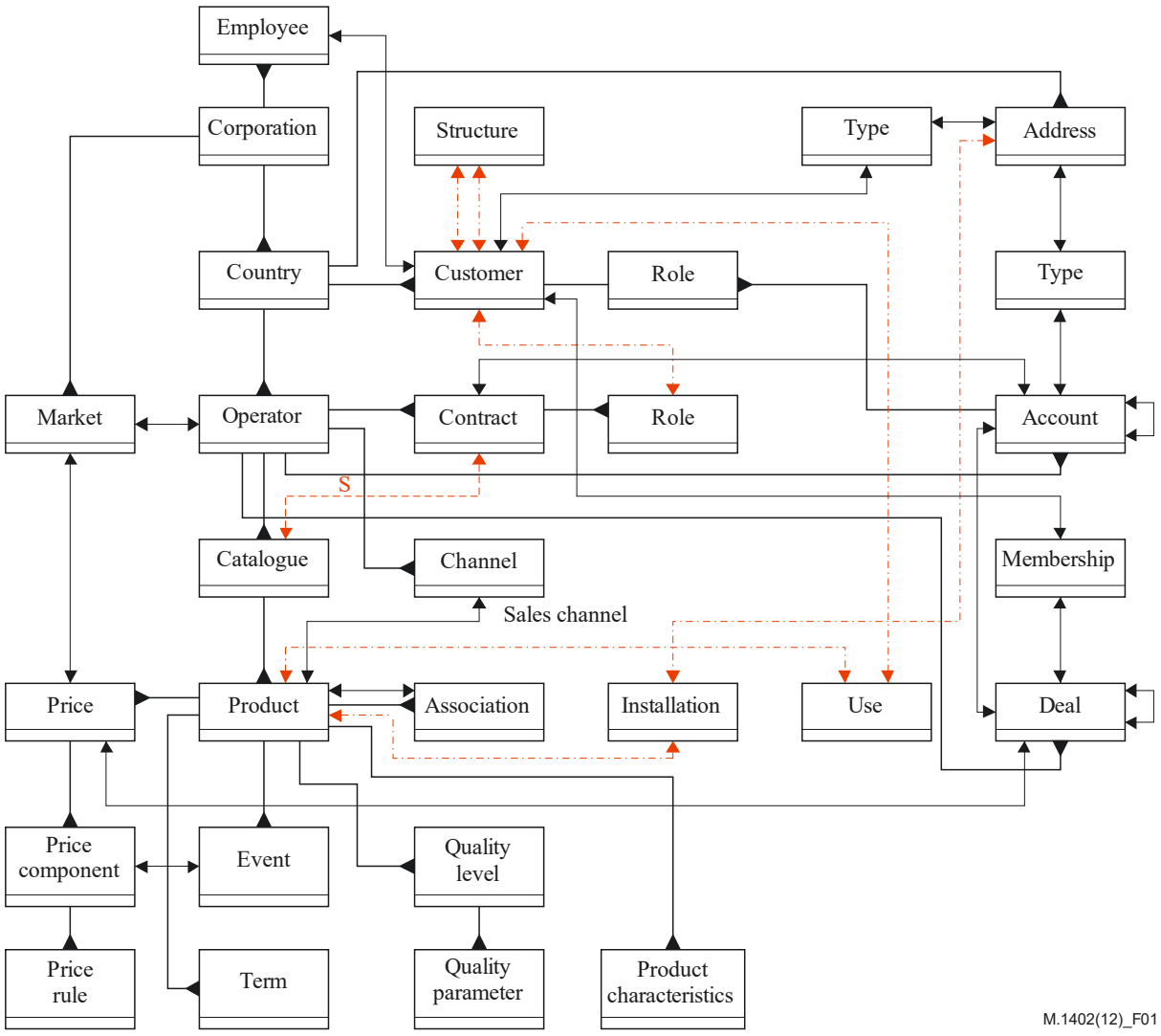

Figure 6-1 External terminology schema for M.1402

The above Figure is the most important Figure of this Technical Report. When you have understood this, you have understood the entire report. The Figure serves as a summary of the entire report.

For each box and line in the diagram, we in this Technical Report may

- show where we are in the diagram
- state definition of each item
- explain how each item is identified
- Show example screen shots
- show an example instance diagram of these resources
- add explanations when needed

In the examples of identifiers, we have shown both headings and values. We have used no abbreviation, even if use of abbreviations will be convenient in practical implementations. Also, we have adjusted both texts and numbers to the left, even if implementations may do otherwise.

7 Notation

This section defines a graphic notation for defining data classes and show example data instances.

We will use the following notation:

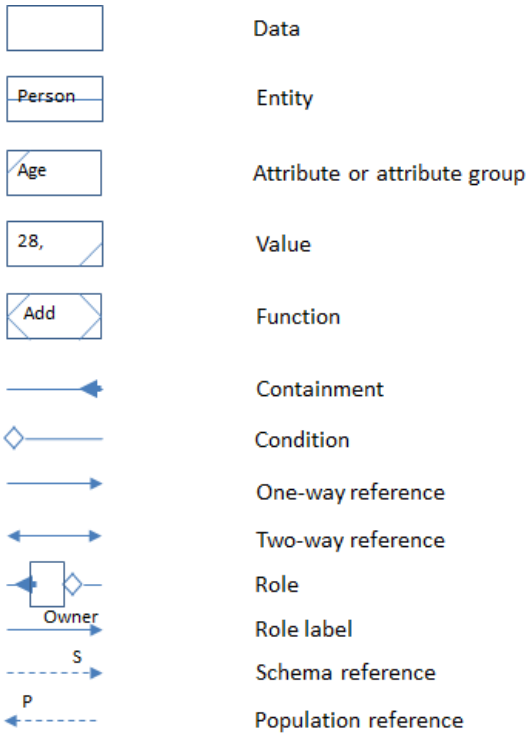

Figure 7-1 Graphic notation

The relationships are stated as

- Containments, or
- Two-way references

Containments tell that the contained class has a class label local to the superior class. An entity class can only be local to one superior entity class. An attribute group class is local to its entity class or another attribute group class. An attribute class is local to its entity class or its attribute group class.

Instances are created by taking copies of the class and its superior containment. If class B is contained in class A, then an instance of a class B must be contained in an instance of class A. This applies for entities, attribute groups and attributes.

Moreover, if class B is contained in class A, then the value of the identifying attribute (group) of an instance of a class B shall be local to the value of the identifying attribute (group) in an instance of class A. Hence, the containment shows how entities are identified local to each other. We will see many examples of this in the texts on each entity class.

In the instance diagrams we will frequently show identification values only, and skip attribute and entity labels, as these will overload the drawings.

In this report, we will use two-way references. Reference classes between entity classes are copied into reference instances between entity instances.

The copying described in the previous paragraphs is called an instantiation mechanism, and it explains how the contents of populations are created.

We will additionally use the one-way schema reference in this report. We will explain this reference when it will be needed.

The reader may find more explanation of the notation in Appendix III of [2].

8 Administrative domain

8.1 Overview

The following Figure gives an overview of the Administrative domain.

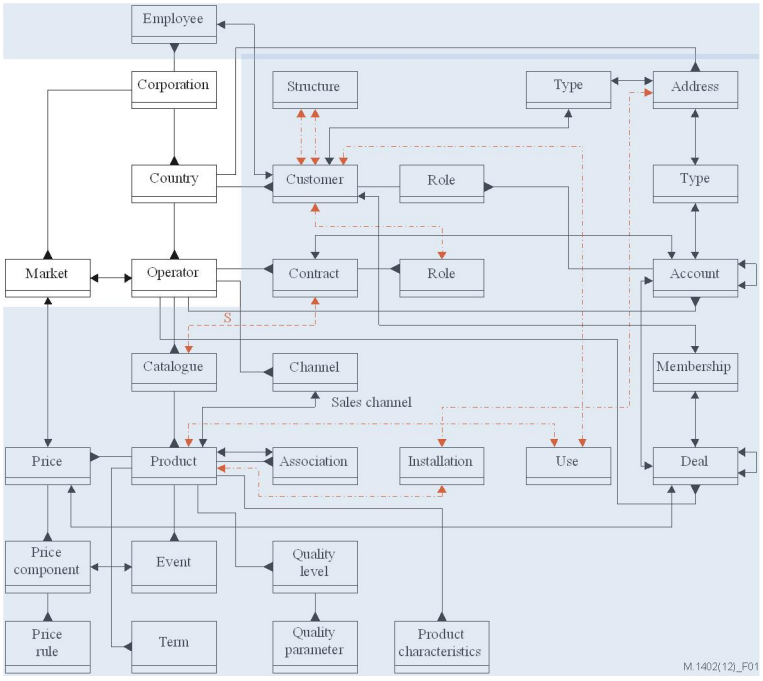

Figure 8-1 Schema of Administrative domain

8.2 Corporations

A Corporation is a company registered in one Country. M.1402 does not prescribe how Corporations are identified.

Corporation
Name
TELENOR.

Figure 8-2 Example layout of Corporation

The Corporation may operate in many Countries. Each operation is called an Operator. An Operator cannot span more than one Country. The Corporation may have more than one Operator in a Country, eg. one Operator for each brand.

The Operator can only sell services to citizens or businesses of this Country. However, this citizen or business may have Addresses in other Countries.

Employees may be registered per Corporation, and not per Operator. See more on Employees.

Markets may be registered per Corporation. The Markets may correspond to Countries, Operators or other Customer and Product segments. See the reference between Market and Operator, and see below.

Customers and Addresses are registered per Country of a Corporation, and not per Operator. The Customer and Address instances are not common across Corporations.

Products, Contracts and Accounts are registered per Operator within a Country within a Corporation.

8.3 Countries

A Country is an area that is assigned a Country Code according to ISO 3166.

All telecommunication operators, sites and resources are identified within Countries. When defining connections between sites in two Countries, you will have to identify both these Countries. This is why Country is such an important notion for designation of telecommunication resources. Note that Products in M.1402 include such connections. M.1401 shows their designations.

Each Country is identified by an ISO 3166 alpha-3 code called Country Code (CC). This is a three letter alphabetic code assigned by the ISO 3166 Maintenance Agency.

The next Figure shows an example layout.

Country
CC
NOR.

Figure 8-2 Example layout of Country Codes

Note in Figure 8-1 that a Country is contained in a Corporation.

A Country is identified within a Corporation as follows:

Corporation	Country
Name	CC
TELENOR.	NOR.
	SWE.
TELIA	NOR.

Figure 8-3 Example presentation of Countries within Corporations

Here we see that TELENOR (NOR and TELIA (NOR are treated as two different entity instances.

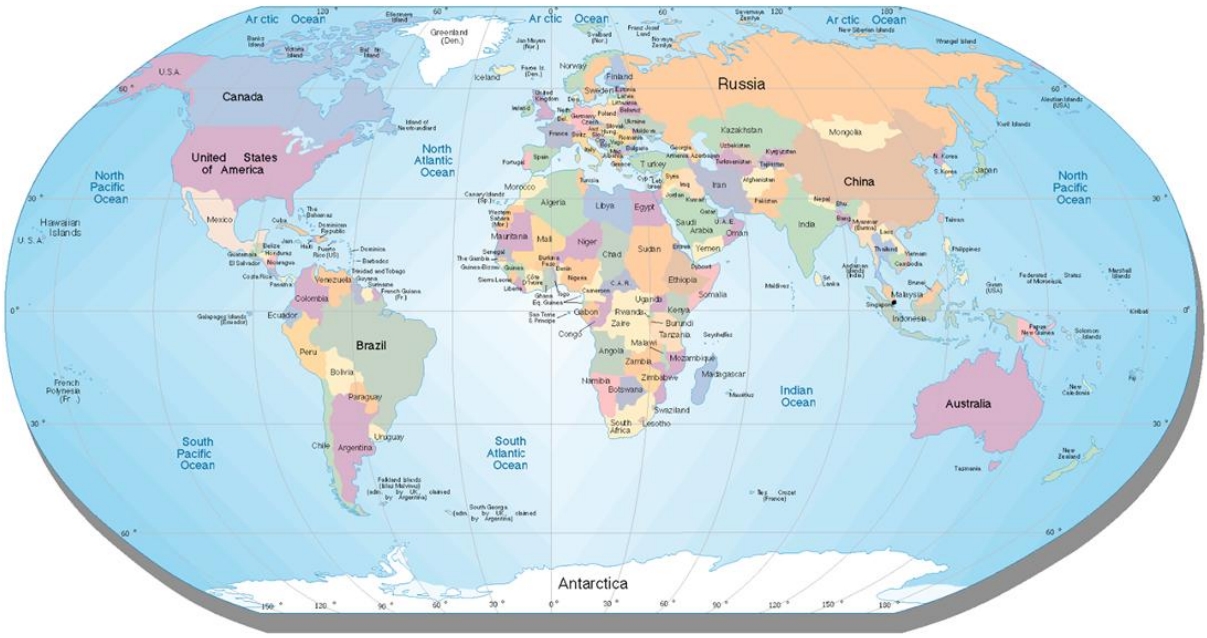

Figure 8-3 World map indicating Countries

Each Country has a telecommunication authority. This may be a ministry. In ITU, this authority may be referred to as an Administration. An Administration is an agency of a government, designated to represent that government and its various interests in the ITU. Sometimes the Administration is the telecommunication regulator of that country, sometimes the Administration is another agency. The Administration acts to administrate the ITU international designations, numbering, addressing, accounting, etc. in coordination with the ITU.

The Administration may delegate responsibilities to various entities. Examples hereof have been:

1. Regulatory responsibilities within a ministry (e.g., Japan, China etc. many in the Asia & Pacific);
2. A fully autonomous & independent regulator (e.g., the USA, the UK, Hong Kong)
3. A semi-autonomous independent regulator (e.g., Canada);
4. Industry self-regulation (e.g., some developing countries etc.);
5. No telecommunication-specific regulation (e.g., New Zealand)

The particular arrangement in each Country needs to be known in order to apply for license to operate telecommunication networks and services, and to become registered as a recognized operating agency, being assigned an ITU Carrier Code. See the subsequent section on Operator.

Customers are registered per Country of a Corporation. Hence, a Customer may be common for several Operators of the same Corporation within that Country.

Addresses are registered per Country.

8.4 Operators

An Operator is an organization responsible for identification and management of telecommunication services or resources. The Operator must be legally recognized by the telecommunication Administration of the Country, or delegation thereof.

An Operator can be a network operator or service provider.

Both the national authority and the Operators should note that each Operator would be registered as a separate entity in each Country in which it operates, and is required to report country-specific information to each national authority. This may or may not imply a need for a multinational Operator to use different ITU Carrier Codes (ICC) in each Country. See on ICCs below. Multi-national Operators are encouraged to ensure that they are registered in each Country in which they operate.

Each Operator in a Country shall be assigned an ICC. A request for an ICC is sent to the regulator, who forwards this information to ITU, which assigns the ICC. The ICC is a six letter code that is unique within the Country.

The ICC shall be used in all communication between Operators on management of their networks, sharing of network resources and provisioning and reporting on connections between these networks.

A list of all ICCs world wide appears at itu.int.

The ICC provides a unique identifier of an Operator within a Country. The assignment of ICCs may be delegated to a lower-level regional administration (for example, a province).

The ICC is used as an indispensable part in the assignment of identifiers for interworking between Operators. It is permissible that an Operator uses one ICC for assignment of identifiers, while other operator codes are used for ordering, billing, etc.

ICC is a 1- to 6-character alphanumeric field.

See ITU-T Recommendation M.1401 Appendix I on registration of ICCs. Here you also will learn about the particular arrangements in North America.

Corporation	Country	Operator
Name	CC	ICC
TELENOR.	NOR.	TELNOR.
		TELMOB.
	SWE.	TELMOB.
TELIA .	NOR.	TELIA .
	SWE.	TELIA .

Figure 8-5 Example Operators in the same and different Countries

In the above Figure, we have listed five different Operators, within two Countries within two Corporations.

Also, the presentation may use global distinguished names.

Operator		
Name	CC	ICC
TELENOR.	NOR.	TELNOR.
TELENOR.	NOR.	TELMOB.
TELENOR.	SWE.	TELMOB.
TELIA .	NOR.	TELIA .
TELIA .	SWE.	TELIA .

Figure 10-6 Example use of global distinguished names

An Operator may register itself, its competitors and collaborators within the same or other Countries. Not all of these may be recognized as Operators, but the Operator may treat them as such, eg. by prefixing the non-operator entities. Such entities may be the national defence, the national railway etc, having their own network, and who may have network sharing with the Operator. The Operator may need to register other Operators when trails are routed through these Operators' network.

Every Operator in the Corporation may be assigned a unique Local identifier (LID). The LID may correspond to a NECA business code. The LID may be used to register not-recognized Operators, vendors and partners. See under the Product domain on registration of Products of these entities.

Every Operator may have a default Billing system for all Accounts under this Operator.

Every Operator may have a default Rating system for all Contracts under this Operator.

An Operator may use several communication Channels to its environment. Such Channels may be retailers, distributors, call centers, self service, emails, bills, public media etc. See on this under the Products domain.

Contracts and Accounts are registered per Operator. See on this under the Customer domain.

Also, an Operator may have several Deals regulating Prices with selected Customers and their Accounts. See on this under the Pricing domain.

8.5 Markets

A Market may be defined as any Customer or Product segment of interest to the Corporation.

The Market may be defined by Customer categories and Product types. The Market may be a part of a Country, a whole Country or go across several Countries.

The Market may be defined by Prices, and the Prices may be dependent on the Market. Therefore, we have a relationship between Market and Price.

The Market may be within an Operator or go across several Operators. The pricing will be dependent on Operator. Therefore, we have a relationship between Market and Operator.

9 Customer Domain

9.1 Overview

The following Figure adds the Customer domain to the Administrative domain.

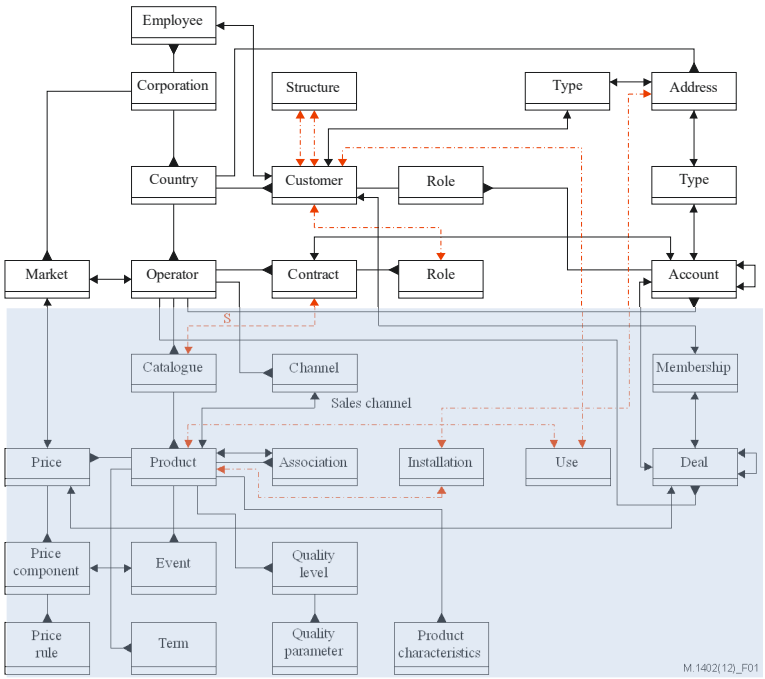

Figure 9-1 External terminology schema of Customer domain

9.2 Customers

A Customer indicates a legal person whom the Operator relates to.

The legal person may have one or more roles to Accounts, Contracts, Deals or Employees in or from the Operator.

A Customer may be a private person or a legal organization. The Customer's internal organization may indicate family, cohabitant and friendship relations, or indicate organization of geographically distributed companies and collaborative relations. The Operator may not want or be able to record this internal organization of the Customer. See the Structure entity in Figure 9-1, and see text below.

Role notions allow the Operator to register the Customer relative to Contracts and Accounts and not to register the Customer's internal organization structure, which may change without the Operator's knowledge.

A Customer is identified by a Customer identifier, KID, within the Operator. This is shown in Figure 9-2.

Customer
KID
123456789

Figure 9-2 Identification of Customers

The KID provides a unique identifier of the legal person within the Operator. This same identifier is used for all contacts with this legal person. The legal person may have other identifiers, eg. the person's Name and Address; however, these may be written in different ways, and may even change throughout the life time of the legal person. The KID does not.

Some Countries require that the passport number or other official identification is used for the KID. Other countries disallow such usage of identifiers of persons.

Each Customer shall have a Name within a Country. Several Customers may have identical Names, and Names may be misspelled. Customers may be distinguished by Address or other information. The Name and Address may be used for search, but are not sufficient for identification.

A Customer cannot be moved between Operators. However, the legal person may be a Customer of several Operators within the same Country.

Note that TELNOR (NOR (123456789 and TELIA (NOR (123456789 are different Customers. Also, TELNOR (NOR (123456789 and TELNOR (SWE (123456789 are different Customers.

The legal person of a Customer must be validated prior to accepting the legal person as a Customer. Different Countries require different documentation and procedures for this validation.

Customers may have one or more Business roles. This attribute may have the following value set:

A – Employee in the corporation
F – Retailer of the corporation
D – Distributor of the corporation
L – Operator in the corporation
K – Competitor of the corporation

or be left blank, which is a normal value.

Customers may have one or more Customer status. Customer status is provided by the business register within the Country.

K – valid customer
UK – (terminated) bankrupt
UF – (terminated) wrong registration
US – (terminated) merged
UO – (terminated) closed
UT – (terminated) legal action.

For persons, the following states are used:

K – customer
KU – customer emigrated
KF – customer disappeared.

Customers may have one Importance attribute. The value

V tells that this is a Very Important Person

in his relationship to the Corporation.

A Customer may be involved in Subordinate structures and/or Superior structures. Each Structure may have one of the following Types:

V – indicates subordinate business unit
M – indicates Member of a family
U – indicates subordinate Unspecified membership
H – indicates historical entity of the current entity

These Structures of Customers are difficult to maintain, as Customers typically do not inform about their change of Structures. Therefore, the Structures may be used as additional information, but for formal contact and billing, use of Roles to Contracts and Accounts are the preferred means. See on Roles below. Also, Type of Address is relevant.

A Customer may have one or more Types of Addresses. The Types are

S – location
P – postal address

A Customer may be registered as being an Employee of the Corporation.

A Customer may have one or more Memberships in Deals, and the Customer may have different Category Types of the Memberships, eg.

K – contact
M – member
E – owner.

A Customer may have one or more Roles to one or more Contracts. Each Role has a Category Type:

E – owner
B – payer
Y – authorized
K – contact
V – very important person
U – user
A – provider agent
R – responsible employee

A Customer may have one or more Roles to one or more Accounts. Each Role has a Category Type:

E – owner
B – payer
Y – authorized
K – contact
V – very important person
U – user
A – provider agent
R – responsible employee

See under Contracts and Accounts on use of these Category Types.

9.3 Contracts

Contract is an arrangement with a Customer on delivery of services, which are called Product instances in this data structure.

Permissible Product types for sale can be found in two ways:

- By the schema reference (S) to Catalogues
- If no explicit schema reference is provided, the permissible Product types are found under the latest Catalogue under the Contract's superior Operator.

Also, the Catalogues may be labelled with information on which one is new or old. This information is missing in M.1402.

Note that older Product types under the current Contract may come from older Catalogues under the current Operator.

A Customer may have one or more Contracts. Sometimes it may be convenient to have one Contract per Account.

Additionally, it may be convenient to have one Contract per fixed line, ie. per Trail. One of the Products under the Contract may refer to the Trail. Other Product instances under the same Contract may refer to Trails routed in the network of other Operators, may refer to termination at the Customer premises, to equipment by the Customer, to a phone number, and to services used on this Trail. These Product instances within the Contract may hide sensitive business information from the

transmission management of the Operator or from other Operators. See more on Products under the Catalogue domain.

The Contract notion allows for letting the Customer Relationship Management act as a hub for communication with other IT systems, networks, employees and partners.

A Contract is identified by a Contract number within the Operator.

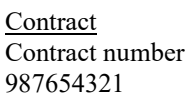

Figure 9-3 Identification of Contracts

A Contract number may by default be identical to a phone number used to identify a Product instance subordinate to the Contract. If a Contract number is assigned this way, one must avoid that this number comes in conflict with numbers assigned by different processes. Also, you may have to create a new Contract if the phone number is changed. Therefore, to mix the notions of Contract and phone number is not a good idea, even if many software packages and Operators are doing just this.

The Contract notion allows Product instances to be grouped according to their Product instance contents and relationships to Customers. However, many commercial software packages do not distinguish between Contracts and Accounts.

The Contract notion allows information on Account numbers to be hidden from the charging engine. Only the Operator has information about all its Accounts. The Customer has information about its billing Accounts.

The Customer may not always be informed about the Contract number, and the Operator may not expect that the Customer is able to inform about his Contract number. Rather, the Customer may take contact about a Number of a Product instance, e.g., a phone number.

A Contract may have an Amount, stating the available amount for usage under this Contract.

A Contract may have an Internet address that tells how to contact the Customer on Product instances and Accounts under this Contract. This Internet address may be different from the Customer’s subscribed internet addresses, which are recorded as Product instances under this or other Contracts.

The Internet address of a Contract may be used to send customer notifications on faults and important events of the Product instances under this Contract.

A Contract may contain an attribute telling about the confidentiality of the information in the Contract. A hidden number value may tell that the contained Product instances, including the phone number, the referenced Customers or the referenced Accounts shall not be exposed to the public. This information is not covered by M.1402.

The Contract may also contain information on how information shall be presented in the telephone directories. This information is not covered by M.1402.

A Contract may refer to one or more Accounts. Normally a Contract refers to one Account only. One Account may refer to multiple Contracts, and thus summarize usage expenditure across these Contracts.

Some Contracts may refer to several Accounts. Hence, the usage on the Contract is split on several bills. Rules given for each Account specifies which costs go where, e.g., before 17.00 hours the cost is put on the company Customer Account and after 17.00 hours it is put on the private Customer Account.

Sometimes, it may be more convenient to split the bill by splitting the Product instances on different Contracts, eg. one Product instance during working hours, and another Product instance outside office hours. This solution requires that the charging engine accepts the split.

Finally, the Account hierarchy may be used to split the bill. See on this below. Note that the Product instances will be contained in Contracts. Hence, the charging engine will get no information on Customers or Accounts. The Contract notion hides all private information.

A Contract may have one or more Roles to one or more Customers.

A Contract may only have one Role of Type owner (E), payer (B), user (U) or provider agent (A), but may have several Roles of Type contact (K) and very important person (V). The following rules apply:

1. If some Roles are not explicitly provided for a Contract, and the Contract has one Account only, then the Roles of the Contract are copied from the Roles of the Account.
2. If the Role of Type user (U) is not indicated for the Contract, but is the same for several subordinate Product instances, then this role is copied to the Contract.
3. If owner (E) is provided, and payer (B) is missing, then the payer is equal to the owner.
4. If owner (E) is provided, and user (U) is missing, the user is equal to the owner.
5. If payer (B) is provided, and owner (E) is missing, the owner is equal to payer.
6. If payer (B) is provided, and user (U) is missing, the user is equal to payer.
7. If user (U) is provided, and owner (E) is missing, the owner is equal to user.
8. If user (U) is provided, and payer (B) is missing, the payer is equal to user.
9. If Use is not provided for some subordinate Product instances, then these are copied from the user (U) Type of the Contract.
10. If some Roles are not provided for a related Account, and this Account has only this Contract, then the Roles of the Account are equal to those of the Contract.

Derived Types may, e.g., be indicated by a separate colour at the user interface, or they may be hidden.

9.4 Accounts

An Account provides information on agreed, delivered and outstanding payments by the Customer.

Accounts are identified by Account numbers within the Operator.

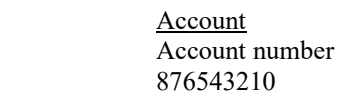

Figure 9-3 Identification of Accounts

Note that Accounts are per Operator, and they cannot go across Operators. There is no such notion as a common Account or common bill of Customers across several Operators, even if they belong to the same Corporation. However, it is possible to define references between Accounts of different Operators, but this could lead into uncharted terrain. See below on Subordinate account and Superior account.

Accounts are used for billing. Additionally, Accounts are used for reporting to accounting and fraud surveillance.

Some Accounts may be created just to give overviews of information to the Customer, without resulting in a bill on this Account. An example of such Accounts may be Accounts per department of a business Customer.

M.1402 does not define appropriate attributes of Accounts, such as Balance. See on bills at the end of this subsection. See on call records under Price components.

An Account will be specified by a Payment form, either

E - Prepaid or
O - Postpaid

An Account may have one or more Roles to one or more Customers. See the values of Category Types under Customer. The following rules apply:

1. An Account may only have one Role of Type owner (E), payer (B), user (U) or provider agent (A), but may have several Roles of Type contact (K) and very important person (V).
2. If some Roles are not explicitly provided for an Account, and the Account has one Contract only, then the Roles of the Account are copied from the Roles of the Contract.
3. If owner (E) is provided, and payer (B) is missing, then the payer is equal to the owner.
4. If owner (E) is provided, and billing address (F) is missing, then the billing address is equal to the owner.
5. If owner (E) is provided, and user (U) is missing, the user is equal to the owner.
6. If payer (B) is provided, and owner (E) is missing, the owner is equal to payer.
7. If payer (B) is provided, and user (U) is missing, the user is equal to payer.
8. If user (U) is provided, and owner (E) is missing, the owner is equal to user.
9. If user (U) is provided, and payer (B) is missing, the payer is equal to user.
10. If some Roles are not provided for a related Contract, then the Roles of the Contract are equal to those of the Account.

Derived Types may be indicated by a separate colour at the user interface, or they may be hidden.

An Account may have an Address type that refers to an Address. The Types can be

A – address of responsible person
B – billing address

Note that a business Customer, eg. Securitas, may have many Accounts. The billing Address of each Account may be different, eg. for each department in different towns. Hence, the bill is sent to Securitas in each town, without the Operator knowing the name of the individual department in that town.

An Account may refer to one or more Contracts. The Account manage expenditures, payments and outstanding payments across these Contracts.

Note that the Account may be owned by an employee through its Customer role, and have a reference to the Contract of this Customer. The company in which the person is employed may have another Account that refers to the same Contract. The company Account may have references to a Contract of each employee Customer.

An Account may have a reference to one or more Deals. This reference is typically established when the Deal is settled between an organization Customer and the Operator. The Deal refers to special Prices of Products in a product Catalogue of the Operator. When a member of the organization uses the Deal, the Price is instantiated to a Product instance under a Contract that has a reference to one or more Accounts. One of the Accounts may be the Account that has the reference to the Deal.

An Account may have several Subordinate accounts, which each indicates a role of another Account. The account structure indicates how the Customer wants his payment information to be structured and split. Amounts from the Subordinate accounts may be summarized to the current Account. An attribute indicates if the Account shall result in a bill or not. This attribute is not covered by M.1402.

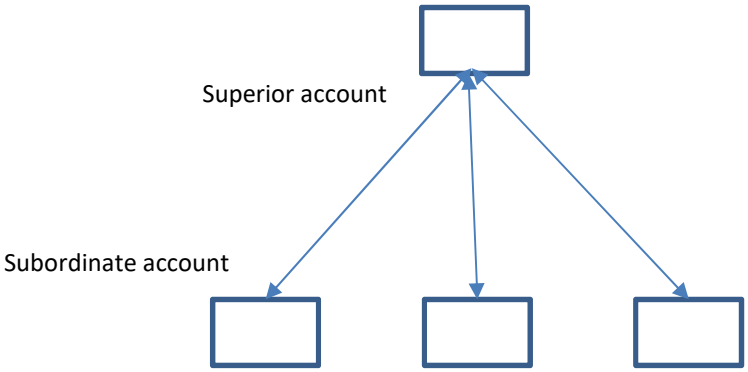

Figure 9-4 Subordinate accounts may be added up to one billing Account

An Account may have a Superior account, which indicates a role of another Account. In this case, the current Account provides information on parts of the Superior account. An Account may have several Superior accounts. This allows for split bills. Figures from the current Account may then be calculated to all Superior accounts. Therefore, the account structure may have several top nodes, but one Account cannot be contained (recursively) via different paths within the same node. An attribute indicates if the Account shall result in a bill or not, and how the amounts shall be split. This attribute is not covered by M.1402.

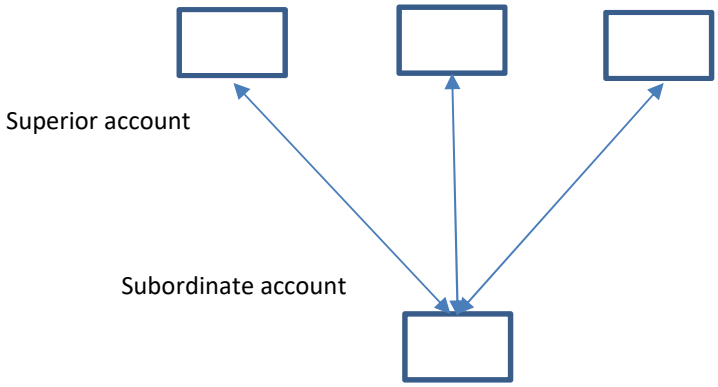

Figure 9-5 An Account can be split on several Superior accounts

One Customer may receive several bills on the same Product instance and the same period, e.g., the Product is partly prepaid and partly postpaid.

Accounts may have a log of bills and payments for this Account. The bills and payments may be created as Orders. See the brown lines in Figure 12-1.

9.5 Addresses

An Address identifies a geographical place.

The Address is local to a Country, and the identifier structure and values are defined by the authorities.

Country	Address
CC	Identifier
NOR	Storgata 2 0153 Oslo

Figure 9-3 Identification of Addresses

There are considerable difficulties with the spelling of Addresses. Therefore, some authorities have issued a code for each Address. These Addresses are located in geographical maps.

The Address notion does not cover bank accounts or similar notions. However, the post box notion is covered by this identifier. Also, postal addresses given by place and postal code/post area are covered by the identifier.

In M.1402, Address is treated as one entity class only; Most implementations split the Address into several entity classes, eg. for streets, local areas, etc.

An Address may have one or more Installations, which each defines a role of a Product instance in this Address. Each Installation contains a Change State

H – historic
I - Installed

An Address may have one or more subordinate Types that indicates a role of a Type of a Customer. See on the Types under Customers.

An Address may additionally have one or more subordinate Types that each indicates a role of a Type of an Account. See on the Types under Accounts.

9.6 Employees

Employees are persons who are employed by an Operator within a Corporation.

The Employee number is assigned by the Corporation, to be unique within the Corporation. This does not prohibit that the creation of Employee numbers is delegated to the Operator or localities within the Operator. However, the Employee number should preferably be valid when visiting another Operator within the same Corporation. However, we have seen that many Operators do not allow for Corporation wide use of the Employee number. Outside the Corporation, the Employee number has no validity.

Corporation	Employee
Identifier	Employee number
TELENOR	345678901

Figure 9-6 Identification of Employees

Most Employees will additionally be registered as a Customer of the Operator. The Employees' role relative to other Customers will be registered as Roles from the employee Customer to the Contract of the other Customer or as Roles from the employee Customer to the Account of the other Customer. Such a Role may be

R – responsible employee

10 Product domain

10.1 Overview

The carved out addition in following Figure gives an overview of the Product domain.

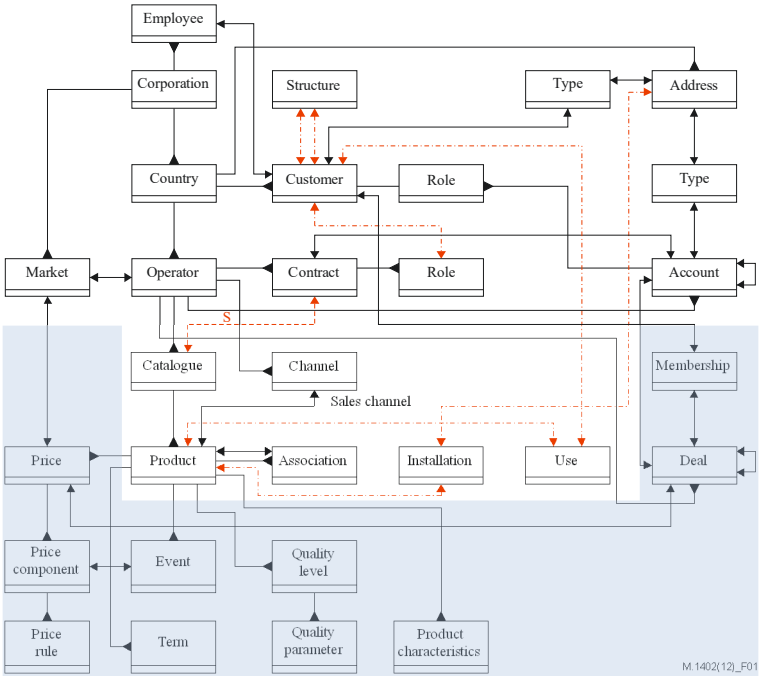

Figure 10-1 External terminology schema of Product domain

10.2 Catalogues

A Catalogue within an Operator contains a list of all Product types that are available from the Operator. Due to needs to manage several versions of offers, the Operator may have several Catalogues.

Additionally, separate Catalogues may be defined for technical Products that are inherited, or by other means to be used to construct Products for sale.

A Catalogue is identified by a unique Identifier within the Operator.

Catalogue
Identifier
2017new

Figure 10-2 Identification of Catalogues

Note that if we want to refer to a Product from a Catalogue of another Operator within the same Country, we have to include the Operator ICC.

Operator	Catalogue
ICC	Identifier
TELIA	2016telia

Figure 10-3 Identification of Catalogues of other Operators

A Product from one Catalogue may be inherited or composed of Products from other Catalogues. Some of the Catalogues may belong to other Operators, such as vendors and partners.

A Catalogue has a unique Identifier within the Operator. If there are several versions for a product Catalogue, version information should be contained in the Identifier.

The attribute Valid period tells for which period the Catalogue is valid. An Operator may support and charge Products that are out of sale, ie. outside the Valid period.

Most Operators manage Catalogues for different purposes within the same Valid period, eg. one Catalogue for customer self service, one for retailers, one for CRM, one for billing etc. This proliferation of Catalogues should be avoided, but is demanding to achieve. However, use of Products from vendors and partners is needed, and inheritance mechanisms are provided for this purpose at the Product level.

Catalogues serve as schemata of Contracts. This means that the contents of a Contract is copied as a subset of the contents of some Catalogue. See on this under Contracts.

10.3 Product types

A Product type is a type of service with a predefined functionality having a predefined set of Prices.

A Catalogue contains a list of Product types. The Product types are called just Product in the Catalogue, due to that they will be copied into instances under Contracts. See below on Product instances.

A Product type is an instance of the Product class in Figure 10-1. Hence, Product instances under Contract are instances of Product type instances under Catalogue instances. This recursive instantiation may be difficult to follow, but is important. Most software vendors have not understood this, and have not implemented it.

Product types are normally for sale, and only those for sale are visible in the customer Channels. Channels apply only for Product types in Catalogues, and may not apply for Product instances under Contacts.

Each Product will have an Identifier within the Catalogue. The unique Identifier within the Operator consists of

Catalogue,
Type and
Number.

The Number is only used for Product instances under a Contract. See below on Product instances. Also, the Catalogue Identifier needs not be shown for Products within that Catalogue, but may be needed for Product instances within a Contract..

Figure 10-4 Identification of Product types within a Catalogue

Each Product will have a Type that is unique within the superior Catalogue. The Type is the identifier of the Product within the Catalogue.

Note that the Type is the term used in customer facing, and may not be used internally or in the service platform. However, there must be a one to one mapping to the internal Product type. M.1402 does not give a term for the internal Product type.

An Operator may use unique Types across all its Catalogues, but Figure 10-1 does not require so.

In addition to the Type attribute, a Product may have a Resource type identifier, provided by the Operator, and which tells that an instance of this Resource type may be replaced by another instance of the same Resource type, but of a different Product type. Hence, Products of different Types may replace each other. The term Resource type is somewhat unfortunate, as we could have used this for the internal Product type.

A Product will have a Name. A Name is a word, term, or phrase by which the Product is known and distinguished from other Products. The Name attribute is applicable for both Product types and Product instances.

The Product will have a Description. This is an explanation of what the Product is.

The Product type may contain information on how to present the Product type on customer facing webs. This information is not covered by M.1402.

The Technology type is a categorization of the Product, such as fixed phone, wireless, data communication, etc.

The Customer base tells what industry or customer base the Product is designed for, such as government, education, finance, family, etc. This attribute is only applicable for Product types.

The Valid period tells the period during which the Product is applicable.

The Status states the condition in which the Product exists. This attribute is applicable for both Product types and Product instances. However, Product type and Product instance may have different status values. For example, Product type may take such values as planned, obsolete, active, while Product instance may take such values as planned, designed, activated, disconnected.

Separate Products may be defined as technical Products, which are used to implement Products for sale in other Catalogues. Also, particular Products may be defined for inheritance of definitions. See below on the Association Type I.

The Product may be defined as a market Product or a technical Product. This allows one Operator A to deliver a Product to another Operator B. The delivered Product may be a technical Product from A that is used to deliver a bundled market Product from B, and the market Product only is visible to the Customers of B. This way, Products may be bundled in several steps.

Due to the bundling mechanisms indicated above, there is no distinction between Product, service and resource. Products may have service attributes, and Products may have attributes that refer to resources. Some software vendors make a mess out of this.

Note that a service may be defined to be a Product that is delivered from a service platform to a user. However, this notion is outside the scope of the Recommendation. See M.1402 Appendix I.

A product may contain one or more Associations.

A Product's Association to one or more other Products may be of the following Types:

- B – Bundle, where the other Products are components
- M – Main product, where the other Products are additional products
- C – Constrained product, where the other Products are excluded products
- I – Product instance, where the other Product is the Product type of the current Product
- N – Next service; if the user is not present at the current service, try the next

The Associations may be seen from a contained Product, as well. The Superior association refers to the Association of the containing Product.

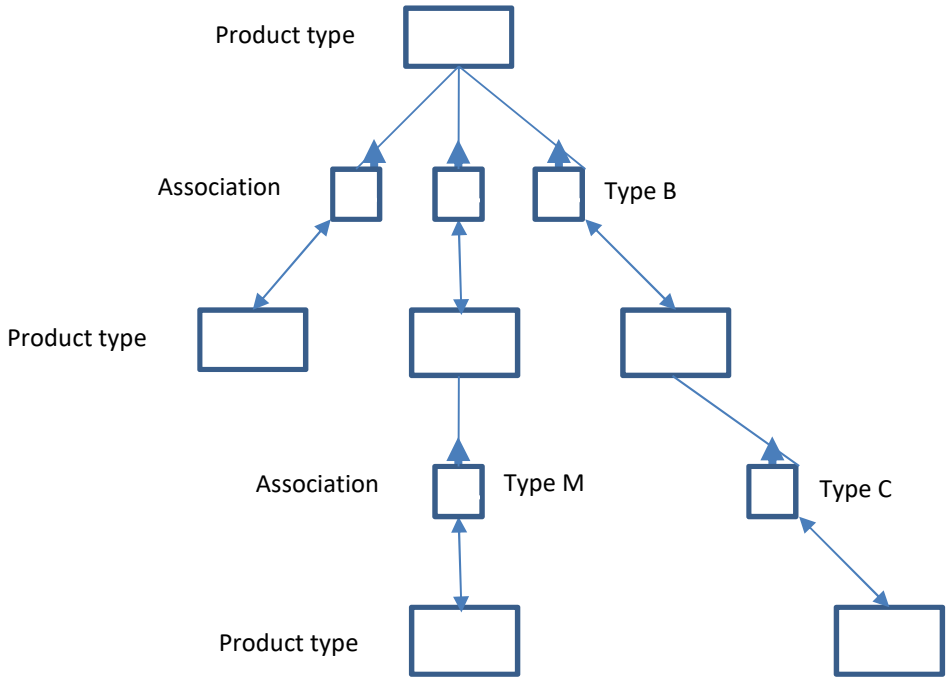

Figure 10-5 A bundled (B) Product type containing three other Product types;
the second contained Product type has an additional (M) Product type,
the third contained Product type cannot (C) be combined with the last Product type

Note in Figure 10-5 that the shown Product types may be involved in still other Associations.

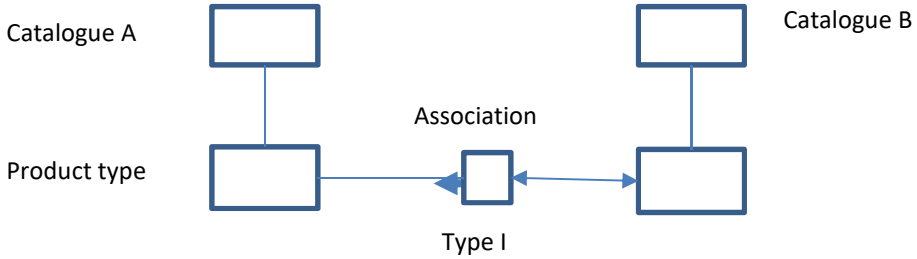

Figure 10-6 Inheritance of a Product type in Catalogue A to a Product type in Catalogue B

Note in Figure 10-6 that Inheritance is treated as instantiation. All properties and relations of the Product type in Catalogue A may be changed in Catalogue B. For example, the Type of the Product type may be overwritten and the Price may be changed.

Only Products may have Prices. If the parameters of the resources imply different pricing, a separate Product must be defined for each of these parameter values. This means that if the service platform has a parameter for bandwidth and the different bandwidths have different Prices, then a separate Product will have to be defined for each bandwidth.

The Product may be an individual Product or be bundled as a predefined set of other Products. See below on Association Type B.

A Product may have a set of permissible additional Products. See below on Association Type M.

The Product may exclude use of other Products. See below on Association Type C.

The topmost node in the Product hierarchy, by bundling or inheritance, i.e., Association Type B or I, shall itself be an ordinary Product (for sale), and the hierarchy shall not be used for categorization of Products, shall not indicate organization, deals or other information.

A Product may inherit properties or subordinate structures from several other Products.

A Product type may be offered through one or more Sales channels. The Channel may belong to the Operator of the current Catalogue or to some other Operator.

A Product may have one or more Prices that may consist of several Price components. One Price component may be for installation, one for subscription, one for traffic etc.

The Product may have different Prices for different Markets and Deals. See on the Pricing domain.

The Product may be purchased and used at specific Terms.

The Product may be charged at specific Events. The Events may be First month, Subsequent month, Minute, 10 gigabytes, billing etc.

The Product may support certain Quality levels that may contain a Service class and a Description.

A Quality class may contain several Quality parameters that each may contain an Identifier, a Name, Value unit, Value, Description and Parameter area.

Note that the references from Product through Use-es to Customers. Note also the references from Product through Installations to Addresses. These references ally only for Product instances.

10.4 Product instances

Product instances appear under a Contract only.

Each Product instance will have an Identifier containing references to the Catalogue and Product type being used to create the Product instance.

This Identifier is made up of

- a Catalogue,
- a Type and
- a Number.

The Catalogue attribute indicates from which Catalogue the Product type is taken to generate this Product instance. Also, the Catalogue makes the Identifier unique within an Operator. The Number may be a telephone number, circuit number or a similar resource identifier. In addition, the Product instance will have a Resource type attribute.

Note that generalization of Number to an alphanumeric field is for further study.

M.1402 does not state how Product instances are represented in service platforms. Transformation of the Identifier, similar to that of Trail, should be expected. See above.

The full Identifier of a Product instance within an Operator is shown in the next Figure.

Product Identifier			
Contract	Catalogue	Type	Number
987654321	2016new	TRAIL	456789priv

Figure 10-7 Identification of Accounts

The Numbers may be local to the Contract, and be chosen 1, 2, 3, … within this Contract.

In case the network side needs other kinds of identification, the Product instance may contain this identification as separate attributes, maybe combined with an algorithm for conversion. The network identification may be called a Resource identifier. This should not be mixed up with Resource type. See under Product types. The Resource identifier is not covered by M.1402.

The construction of network identifiers, eg. for Trails, is prescribed in M.1401.

Note that the Contract number and Type may contain strings that goes into the network identifier. If there is only one Product instance of each Product type in a Contract, then even the Number may be left out.

The Product instance may identify a resource, such as a phone number. In this case, the phone number goes into the number. M.1402 does not cover repositories of resources, such as of phone numbers.

A Contract may contain several Product instances of the same or different Product types. Note that many commercial software packages allows for one Contract, or even Customer, having only one Product instance of the same Product type.

A Product instance is an instance of exactly one Product type, and there may be several instances of the same Product type. This is prescribed through the S(chema) reference from each Contract instance to Catalogue instances. This reference specifies that:

- All Products subordinate to the referenced Catalogue instances are permissible Products to become subordinate to the Contract instance;
- The S(chema) reference itself may be instantiated to refer from a specific Contract instance to one or more specific Catalogue instances;
- The data instances in the tree subordinate to a Contract are instances of the data under the referenced Catalogues, i.e., Product instances are instances of Product types.

As an alternative to the explicit S(chema reference), the path from the Contract via Operator to its first Catalogue may be used.

A Product instance may be contained in a Product instance bundle, may have additional Product instances and exclude use of others according to the prescriptions in its Product types. The following rules apply:

1. A Product instance may only be contained in one superior product bundle, while a Product type may be contained in several superior product bundle types.
2. A Product's Superior association is an Association that references another Product.
3. This recursive reference via an Association between the Product types put constraints on the Product instances under Contracts. The service platform may not support the Association notion. Hence, the information in the Associations have to be transformed by some algorithm to properties of the Products.
4. A Product instance must contain exactly one Association having the Type Bundle, if the corresponding Product types have some Associations with the Type Bundle.
5. A Product instance may only have a subset of the Association Type Main Product that are given for the Product type.
6. A Product instance may not contain or be part of a Product that is associated by the Association Type Constrained product of the Product type.
7. A Product instance can only have the Association Type Next to other Product instances if this is prescribed between the Product types
8. A Product instance obeys all rules that are stated for its Product type, as well as the Product type's recursively superior Product types via Associations having Type I.

When the Association of Type I is between two Product types, it is used to inherit properties. When it is between a Product instance and a Product type, it is used for instantiation. Inheritance may be seen as a constrained form of instantiation. Inheritance is one-to-one copying; instantiation is one-to-many. Note that the Association from Product instance to Product type may or may not be explicitly stored. The Type of the Product instance Identifier will tell about this Association.

A Product instance may have references via one or more Use-es to Customers. Note that for a leased line Product instance, there may be a Use Customer in each end; therefore, there may be two Use-es to different Customers.

A Product instance may have references via one or more Installations to Addresses. Note that for a leased line Product instance, there may be Installation Addresses in each end; therefore, there may be two Installations to different Addresses. If you want to know which Customers are users at this Address, then this is registered be a reference from Address via the Type to Customer.

Use-es and Installations of Products apply only for Product instances.

Product instances may have a log of customer notifications. These notifications may be treated as Orders. See the brown lines in Figure 12-1.

11 Pricing domain

11.1 Overview

The following Figure includes the Pricing domain.

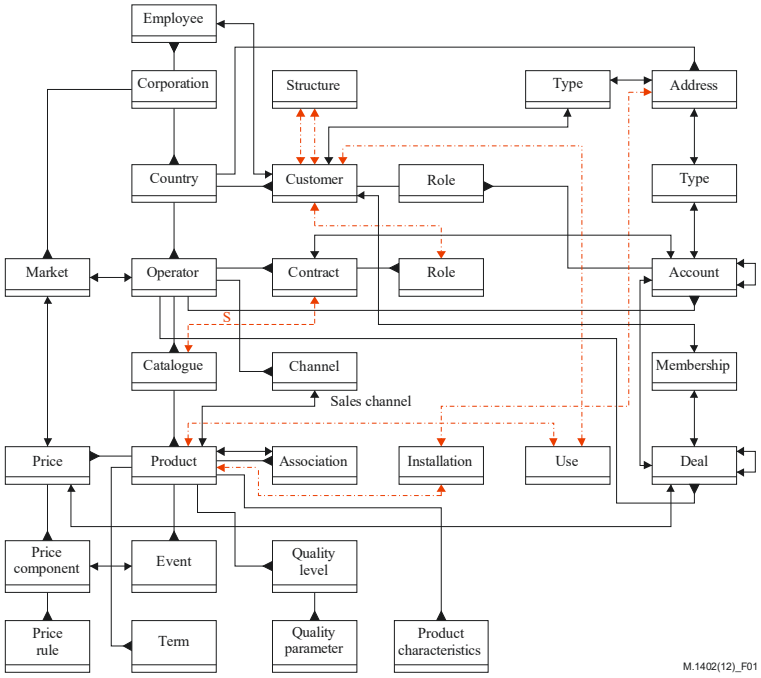

Figure 11-1 External terminology schema including the Pricing domain

11.2 Prices

A Product type may have one or more Prices.

Each Price may be identified by a Price code. This Price code is not identified in M.1402. The Price code may be local to the Product type.

Product	Price
Identifier	Price code
Type	
PHONE	1
	2
	3

Figure 11-2 Identification of Prices

The Prices may apply for one or more Markets or one or more Deals or a combination of a Market and a Deal. These dependencies are stated as references to these. The information about Market or Deal may not be instantiated for the Price of a Product instance. Only the selected Price may be instantiated.

While a Product type may contain many alternative Prices, the Price instance can only have one active Price at a certain moment. See more on this under Price components.

The Customer may be registered for one or more Deals. One of these have to be picked up during ordering of the actual Product. Also, the appropriate Market has to be identified for this Customer; this reference is not shown in Figure 11-1.

The Price entity does not contain any information on amounts. This information is delegated to the Price components.

11.3 Price components

A Price is made up of a set of Price components. Normally, Price components are added to make up the Price. However, the Price type may indicate subtraction, multiplication or other operations.

A Price component may or may not have an identifier.

A Price component may have a Price type that describes this price element, such as recurring, discount, allowance, penalty, and so forth.

The Price component is specified by the Unit to be used for this Price component. The Unit may be given in a certain currency, minutes, SMSes or other.

The Price component contains the Amount of Units for this Price component.

The Price component may contain a Description, ie. a narrative that explains in detail the semantics of this Price component.

The Price component may contain a Valid period for which the pricing is valid. The Valid period may be constrained to the time of the day, and another Price component may be valid at other hours.

Price components are instantiated under the Price of the Product instance, and are used to record usage and charges for this Price component in a certain period. Data for this recording are not defined in M.1402.

The usage recording takes place at certain Events, being prescribed under the Product type. Each record at such an Event instance is typically called a call record. Also, the billing system may use the same data structure to record the details of its accounting. Call records are only relevant for Product instances, and not for Product types.

A Price component is governed by Price rules. Each Price rule consist of three types of components, which define the operation of the rule. These components are Event, Price condition and Price action. The Event refers to the Event of the Price component. The Price condition state if and when the Price action calculations shall take place.

A Product instance will have only one active Price out of multiple options under the Product type. Hence, a Product instance will have minimum one Price, and needs not have multiple Prices as of the Product type. However, the Price may change, eg. due to new Deals with the Customer. Or the Customer has one Deal for daytime and is an ordinary private Customer for other Hours. This may imply that the Product instance will have several Prices.

A Product instance may have several Price components. All these Price components of the Price will contribute to the total cost to be charged for use of the Product instance. Some Price components will be charged by the service engine. Other Price components will be charged by the billing system.

The second previous paragraph states that the Product instance has a disjunction of Prices. The last previous paragraph states that each Price consists of a conjunction of Price components.

12 Order domain

12.1 Overview

The following Figure extends Figure 7-1 with data of corresponding Orders.

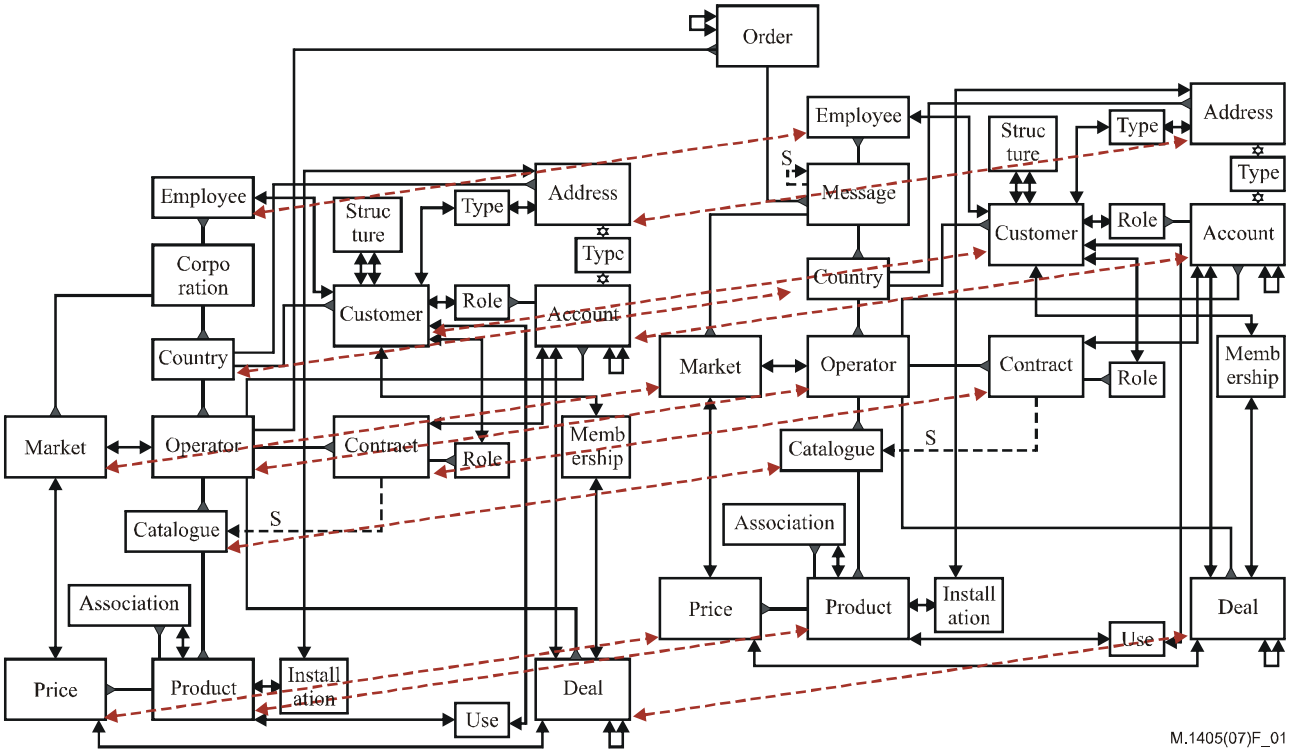

Figure 12-1 External terminology schema of CRM Orders

Note that Figure 12-1 is not updated with all the data of the latest version of M.1402.

12.2 CRM Orders

This section gives an overview on how Orders are defined and used for the service management. These Orders may be called CRM Orders. CRM stands for Customer Relationship Management.

This section is not a full tutorial on these Orders.

The text in this section is based on ITU-T M.1405. Formalization of orders for service management [4].

An Order identifies a long transaction between IT systems, organization (unit)s or users. A long transaction means that it may visit several of these actors, or the same actor several times, before the transaction terminates.

The Order is created and given an Identifier by an Operator whenever a request comes to the Operator. The Order is only valid within the domain of this Operator, and across systems, organizations and users of this Operator. So, the Order may be used towards the customers and partners of the Operator. These are free to define derived Orders within their domain, but have to

answer through the original Order. So, Orders may have derived Orders, and this reference between Orders needs to be managed.

A CRM Order may be one of the following kinds:

- Customer request
- Customer notification
- Network order
- Activation
- Bill
- Payment

Note that each kind of Order may only contain a subset of the data shown in Figure 12-1. Also, each of these Orders may be split in Orders to different destinations, eg. to another network Operator.

The management of work flows will not be addressed in this Technical report.

Orders may also be split into sub-orders, called Messages. A Message can be compared with a statement, some may be short and simple, while others may be long and complex. Different Messages may be sent on different paths through the domain of the Operator. Typically, a work Order may contain only one Message or few Messages.

The Messages may contain data. The data are the same as what is described in the previous sections of this book. The general schema of the data is also the same, except all classes are local to the class Message. Each data instance in an Order is local to a Message instance.

The Messages bring information about events which are going to happen, and give reports on what has happened. Hence, the Orders are means for the management of events, and they identify who will do what and has done what. Therefore, the Orders are essential means to ensure data quality. The Orders give a full overview of what is planned to happen, and of what has happened.

It is also possible to define Message templates that allows for only a subset of the generic data structure of a Message. Hence, you may have one Message class for provisioning, one for trouble-ticketing, one for notification of customers, one for orders to partners etc.

In Figure 12-1, the Main register is depicted to the left, and the Order register to the right. Note that Orders are contained in the Operator.

The brown dashed diagonal arrows depict the time dimension of each entity in the Main register. All mentioning of this entity instance in various Orders are listed under this entity instance.

If you look up on an entity instance within an Order, you find a reference to the current status of this entity instance in the Main register.

In the above Figure, we have not shown a data structure for data flow. This could also be a topic for a larger report on Orders. You may find appropriate data structures for this in documentation of Work flow engines. But note that the entire work flow may be defined as a database application, and a process engine is not needed.

Note that the Order register and work flow should be implemented together with the CRM main register in one CRM system. The Order management and the work flow management should not be implemented as separate systems.