Data for a Telecom Operator

ARVE MEISINGSET

Arve Meisingset is Senior Research Scientist in Telenor Group Business Development and Research

Any IT system can only read and write data and carry out functions on these data. Unfortunately, many designers define the functions within the mappings to input-outputs, and not as mappings between data within the system. This makes them believe that IT systems may carry out work processes in addition to managing data. We advocate for a data centric view of IT systems, and strongly discourage the use of the process and workflow perspectives. This paper gives an overview of data definitions in ITU-T Recommendations M.1401-M.1405 for services, networks and orders of a telecom operator. A separate Annex about The Language tells how these data are defined and used.

Motivation

Process designers are violating the so-called 100% principle, that all functions should be defined among the data, not from data to input-output. The 100% principle ensures that all data definitions and their functions are centralised and you cannot derive or insert inconsistent data. The 100% principle does not allow you to bypass any function. Thus it does not allow you to construct a system that may lead to inconsistencies.

In a process oriented design, the functions are split on various processes, and what functions are executed will then depend on what processes are activated. Execution of different processes may lead to inconsistent data.

The 100% principle encourages design of IT systems like when defining spreadsheets. In spreadsheets, functions are defined as formulas between the cells within the spreadsheet. In database applications you often define and use more complex data structures than offered by spreadsheets. The data centric view that follows from the 100% principle may reduce complexity to a fraction of designs using workflow and process thinking. And the solution will become much more flexible in use.

In the data definitions presented in this paper, all functions are defined subordinate to data. This allows you to modify any data in any sequence as long as you do not violate the constraints stated subordinate to the data. This means that you will not need to follow a strict workflow to carry out a task. The functions (among the data) will check if your modifications are valid and warn you if you make mistakes.

In this paper we define data as perceived by the human user at his human-computer interface. Whether data are permanently stored or not is immaterial for the specification. Derived data are included in the specifications. This makes the specification different from other approaches that only define concepts or data that are stored.

Conventions

The main body of this text uses a graphical notation only. The figures use rectangles to indicate object classes. Lines with reversed arrowheads indicate subordinate object classes. Two-way arrows indicate references between object classes. A dashed one-way arrow supported with an S at the arrowhead indicates a schema reference from a population. Each element subordinate to a schema may be instantiated to several elements subordinate to the population. And, the schema reference may itself be instantiated, like any other reference. See more on the notation in The Language Annex. Orange dotted lines indicate references between the main register part and the order register part.

Formalization of Data for Service Management

ITU-T Recommendation M.1402 contains definitions of data for service management. The Recommendation focuses on defining data for products, customers, accounts, contracts, deals, addresses, prices, and various segments and relationships between all these. Some attributes that are essential for identifying objects and associations between them are defined, as well. Order data and detailed mappings to network elements are not covered. Also, billing, marketing, sales, retailer, distributor, number management, device management, location management, logistics and finance are not covered. Finally, alarms, trouble tickets, call records and other call centre information are missing.

Boxes that do not have a reversed arrow head are contained in the root node, ie. in the schema itself. Derivation of data is indicated by the combined dashed and dotted lines.

 ISSN 0085-7130

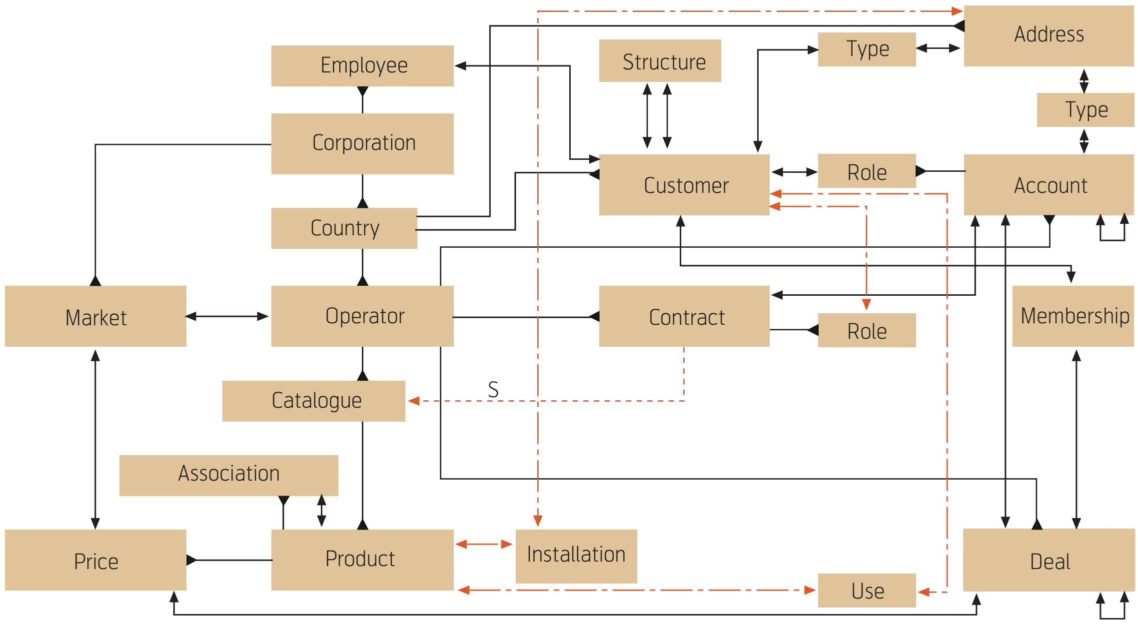

Figure 1 External Terminology Schema for service management

Corporation is included such that data of different Corporations may be executed on the same service platform. Countries indicate all areas that the Corporation has Operators in, has Customers in or has relationships to, eg. interconnect to external Operators. Countries are identified by Country Codes (CC) assigned by ISO. Operators are identified by ITU Carrier Codes (ICC) assigned by the Administration in that Country. Note that an Operator operates within this Country only, under the jurisdiction of the Country.

We observe that Country in this graph is not a Country that exists independently of Corporation. The Country is here a means of structuring the Corporation, and so is Operator. The Country may have an attribute Number of customers. This is the Number of customers of that Corporation in that Country. Another Corporation may have a different Number of customers in that same Country. Since we are defining a tree of data (with references between the nodes), we do not need to define ternary (or *n*-ary)

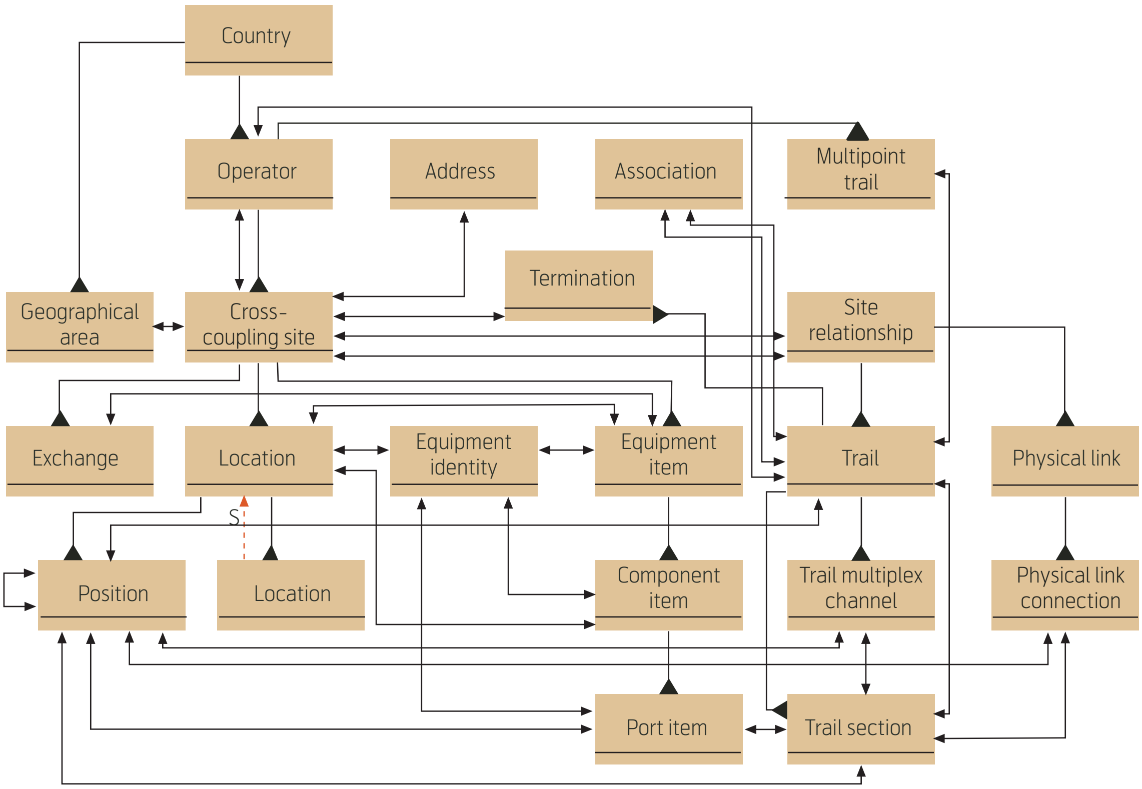

Figure 2 External Terminology Schema for network resources

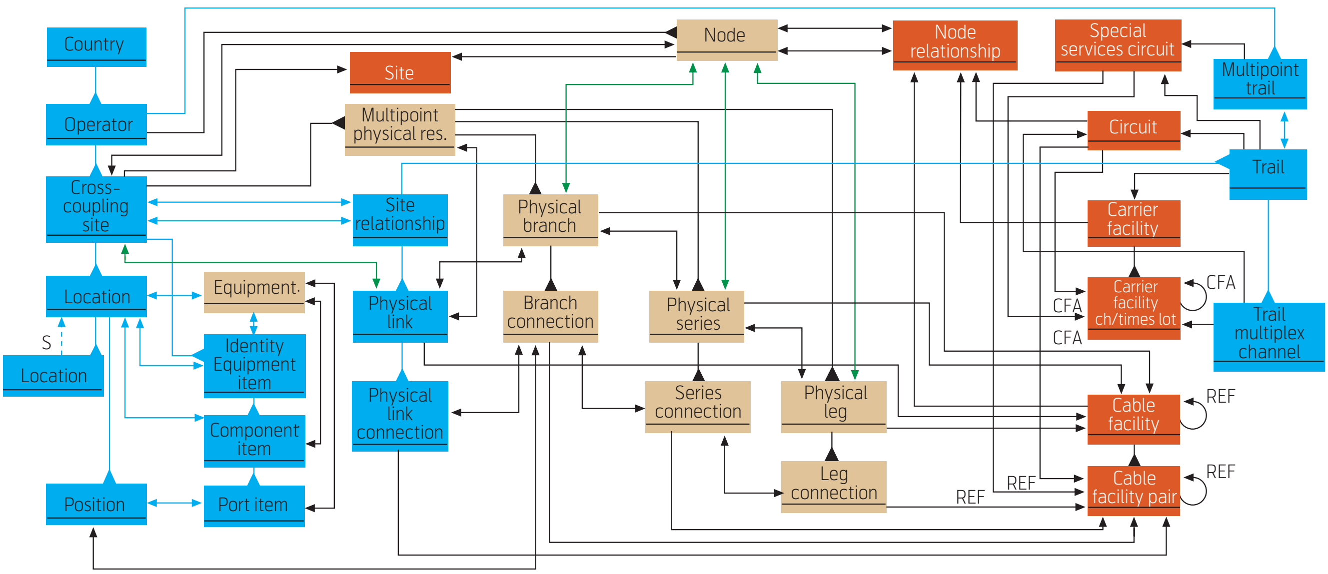

Figure 3 Identification of Physical Network Resources

associations, for attributes of Countries of different Corporations, as would be needed in a conceptual approach using flat name spaces.

Formalization of Interconnection Designations among Operators' Telecommunication Networks

ITU-T Recommendation M.1401 contains data definitions for designations of interconnections and other information about network resources that are required to be communicated between operators.

Note that in Figure 2, Corporation is not included. This may mean that it is just suppressed in the presentation, or that Corporation is not needed. If it is suppressed, the top structure is as of service management. If it is excluded, it may mean that the network resources are identified independently of the Corporation owning the resources.

Identification of Physical Network Resources

In Figure 3 the blue boxes are from M.1401. The brown boxes are extensions for the Physical network, ie. for outdoor plant. The orange boxes depict interpretations of ANSI T1.253. The mappings from the two first to the last are shown as one-way references by one-way arrows.

Formalization of Generic Orders

Except for Order and Message and their references, the right-hand part of Figure 4 is a copy of the left-hand part. The orange arrow shows the correspon-

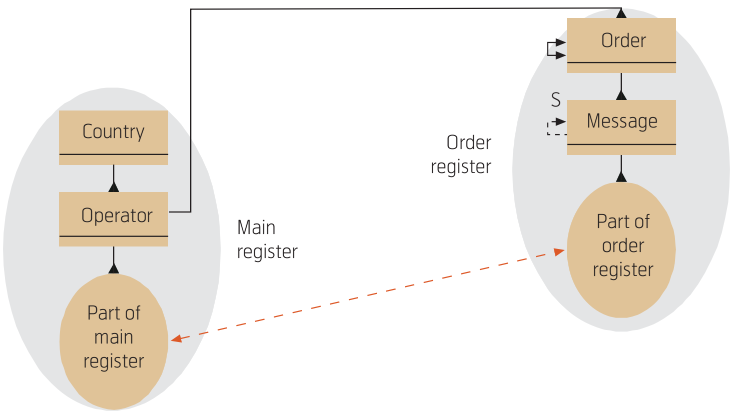

Figure 4 Orders and their relation to their main register

dences. Country may be contained in Corporation; this is not elaborated in this Recommendation.

The reader should note that data definitions for the contents of (Messages of) Orders are the same as for the main register. The main difference is the scope, as order data are contained in the Messages and not within the system – as of the main register.

A Message is treated as a set of statements in one sentence. The statements are elementary and correspond to predicates in predicate calculus. An example statement is "Customer abc has Role Owner of Account 123", or in predicate calculus "Role(abc, Owner, 123)". In the corresponding Message type, we find the template "Customer x has Role y of Account z", ie. "Role(x, y, z)".

The main register consists of a set of elementary statements and so does the order register. In the main

ISSN 0085-7130 ©Telenor ASA 2009

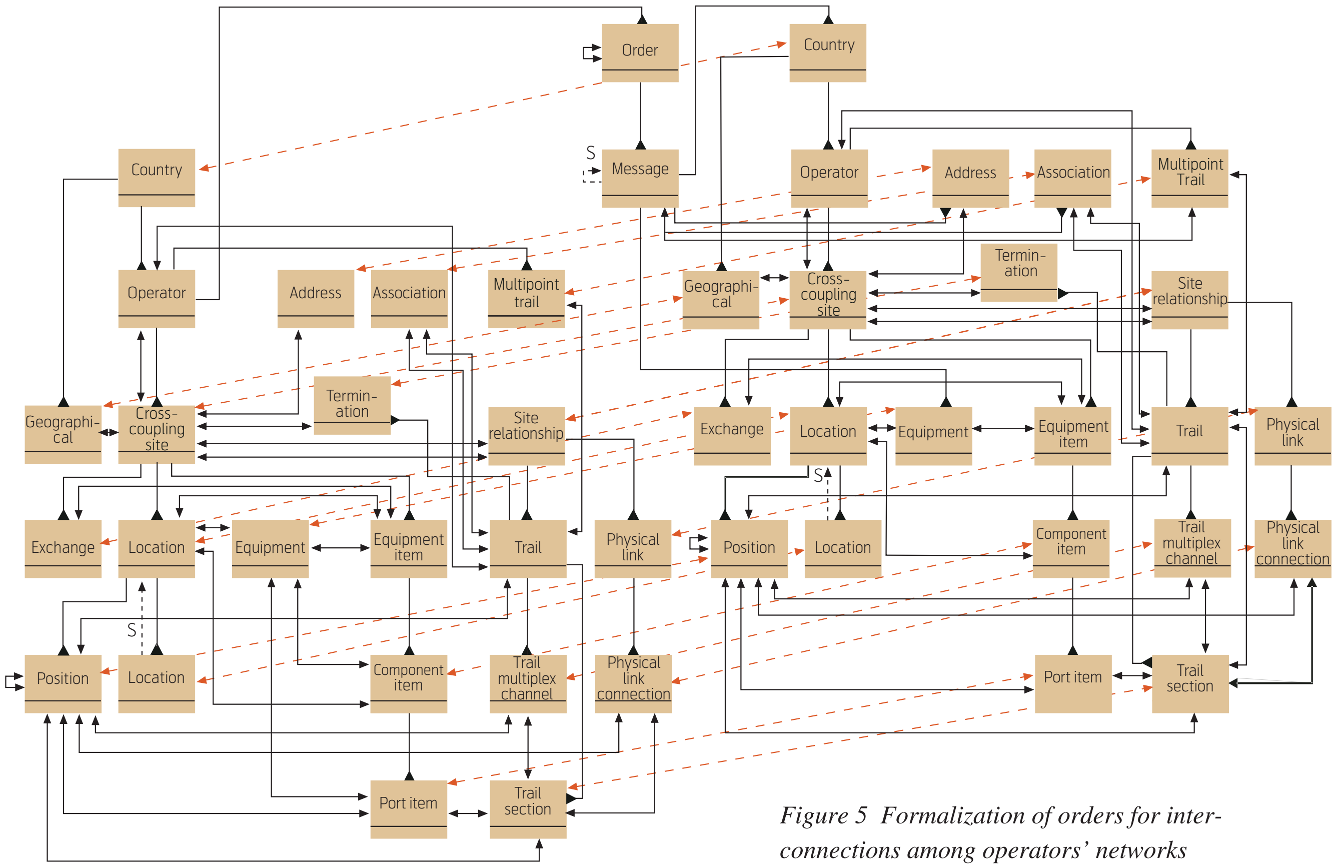

Figure 5 Formalization of orders for inter-connections among operators' networks

NOTE: The left-hand part is identical to [ITU-T M.1401]. Except for Order and Message and their references, the right-hand part is a copy of the same. The orange arrows show the correspondences. Country may be contained in Corporation; this is not elaborated in this Recommendation.

register, the schema of the data classes is instantiated into a population of data instances. In the order register, the message types are instantiated into message instances. A message type may contain most of the data definitions that are contained in the schema of the main register, and a message instance may contain some of the data instances that are contained in the population of the main register.

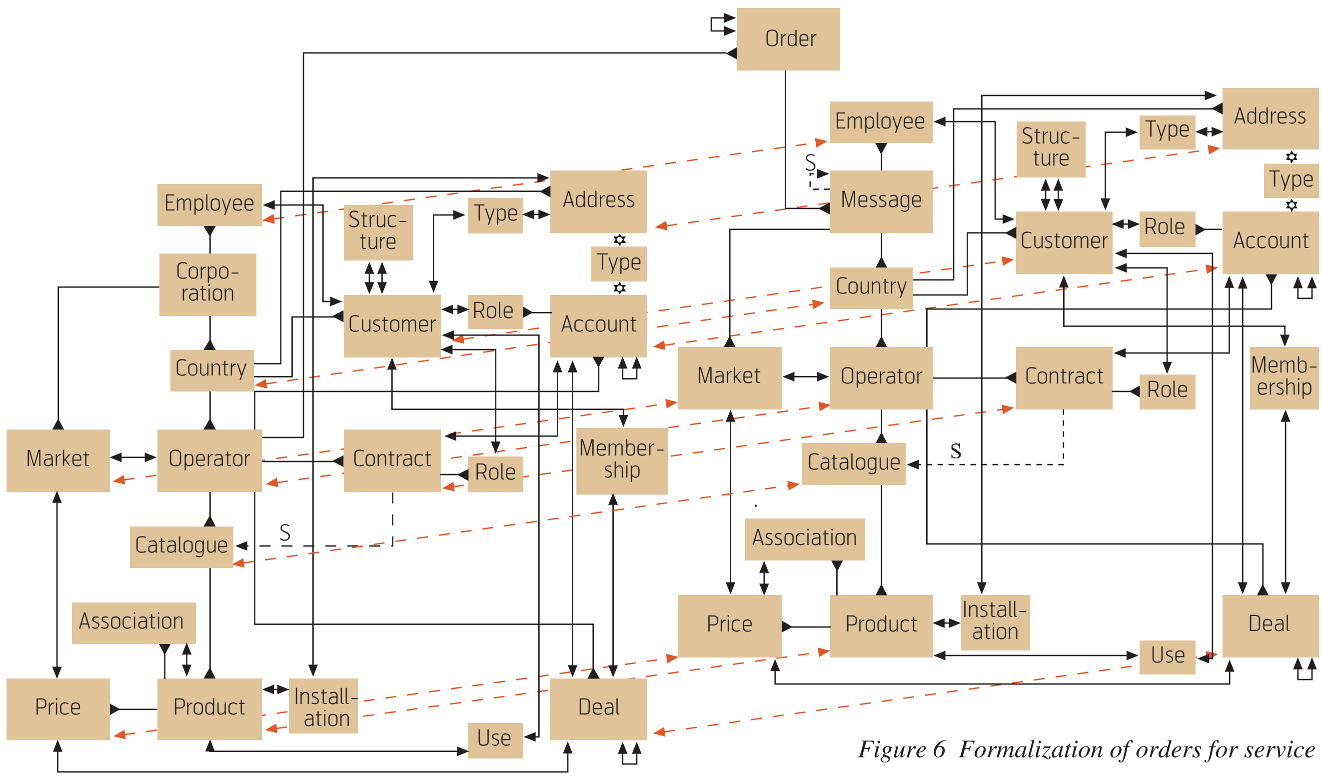

Figure 6 Formalization of orders for service management among operators

References

ATIS TMOC. *Identification of Physical Network Resources*. December 7, 2004.

ANSI T1.253. *Location Entity ID*. American National Standards Institute, 2001.

ITU. *Data oriented human-machine interface specification technique – Scope, approach and reference model*. Geneva, International Telecommunication Union, March 1993. (ITU-T Rec. Z.352)

ITU. *Data architecture of one software system*. Geneva, International Telecommunication Union, Feb 2007. (ITU-T Rec. Z.601)

ITU. *Formalization of interconnection designations among operators' telecommunication networks*. Geneva, International Telecommunication Union, July 2006. (ITU-T Rec. M.1401)

ITU. *Formalization of data for service management*. Geneva, International Telecommunication Union, Aug 2007. (ITU-T Rec. M.1402)

ITU. *Formalization of generic orders*. Geneva, International Telecommunication Union, Aug 2007. (ITU-T Rec. M.1403)

ITU. *Formalization of orders for interconnections among operators' networks*. Geneva, International Telecommunication Union, Aug 2007. (ITU-T Rec. M.1404)

ITU. *Formalization of orders for service management among operators*. Geneva, International Telecommunication Union, Aug 2007. (ITU-T Rec. M.1405)

Annex: The Language

ITU-T Recommendations M.1401-M.1405 define External Terminology Schemata according to ITU-T Recommendation Z.601. The notation used is introduced in ITU-T Recommendation M.1401 Appendix III. A more complete exposition of the notation is found in *The HMI specification technique*, Telenor FoU N 54/96. The following explanation uses the style used in contributions to ITU-T SG4 Question 2, Designations for interconnections among network operators.

An External Terminology Schema defines terms and grammar of presentations at human-computer interfaces to an application system. One system may have several External Terminology Schemata; one for each national or company terminology being supported by the system.

Figure A.1 depicts an extract of an External Terminology Schema. The indentations replace opening parentheses. Closing parentheses are indicated by line shifts. Underlining indicates object class. Other entries indicate attributes or attribute groups. Specification of value types or value sets are not shown in these examples.

A schema consists of class definitions. The class labels are organised in a branch of a data tree (possibly with references between the nodes of the tree). The first example shows no reference. Class labels are local to the superior node of the tree; hence, similar class labels – like Identifier, Identifier and Identifier – may be reused for different purposes and have different definitions. This is an important feature of end user terminologies, that the terms are not globally unique, but local to each other. Therefore a subordinate node can (in principle) never exist without its superior node, but we will in the concrete representation allow for some simplifying suppressions like we also do in the graphical notation for the schema where some lines are suppressed.

The subordinate entries within a list like CC (Country Code), Operator and Town (of Country) state three alternatives, which means that some, but not all of these entries may exist in the instantiated data. So, the lists state disjunctions while the superior node states a strong unbreakable condition. The entries of the lists of the External Terminology Schema have a similar role as of elementary statements in a sentential logic. However, this is not sentential logic; it is a noun logic; all you state is an organisation of terms. And this organisation of the External Terminology Schema of the overall data tree is a normalised organisation of the data. So, the External Terminology Schema is something like a dijunctive normal form. This is analogous to normalisation in the Relational model of databases.

The External Terminology Schema may contain derived data, but use of derived data will not be

```
Corporation
    Identifier
    Country
        CC
        Operator
            ICC
            Cross-coupling site
                Identifier
        Town
            Identifier
```

Figure A.1 Example External Terminology Schema without references shown or explained in this text. An External Terminology Schema contains only elementary statements of data. Compound statements are defined in Contents Schemata. A Contents Schema defines the abstract syntax of and permissible operations in a screen picture or report. So, one screen picture is considered to be made up of one compound sentence only. This sentence will have a subject, ie. the object that you talk about. We call this object the base, and we introduce a Base function to define the Start and the Stop of the global distinguished identifier of this base object.

```
Corporation
    Base ( Start
    Identifier
        Base (Stop
    Country
        CC
        Operator
            ICC
            Cross-coupling site
                Identifier
        Town
            Identifier
```

Figure A.2 Example Base function

In this Contents Schema, we have chosen a Corporation as the base, and we will list all its Countries with subordinate Operators (with their Cross-coupling sites) and Towns of each. Note that the Base function serves as a special kind of parenthesis.

In the next example, we choose an Operator as the base.

In Figure A.3, we will look up a particular Operator in order to verify whether it exists and list all its Cross-coupling sites.

The abstract syntax of the response may look as in Figure A.4. Note that instantiation is done by copying from the schema. Therefore, we say that the schema is homomorphic to its population (of instances). Note also that the copying leads to the use of significant duplicates (in lists). So all three Cross-coupling site

```
Corporation
    Base ( Start
    Identifier
Country
        CC
        Operator
            ICC
                Base ( Stop
            Cross-coupling site
                Identifier
```

Figure A.3 Example Contents Schema

```
Corporation
    Base ( Start
    Identifier
        TELENOR
    Country
        Identifier
            NOR
        Operator
            ICC
                TELENOR
                Base ( Stop
            Cross-coupling site
                Identifier
                    C1
            Cross-coupling site
                Identifier
                    C2
            Cross-coupling site
                Identifier
                    R14
```

Figure A.4 Example Contents Population

entries appear at the same level in the above example. They may or may not contain unique identifiers.

In the above example we have filled in values in capital letters, but they may have any format. Note that if no constraint is attached to a (leaf) node of the data tree, you may attach any subordinate data. This is why we call this an attachment grammar. The data tree acts as a syntax tree according to a rewriting grammar. In a rewriting grammar you throw away the superior intermediate nodes and are only left with the leaf nodes in the final production. In the attachment grammar you do not throw away the superior nodes, as they constitute the most important part of the expression, ie. the context of the values. The entire Contents Schema in the above example expresses the identifier of one object instance, the Operator TELENOR, and lists its Cross-coupling sites. If you do not express all this, you will have to leave the interpretation to the human user and/or the programs. This is what happens in (all) alternative data definition languages. The shown Contents Schema makes the interpretation unambiguous for humans and machines.

In alternative data definition languages, you first list all the terms in a glossary, and most often see to it that they are globally unique. Then you define the grammar, ie. all the production rules. When these rules are executed, you produce a syntax tree. And finally you use only the leaf nodes in the final production.

In the attachment grammar, you place the terms in a data tree, not in one list. The data tree with references states a template for what is permissible instances. So, the data tree is the production rules and the syntax tree. Constraints on the nodes in the tree are stated

```
Corporation
    Identifier
    Country
        CC
        Operator
            ICC
            Cross-coupling site
                Identifier
                Town
                    <> ' Cross-coupling site '
                    Operator ' Country ( Town
        Town
            Identifier
```

Figure A.5 Example External Terminology Schema with a one-way reference

subordinate to each node and state references between the nodes. This is similar to attribute grammars, where functions are stated subordinate to the nodes of the syntax tree.

Now we are ready to give an example of references between objects. See Figure A.5. We start with a one-way reference defined in an External Terminology Schema.

```
Corporation
    Identifier
        TELENOR
    Country
        CC
            NOR
        Operator
            ICC
                TELNOR
            Cross-coupling site
                Identifier
                    C1
                Town
                    <> ' Cross-coupling site '
                    Operator ' Country ( Town
                    ( Identifier ( OSLO
            Cross-coupling site
                Identifier
                    C2
                Town
                    <> ' Cross-coupling site '
                    Operator ' Country ( Town
                    ( Identifier ( OSLO
            Cross-coupling site
                Identifier
                    R14
                Town
                    <> ' Cross-coupling site '
                    Operator ' Country ( Town
                    ( Identifier ( BERGEN
        Town
            Identifier
                OSLO
        Town
            Identifier
                BERGEN
```

Figure A.6 Example External Terminology Population with one-way references

Town subordinate to Cross-coupling site states that the site may have subordinate Towns. We want these Towns to be roles of references to Towns subordinate to Country. Note that due to the use of local class labels, the role label Town needs not be different from the label of its referenced object Town.

> <> states that there is a condition on the Town (role) that it can only exist if the condition is satisfied.
>
> ' states a navigation up to the superior node, here first to the Cross-coupling site.
>
> (states navigation down to a subordinate node, here to the (referenced) Town.

Figure A.6 shows an instantiation of the same.

We may now state a Contents Schema from the previous External Terminology Schema, see Figure A.7.

```
Corporation
    Base ( Start
    Identifier
    Country
        CC
        Operator
            ICC
                Base ( Stop
            Cross-coupling site
                Identifier
                Town
                    {<> ' Cross-coupling site '
                    Operator ' Country ( Town}
```

Figure A.7 Example Contents Schema with a one-way reference

In Figure A.7, we have put in control scope brackets {, } to state that the condition shall not be executed in the Contents schema, ie. there is no list of Towns under Country in this schema, while the expression in the control scope parenthesis expresses the correct navigation in the External Terminology Schema. The External Terminology Schema is homomorphic to its Contents Schemata. The resulting instantiation may look as shown in Figure A.8.

How the above data are finally presented as concrete syntax at the human-computer interface depends on the style guide used. However, we would like to fold out the data tree from the Contents Population such that the user may uniquely interpret the data. Figure A.9, without explanation, shows how this may be done.

```
Corporation
    Base ( Start
    Identifier
        TELENOR
    Country
        CC
            NOR
        Operator
            ICC
                TELNOR
                Base ( Stop
            Cross-coupling site
                Identifier
                    C1
                Town
                    {<> ' Cross-coupling site '
                    Operator ' Country ( Town}
                    ( Identifier ( OSLO
            Cross-coupling site
                Identifier
                    C2
                Town
                    {<> ' Cross-coupling site '
                    Operator ' Country ( Town}
                    ( Identifier ( OSLO
            Cross-coupling site
                Identifier
                    R14
                Town
                    {<> ' Cross-coupling site '
                    Operator ' Country ( Town}
                    ( Identifier ( BERGEN
```

Figure A.8 Example Contents Population with one-way references

Figure A.9 shows an Example Layout Population, while we have not shown the corresponding Layout Schema. Except for suppressions, the Contents Schema is isomorphic to its Layout Schemata.

Note that from the above presentation, the user may read the data structure and follow the schema graph line by line, and item by item. The entire screen starts at one node and spreads out like a (connected) tree from this base.

```
Operator
Country           ICC
Corporation   CC
Identifier
TELENOR       NOR  TELNOR

Cross-coupling site
Identifier        Town
                  Country                Identifier
                  Corporation     CC
                  Identifier
C1                TELENOR         NOR    OSLO
C2                TELENOR         NOR    OSLO
R14               TELENOR         NOR    BERGEN
```

Figure A.9 Example Layout Population of an Operator, its Cross-coupling sites and their Towns

Operator

Corporation	Country	ICC
TELENOR	NOR	TELNOR

Cross-coupling site

Identifier	Town	'Country	'Corporation
C1	OSLO	NOR	TELENOR
C2	OSLO	NOR	TELENOR
R14	BERGEN	NOR	TELENOR

Figure A.10 Example screen picture

The reference from Cross-coupling site to Town may be misleading in Figure A.9, as the user of the screen may think that the site refers to Corporation. Therefore, often reversed local naming is used, as shown in Figure A.10. Note also that many suppressions are used here to simplify the presentation.

Hopefully, now you have learned some of what is behind a human-computer interface.

Note that the class labels (in the schemata) appear as headings on the screens. You write nothing in the specification that will not appear on the screens in one way or another. Hence, there is no room for scribbling concepts in the specifications. Every item will appear as data somewhere. You are not allowed to use computerish names, but will have to use the same labels as the users see.

The motivation of the presented way of specification is that we want to define the terminology and grammar of the presentations at the human-computer interfaces. And we want to give the users access to these specifications, using their own terminology.

If you study the previous example schemata carefully, you may realise that there is very little that needs to be specified in order to create a huge system, but everything has to be to the point, and be right.

To create good data designs is an art. ITU-T Recommendation Z.352 Annex A gives an introduction to this art. You should bear in mind that every data item will have to be managed and read by some user, and it is your responsibility to make this management efficient, flexible and appropriate. These goals are achieved through the data design itself, and not through management functions of the data. In conceptual approaches you tend to disregard these design considerations, and hope that they may be dealt with later. They cannot.

For an introduction of Arve Meisingset, please turn to page 80.