INFORMATION MANAGEMENT FOR THE EXCHANGE OF DATA AMONG ELECTRIC UTILITIES: PROPERTY MODELLING

Rainer Bacher
ETH Zürich
The Swiss Federal Institute of Technology,
Zürich, Switzerland
Rainer.Bacher@eeh.ee.ethz.ch

Dirk Lambrecht
Forschungsgemeinschaft für Hochspannungs- und Hochstromtechnik e.V.,
Mannheim, Germany
lambrecht@fgh-ma.de

Eirik Selvik
SINTEF Energy Research,
Trondheim, Norway
eirik.selvik@energy.sintef.no

KEYWORDS

Information modelling, Data modelling, Data exchange, Information sharing

ABSTRACT

For most electric utilities the exchange of data within the utility, among utilities and between utility and supply industry is a major concern. The exchange of data implies a common understanding of data pieces: names, attributes, values and methods and their exact meaning. Thus data has to be modelled on a higher level, using ”information modelling”. This paper presents a model for the description of properties enabling the user to exchange properties on basis of a common information model.

1. INTRODUCTION

Due to the opening of the electricity market and the general tendency of integrating information systems the importance of a formalised information and data exchange is growing. The ElectroNet[1] project aims at defining a common European information and data model for the exchange of data for system and operational planning of HV and EHV networks [1]. Important aspects of this information model have been described in [2, 3, 4, 5].

In order to understand the difference between information, data and their associated models, the following definitions must be given [6]: ”*Information is knowledge of ideas, facts and/or processes; Data are symbols which represent information for processing purposes, based on implicit or explicit rules. An information model is a formal description of ... information ... which forms a model of a portion of interest of the real world. A data model is an information model that is specialized to take account of a particular instantiation method*”.

In this paper we describe a conceptual information model which is independent on any particular instantiation form.

The main features of the ElectroNet model are summarized as follows.

1) A model for the things in a network and how they are connected.
2) **A model for the properties possessed by the objects in the network.**
3) A model for the state of the network and the way in which the connections of the objects in the network and their properties can depend upon the state.
4) A model which shows how electrical functions and equipment are presented on a schematic diagram.

In this paper, we describe important parts of the above model 2): *Properties are aspects of physical objects that can be measured or observed*. They are representing the core of exchanged information. As a consequence the topical information model for the description of properties is the most important part of an overall information model. Its final quality may limit the applicability towards practical, real-world problems – or – it may allow flexible and easy to handle future possibilities.

Today, different standards and quasi-standards exist for the description of properties. The most significant ones are IEC 61360 [7], AP212 [8] and the EPRI-CIM [9] / IEC 61970 [10]. However for the field of electrical power engineering a wider set of properties will be needed.

This paper first gives an overview about these existing standards in section 2. In section 3, the most important aspects of the information modelling language EXPRESS are summarized. This is necessary in order to understand the notation of the ElectroNet property model parts given in section 4. The paper ends with concrete property modelling examples in section 5 and conclusions in section 6.

---

[1] ElectroNet: An ESPRIT IV European Community project No 22297, 1996-1999 (http://www.eus.ee.ethz.ch/electronet)

2. STATE OF THE ART

Nowadays data and thus properties of electrical energy supply systems are exchanged on the basis of commonly used and well-known exchange formats, like the ones of UCPTE [11] and IEEE [12] or on the basis of data models that are defined using the relational paradigm. These exchange formats and models explicitly describe network components together with the attributes, thus providing a fixed model for the definition of a network. The attributes (properties) are considered to be independent on each other and are of the types: real, integer, text string and flag. They have fixed units and they are used to represent one fixed level of a property (maximum or minimum or nominal etc.).

The actual situation using these “hard-coded” data models and exchange formats does not meet the requirements of modern information technology. A much wider range of properties including the description of dependent properties, matrix properties and some flexible grouping mechanisms should be covered. There is also a need to have a mechanism for various alternative representations of network components in the same model framework.

The property model presented here has been created in the context of the ElectroNet project taking into account already existing standards and quasi-standards. The most important ones are AP212, IEC 61360 and the EPRI CIM/IEC 61970.

In order to be able to judge about the possibilities offered by these standards, they are briefly reviewed in this section.

2.1 IEC 61360

IEC 61360 ”Standard data element types with associated classification scheme for electrical components” [7] is a standard defining rules on how to define data element types (DETs) (properties, property types, attributes, parameters ..). Each DET shall be provided with a clear definition. No ambiguity shall arise. Attached to the definition is also information concerning the form of presentation of values of the DET, the unit of values, name, symbol, references to source documents for the definition etc.

IEC 61360 distinguishes between two kinds of DETs: **Quantitative DETs** and **Non-quantitative DETs**. A quantitative data element type represents a physical quantity, a quantity of information or a count of objects. A quantitative DET is represented by a numerical value. A non-quantitative data element type identifies or describes an object by means of codes, abbreviations, names, references or descriptions.

IEC 61360 recognises the fact that the actual value of a DET can be influenced by the actual value of another DET. In these cases, the first DET is conditioned by the latter. IEC 61360 requires that such conditions are specified as part of the definition of a DET.

2.2 AP212

ISO 10303 ”**St**andard for the **E**xchange of **P**roduct Model Data” is intended to provide an unambiguous representation of computer interpretable product information throughout the life-cycle of products [13].

STEP does not only define basic structures for product representation, but also information models dealing with specific problems of particular application areas, the so-called APs (application protocols).

Application protocol AP212 ”Electrotechnical design and installation” [8] is concerned with aspects of electrical engineering. Here predefined data models are given, which can be used to describe different aspects of electrotechnical installations. One part of AP212, the UoF (unit of functionality) ”data element” deals with the representation of properties.

These data elements can be attached to entities in a very flexible way. So nearly all main entities can be equipped with an arbitrary number of data elements and there is no limitation concerning types of data elements.

Three different types of data elements are distinguished, namely **standardised data elements, user-defined data elements** and **predefined data elements**. They may have hierarchical or dependency relationships.

Standardised data elements are described by means of their data element code and the name of the corresponding standard. So particularly IEC 61360 is supported.

User-defined data elements are represented by their data element code and definition. They additionally reference to source documents and custodians.

Both standardised and user-defined data elements are described as single values or lists of values. Single values are further tagged as minimum, maximum, nominal, specified or typical level. Single values may be strings, numbers (positive or negative integers or real numbers), logical values, binary values (sequence of binary digits) or measures with unit.

Predefined data elements are explicitly describing mainly geometrical and environmental properties. The only electrotechnical properties defined here are rated power, rated current and rated voltage distinguishing AC and DC values.

2.3 EPRI CIM / IEC 61970

The EPRI-CIM (Common Information Model) is a data model developed for EMS (Energy Management System) purpose [9]. [10] (IEC 61970) is a follow-up project.

In this data model a fixed attachment of properties to entities is given. For example: A transformer (model) consists of a resistance R1, two reactances X1 and X2 and a shunt capacitance B1. The connectivity of this transformer model is implicitly assumed to be known. This model can easily be mapped to structures of relational data bases. As a consequence each entity is characterised by means of the attributes describing it.

One special feature of the EPRI-CIM (not found in other models) is the so-called curve concept. Curves are relationships between one independent variable (X-axis) and one or two dependent variables (Y1- and Y2-axis). They are mainly used as schedules. This concept is reused in the current project in slightly extended form.

3. INFORMATION MODELLING LANGUAGE

In order to describe an information model, a language must be chosen: EXPRESS [6] is a data description language defined as a part of the STEP standard (ISO10303-11 [13]). EXPRESS-G is a subset of EXPRESS, its graphical representation. EXPRESS has similar objectives to other modelling languages, such as NIAM, IDEF1X, ER or UML. In order to stay within the framework defined by STEP [13] information models given herein are depicted using this EXPRESS[-G] notation.

3.1 EXPRESS-G symbols

Figure 1 summarises the most important symbols needed to understand EXPRESS-G notation. Here each rectangular box corresponds to an entity (class in Object Oriented Programming (OOP)).

The small circles always indicate the end of the line to which they are attached. Boxes with a line at the right hand side are showing simple types such as REAL, INTEGER, STRING, NUMBER, BINARY, LOGICAL and BOOLEAN. Boxes with dashed border and a dashed line at the right hand side are indicating enumeration types, representing a limited list of predefined attributes, e.g. an enumeration type ”colours” may comprise a list: ”green, black, blue”.

Structures are indicated by means of lines connecting the corresponding boxes.

Thin lines are showing relationships. They are implying a direction, so one instance of the entity at the non-circled end ”points” to at least one instance of the entity at the circled end of the line, where the corresponding attribute belongs to the first entity.

One instance may as well point to an aggregation of instances. Basically aggregation types: set, bag, list and arrays are distinguished. The most important aggregation types used here are sets and lists.

Within a set one instance may only appear once. The sequence of instances is not relevant. This is represented using the notation *S[a:b]*, where *a* describes the minimum and *b* maximum number of instances. If sets should not be limited, this is indicated by means of a question mark instead of letter *b*.

Within a list the sequence of instances becomes relevant. Lists are indicated by means of the notation *L[1:?]*.

Relationships may also have inverse attributes; this is indicated by means of the term (INV). So the corresponding attribute belongs to the entity (“has-a”) at the circled end of the connecting line. Thick lines are representing superclass – subclass hierarchies (”is-a”).

In Figure 1, entity A has an attribute 1 (a enumeration type) and an attribute 4 (a list of one or two real number values). Entity A is an Entity B type (the thick line). Entity A has also an attribute 2 of type entity B. Entity B has an attribute 3 (a set with at least one element) which points back to entity A.

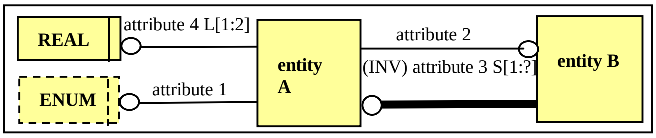

Figure 1. Most important EXPRESS-G symbols

4. ELECTRONET PROPERTY MODEL

Based on the standards and quasi standards reviewed, the ElectroNet consortium has created a very flexible topical information model for the representation of both numerical and non-numerical properties. As the information model should be made as generic as possible, but still directly applicable to practical systems, not only requirements of planning applications must be met, but other technological areas must be supported as well.

The property information model is organised using schemas. Each schema depicts the information model needed for the representation of particular aspects of properties. They are given in the following subsections.

In general all properties are described by means of their classification, stating what an instance of the entity property is intended to represent. An optional attribute is indicating the admitted levels, distinguishing, whether a property is to be understood as a maximum, minimum, typical, nominal, setting, calculated, actual or estimated value. One property may be conditioned by one or more other properties.

The attachment of properties to objects – to which they belong - is done by means of the entity possession_of_property. E. g. all properties needed to describe the object transformer, are attached using this mechanism (E.g. “The propery *reactance* is possessed by the object *transformer*”).This is a significant difference to the EPRI-CIM model where the properties of a transformer are attached as attributes directly.

In Figure 2, the expression (ABS) indicates that this entity cannot be instantiated directly. It corresponds to an abstract (ABS) class in OO programming. Thus, the entity possession_of_property can only point to one (“1” near the thick line) of the instances of the non-(ABS)-entities below “(ABS)property”. Note that “(ABS) simple_property” is again an abstract entity. As a consequence attributes of possession_of_property can even point to lower levels than simple_property, see Figure 3.

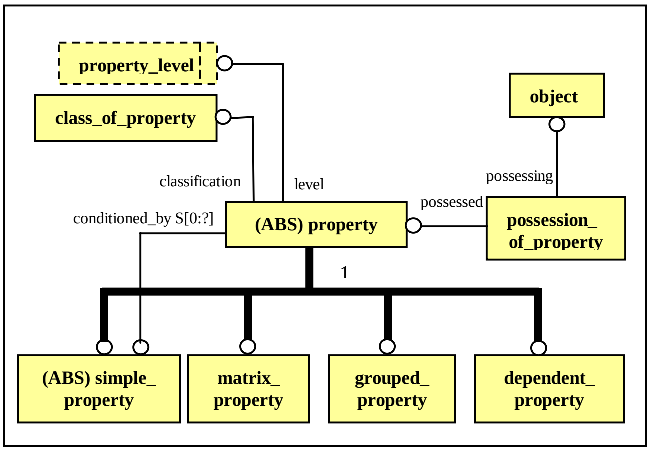

Figure 2. Overview

4.1 Simple properties

Simple properties can be of boolean, string or numeric type. For string properties the language can also be given. In addition, a property representing a timestamp is considered to be a simple property (s. Figure 3). Due to the fact, that all attributes belonging to a timestamp are optional attributes (year, month, day, hour, min, s, ms) the user can adopt the accuracy of information to his requirements.

Numeric properties are further subdivided into real, integer and complex values, where the complex values may be described using the cartesian or the polar representation. A ”unit_of_property” can be assigned to numeric properties, see section 4.5.

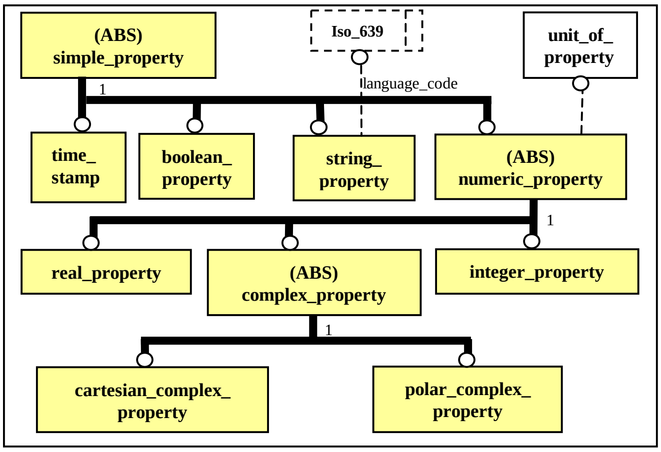

Figure 3.Simple properties

The dashed line in Figure 3 indicates an optional attribute (ISO_639): It can, but need not be given.

4.2 Dependent properties

Dependent properties can be described by means of formulas and data points. Formulas may be specific formulas with predefined parameters or very generic formulas depending on programming language expressions. Data points are represented by means of the entity numeric_space_property_description. (s. Figure 4).

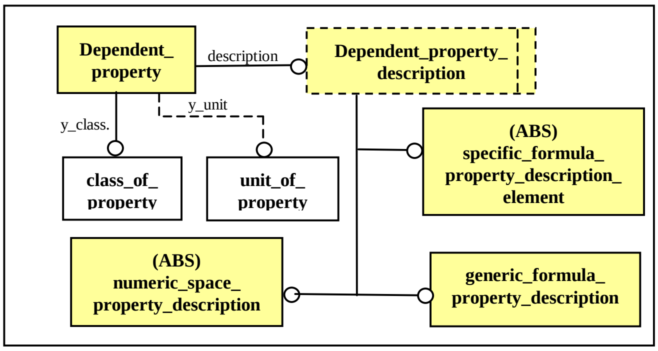

Figure 4. Dependent properties

The description using data points (s. Figure 5) can be two or three dimensional. The corresponding values are given as list. As intervals between single values do not need to be equidistant, attributes can be given to indicate the increment of values. Furthermore different types of interpolation are supported e.g. linear or logarithmic. To be as general as possible, different coordinate systems can be described as well.

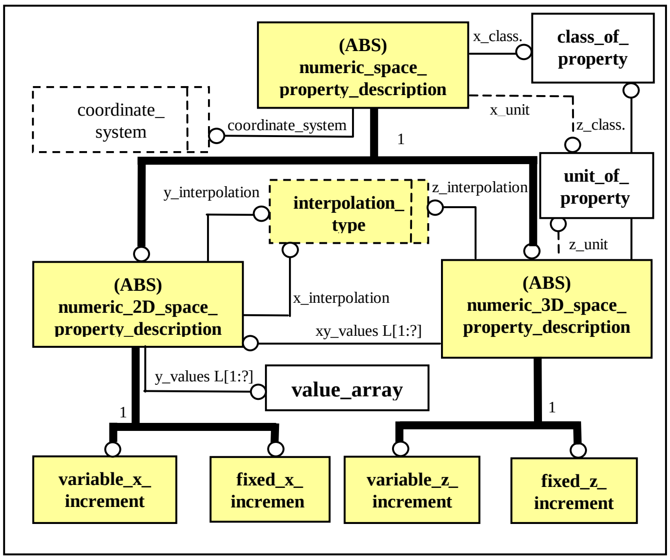

Figure 5. Data point properties

For the representation of values within an interval [x_start, x_end] two basic descriptions of specific formulas (s. Figure 6) are provided :

Polynomial formulas:

$$Y = A + B_1X^{C1} + B_2X^{C2} + \dots + B_nX^{Cn} \qquad (1)$$

Exponential formulas:

$$Y = A + B_1D_1^{E1X} + B_2D_2^{E2X} + \dots + B_nD_n^{EnX} \qquad (2)$$

These descriptions can as well be combined:

$$Y = A + \Sigma B_iX^{Ci} + \Sigma B_jD_j^{EjX} \qquad (3)$$

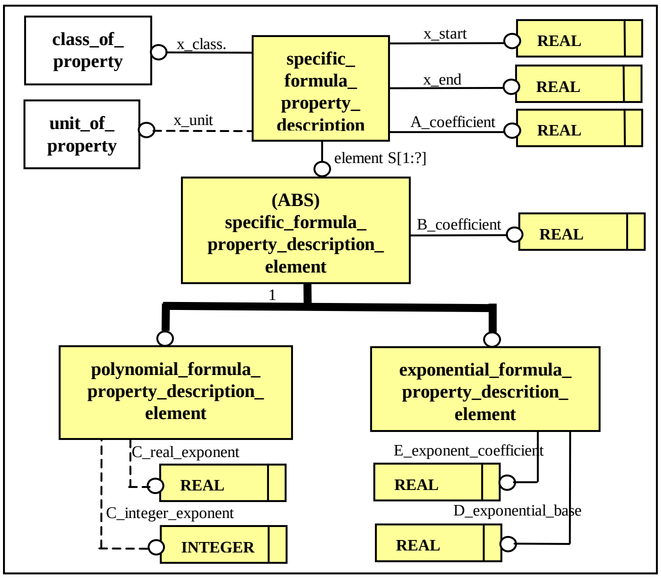

Figure 6. Formula properties

Furthermore a generic description of formulas by a programming language expressions can be given (s. Figure 7), for formulas of the kind:

$$Y = f(X_1, X_2, X_3, \dots, X_n) \qquad (4)$$

(where X_n represents different property types)

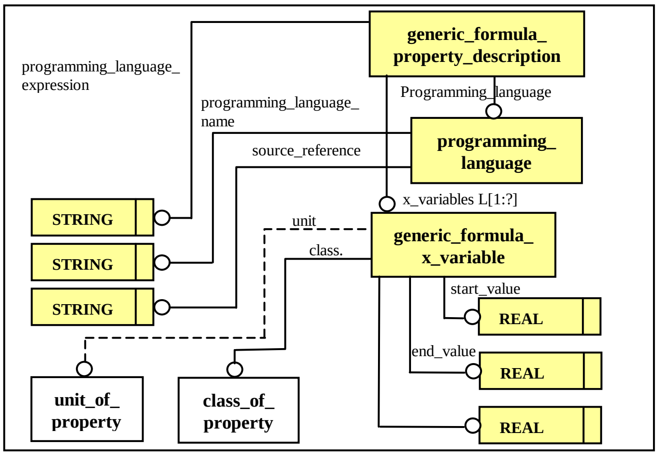

Figure 7. Generic formula property

4.3 Grouped properties

Simple properties may be grouped as arrays, where each component is a list of the same basic value type i.e. boolean, complex etc. Arrays can be further summarised as grouped properties, that may be grouped again (s. Figure 8).

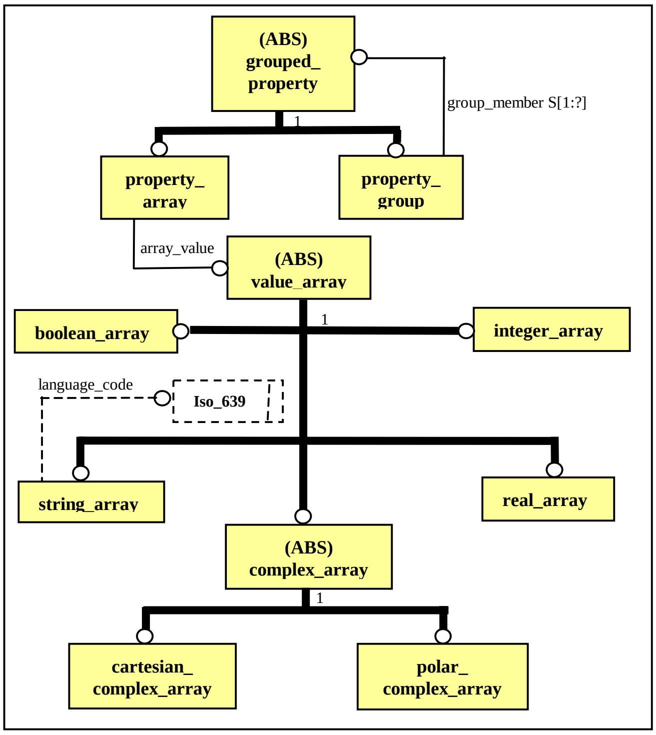

Figure 8. Grouped property

As the ElectroNet project is concerned with the exchange of data, it provides an encoding mechanism to transfer groups of properties, which are exchanged several times, where only the values change, but the meaning of the position within a group stays the same (s. Figure 9).

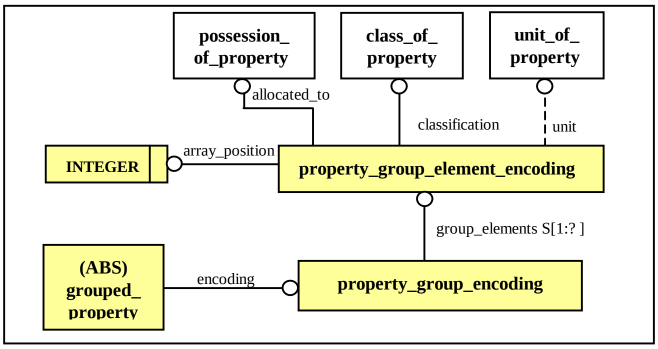

Figure 9. Encoding mechanism

4.4 Matrix properties

The ElectroNet property model recognises a matrix as a specific kind of property. One major reason for this is that it is not possible to deal with an element of a matrix without the knowledge of all the other elements, i.e. the matrix is needed as a property on its own. In order to avoid any inconsistencies matrices are defined as property types but not the elements of the matrices (s. Figure 10). Matrices are described by means of their unit of property, the numbers of rows and columns, their bandwidth and the type of matrix, distinguishing, whether a matrix is a full, upper, lower or diagonal matrix.

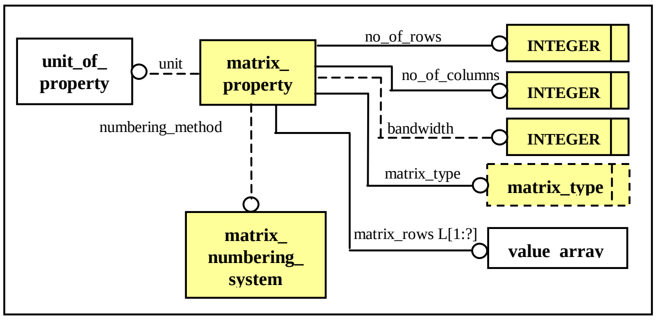

Figure 10. Matrix properties

Elements of the matrix are described as rows, where each row corresponds to a value array. The attachment of individual matrix elements (represented by two sub-indices of the matrix) to objects, is done by means of a matrix numbering system.

4.5 Unit of property

IEC 61360 specifies that for properties which require a unit of measure, the unit shall be the SI unit such as Ω, V[oltage], A[mpere]. In the context of the project, this requirement has not been considered very useful. It should be possible to specify units in terms of the SI unit together with one of the standard prefixes n[ano], μ, m[illi], K[ilo], M[ega], G[iga], T[era] etc. As a consequence a part of the property model allows the specification of types of units by the ”si_prefix” (s. Figure 11).

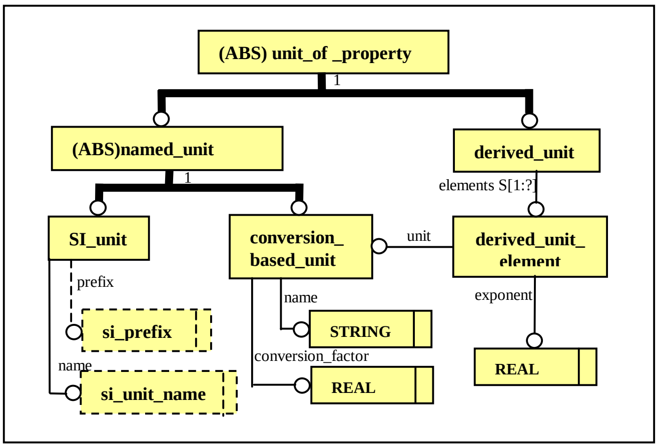

Figure 11. Unit of property

The "unit_of_property" schema handles also the need for exchanging properties in terms of the p.u. (per unit) system. The p.u. can therefore be treated like a unit.

4.6 Property constraints

There exist cases when the instantiated values of a class_of_property (i.e. a DET) are constrained. These constraints represent limitations on the possible values which an instance of a class_of_property can take.

The ElectroNet property model contains a separate schema which allows constraints to be defined. The model also includes rules for checking that instantiated values follow these rules. The constraints that can be defined are such that the instantiated value of a class_of_property shall be either

- higher than a lower limit
- lower than an upper limit
- within a range defined by a lower and an upper limit
- one value out of a set of discrete values.

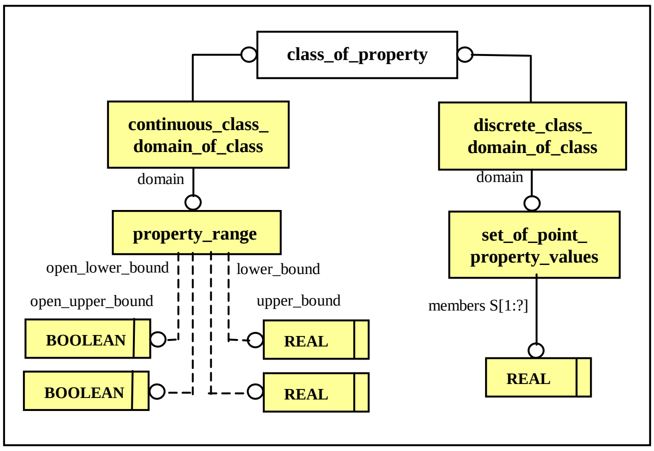

Figure 12. Constraints

An important issue of the constraint modelling is that an instantiated value of a class_of_property can only be checked for constraints violations if the instantiated value is not associated with a unit_of_property. Thus, the unit_of_property is established as part of the definition of the class_of_property.

5. EXAMPLES

The concepts presented in this paper are by definition abstract, “conceptual”, “high-level”. This section intends to clarify the application of the ElectroNet property model by giving a few concrete examples.

5.1 Property information modelling

Dependent property (Figs. 2, 4, 7): Typical dependent properties for a simple AC high voltage transmission line are the various dimension properties of the tower (height, distance between conductors, number of conductor strands in the bundle, etc.) and the resulting model properties for R [Ω], X [Ω], B [1/Ω]. There are well defined functional dependencies (formulas) between these properties. The formulas can be formulated as part of the generic_formula_property_description, see Figure 7.

Grouped property (Figs. 2, 8 and 9): The typical transmission Pi-line-model properties R, X and B can be grouped together to allow a triplet of real values to be exchanged for each instance of this Pi-line-model.

Unit of property (Figure 11): Power transmission occurs at very high levels, e.g. the voltage is in the range of thousands Volt, powers are in the range of hundred-thousands Volt-Amperes. This fact can be used when defining voltages with kV and powers with MVA units.

Property constraints (Figure 12): Many AC line models include the modelling of resistances R. When defining R to be a class of property and associating this class of property to a real number type and the associated constraint of type "≥ 0", one clearly prevents the use of line models with negative resistances. For the case where an equivalent line model should, however, be used, one must create a special "equivalent" line model where no value constraints are set on R.

5.2 Data exchange

A particular object "transmission_line_model" is taken as an example: A single property of 5.2 Ohm (a resistance) is attached to this transmission_line_model. This fact can be modelled as follows using the property information model of this paper:

The value 5.2 is a real number. Thus an instance of a **real_property** (instance #1, see "code" at the end of this subsection) is needed, having the real number 5.2 as one of its attributes. Since **real_property** is a subtype of **numeric_property**, **simple_property** and **property** it also represents a potential instance of these entities. Due to the fact that only real_property is non-abstract, only this entity is instantiated.

The unit 'Ohm' is a SI defined unit, so there is an instance of the entity **SI_unit** (instance #2) also representing the entities **named_unit** and **unit_of_property**. The SI_unit instance has the string 'Ohm' as the attribute **name**.

In order to describe the knowledge that 5.2 Ohm is a "resistance", there is an instance of the entity **class_of_property** (instance #3) defining resistance as a data element type (DET).

The object 'transmission_line' is represented in the model as an instance of the high-level ElectroNet entity **object** (instance #4). This instance #4 possesses the instance of the **real_property** (instance #1) through one instance of the entity **possession_of_property** (instance #5). Finally, the instance of the **real_property** (instance #1) holds the instances of **SI_unit** (instance #2) and **class_of_property** (instance #3) as attributes.

A pseudo ASCII-STEP Physical File (SPF, [13]) representation of this example will look like:

```
#1=real_property(#3, $, $, #2, 5.2);
#2=SI_unit('Ohm', $);
#3=class_of_property('resistance');
#4=object('transmission_line_model');
#5=possession_of_property(#4, #1);
```

The STEP physical file (SPF) is a standardised exchange format. Its use is implied when applying STEP technology.

The "$"-signs in the above five lines are part of the language syntax of this standard SPF-Format. The meaning is not relevant here. The above five lines define in a unique way the fact that the transmission_line has a resistance of 5.2 Ohm. For a computer program, this is, however, not enough: Together with the information model concepts, the computer program knows from these five lines, that 5.2 is a real number, it knows that 'Ohm' is a SI_unit (this is a standard defined outside of the ElectroNet project and is just referenced here), it knows that 'resistance' is a class of a property.

Note that the "words of the language" in these five lines are semantically defined by the concepts presented in this paper. The syntax of this language is defined by STEP [13]. Thus, STEP not only allows the formal description of concepts by Express-G diagrams (or the language EXPRESS, not used in this paper), but also the formalised use of the concepts to exchange data which conform to the information model presented before.

Other aspects of the transmission line can be formulated within the ElectroNet information model: The connectivity, different states and schematic drawings of the line. These information models, however, are outside of the scope of property modelling and thus outside of this paper.

6. CONCLUSION

This paper presents a model for the definition and description of properties enabling the user to exchange properties on basis of a common information model developed in the context of the ESPRIT IV ElectroNet project No. 22297. An information model which allows the exchange of ”property data” represents only one part of the ElectroNet information model which also includes a model for “things in a network and their connectivity”, a model for “states of the network” and a model for “the presentation of electrical functions and equipment” [4, 14]. It has, however, its own existence and represents the core of the whole information model: It allows the exchange of ”real world values” for any objects of concrete power systems. The presented information model provides a very flexible and extensible concept for the description of properties of any network components.

The model allows the creation of new properties in a consistent way. Properties defined in the model should be understood in a clear and unique way due to the fact that the semantics of each aspect of a ”property” are part of the information model. As a consequence, data associated to properties within the world of electrical power systems can be stored and be exchanged in an unambiguous and non-redundant way.

Future work will concentrate on the implementation of the information model in order to demonstrate the real-world-applicability and its performance.

This topical information model is also part of a model presented in IEC TC57 (Power system control and associated communications) and IEC TC3 (Documentation and graphical symbols). Efforts will be made to establish the ElectroNet information model as an international standard.

ACKNOWLEDGEMENT

The work presented in this paper has been performed in the context of the collaborative ElectroNet project (No 22297) under the Esprit Program, funded partly by the E.C. over 3 years and due to finish in September 1999. The Swiss part (ETH Zürich) of this project has been sponsored by the BBW (Bundesamt für Bildung und Wissenschaft), Bern, Switzerland (BBW Project No. 95.0840). The Electro-Net consortium members gratefully acknowledge the support being received for this work.

REFERENCES

[1] ElectroNet Project technical annex ”High Voltage Information Exchange for Planning and Analysis” ESPRIT Project EP 22297, Part II- description of the RTD content, Note No. 98.90/96, edited by Bertin & Cie (July 1996)

[2] S. Massucco, R. Bacher and A. Azarian: ElectroNet: “High Voltage Electrical Network Information Exchange for Planning and Analysis” Proceedings of PDT Days 1997 (April 16, 1997, Sophia Antipolis, France)

[3] T. Dreyer, A. Azarian, R. Bacher, D. Lambrecht and D. Brown: “Information Exchange Between Electric Power Systems: Achievements within the ElectroNet Esprit IV Project”. Proceedings of PDT Days 1998 (25/26 March 1998, Garston, Watford, UK)

[4] David Leal, Inaki Laresgoiti, D. Lambrecht and R. Bacher: “An Open System Approach to Power Systems Information Exchange”; Proceedings of EPSOM'98 (23-25 Sept. 98, Zürich, Switzerland)

[5] L. Baggini, A. Invernizzi, B. Cova, B. Delfino, G.B. Denegri and S. Massucco: “Electrical Network Studies and Data Exchange Requirements and Tools”; Proceedings of EPSOM'98 (23-25 Sept. 98, Zürich, Switzerland)

[6] D. Schenck, P. Wilson, Information modelling: The EXPRESS way, Oxford university press (1994)

[7] IEC 61360: Standard data element types with associated classification scheme for electrical components

[8] AP212: Electrotechnical design and installation, IEC/TC3 – ISO TC184/SC4 JWG9 N23-9

[9] EPRI-CIM IEC Draft - EMSAPI: Energy Management System Application Program Interface, Document 1, Common Information Model, Version 1, December 1996 (ftp://ftp.macro.com/epriapi/api1v01a.pdf)

[10] Draft IEC 61970: Energy Management System Application Program Interface (EMS-API) – Part 301: Common Information Model (CIM), Draft Revision 1, IEC, 1998 (ftp://ftp.macro.com/epriapi)

[11] UCPTE Datenaustausch für Lastflußvorschaurechnung in der UCPTE, June 1996

[12] IEEE common format for exchange of solved load flow data, IEEE Transactions on Power Apparatus and Systems, 1973, pp. 1916-1925

[13] ISO 10303: Standard for the exchange of product model data

[14] ElectroNet-Esprit IV Project No. 22297 public documents internet site: http://www.eus.ee.ethz.ch/electronet/