PUBLICLY AVAILABLE SPECIFICATION

IEC PAS 62424

Pre-Standard

First edition
2005-06

Representation of process control engineering requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools

Reference number
IEC/PAS 62424:2005(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

PUBLICLY AVAILABLE SPECIFICATION

IEC PAS 62424

Pre-Standard

First edition
2005-06

Representation of process control engineering requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XE

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

REPRESENTATION OF PROCESS CONTROL ENGINEERING REQUESTS IN P&I DIAGRAMS AND DATA EXCHANGE BETWEEN P&ID TOOLS AND PCE-CAE TOOLS

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

A PAS is a technical specification not fulfilling the requirements for a standard, but made available to the public.

IEC-PAS 62424 has been processed by IEC technical committee 65: Industrial-process measurement and control.

The text of this PAS is based on the following document:	This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document
Draft PAS	**Report on voting**
65/356/NP	65/361/RVN

Following publication of this PAS, which is a pre-standard publication, the technical committee or subcommittee concerned will transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of three years starting from 2005-06. The validity may be extended for a single three-year period, following which it shall be revised to become another type of normative document or shall be withdrawn.

INTRODUCTION

Efficient process engineering requires highly sophisticated tools for the different needs of the departments involved. These engineering tools are normally specialised in process design (PD), in process control engineering (PCE), etc. Therefore, working interoperability is essential to optimise the engineering process in total. Thus, the definition of a harmonised interface and data management is a core task to ensure a smooth workflow throughout the whole project and to guarantee data consistency in the different tools.

This document defines procedures and specifications for the exchange of PCE relevant data provided by the P&ID module. The requirements for a change management procedure are described. A generally accepted technology for machine information exchange, the Extensible Markup Language (XML) is employed. Hereby, a common basis is given for information integration.

However, a definition is still necessary for uniform semantics. CAEX (**C**omputer **A**ided **E**ngineering e**X**change) as defined in this document is an appropriate data format for this purpose. This concept of data exchange is open for different applications.

The main task of a data exchange is transporting/synchronizing information from the P&ID database to the PCE databases and vice versa. The owner's naming system is the key for a unique identification. For detailed information about representation of PCE loops in P&I diagrams, see Clause 6.

The data exchange system may be a stand-alone, vendor-independent application or a module in an engineering environment. The data between a P&ID tool and a PCE tool and vice versa is exchanged via CAEX.

After the data exchange, there are three places where information about the plant is stored. Both the proprietary databases of the tools considered include private and common information. Both are stored at different places and different divisions that are working on them. Hereby, the intermediate database CAEX only stores common information. In a wider approach, the intermediate database can store both common and private information. This becomes important if a third application is connected to the neutral database. If the intermediate database is used as a temporary data stream only (without storing the information in a file), the information will be lost after processing the data conciliation.

Figure 1 illustrates the information flow for the P&ID and the PCE database reconciliation. The data exchange is carried out via a neutral intermediate CAEX database, not direct from database to database. The intermediate CAEX database can be a file (for file-based data exchange) or a stream (for network-based data exchange). The term "CAEX database“ within this specification has to be understood in this way, it does not denominate a database product such as SQL.

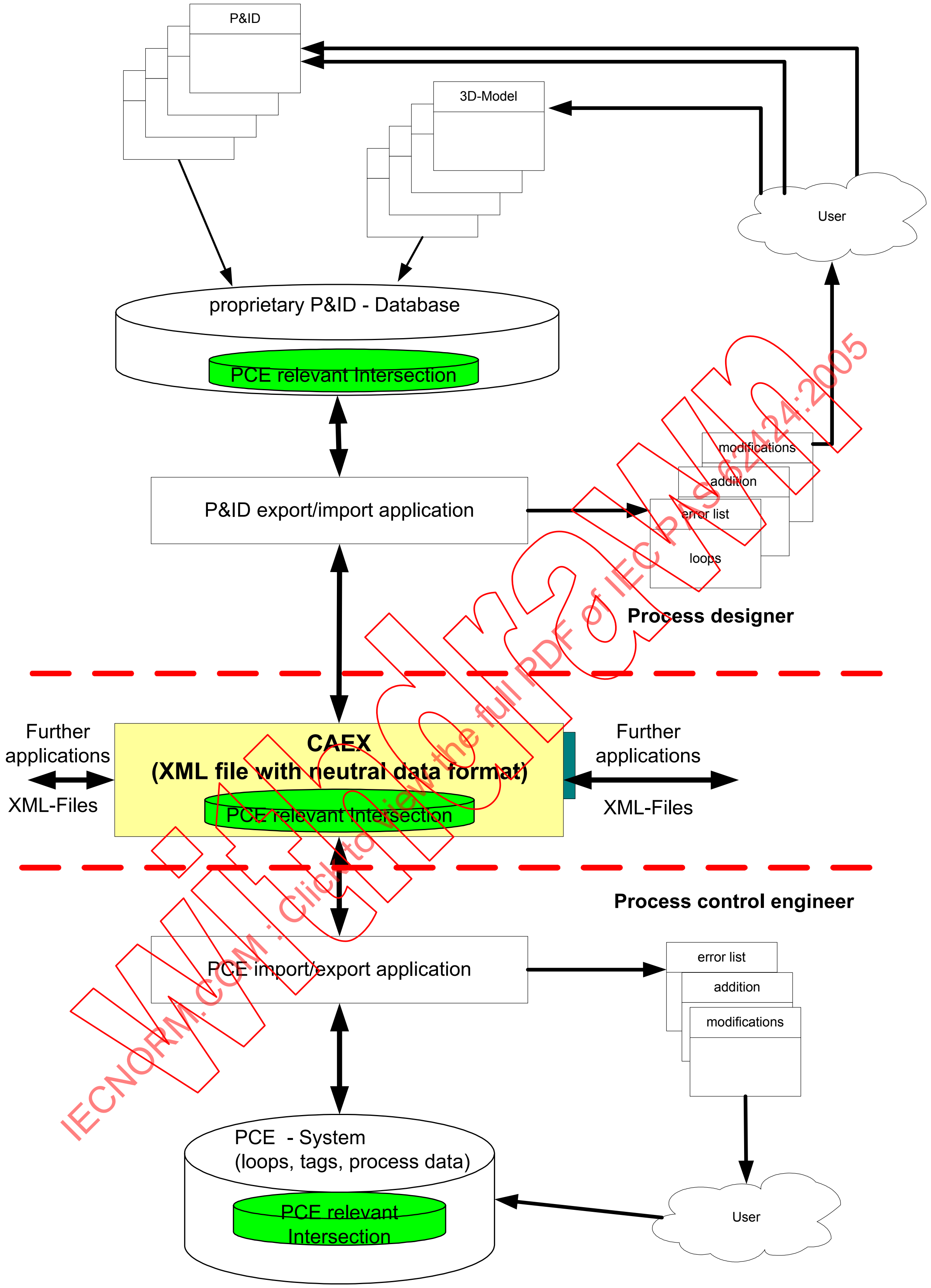

P&ID
3D-Model
User
proprietary P&ID - Database
PCE relevant Intersection
modifications
addition
error list
loops
P&ID export/import application
Process designer
Further applications
XML-Files
CAEX
(XML file with neutral data format)
PCE relevant Intersection
Further applications
XML-Files
Process control engineer
PCE import/export application
error list
addition
modifications
PCE - System
(loops, tags, process data)
PCE relevant Intersection
User

Figure 1 – Information flow of P&ID and PCE tools

REPRESENTATION OF PROCESS CONTROL ENGINEERING REQUESTS IN P&I DIAGRAMS AND DATA EXCHANGE BETWEEN P&ID TOOLS AND PCE-CAE TOOLS

1 Scope

This specification describes how process control engineering requests are represented in a P&I diagram.

It also defines the exchange of process control engineering request relevant data between a process control engineering tool and a P&I tool by means of a data transfer language (called CAEX). These provisions apply to the export/import applications of such tools.

The representation of the PCE functionality in P&I diagrams is defined by a minimum number of rules to clearly indicate their category and processing function, independent from the technique of realization (see Clause 6). The definition of graphical symbols for process equipment (for example vessels, valves, columns, etc.), their implementation and rules for the numbering system are not the scope of this standard. These rules are independent from this standard and can be found, for example, within DIN EN ISO 10628 or ISO 14617.

Clause 7 specifies the data flow between the different tools and the data model CAEX.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60050-826, *International Electrotechnical Vocabulary – Part 826: Electrical installations of buildings*

IEC 61346-1 (all parts), *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designation*

IEC 61511-1 (all parts) , *Functional safety – Safety Instrumented Systems for the process industry sector*

ISO 10628, *Flow diagrams for process plants – General rules*

ISO 14617 (all parts), *Graphical symbols for diagrams*

EN 13480-1, *Metallic industrial piping – Part 1: General*

EN 1594, *Gas supply systems – Pipelines for maximum operating pressure over 16 bar – Functional requirements*

EN 982, *Safety of machinery – Safety requirements for fluid power systems and their components – Hydraulics*

Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 (available at <http://www.w3.org/TR/2004/REC-xml-20040204/>)

3 Terms and definitions

For the purpose of this document, the following terms and definitions, as well as those given in 60050-826, apply

3.1
process control equipment
equipment, having a process control function

3.2
process control function
function to work on process variables, which is composed of basic functions of process control, specific to units of the plant

NOTE In addition to process control functions associated with single sensors and actuators, there can also be process control functions that link input and output variables across several sensors and actuators. For instance, a process control function in the feedback path with the controlled variable as input variable and the manipulated variable as output variable, describes the action path from the sensor via the controller to the final controlling element.

3.3
PCE request
a PCE request describes requirements for process control equipment. Each PCE request is graphically represented by a bubble which collects all information on the functional requirements

3.4
sensor
functional unit that senses the effect of a measured variable at its input and places a corresponding measurement signal at its output

NOTE Examples of sensors are:

a) Thermocouple; b) Foil strain gauge; c) pH electrode.

3.5
Actuator
functional unit that generates from the controller output variable the manipulated variable required to drive the final controlling element

NOTE If the final controlling element is mechanically actuated, it is controlled via an actuating drive. The actuator drives the actuating drive in this case.

EXAMPLE A practical example of an actuator acting directly on the final controlling element is a d.c. drive. The control unit takes the function of an actuator. The final controlling element is formed by the thyristor assembly that delivers a variable d.c. voltage as an output variable. The control unit and the thyristor assembly together form the final controlling equipment.

3.6
CE loop
collection of PCE requests and PCE control functions depicting their functional coherence

3.7
Bubble
symbol used to denote and identify a process function. It contains an identification

3.8
process function
function in a process
[IEC 61512-1]

3.9
PCE control function
function in a PCE control
[IEC 61512-1]

3.10
process category
letter that designates the kind of process control request

4 Abbreviations

Table 1 shows the abbreviations used in this specification.

Table 1 – Abbreviations

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAEX	Computer Aided Engineering eXchange
CCR	Central Control Room
E&I	Electrical and Instrumentation
ERP	Enterprise Resource Planning
GMP	Good Manufacturing Practice
N.A.	Not applicable
PCE	Process Control Engineering
PCS	Process Control System
P&ID	Piping and Instrumentation Diagram
PD	Process design
PU	Package Unit
SIL	Safety Integrety Level
XML	Extensible Markup Language

5 Conformity

To claim conformity to this specification with respect to the graphical representation of PCE requests in P&ID, the requirements of Clause 6 shall be fulfilled.

To claim conformity to this specification with respect to the PCE relevant data exchange, the requirements of Clause 7 and the following requirements shall be fulfilled.

The data exchange shall be performed by a separate or integrated **import/export application** that provides for the data exchange between the related tool and CAEX.

NOTE The goal of the import/export application is to provide for a data reconciliation for the intersection of the source and target databases. It is able to read the proprietary database of the considered tool and to reconcile the data with the neutral CAEX database.

The export/import application shall check, report and provide the intersection data of both databases. The neutral database shall be open for additional applications.

The data import function shall enforce a configurable checking step (for example,. rule-based) during the import process; it shall not allow unguided automatic changes. The configurable checking step shall include functionality for automatic or manual acceptance of data changes, allowing single decisions up to bulk data management.

All changes in the proprietary database and all data inconsistencies discovered shall be reported by the import application. The generation of the report shall be configurable. The import/export application shall assure that the intersection of the different databases holds the same information and that additional division specific data is handled in a consistent way. Data manipulation by a project division is an ongoing process during the whole project and beyond it. Thus, the creation, changing and deletion of data shall be possible during the life-cycle of the plant.

CAEX databases shall be consistent. This requires a consistency check before exporting the data. This procedure shall be followed after a successful data manipulation in a P&ID-tool or PCE-tool in order to bring the new information into the neutral database or vice versa. Before any data changing action is carried out, the user shall be informed and asked for confirmation. The consistency check shall encompass at least the following steps and fulfil the following requirements.

a) Data export from source database to neutral database

   1) Check P&ID and PCE database for at least

      i) duplicate PCE requests or loop designations;

      ii) mandatory fields being filled in;

      iii) correct use of numbering system of the PCE requests;

   Inconsistent data shall not be exported

   2) Generate PCE relevant information.

   3) Check for changed information in comparison with previously stored data in the neutral database.

   4) Renaming of PCE request shall be supported by the export functionality.

   5) Perform data export from proprietary into neutral database

      i) e. g. if the PCE request has been changed, the old PCE request within the neutral database shall be deleted and the new one can be exported from the proprietary database into the neutral database. The old PCE request information may be stored in a backup.

      ii) Other changes shall be performed with the existing object.

   6) Generate reports after each data exchange

      i) new PCE requests list, missing PCE requests list, changed PCE requests list, deleted PCE requests list, problems and error list.

b) Data import from neutral database into the target database

   1) Generate PCE relevant information from neutral database.

   2) Check for changed information by comparing the neutral database with the target database.

   3 Perform data import from the neutral into the proprietary database.

   4 Renaming of PCE request shall be supported by the import functionality.

   5 Generate reports after each data exchange.

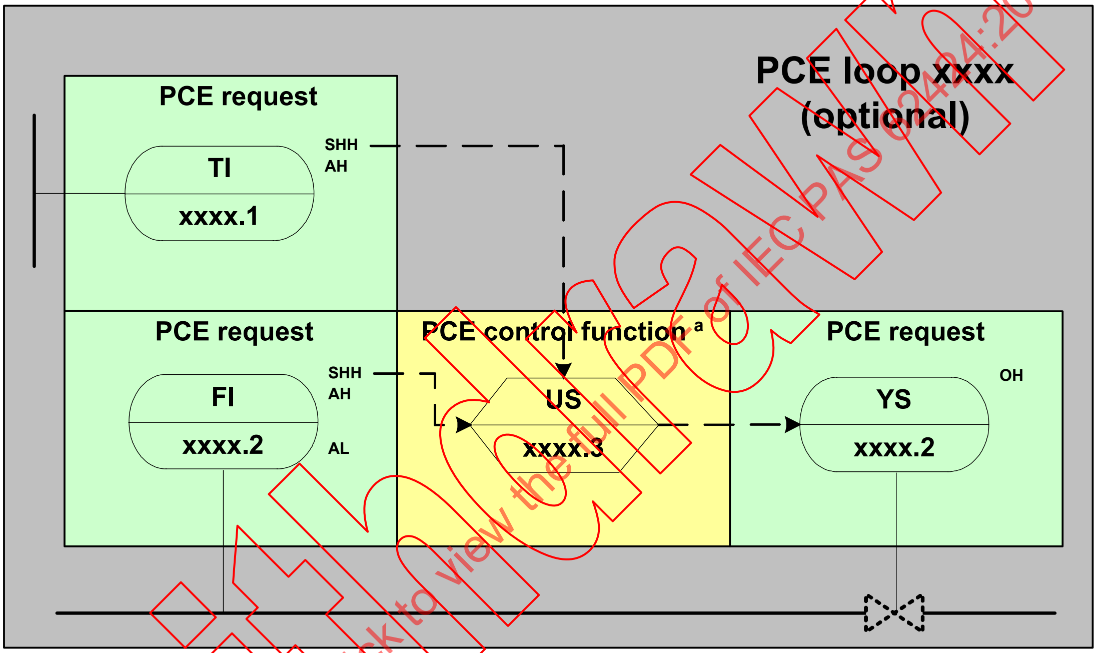
      i) error lists:

      ii) inconsistencies due to imported data can be detected by the target application during the import process and are not considered within this specification.

6 Representation of PCE requests in P&ID

6.1 PCE request and PCE loop

In a P&ID the functional design of a plant is determined. Details of technical equipment are given only if functions are correlated with the design of specific equipment. Consequently the P&ID describes requirements for the design of the process control equipment. Each PCE request shall be illustrated in the P&ID with an individual identification. In order to meet the requirements of data handling, the same identification shall not be used for different PCE requests. Functional coherence may be depicted by collecting the individual PCE requests in a PCE loop. A PCE loop does not have a graphical representation but is embedded into a numbering system. Depending on the engineering strategy, a PCE loop thereby consists of at least one, but can also combine several, PCE requests. If PCE loops are used, these shall be represented in the identification of all PCE requests concerned. An example of this concept is given in Figure 2.



[a] The PCE control function used in Figure 2 is defined in 6.3.10.

Figure 2 – Organisation of PCE requests

6.2 Objectives and principles

This subclause defines how to represent the process control engineering functionality in P&ID. Technical details of the equipment used shall not be depicted in general. This is due to the goal to ensure a smooth engineering workflow by separating process and instrumentation design.

Therefore, it is necessary to clarify

a) the PCE category and function;

b) the numbering system for PCE requests in P&ID;

c) the graphical representation of PCE requests in P&ID;

d) the type of functional connection between the PCE requests: the control functions;

e) the graphical representation of signals in P&ID.

Detailed information on complex control functions shall not be part of the P&ID. Therefore, additional documentation shall be prepared to define the required functionality. A control function shall also be individually identified and may be represented on the P&ID.

6.3 Requirements for the identification and representation of PCE requests

6.3.1 General

Each PCE request shall be graphically represented by a bubble, which collects all information about the functional requirements. Three data fields inside and 10 data fields outside the bubble are defined to hold all the information about a PCE request (see Figure 3). For detailed information, see 6.3.3-6.3.9.

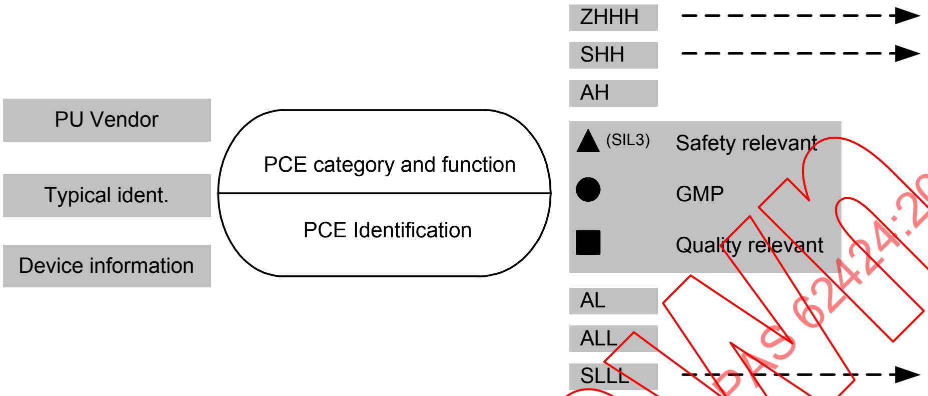


Figure 3 – General representation of a PCE-request in P&ID

As stated before, only the PCE functionality shall be depicted on P&ID, not the PCE implementation. In exceptional cases however, there might be constellations where the presentation of detailed realization information is inevitable. For example, in the case of a multi sensor element which means an instrument that produces measurements for different categories, every category shall be represented by its own bubble. The bubbles are stacked up, as shown in Figure 4.

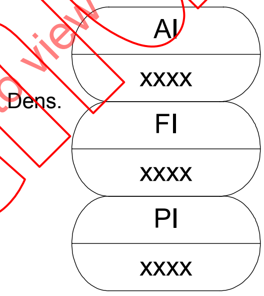


Figure 4 – Multi sensor element

In all cases where the PCE request is connected to the equipment or pipe this shall be shown by a full line, connecting the bubble with the equipment or pipe.

6.3.2 Types of lines

Signal lines are used to illustrate the functional relationship between PCE requests. A signal line shall be depicted as a dashed line with an arrow to indicate the information flow. The source of the information flow shall be a bubble of a PCE control function or request or a switching action consisting of the six fields right outside the bubble. The sink of the information flow shall be a bubble of a PCE request or a control function.

Process connections shall be depicted by a solid line without a direction. Multisensor instruments with only one process connection shall have an extra bubble for each category and only one process connection.

6.3.3 Displaying the location of the operator interface

Each PCE request is graphically represented by a bubble. This standard distinguishes the location of the operator interface between local, the local control panel and a central control room. The location does not reflect any realization in systems.

A local interface shall be represented by a plain bubble as shown in Figure 5. It could be a pressure gauge, for example.

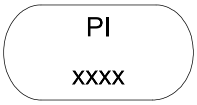


Figure 5 – Local interface

Operator action/information on a local control panel shall be represented as shown in Figure 6.

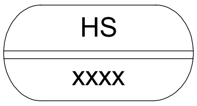


Figure 6 – Manually operated switch in local control panel

Remote requests operated in a central control room shall be represented as shown in Figure 7.

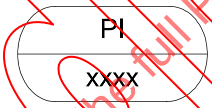


Figure 7 – Pressure indication in a central control room

6.3.4 PCE categories and processing functions

6.3.4.1 Indication of PCE categories and processing functions

The upper part of the bubble shall show the information of the PCE category and its PCE processing function. Each bubble shall have at least one PCE category and one PCE processing function. As an exception for the processing function, see 6.3.4.3.

6.3.4.2 PCE categories

The first character shall be selected according to the measured or manipulated variable. Table 2 shows the list of PCE categories used.

Table 2 – PCE categories

Letter	PCE category
A	Analysis
B	1)
C	1)
D	1)
E	Electrical value (detail definition by user)
F	Flow
G	Distance, length, position
H	Manual and manually initiated operation
I	1)
J	1)
K	Time-based function
L	Level
M	1)
N	Actuation setting (motor)
O	1)
P	Pressure
Q	Quantity or counter
R	Radiation
S	Speed or frequency
T	Temperature
U	N.A. (see 6.3.10)
V	Vibration
W	Weight, mass, force
X	2)
Y	Actuation setting (valve)
Z	1)

[1] These letters can be defined by users.

[2] The unclassified letter X is intended to cover unlisted meanings that will be used only once or used to a limited extent. If used, the letter may have any number of meanings as a PCE category and any number of meanings as a PCE function.

6.3.4.3 PCE processing functions

Starting with the second character, the successive letters in the upper part of the bubble shall represent the processing function of the PCE request. The letters given in Table 3 shall be used to indicate the processing function of a PCE request.

Table 3 – PCE processing function

Letter	Processing function
A	Alarm, Message
B	Restriction
C	Control
D	Difference
E	N.A .
F	Ratio
G	N.A.
H	High limit, on, opened
I	Indication of analog values
J	N.A.
K	N.A.
L	Low limit, off, closed
M	N.A.
N	N.A.
O	Local or DCS status indication of binary signals
P	N.A.
Q	Integrating or counting
R	Recorded value
S	Binary control function or switching function (not safety relevant)
T	N.A.
U	N.A.
V	N.A.
W	N.A.
X	[2]
Y	Computing function
Z [1]	Binary control function or switching function (safety relevant)

[1] The triangle may also be used to indicate in a redundant way that the processing function is safety relevant (see Figure 3).

[2] The unclassified letter X is intended to cover unlisted meanings that will be used only once or used to a limited extent. If used, the letter may have any number of meanings as a PCE category and any number of meanings as a PCE function.

The letters I and R refer to the result of the preceding processing function, for example FIQI means the indication of a flow and its quantity.

To avoid redundant information, the PCE processing functions A, H, L, O, S and Z shall only be used outside the bubble. In this case, it is possible that the category may be depicted without a process control function.

The processing control functions shall be used in the sequence given in Table 4.

Table 4 – Sequence combinations

Category		**1**	**2**	**3**	**4**
See Table 3	**1**	F	D	Y	C
	2	B	Q	X	

NOTE The table hierarchy should be from left to right and per column top down.

6.3.4.4 PCE processing functions for actuators

The PCE processing functions can be used for actuators in the same way as for sensors. Some examples are shown in Table 5.

Table 5 – PCE processing functions for actuators

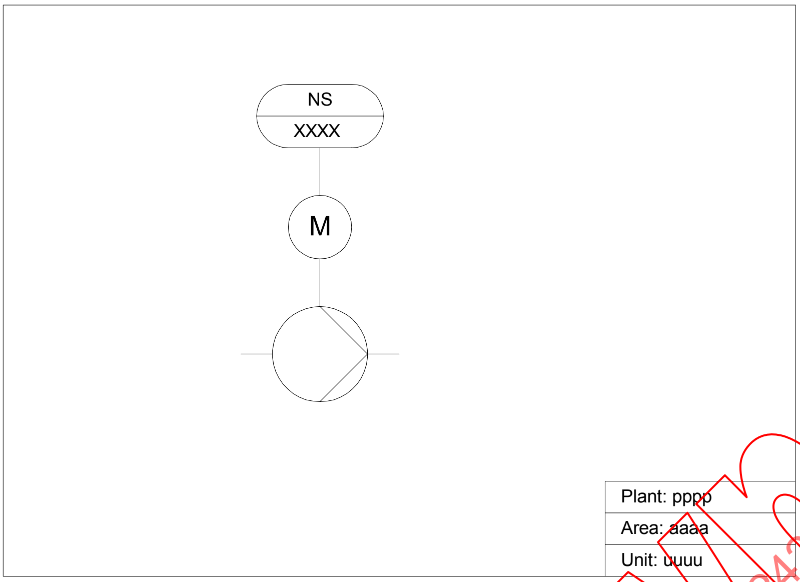
Letter	**Processing Function**
YS	On/off valve
YC	Control valve
YCS	Control valve with on/off function
YZ	On/off valve (safety relevant)
YIC	Control valve continuous position indication
NS	On/off motor
NC	Control motor

The letters C, S, Z may be bound by parantheses in order to indicate that they describe a property of the actuator and not an additional processing function.

The valve operating mechanism, for example electric, pneumatic or hydraulic, is not specified by its presentation in the P&ID.

6.3.5 PCE request numbering system

A general numbering system (see IEC 61346) shall be used in order to identify the PCE request unambiguously. This identification shall be independent of the PCE processing function of the PCE request and depicted in the lower part of the bubble. Preceding identification levels (for example, site, plant, unit, area) can be omitted in the bubble if the uniqueness of the request within the context of the P&ID is guaranteed (see Figure 8). If PCE requests are combined in a PCE loop, their identification shall have separate levels for the loop and the request.



NOTE In the bubble of the depicted request only the last level of the identification system is shown. The plant, area, unit information can be taken from the lower left corner. Thus, the complete identification of the request is: pppp-aaaa-uuuu-xxxx

Figure 8 – Example for a PCE request identification

6.3.6 PU-vendor and typical identification

Above the horizontal line, but outside the bubble on its left side, PU-vendor information may be given as shown in Figure 9. This field may also be used to show other-project specific information.

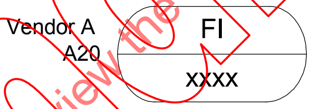


Figure 9 – Example: Flow measurement delivered by vendor A specified by typical A20

To support automatic loop, request and tag generation with the PCE CAE tool, requests, especially motor requests should be indicated by a "typical" number on the centre left side, outside the bubble. These "typicals" are fixed by the project team and are used to determine the composition of the request, for example how the motor drive should be switched (with start-stop only, with start-stop and running indication, with current measurement, etc.).

6.3.7 Device information

Device information can be indicated in the lower zone outside the bubble on the left side (see Figure 10).

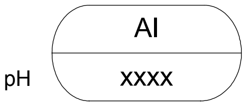


Figure 10 – Example: pH-measurement

6.3.8 Alarming, switching and indicating

The characters H and L as PCE processing functions, indicating the high or low limit, shall be used in combination with A, O, S or Z only if an automatic action (S or Z), an operator action (A) or an indication (O) is activated when the limits are reached. In each level (for example H, HH, HHH) it shall be possible to combine alarming and switching function, for example AS or AZ. These functions shall always be indicated outside the bubble, as shown in Figure 11. Up to three levels for high and also for low alarm/switching/indication shall be possible. If necessary the safety integrity level (SIL) according to IEC 61511 may also be indicated in parantheses.

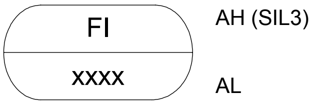


Figure 11 – Example: Flow measurement with safety relevant high alarm and a low alarm

The representation shall be: <processing function><alarm level>, whereas the order of the processing function shall be O, A, S, Z.

The Z may be used to mark an alarm being part of a safety function. If there is not only an alarming (A) but also a switching function (S), the corresponding limits may easily be documented by a controlling function U (see 6.3.10). On the other hand, the limits of the switching function may be indicated on the P&ID. If so, it shall be in an unambiguous way and shall be connected with the controlling function or the actuator starting with the SH, SHH, SL or SLL symbols as shown in Figure 12.

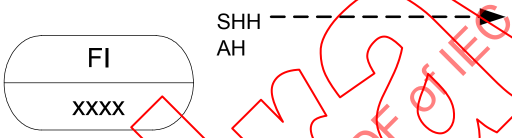


Figure 12 – Flow measurement with high alarm and a high-high switching function

The combination of Figures 11 and 12 is possible as shown in Figure 13.

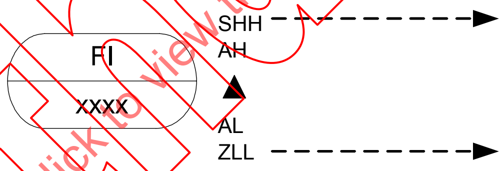


Figure 13 – Flow measurement with a high-high switch limit, a high alarm, a low alarm and a low-low switch limit for a safety function

6.3.9 Safety-relevant, GMP and quality-relevant PCE requests

Outside the bubble, a circle symbol should be used as an indication for GMP relevant sensors or actuators and a square for the indication of a quality-relevant PCE request. A triangle should be used for a safety function (see Figure 14).

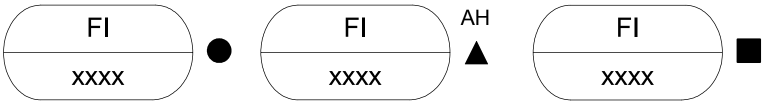


Figure 14 – A GMP relevant, a safety relevant and a quality relevant flow measurement

These symbols shall be placed as closely as possible to the bubbles.

6.3.10 PCE control functions

PCE control functions essentially contain the functional relationship between sensors and actuators. These control functions are the “building stones”, the elements of the entire process-functionality. Mostly they are technically achieved via control system configuration. In simple configurations, for example one sensor and one actuator, where the relation is unambiguously represented in the P&ID, the PCE control function may be omitted.

The symbol for the PCE control function is the hexagon. This hexagon (see Figure 15), symbolises the control functionality, which has one or more sensors as inputs, and one or more actuators as outputs. The operators’ inaccessible PCE control functions may be depicted by using the same symbol but with a dashed h orizontal line.

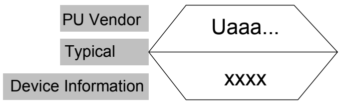


Figure 15 – Control function

The hexagon symbol shall be connected with dashed lines and arrows to the various bubbles that represent the relevant PCE requests (see examples). The arrows indicate the direction of the information (Sensor to U and U to actuator). As for PCE requests, information about the PU vendor and a typical identification may be given on the left side (see 6.3.6).

The PCE control functions shall be identified separately. A general identification system shall be used in order to identify the PCE control function unambiguously. This identification shall be independent of the PCE processing function of the PCE control function and depicted in the lower part of the hexagon. Preceding identification levels (for example site, plant, unit, area) may be omitted in the hexagon if the uniqueness of the control function within the context of the P&ID is guaranteed (see 6.3.5). If control functions are integrated in a PCE loop, their identification shall have different levels for the loop and control function.

The detailed and complete function of the U shall be documented in a separate document, entitled with the U identification.

The upper part of the hexagon symbol shall contain U*aaa*, where *a* is one or more of the processing functions A, C, D, F, Q, S, Y or Z (see Table 3).

It is possible, for example, that a US has a partial UZ character. In that case the U shall become the designation USZ. Every USZ shall have at least one sensor and one actuator which is safety relevant; this means that at least one sensor and one actuator connected to a USZ has the Z as a processing function.

7 Neutral data exchange of PCE relevant P&ID information

7.1 Objectives

P&I drawings include a variety of information relevant for process control engineering purposes. Clause 6 defines as to how basic information concerning PCE requests and their process-relevant functionality shall be represented in a P&I drawing. The specification given concerns primarily the graphical notation, but of course, this establishes structural and semantic fixings too. In this clause these structural and semantic fixings shall be mapped to a semi-formal form. To do this, the CAEX-system-description-language (see Annex A) is used. For this language, an XML-representation is given in Annex C, which allows an open exchange of the modelled data between the P&I drawing system and the PCE systems.

7.2 Meaning of P&I drawing elements

P&I drawings show a plant (or a part of it) in its function as a physical framework. Some aspects are the material flow through vessels and pipes, physical actuations (pumps, stirrers, electrical heating), the coupling between the physical and the control world (PCE requests), and the main dependencies between the control functions.

P&I diagrams show functional requirements (roles) and not the assembly of equipment. A shown pump symbolises not the equipment "pump" but the requirement that at this place a "pumping functionality" is needed. Additional attribute-requirements concerning this pumping functionality like "flow rate", "inlet pressure" and so on can be added.

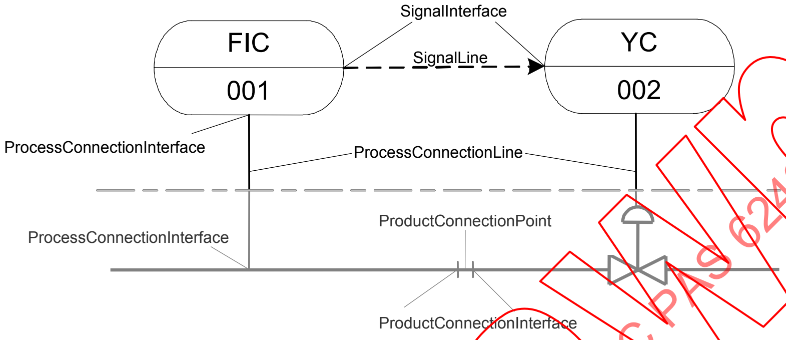


Figure 16 – P&I elements and associations

NOTE Relevant items are shown in dark lines

P&I diagrams show graphically the functional associations between the elements. In the example given in Figure 16 four main classes of associations are shown.

a) Signal association

Symbolizing the requirement that the functionality of one PCE request is influenced by the functionality of another PCE request. The signal association is notated as declared in Clause 6 by a dashed line, the so-called "SignalLine". The SignalLine only symbolizes the functional influence and not the electrical wiring.

b) Process connection association

Symbolizing the coupling point at which information from the control world is transformed to a physical effect and vice versa. A process connection association always links one PCE request with one process plant element. The process connection association is notated as declared in clause 6 by a plain line, the so-called "ProcessConnectionLine". The ProcessConnectionLine only symbolizes the functional coupling and not an actuation element. The length of the line is without any meaning.

c) Product association

Symbolizes the coupling of two pieces of equipment with the possibility of material transfer between them(pipe-pipe, pipe-vessel). The properties of this kind of association are not subject of this document.

d) Mechanical association

Symbolizes mechanical coupling within actuation elements (drive-valve, motor-pump).The properties of this kind of association are not the subject of this document.

7.3 PCE relevant information of P&ID tools

Besides general structural and functional information, P&ID tools handle a variety of information, which are of direct interest to the basic process control engineering.

a) Control relevant information

PCE requests, process connections, signal lines with all their attributes and interfaces described in clause 6 comprises the process-relevant information needed for the process control engineering.

b) Additional information

In many cases, the P&ID tools support additional process-relevant or technology-relevant functional requirements concerning the process connections. Examples are maximum pressure, pipe diameters, information concerning the medium and so on. Some of this information can be important for the process control engineering system too. Clause 8 gives a list of relevant additional parameters.

7.4 Formal description of PCE relevant information of P&ID tools

7.4.1 General

NOTE The P&I diagram is the most important interface between process engineering and process control engineering. It is of fundamental interest to standardize not only the graphical notation of the PCE relevant information but an exchange format too, which supports an open information flow from the P&ID tools to the PCE tools and vice versa.

The PCE data model for PCE relevant information as described in Clause 6 is shown in Figure 17.

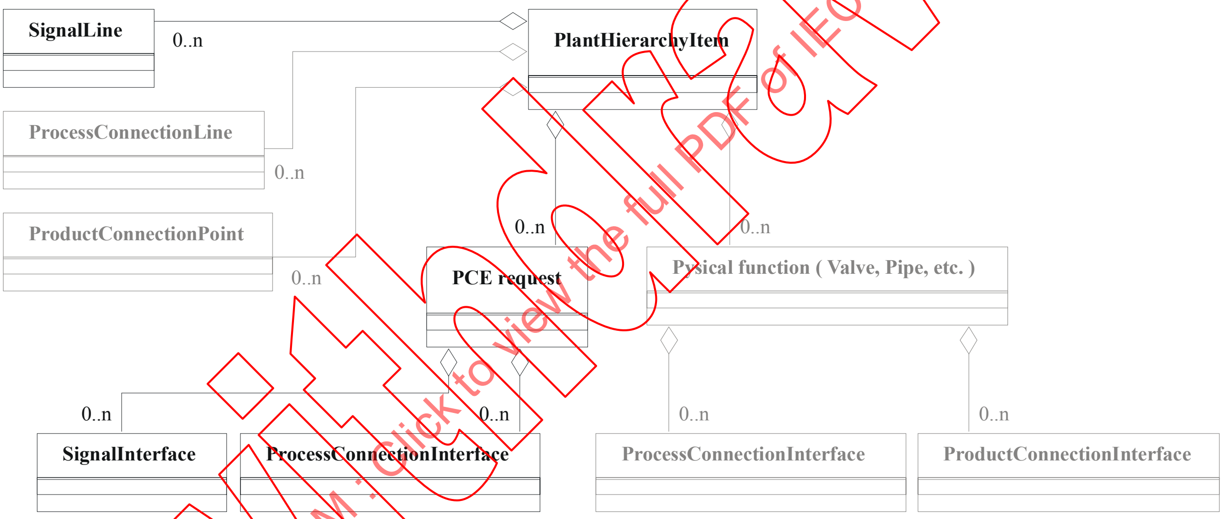


Figure 17 – Process data model

NOTE PCE relevant items are shown in dark lines

The specifications in Clause 6 provide that

PlantHierarchyItems are defined as composite objects that may contain further Plant-HierarchyItems (this allows for the the creation of a hierarchical plant design).

- Each PlantHierarchyItem can contain PCE requests and/or SignalLines. Furthermore, physical functions like valves and pipes may be presented here, if desired, including their interfaces, but they are not within the scope of this document.
- Each PCE request contains 0...n ProcessConnectionInterfaces.
- Each PlantHierarchyItem, PCE request, Physical Function, SignalLine, ProcessConnectionInterface and SignalInterface can have a set of attributes.
- Control functions can be handled in the same way as PCE requests but does not include ProcessConnectionInterfaces.

7.4.2 Using the CAEX system description language

The CAEX system description language supports an exchange of CAE data by an XML file. It is schema-based. The semantic of the CAEX System description language allows for the exchange of instance data (plant data), type data (class data) and complete libraries as well. It comprises a means to support the change management process.

The schema model of the CAEX system description language is given in Annex A.

7.4.3 CAEX System description language mapping

Each PCE request is part of one and only one PlantHierarchyItem (which may of course be itself part of one and only one higher PlantHierarchyItem).

7.4.4 Basic mappings

The PlantHierarchyItem is represented by a CAEX-SystemUnitClass.

Each PCE request, which is part of this PlantHierarchyItem, is represented in CAEX by a nested role CAEX RoleObject within the SystemUnitClass.

Each PCE request possesses either a SignalInterface (Source) "main" or a ProcessConnectionInterface "ActuatorSource" with respect to the signal output of its processing function. A PCE request without any interface makes no sense.

Each defined additional switching function (SH, SHH,..,SL,..,ZH,..) implements an additional SignalInterface (source) within the PCE request.

Each ending SignalLine implements an additional SignalInterface (sink) within the target PCE request.

Each SignalLine between two PCE requests of the same PlantHierarchieItem implements an additional Link-Object, linking the corresponding interfaces of the two PCE requests within the SystemUnitClass.

Each defined alarming function (AH, A, ALL..) implements an additional AlarmInterface (source) within the PCE request.

Each PCE request shall have the following attributes (mandatory):

–	PCE category	(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z) (see Table 3)
–	Location	(Local, Local Panel, Central)

Each PCE request may have one or more of the following attributes (optional):

–	PU vendor	(string)
–	Typical identification	(string)
–	Device information	(string)
–	C processing function	Boolean
–	F processing function	Boolean
–	D processing function	Boolean
–	Y processing function	Boolean
–	B processing function	Boolean
–	Q processing function	Boolean
–	X processing function	Boolean
–	GMP relevant	Boolean
–	Safety relevant	(string)
–	Quality relevant	Boolean

Each attribute specified for a certain PCE request has to be included in the attribute list of the corresponding CAEX-Role-Object in the given order.

The symbol for a PCE request – bubble or hexagon – carries no additional information and is not mapped to the CAEX-Model.

The local name of the PCE request within the PlantHierarchyItem shall correspond to the RoleName within the SystemUnitClass.

Process connections represent associations between PCE requests and ProcessElements. They are outside the scope of PCE and are not mapped to the CAEX model within this standard.

Each end of a process connection at a PCE request implements an additional ProcessConnectionInterface within this PCE request.

All additional information given by the P&ID tool with respect to this process connection shall be mapped to attributes of the corresponding ProcessConnectionInterface.

If defined, the attributes listed in Clause 8 shall be the first in the given order.

Each defined indication function (I, O, OH, ….) implements an additional IndicationInterface.

NOTE The function OSH creates an IndicationInterface and a SignalInterface as well.

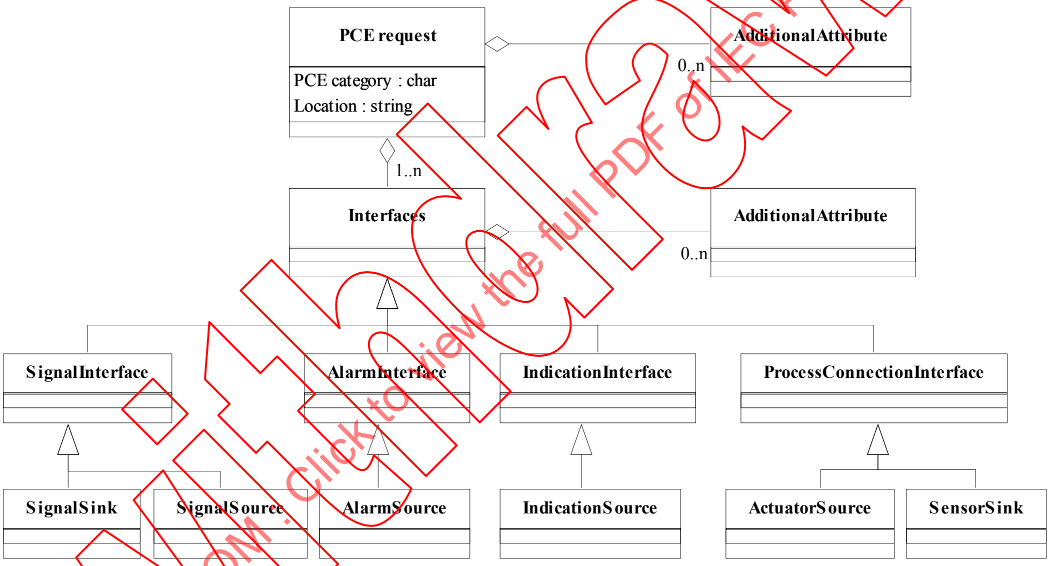


Figure 18 – PCE request data model

7.4.5 Advanced mapping

This subclause gives information only.

7.4.5.1 Connections between elements belonging to different PlantHierarchyItems

Figure 19 gives an example in which a SignalLine couples a PCE request of PlantSection A with a PCE request of PlantSection B. In this case, the plant sections themselves get external signal interfaces. PlantSection B has an external SignalSource and PlantSection A an external SignalSink.

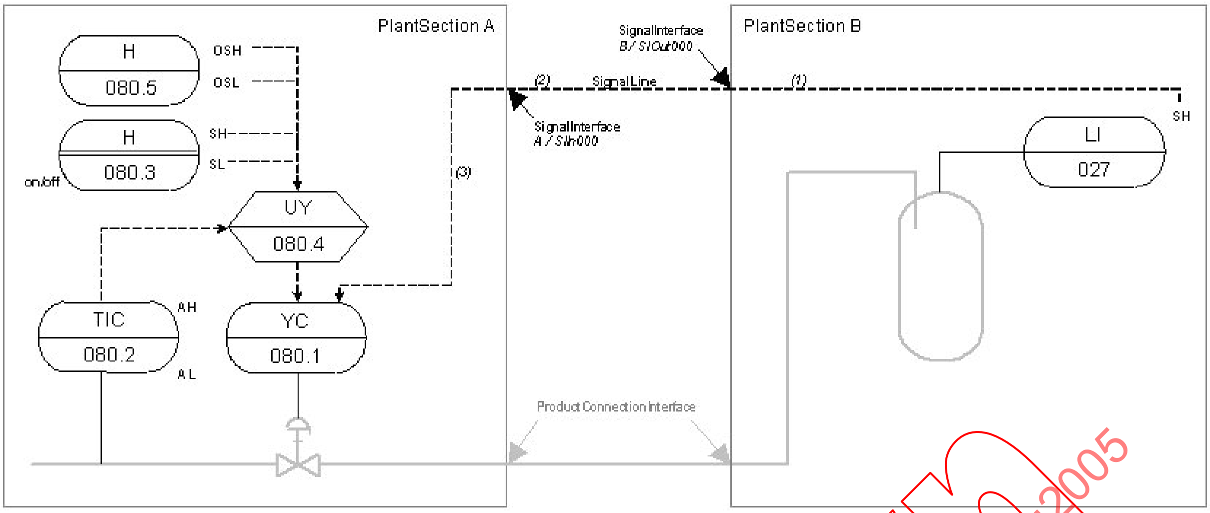


Figure 19 – Example of two parts of a plant and their connections

The SignalLine in this case is mapped to three objects:

a) a link which is part of PlantSection B, connecting B/027/SH with B/SIOut000;

b) an external link which is part of the external plant, connecting B/SIOut000 with A/SIIn000;

c) a link which is part of PlantSection A, connecting A/SIIn000 with A/080.1/In001.

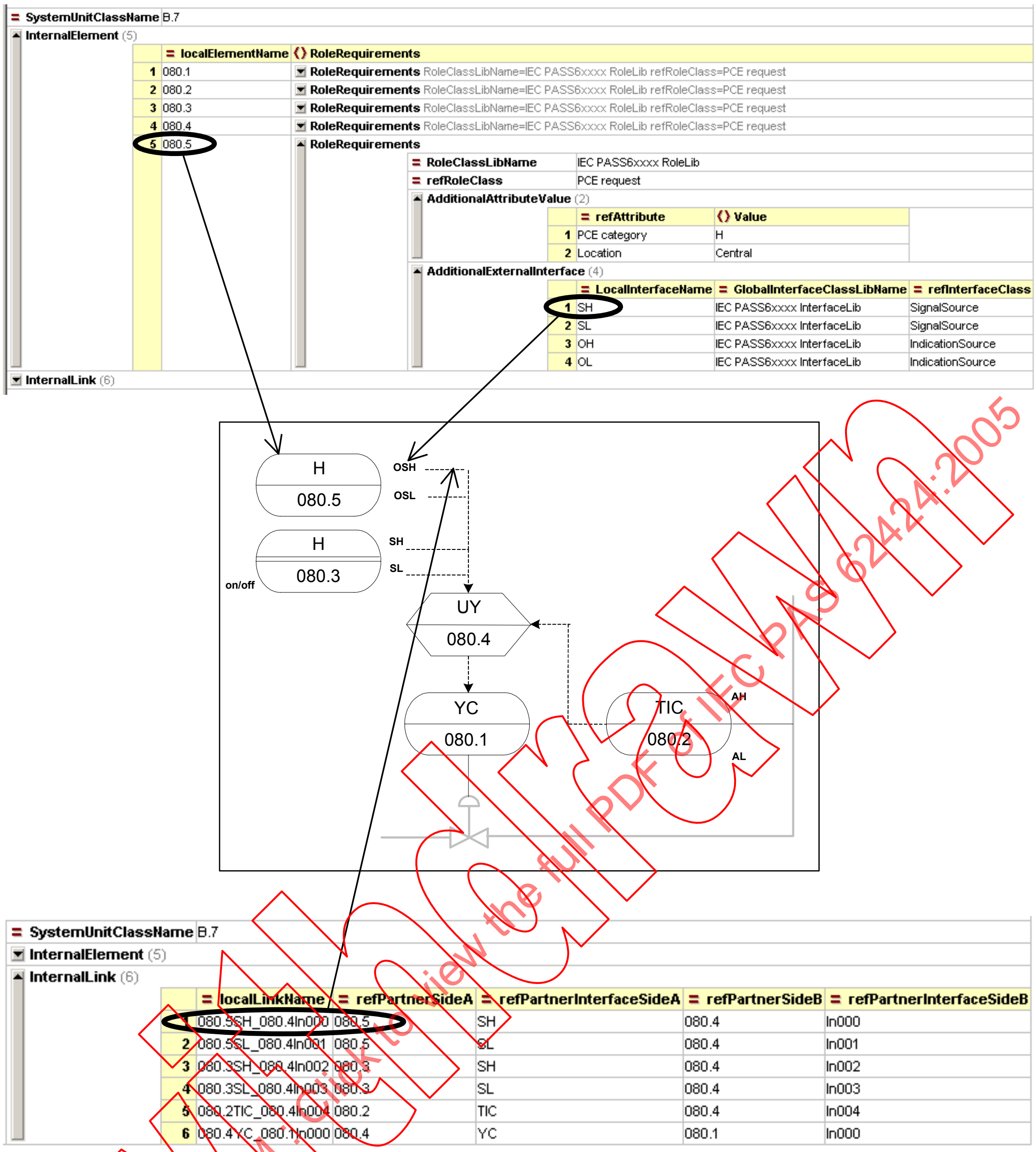
7.4.5.2 PCE loops

PCE loops are identified by naming conventions. PCE loops will not be mapped to structural elements.

The consuming application shall know the special meaning of the naming convention to be able to identify PCE loops.

7.4.6 Example

Figure 19 is an example for the CAEX mappings given in Annex B (see Clause B.7). For example, the PCE request 080.5 is mapped as internal element under the system unit class. This means the PCE request is an element of a plant. It contains the required attributes and the external interfaces, which are linked with the external interfaces of the other PCE requests or technical units. The PCE request may be extended with optional attributes.



SystemUnitClassName B.7

InternalElement (5)

	localElementName	RoleRequirements
1	080.1	RoleRequirements RoleClassLibName=IEC PASS6xxxx RoleLib refRoleClass=PCE request
2	080.2	RoleRequirements RoleClassLibName=IEC PASS6xxxx RoleLib refRoleClass=PCE request
3	080.3	RoleRequirements RoleClassLibName=IEC PASS6xxxx RoleLib refRoleClass=PCE request
4	080.4	RoleRequirements RoleClassLibName=IEC PASS6xxxx RoleLib refRoleClass=PCE request
5	080.5	RoleRequirements

RoleClassLibName IEC PASS6xxxx RoleLib

refRoleClass PCE request

AdditionalAttributeValue (2)

	refAttribute	Value
1	PCE category	H
2	Location	Central

AdditionalExternalInterface (4)

	LocalInterfaceName	GlobalInterfaceClassLibName	refInterfaceClass
1	SH	IEC PASS6xxxx InterfaceLib	SignalSource
2	SL	IEC PASS6xxxx InterfaceLib	SignalSource
3	OH	IEC PASS6xxxx InterfaceLib	IndicationSource
4	OL	IEC PASS6xxxx InterfaceLib	IndicationSource

InternalLink (6)

SystemUnitClassName B.7

InternalElement (5)

InternalLink (6)

	localLinkName	refPartnerSideA	refPartnerInterfaceSideA	refPartnerSideB	refPartnerInterfaceSideB
1	080.5SH_080.4In000	080.5	SH	080.4	In000
2	080.5SL_080.4In001	080.5	SL	080.4	In001
3	080.3SH_080.4In002	080.3	SH	080.4	In002
4	080.3SL_080.4In003	080.3	SL	080.4	In003
5	080.2TIC_080.4In004	080.2	TIC	080.4	In004
6	080.4YC_080.1In000	080.4	YC	080.1	In000

Figure 20 – Basic mappings

In the following, the full XML-text is shown for this example.

```
<CAEXFile FileName="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\CAEX-Projekt\Dke 941\CAEX_ClassModel_V_1.11.xsd">
    <SystemUnitClassLib GlobalSystemUnitLibName="AnnexLib">
        <SystemUnitClass SystemUnitClassName="B.7">
            <InternalElement localElementName="080.1">
                <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib" refRoleClass="PCE request">
                    <AdditionalAttribute LocalAttributeName="C processing function">
                        <DefaultValue>true</DefaultValue>
                    </AdditionalAttribute>
                    <AdditionalAttributeValue refAttribute="PCE category">
                        <Value>Y</Value>
                    </AdditionalAttributeValue>
                    <AdditionalAttributeValue refAttribute="Location">
                        <Value>Central</Value>
                    </AdditionalAttributeValue>
                    <AdditionalExternalInterface LocalInterfaceName="In000" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
                    <AdditionalExternalInterface LocalInterfaceName="Y" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="ActuatorSource"/>
                </RoleRequirements>
            </InternalElement>
            <InternalElement localElementName="080.2">
                <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib" refRoleClass="PCE request">
                    <AdditionalAttribute LocalAttributeName="I processing function">
                        <DefaultValue>true</DefaultValue>
                    </AdditionalAttribute>
                    <AdditionalAttribute LocalAttributeName="C processing function">
                        <DefaultValue>true</DefaultValue>
                    </AdditionalAttribute>
                    <AdditionalAttributeValue refAttribute="PCE category">
                        <Value>T</Value>
                    </AdditionalAttributeValue>
                    <AdditionalAttributeValue refAttribute="Location">
                        <Value>Central</Value>
                    </AdditionalAttributeValue>
                    <AdditionalExternalInterface LocalInterfaceName="TIC" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
                    <AdditionalExternalInterface LocalInterfaceName="AH" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="AlarmSource"/>
                    <AdditionalExternalInterface LocalInterfaceName="AL" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="AlarmSource"/>
                    <AdditionalExternalInterface LocalInterfaceName="In000" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="SensorSink"/>
                    <AdditionalExternalInterface LocalInterfaceName="I" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="IndicationSource"/>
                </RoleRequirements>
            </InternalElement>
            <InternalElement localElementName="080.3">
                <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib" refRoleClass="PCE request">
                    <AdditionalAttribute LocalAttributeName="Device Information">
                        <DefaultValue>on/off</DefaultValue>
                    </AdditionalAttribute>
                    <AdditionalAttributeValue refAttribute="PCE category">
                        <Value>H</Value>
                    </AdditionalAttributeValue>
                    <AdditionalAttributeValue refAttribute="Location">
                        <Value>Local panel</Value>
                    </AdditionalAttributeValue>
                    <AdditionalExternalInterface LocalInterfaceName="SH" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
                    <AdditionalExternalInterface LocalInterfaceName="SL" GlobalInterfaceClassLibName="IEC
                    PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
                </RoleRequirements>
            </InternalElement>
            <InternalElement localElementName="080.4">
                <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib" refRoleClass="PCE request">
                    <AdditionalAttribute LocalAttributeName="Y processing function">
                        <DefaultValue>true</DefaultValue>
                    </AdditionalAttribute>
                    <AdditionalAttributeValue refAttribute="PCE category">
                        <Value>U</Value>
                    </AdditionalAttributeValue>
                    <AdditionalAttributeValue refAttribute="Location">
                        <Value>Central</Value>
                    </AdditionalAttributeValue>
```

```
          <AdditionalExternalInterface LocalInterfaceName="Y" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
          <AdditionalExternalInterface LocalInterfaceName="In000" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
          <AdditionalExternalInterface LocalInterfaceName="In001" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
          <AdditionalExternalInterface LocalInterfaceName="In002" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
          <AdditionalExternalInterface LocalInterfaceName="In003" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
          <AdditionalExternalInterface LocalInterfaceName="In004" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSink"/>
        </RoleRequirements>
      </InternalElement>
      <InternalElement localElementName="080.5">
        <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib" refRoleClass="PCE request">
          <AdditionalAttributeValue refAttribute="PCE category">
            <Value>H</Value>
          </AdditionalAttributeValue>
          <AdditionalAttributeValue refAttribute="Location">
            <Value>Central</Value>
          </AdditionalAttributeValue>
          <AdditionalExternalInterface LocalInterfaceName="SH" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
          <AdditionalExternalInterface LocalInterfaceName="SL" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="SignalSource"/>
          <AdditionalExternalInterface LocalInterfaceName="OH" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="IndicationSource"/>
          <AdditionalExternalInterface LocalInterfaceName="OL" GlobalInterfaceClassLibName="IEC
          PASS6xxxx InterfaceLib" refInterfaceClass="IndicationSource"/>
        </RoleRequirements>
      </InternalElement>
      <InternalLink localLinkName="080.5SH_080.4In000" refPartnerSideA="080.5"
      refPartnerInterfaceSideA="SH" refPartnerSideB="080.4" refPartnerInterfaceSideB="In000"/>
      <InternalLink localLinkName="080.5SL_080.4In001" refPartnerSideA="080.5"
      refPartnerInterfaceSideA="SL" refPartnerSideB="080.4" refPartnerInterfaceSideB="In001"/>
      <InternalLink localLinkName="080.3SH_080.4In002" refPartnerSideA="080.3"
      refPartnerInterfaceSideA="SH" refPartnerSideB="080.4" refPartnerInterfaceSideB="In002"/>
      <InternalLink localLinkName="080.3SL_080.4In003" refPartnerSideA="080.3"
      refPartnerInterfaceSideA="SL" refPartnerSideB="080.4" refPartnerInterfaceSideB="In003"/>
      <InternalLink localLinkName="080.2TIC_080.4In004" refPartnerSideA="080.2"
      refPartnerInterfaceSideA="TIC" refPartnerSideB="080.4" refPartnerInterfaceSideB="In004"/>
      <InternalLink localLinkName="080.4YC_080.1In000" refPartnerSideA="080.4"
      refPartnerInterfaceSideA="YC" refPartnerSideB="080.1" refPartnerInterfaceSideB="In000"/>
    </SystemUnitClass>
  </SystemUnitClassLib>
</CAEXFile>
```

8 Recommended attributes

The object of this subclause is to give a set of attributes, which are typically stored in P&ID systems and relevant in the PCE environments. If used, these attributes shall be exchanged in the manner shown in Table 6 via the CAEX data exchange format.

The attributes given in Table 6 describe information with respect to the special process connections. These attributes shall be mapped to additional attributes of corresponding process connection interfaces.

Table 6 – P&ID attributes relevant in PCE environment

Attributes	CAEX mapping	Definition
Medium Code	AdditionalAttribute (see A.17.6)	
Medium code description	AdditionalAttribute (see A.17.6)	
Material balance point	AdditionalAttribute (see A.17.6)	
Pressure rating	AdditionalAttribute (see A.17.6)	According to EN 982
Design temperature	AdditionalAttribute (see A.17.6)	According to EN 1594
Pipe class	AdditionalAttribute (see A.17.6)	According to EN 13480-1
Main nominal pipe size	AdditionalAttribute (see A.17.6)	
Adjusted nominal pipe size	AdditionalAttribute (see A.17.6)	
Connection size	AdditionalAttribute (see A.17.6)	
Heat tracing	AdditionalAttribute (see A.17.6)	
Heat tracing type	AdditionalAttribute (see A.17.6)	
Heat tracing temperature set point	AdditionalAttribute (see A.17.6)	
Equipment/pipe flag	AdditionalAttribute (see A.17.6)	
Equipment id	AdditionalAttribute (see A.17.6)	
Pipe id	AdditionalAttribute (see A.17.6)	
Insulation type	AdditionalAttribute (see A.17.6)	
Insulation width	AdditionalAttribute (see A.17.6)	

The attributes given in Table 7 concern information with respect to the internal object management. They shall be mapped to additional attributes of the corresponding object.

Table 7 – Data handling attributes

Attributes	CAEX mapping	Normative reference
InternalUniqueID	AdditionalAttribute (see A.17.6)	
Short description	AdditionalAttribute (see A.17.6)	
Long description	AdditionalAttribute (see A.17.6)	

Annex A
(normative)

CAEX – Data model for machine information exchange

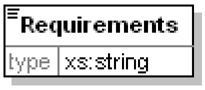
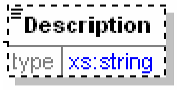
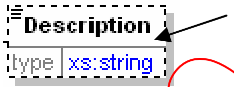
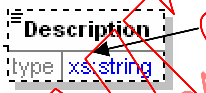
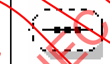
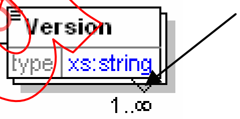
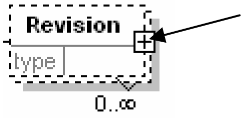
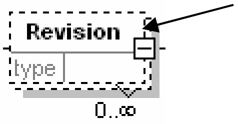
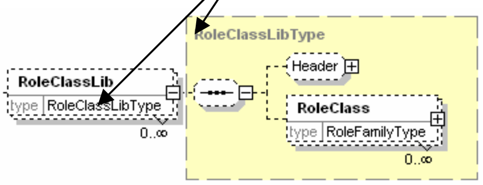
A.1 CAEX and its diagram conventions

The neutral data format CAEX defines structures for the definition of plant elements with their characteristics and its relationships. CAEX is a basis for a general exchange format for CAE planning data and is specified as XML schema.

It allows mixtures from standardized partial solutions and single solutions on all hierarchy levels, it supports the change process; it promotes but does not enforce the structure of libraries and permits the free exchange of data.

The schema diagrams uses the following conventions to illustrate the structure of the schema elements, the types of the elements, the attributes, the rules for optional elements and the repetitions (see Table 8).

Table 8 – Convention

Diagram element	Description	Example
Rectangle with solid border	Indicates an element that shall be implemented (mandatory)	
Rectangle with dashed border	Indicates an Element that may be implemented (optional)	
Datatype	Indicates the datatype of an Element – after the keyword “type” in the second line of an Element	 Datatype of the XML Element
Namespace	Indicates the namespace of the used datatype. (Keyword “xs:”) The described CAEX schema refers only to the namespace of W3C (xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema)	 Used Namespace
Sequence	Indicates that the following elements shall be in the defined order	
Range	Indicates the range in which the element can occur. For example 1 to infinite	 Range of the element
Plus Sign	Indicates that the xml-element contains another element. The containing elements are not shown	 Contains sub-elements
Minus Sign	Indicates that all containing XML-elements are shown	 All containing sub-elements are shown
Greying background with dashed rectangle	Indicates that the elements shown are composed in a defined datatype. The name of the datatype is shown at the top of the dashed rectangle with dashed lines	Datatype of the element shown  RoleClassLibType Header RoleClassLib type RoleClassLibType 0..∞ RoleClass type RoleFamilyType 0..∞

A.2 Definition of terms

In the following, all the elements and datatypes of CAEX are described.

Table 9 – CAEX datatypes and elements

AdditionalInformation	Auxiliary field that may contain any additional information. It may be used in the substructure of the header.
AdditionalAttribute	Characterizes an additional attribute
AdditionalAttributeValue	Characterizes an additional value of a defined attribute and its constraints

AdditionalExternalInterface	Characterizes an additional external interface of a defined interface
Attribute	Characterizes a property of an defined element
AttributeNameMapping	Mapping of an internal used attribute name to an ‘external’ defined attribute name
AttributeValue	Values of predefined attributes may be set
CAEXFile	Root-element of the exchanged data
Constraint	Element to restrict the range of validity of a defined attribute
Copyright	OrganiZational information about the copyright protection
DefaultValue	Predefined value for an attribute
Description	Textual description
ExternalInterface	Description of an external interface of an element
InterfaceClass	Definition of a class of an interface type
InterfaceClassLib	Definition of a library of interface classes. It may contain any class definitions
InterfaceNameMapping	Mapping of an internally used interface name to an ‘externally’ defined interface name
InternalElement	Description of the internal structure of an element
InternalLink	Defines the relationship between internal elements
MappingObject	
NominalScaledType	Element to define constraints of nominal scaled attribute values
OrdinalScaledType	Element to define constraints of ordinal scaled attribute values
SystemHierarchy	Describes a concrete hierarchy of a plant
SystemHierarchyElement	Describes a sub-plant structure
PredefinedRealisation	Element to reference a predefined realization of an element used
Quantity	
refSemantic	Reference to a definition of a defined attribute, for example to an attribute in a standardized library
RequiredMaxValue	Element to define a maximum value of an attribute
RequiredMinValue	Element to define a minimum value of an attribute
RequiredValue	Element to define a required value of an attribute
RequiredValues	Element to define a required value of an attribute. It may be instantiated multiple times as a substructure of the element NominalscaledType
Requirements	Defines requirements as a constraint for an attribute value
Revision	Organizational information about the state of the revision
RoleClass	Definition of a class of a role type
RoleClassLib	Definition of a library of role classes. It may contain any class definitions
RoleRequirements	Defines requirements for a role. It contains substructures for the definition of a required additional attribute, a required additional attribute value, and a required additional external interface
SingletonClassDescription	Definition of a class of a type SystemUnitClass as a unique description. This means it cannot be referenced by another element and it cannot be derived from another class definition
SupportedRoleClass	Characteriz roles a defined class can play
SystemUnitClass	Definition of a class of a SystemUnitClass type
SystemUnitClassLib	Definition of a library of SystemUnit classes. It may contain any class definitions

UnknownType	Element to define constraints for attribute values of an unknown scale type
Value	Element to take a value of an attribute
Version	Organizational information about the state of the version
AttributeType	Defines base structures for attribute definitions
AttributeValueRequirementType	Defines base structures for definition of constraints for values of an attribute
AttributeValueType	Defines base structures for definition of a value of an attribute
ChangeMode	Describes the state of an element. The changeMode may have the following values: state, create, delete and change. This information can be used for further change management applications
Header	Defines a group of organizational information, like description, version, revision, copyright, etc.
InterfaceClassLibType	Defines base structures for an interface library definition
InterfaceClassType	Defines base structures for an interface class definition
InterfaceFamilyType	Defines base structures for an interface family definition. Interface family definitions contain derived interface class definitions of a base interface class
InterfaceRequirementType	Defines base structures for definition of interface requirements
MappingType	
SystemHierarchyElementType	Defines base structures for definition of a substructure of a plant hierarchy
SystemHierarchyLibType	Defines base structures for a plant hierarchy library definition
PropertyType	Defines an attribute to take in information for further development
reference	Defines an attribute to take in a reference to another element and attributes. References to elements are represented as a path of elements divided by a slash. Reference to an attribute of an element is represented by a point between the referenced element and the referenced attribute
RoleClassLibType	Defines base structures for a role library definition
RoleClassType	Defines base structures for a role class definition
RoleClassFamilyType	Defines base structures for a role family definition. Role family definitions contain derived role class definitions of a base role class
SystemUnitClassLibType	Defines base structures for a SystemUnit library definition
SystemUnitClassType	Defines base structures for a SystemUnit class definition
SystemUnitFamilyType	Defines base structures for a SystemUnit family definition. SystemUnit family definitions contain derived SystemUnit class definitions of a base SystemUnit class
SystemUnitInstanceType	Defines base structures for a SystemUnit instance definition

A.3 Definition of elements

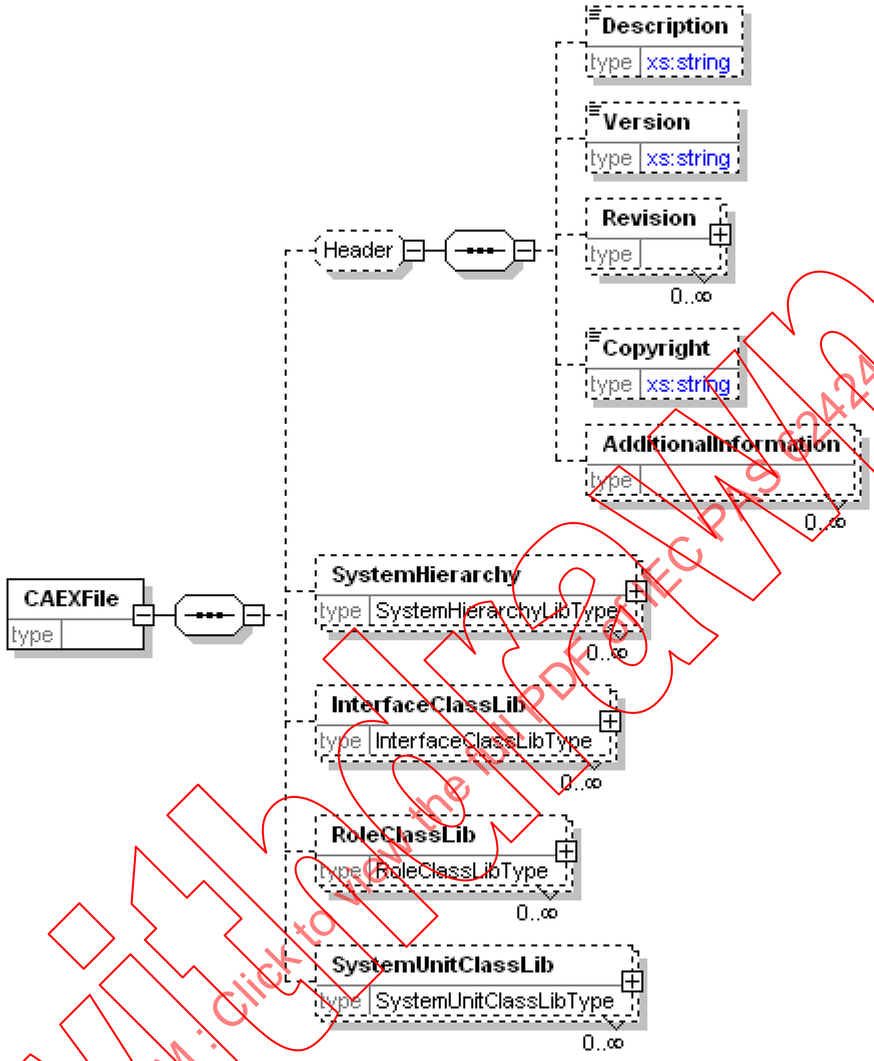
A.3.1 General

The CAEX Model consists of abstract XML elements and attributes for the specification of any plant items. Elements may have subelements and attributes.

Each element has the attributes “PropertyString” and “changeMode”. They are designed to store detailed information about the exchanged data. Usually they would contain structured information. “changeMode”, contains a list of data: {state, create, delete, change} (see Clause A.21).

A.3.2 Element CAEXFile

The element “CAEXFile” describes the root element of the data exchange format. The attribute “FileName” shall be used and stores the name of the transferred file.

Diagram	
Children	**Description Version Revision Copyright AdditionalInformation SystemHierarchy InterfaceClassLib RoleClassLib SystemUnitClassLib**

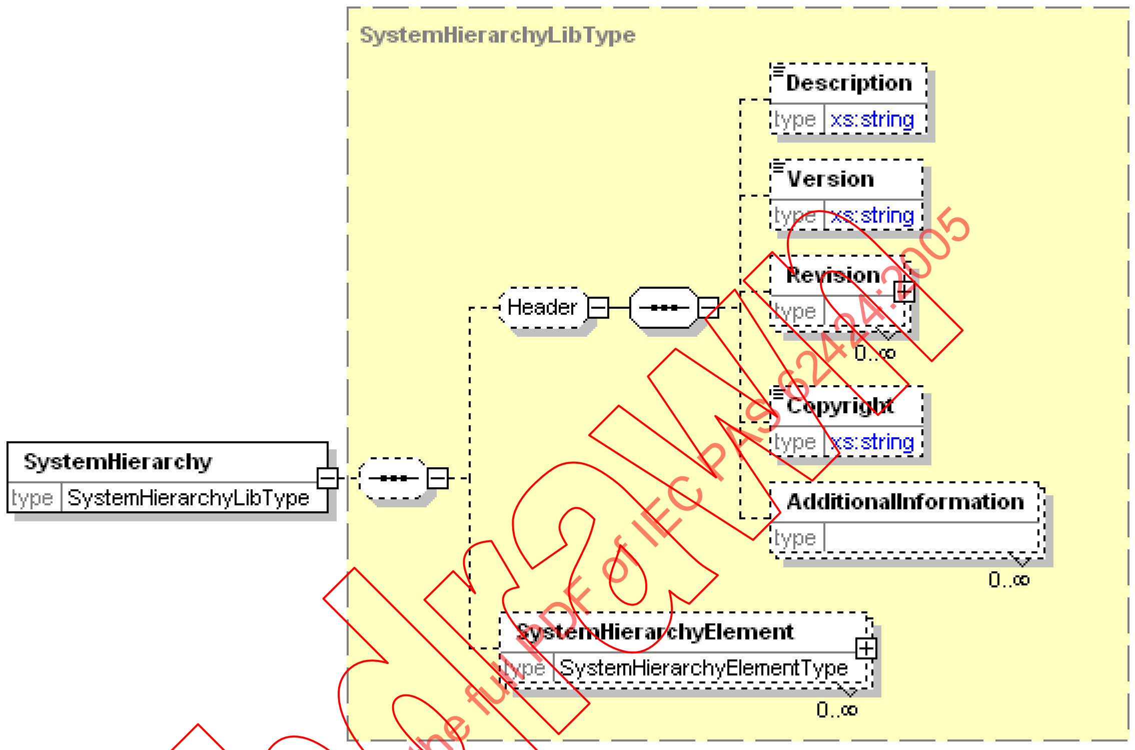
Attributes	Name	Type	Use	Default	Fixed	Annotation
	FileName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		

Source

```
<xs:element name="CAEXFile">
 <xs:complexType>
  <xs:sequence>
   <xs:group ref="Header" minOccurs="0"/>
   <xs:element name="SystemHierarchy" type="SystemHierarchyLibType" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element name="InterfaceClassLib" type="InterfaceClassLibType" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element name="RoleClassLib" type="RoleClassLibType" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element name="SystemUnitClassLib" type="SystemUnitClassLibType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="FileName" type="xs:string" use="required"/>
  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
 </xs:complexType>
</xs:element>
```

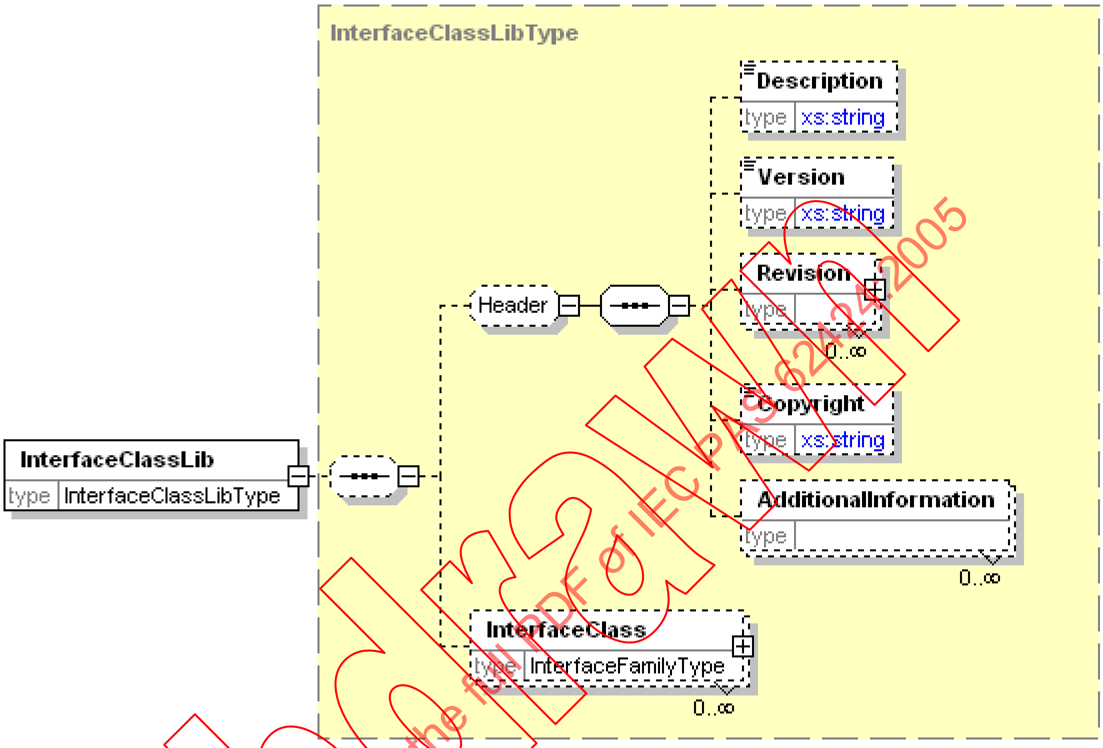
A.3.3 Element CAEXFile/SystemHierarchy

The element “SystemHierarchy” specifies hierarchy structures of plants. Therefore, the attribute “GlobalSystemHierarchyName” shall be used. It describes the identifier of the hierarchy. A detailed description of the hierarchy may be stored in the attribute “description”. The element “SystemHierarchy” is of the type “SystemHierarchyLibType” (see Clause A.16).

<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Type</td><td><b>SystemHierarchyLibType</b></td></tr>
<tr><td>Children</td><td><b>Description Version Revision Copyright AdditionalInformation SystemHierarchyElement</b></td></tr>
<tr><td>Attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>GlobalSystemHierarchyName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Source</td><td><xs:element name="SystemHierarchy" type="SystemHierarchyLibType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

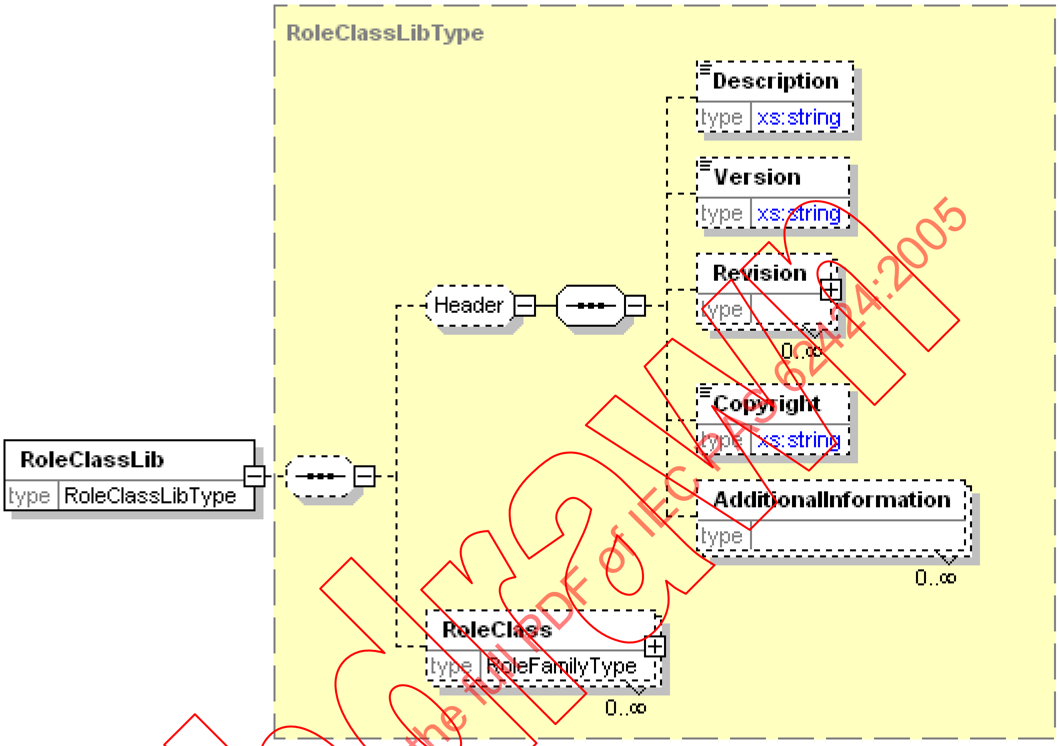
A.3.4 Element CAEXFile/InterfaceClassLib

The element “InterfaceClassLib” is the root element for the definition of interface classes. It may contain multiple interface class definitions. Each interface class library shall have a unique library name “GlobalInterfaceLibName” in the exchanged file. The element “InterfaceClassLib” is of the type “InterfaceClassLibType” (see Clause A.7).

Diagram	 InterfaceClassLibType Description type xs:string Version type xs:string Revision type 0..∞ Header Copyright type xs:string AdditionalInformation type 0..∞ InterfaceClassLib type InterfaceClassLibType InterfaceClass type InterfaceFamilyType 0..∞					
Type	**InterfaceClassLibType**					
Children	**Description Version Revision Copyright AdditionalInformation InterfaceClass**					
Attributes	Name	Type	Use	Default	Fixed	Annotation
	GlobalInterfaceLibName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
Source	<xs:element name="InterfaceClassLib" type="InterfaceClassLibType" minOccurs="0" maxOccurs="unbounded"/>					

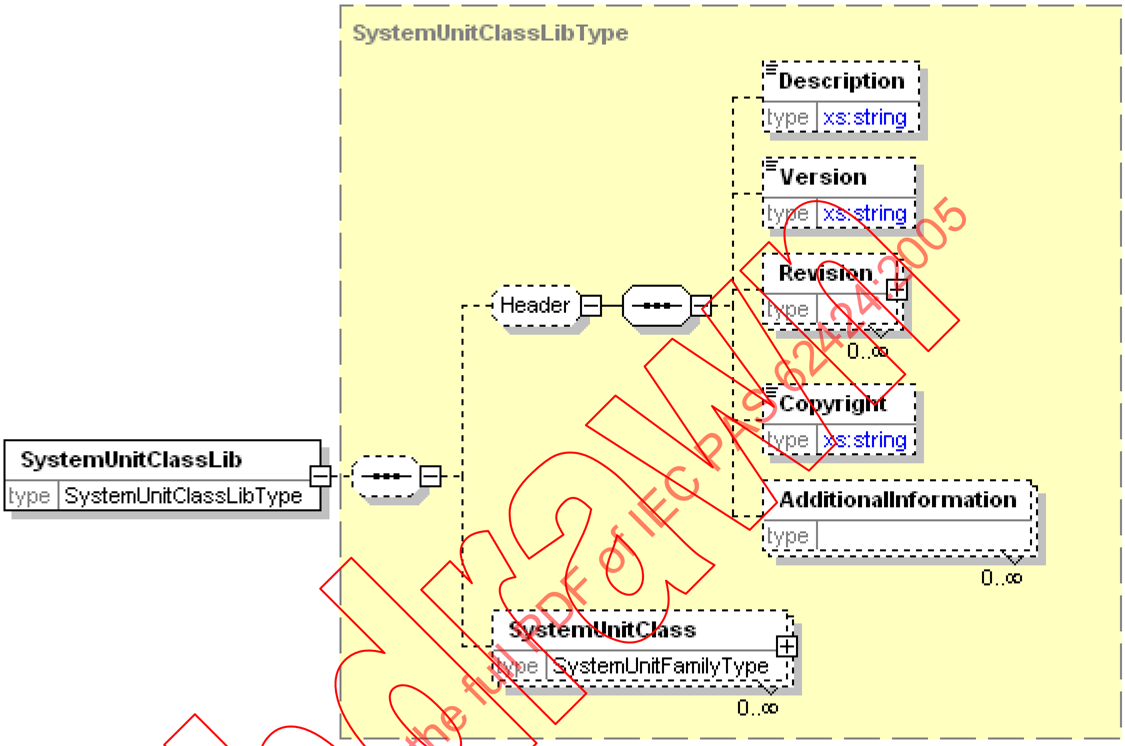
A.3.5 Element CAEXFile/RoleClassLib

The element “RoleClassLib” is the root element for the definition of role classes. It may contain multiple role class definitions. Each role class library shall have an unique library name “GlobalRoleLibName”in the exchanged file. The element “RoleClassLib” is of the type “RoleClassLibType” (see Clause A.12).

Diagram	 RoleClassLibType Description type xs:string Version type xs:string Revision type 0..∞ Header Copyright type xs:string AdditionalInformation type 0..∞ RoleClassLib type RoleClassLibType RoleClass type RoleFamilyType 0..∞					
Type	**RoleClassLibType**					
Children	**Description Version Revision Copyright AdditionalInformation RoleClass**					
Attributes	Name	Type	Use	Default	Fixed	Annotation
	GlobalRoleLibName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
Source	<xs:element name="RoleClassLib" type="RoleClassLibType" minOccurs="0" maxOccurs="unbounded"/>					

A.3.6 Element CAEXFile/SystemUnitClassLib

The element “SystemUnitClassLib” is the root element for the definition of system unit classes. It may contain multiple system unit class definitions. Each system unit class library shall have a unique library name “GlobalSystemUnitLibName”in the exchanged file. The element “SystemUnitClassLib” is of the type “SystemUnitClassLibType” (see Clause A.17).

<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>Type</td><td><b>SystemUnitClassLibType</b></td></tr>
<tr><td>Children</td><td><b>Description Version Revision Copyright AdditionalInformation SystemUnitClass</b></td></tr>
<tr><td>Attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>GlobalSystemUnitLibName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Source</td><td><xs:element name="SystemUnitClassLib" type="SystemUnitClassLibType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.4 Definition of Types

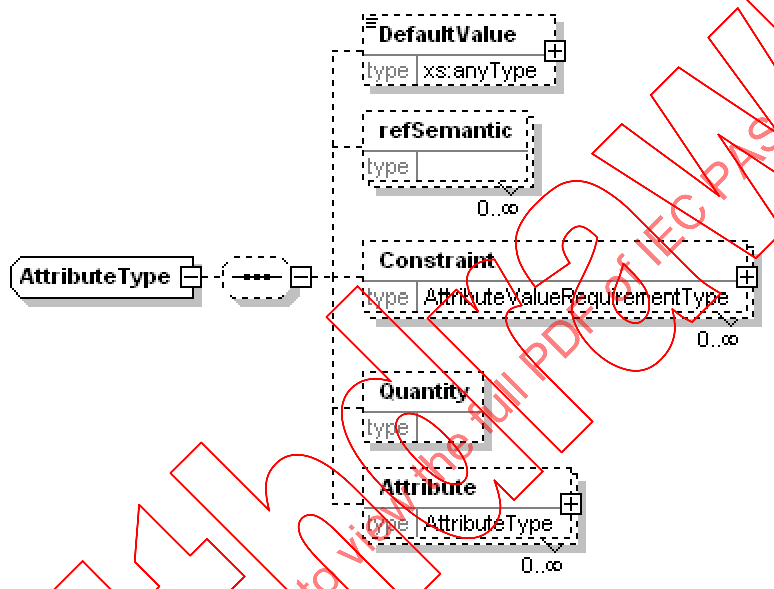
A.4.1 General

In the following, datatypes of the CAEX metamodel are described. Basically XML allows the definition of two different kinds of datatypes: simple and complex datatype. Simple datatypes may enhance or restrict predefined XML datatypes. Complex datatypes may have also subelements. Once defined, dataype may be used multiple times in the CAEX metamodel.

A.4.2 ComplexType AttributeType

Attributes may specifiy characteristics of plant items. Each attribute requires a unique attribute name stored in “LocalAttributeName”. A textual description may be stored in “description”. “attributeDatatype” may contain the type of the specified attribute, for example, integer, single, string, etc. The “dimension” may contain the dimension like cm, m, or °C.

Diagram



children **DefaultValue refSemantic Constraint Quantity Attribute**

used by elements **InterfaceRequirementType/AdditionalAttribute SystemUnitClassType/InternalElement/RoleRequirements/AdditionalAttribute InterfaceClassType/Attribute RoleClassType/Attribute SystemUnitClassType/Attribute AttributeType/Attribute**

attributes

Name	Type	Use	Default	Fixed	Annotation
LocalAttributeName	xs:string	required			
description	xs:string	optional			
dimension	xs:string	optional			
attributeDataType	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

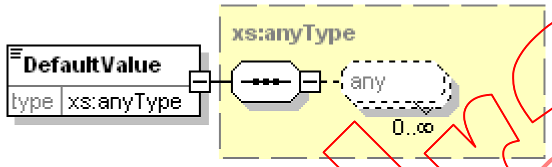
source

```
<xs:complexType name="AttributeType">
 <xs:sequence minOccurs="0">
  <xs:element name="DefaultValue" type="xs:anyType" minOccurs="0"/>
  <xs:element name="refSemantic" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:attribute name="correspondingAttributeName" type="reference" use="required"/>
    <xs:attribute name="GlobalSemanticLibName" type="xs:string" use="required"/>
    <xs:attribute name="description" type="xs:string" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="Constraint" type="AttributeValueRequirementType" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="Quantity" minOccurs="0">
   <xs:complexType>
    <xs:attribute name="quantityName" type="reference" use="required"/>
    <xs:attribute name="GlobalQuantityLibName" type="xs:string" use="required"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
```

<table>
<tr><td></td><td></xs:element>
<xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="LocalAttributeName" type="xs:string" use="required"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="dimension" type="xs:string" use="optional"/>
<xs:attribute name="attributeDataType" use="optional">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="Integer"/>
<xs:enumeration value="Real"/>
<xs:enumeration value="String"/>
<xs:enumeration value="Boolean"/>
<xs:enumeration value="Complex"/>
<xs:enumeration value="Category"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="PropertyString" type="propertystring" use="optional"/>
<xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType></td></tr>
</table>

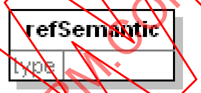
A.4.3 Element AttributeType/DefaultValue

The element “DefaultValue” may store predefined values of the specified attributes.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>**xs:anyType**</td></tr>
<tr><td>attributes</td><td><table><tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr></table></td></tr>
<tr><td>source</td><td><xs:element name="DefaultValue" type="xs:anyType" minOccurs="0"/></td></tr>
</table>

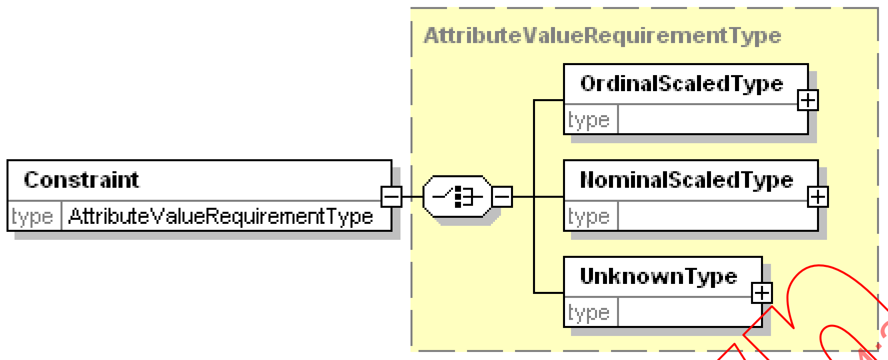
A.4.4 Element AttributeType/refSemantic

The meaning of the specified attribute may be defined in the element “refSemantic”. The meaning may be specified by a reference to a corresponding attribute in a semantic library. Therefore, the element contains the required field “correspondingAttributeName” and “GlobalSemanticLibName”. GlobalSemantic libraries may contain a list of standardized attribute definitions. The field “description” may contain more descriptions of the meaning of the referenced attribute.

<table>
<tr><td>Diagram</td><td>
</td></tr>
<tr><td>attributes</td><td><table><tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr><tr><td>correspondingAttributeName</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr><tr><td>GlobalSemanticLibName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr><tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr><tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr></table></td></tr>
<tr><td>source</td><td><xs:element name="refSemantic" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="correspondingAttributeName" type="reference" use="required"/>
<xs:attribute name="GlobalSemanticLibName" type="xs:string" use="required"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType>
</xs:element></td></tr>
</table>

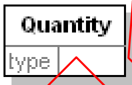
A.4.5 Element AttributeType/Constraint

The element “constraint” may specifiy restrictions of the specified attributes. For each constraint a field “constraintName” shall be used. The datatype of this element is described in A.5.12.

diagram	 AttributeValueRequirementType OrdinalScaledType type NominalScaledType type UnknownType type Constraint type AttributeValueRequirementType					
type	**AttributeValueRequirementType**					
children	**OrdinalScaledType NominalScaledType UnknownType**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	constraintName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="Constraint" type="AttributeValueRequirementType" minOccurs="0" maxOccurs="unbounded"/>					

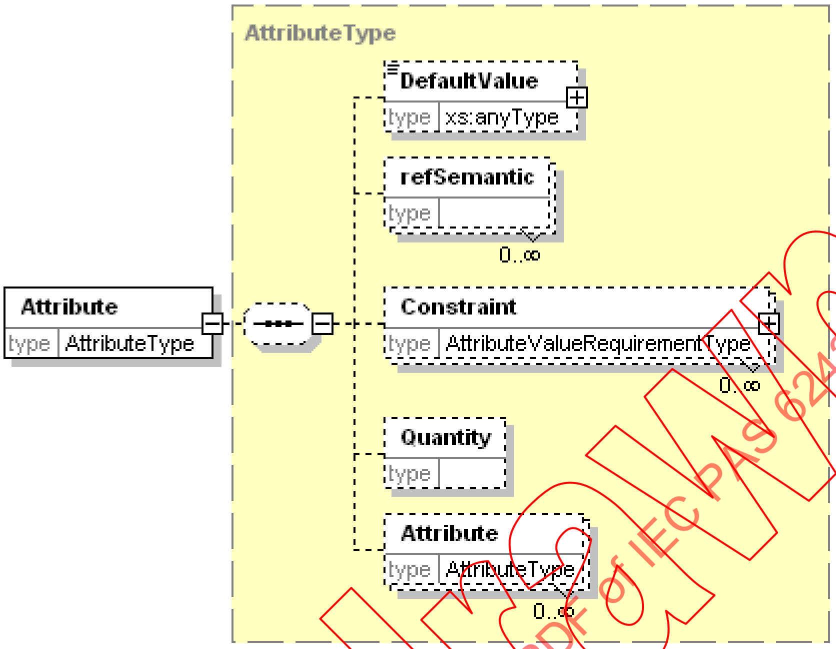
A.4.6 Element AttributeType/Quantity

The element “Quantity” may contain basic dimension description like length or time. The element contains similar to the description of “refSemantic” element a field for the quantity name “quantityName” and a reference to the quantity library “GlobalQuantityLibName” where it is defined.

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	quantityName	reference	required			
	GlobalQuantityLibName	xs:string	required			
	changeMode	changeMode	optional	state		
source	<xs:element name="Quantity" minOccurs="0"> <xs:complexType> <xs:attribute name="quantityName" type="reference" use="required"/> <xs:attribute name="GlobalQuantityLibName" type="xs:string" use="required"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:complexType> </xs:element>					

A.4.7 Element AttributeType/Attribute

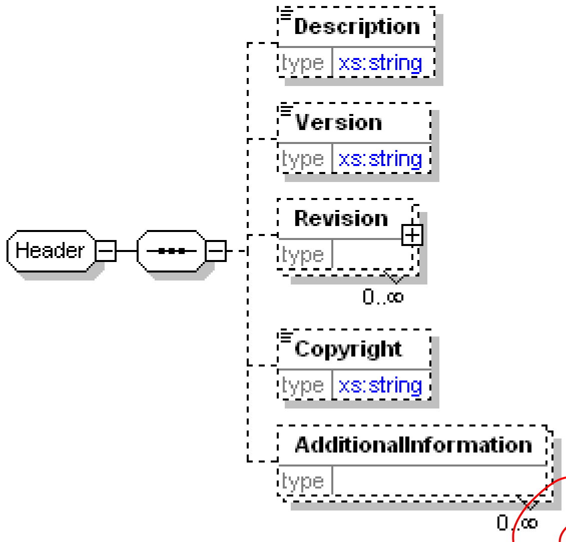
The element “attribute” as a sub-element of the attributetype may specify user-defined categories of attributes. This means a definition of structured attributes are possible. This definition describes a recursive structure of attribute definitions. It has the same attribute datatype definition.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>source</td><td><xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.5 Group Header

A.5.1 General

The group “Header” specifies a general description of organizational information.

diagram	 Description type xs:string Version type xs:string Revision type 0..∞ Copyright type xs:string AdditionalInformation type 0..∞ Header
children	**Description Version Revision Copyright AdditionalInformation**
used by	element **CAEXFile** complexTypes **InterfaceClassLibType InterfaceClassType InterfaceRequirementType SystemHierarchyLibType RoleClassLibType RoleClassType SystemUnitClassLibType SystemUnitClassType SystemUnitInstanceType**

source

```
<xs:group name="Header">
 <xs:sequence>
  <xs:element name="Description" minOccurs="0">
   <xs:complexType>
    <xs:simpleContent>
     <xs:extension base="xs:string">
      <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
      <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="Version" minOccurs="0">
   <xs:complexType>
    <xs:simpleContent>
     <xs:extension base="xs:string">
      <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
      <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
     </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="Revision" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="RevisionDate" type="xs:dateTime"/>
     <xs:element name="OldVersion" type="xs:string" minOccurs="0"/>
     <xs:element name="NewVersion" type="xs:string" minOccurs="0"/>
     <xs:element name="AuthorName" type="xs:string"/>
     <xs:element name="Comment" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="Copyright" minOccurs="0">
   <xs:complexType>
    <xs:simpleContent>
     <xs:extension base="xs:string">
```

	<xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element name="AdditionalInformation" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:attribute name="title" type="xs:string" use="required"/> <xs:attribute name="content" type="xs:string" use="optional"/> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:complexType> </xs:element> </xs:sequence> </xs:group>					

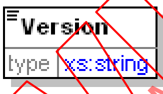
A.5.2 Element Header/Description

The element “Description” may store the organizational information of a unit.

diagram						
type	extension of **xs:string**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="Description" minOccurs="0"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element>					

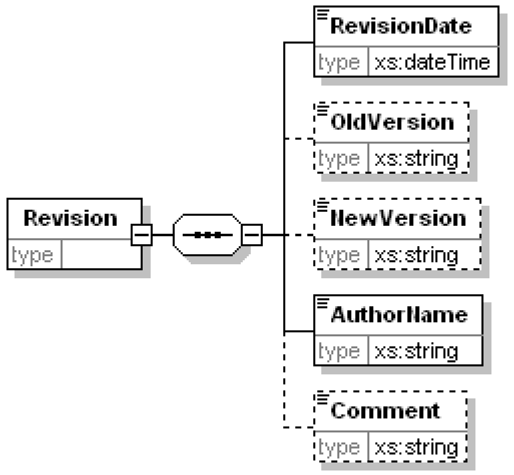
A.5.3 Element Header/Version

The element “Version” may store information about the version.

diagram						
type	extension of **xs:string**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="Version" minOccurs="0"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element>					

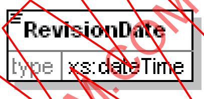
A.5.4 Element Header/Revision

The element “Revision” may store general information about the revision.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>children</td><td><b>RevisionDate OldVersion NewVersion AuthorName Comment</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:element name="Revision" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RevisionDate" type="xs:dateTime"/>
      <xs:element name="OldVersion" type="xs:string" minOccurs="0"/>
      <xs:element name="NewVersion" type="xs:string" minOccurs="0"/>
      <xs:element name="AuthorName" type="xs:string"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
  </xs:complexType>
</xs:element&gt;</td></tr>
</table>

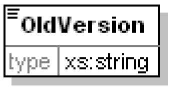
A.5.5 Element Header/Revision/RevisionDate

The element “RevisionDate” may store the date of the revision.

diagram	
type	**xs:dateTime**
source	<xs:element name="RevisionDate" type="xs:dateTime"/>

A.5.6 Element Header/Revision/OldVersion

The element “OldVersion” may store the old version of the revision information.

diagram	
type	**xs:string**
source	<xs:element name="OldVersion" type="xs:string" minOccurs="0"/>

A.5.7 Element Header/Revision/NewVersion

The element “NewVersion” may store the new version of the revision information.

diagram	
type	**xs:string**
source	<xs:element name="NewVersion" type="xs:string" minOccurs="0"/>

A.5.8 Element Header/Revision/AuthorName

The element “AuthorName” may store the name of the author who has filled out the version information.

diagram	
type	**xs:string**
source	<xs:element name="AuthorName" type="xs:string"/>

A.5.9 Element Header/Revision/Comment

The element “Comment” may store comments of the revision information.

diagram	
type	**xs:string**
source	<xs:element name="Comment" type="xs:string" minOccurs="0"/>

A.5.10 Element Header/Copyright

The element “Copyright” may store copyright information.

diagram						
type	extension of **xs:string**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="Copyright" minOccurs="0"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element>					

A.5.11 Element Header/AdditionalInformation

The element “AdditionalInformation” may store additional information of the organizational information. It shall contain the field “title” and may contain the field “content”.

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
title	xs:string	required			
content	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

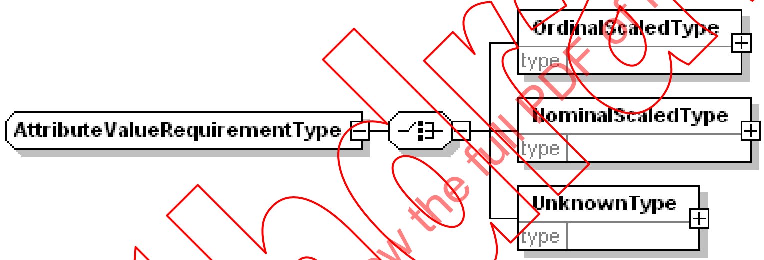
source

```
<xs:element name="AdditionalInformation" minOccurs="0" maxOccurs="unbounded">
 <xs:complexType>
  <xs:attribute name="title" type="xs:string" use="required"/>
  <xs:attribute name="content" type="xs:string" use="optional"/>
  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
 </xs:complexType>
</xs:element>
```

A.5.12 complexType AttributeValueRequirementType

The datatype “AttributeValueRequirementType” may specifiy the requirements for values of attributes. It shall contain the field “constraintName”.

diagram



children **OrdinalScaledType NominalScaledType UnknownType**

used by elements **AttributeType/Constraint AttributeValueType/Constraint**

attributes

Name	Type	Use	Default	Fixed	Annotation
constraintName	xs:string	required			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

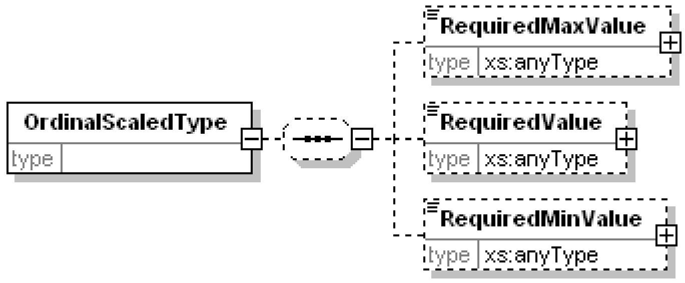
source

```
<xs:complexType name="AttributeValueRequirementType">
 <xs:choice>
  <xs:element name="OrdinalScaledType">
   <xs:complexType>
    <xs:sequence minOccurs="0">
     <xs:element name="RequiredMaxValue" type="xs:anyType" minOccurs="0"/>
     <xs:element name="RequiredValue" type="xs:anyType" minOccurs="0"/>
     <xs:element name="RequiredMinValue" type="xs:anyType" minOccurs="0"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="NominalScaledType">
   <xs:complexType>
    <xs:sequence minOccurs="0">
     <xs:element name="RequiredValues" type="xs:anyType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="UnknownType">
   <xs:complexType>
    <xs:sequence minOccurs="0">
     <xs:element name="Requirements" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:choice>
```

	<xs:attribute name="constraintName" type="xs:string" use="required"/>
	  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
	  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
	</xs:complexType>

A.5.13 Element AttributeValueRequirementType/OrdinalScaledType

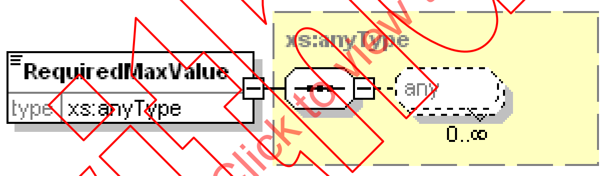
The element "OrdinalScaledType" may store requirements of values of attributes if the datatype of the specified attribute is of an ordinal scaled datatype.

diagram	 OrdinalScaledType type RequiredMaxValue type xs:anyType RequiredValue type xs:anyType RequiredMinValue type xs:anyType
children	**RequiredMaxValue RequiredValue RequiredMinValue**

source	<xs:element name="OrdinalScaledType">
	  <xs:complexType>
	    <xs:sequence minOccurs="0">
	      <xs:element name="RequiredMaxValue" type="xs:anyType" minOccurs="0"/>
	      <xs:element name="RequiredValue" type="xs:anyType" minOccurs="0"/>
	      <xs:element name="RequiredMinValue" type="xs:anyType" minOccurs="0"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>

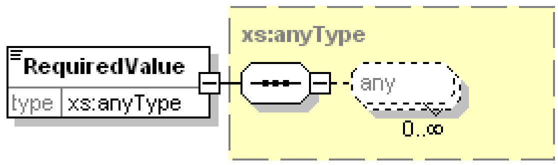
A.5.14 Element AttributeValueRequirementType/OrdinalScaledType/RequiredMaxValue

The element "RequiredMaxValue" may store a maximum value of a specified attribute.

diagram	 RequiredMaxValue type xs:anyType xs:anyType any 0..∞
type	**xs:anyType**
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="RequiredMaxValue" type="xs:anyType" minOccurs="0"/>

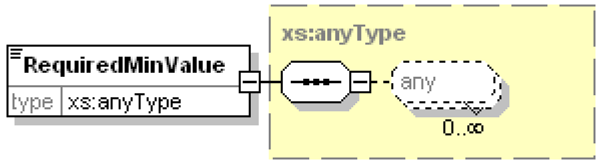
A.5.15 Element AttributeValueRequirementType/OrdinalScaledType/RequiredValue

The element "RequiredValue" may store a required value of a specified attribute.

diagram	 RequiredValue type xs:anyType xs:anyType any 0..∞
type	**xs:anyType**
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="RequiredValue" type="xs:anyType" minOccurs="0"/>

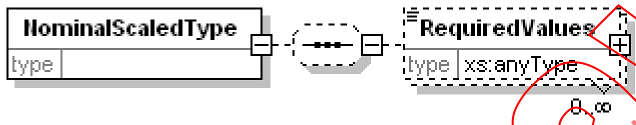
A.5.16 Element AttributeValueRequirementType/OrdinalScaledType/RequiredMinValue

The element “RequiredMinValue” may store a minimum value of a specified attribute.

diagram	 RequiredMinValue type xs:anyType xs:anyType any 0..∞
type	**xs:anyType**
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="RequiredMinValue" type="xs:anyType" minOccurs="0"/>

A.5.17 Element AttributeValueRequirementType/NominalScaledType

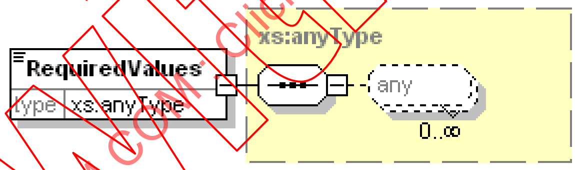
The element “NominalScaledType” may store requirements of values of attributes if the datatype of the specified attribute is of a nominal scaled datatype.

diagram	 NominalScaledType type RequiredValues type xs:anyType 0..∞
children	**RequiredValues**

source	<xs:element name="NominalScaledType">
	  <xs:complexType>
	    <xs:sequence minOccurs="0">
	      <xs:element name="RequiredValues" type="xs:anyType" minOccurs="0" maxOccurs="unbounded"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>

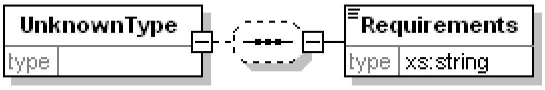
A.5.18 Element AttributeValueRequirementType/NominalScaledType/RequiredValues

The element “RequiredValues” may store values of specified attribute.

diagram	 RequiredValues type xs:anyType xs:anyType any 0..∞
type	**xs:anyType**
attributes	Name Type Use Default Fixed Annotation
source	<xs:element name="RequiredValues" type="xs:anyType" minOccurs="0" maxOccurs="unbounded"/>

A.5.19 Element AttributeValueRequirementType/UnknownType

The element “UnknownType” may store requirements of values of attributes if the datatype of the specified attribute is unknown.

diagram	
children	**Requirements**

source

```
<xs:element name="UnknownType">
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element name="Requirements" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

A.5.20 Element AttributeValueRequirementType/UnknownType/Requirements

The element “Requirements” may store any requirements of a specified attribute.

diagram	Requirements type xs:string
type	**xs:string**

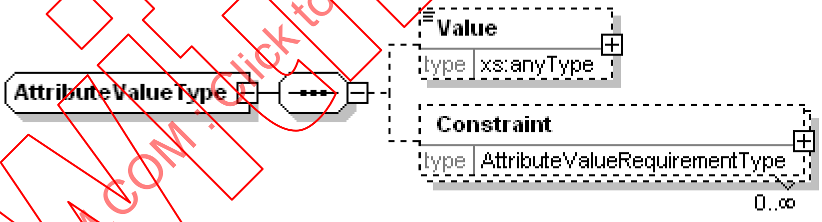
source

```
<xs:element name="Requirements" type="xs:string"/>
```

A.6 ComplexType AttributeValueType

A.6.1 General

The datatype “AttributeValueType” may store values and constraints of defined attributes. Predefined attribute definitions can be enhanced by value definitions or restricted by constraint definitions. It contains the required field “refAttribute” which refers to the definition of the attribute. The description and dimension of the attribute can be stored in the fields “description” and “dimension”.

diagram	 AttributeValueType Value type xs:anyType Constraint type AttributeValueRequirementType 0..∞
children	**Value Constraint**
used by	elements **SystemUnitClassType/InternalElement/RoleRequirements/AdditionalAttributeValue InterfaceClassType/AttributeValue InterfaceRequirementType/AttributeValue RoleClassType/AttributeValue SystemUnitClassType/AttributeValue**

attributes

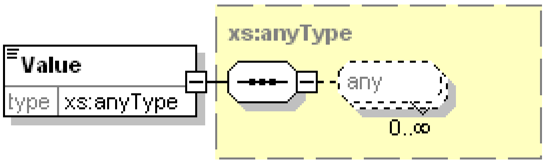
Name	Type	Use	Default	Fixed	Annotation
refAttribute	reference	required			
description	xs:string	optional			
dimension	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:complexType name="AttributeValueType">
  <xs:sequence>
    <xs:element name="Value" type="xs:anyType" minOccurs="0"/>
    <xs:element name="Constraint" type="AttributeValueRequirementType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="refAttribute" type="reference" use="required"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="dimension" type="xs:string" use="optional"/>
  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType>
```

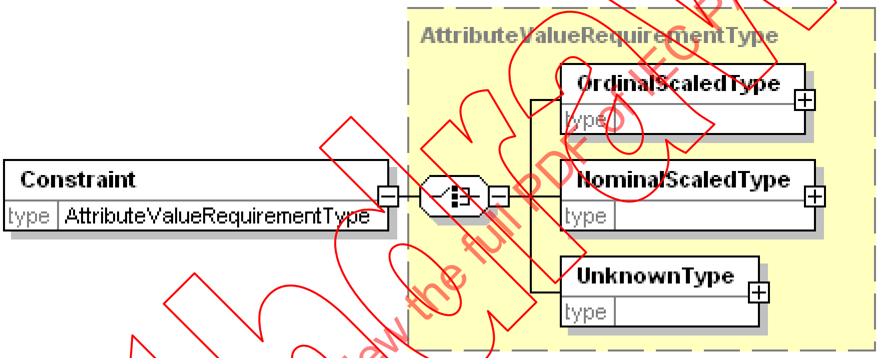
A.6.2 Element AttributeValueType/Value

The element “Value” may store the value of a specified attribute.

<table>
<tr><td>diagram</td><td colspan="6">
</td></tr>
<tr><td>type</td><td colspan="6"><b>xs:anyType</b></td></tr>
<tr><td>attributes</td><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>source</td><td colspan="6">&lt;xs:element name="Value" type="xs:anyType" minOccurs="0"/&gt;</td></tr>
</table>

A.6.3 Element AttributeValueType/Constraint

The element “Constraint” may store restrictions of specified attribute. It is based in the defined dataytpe “AttributeValueRequirementType” (see A.5.12).

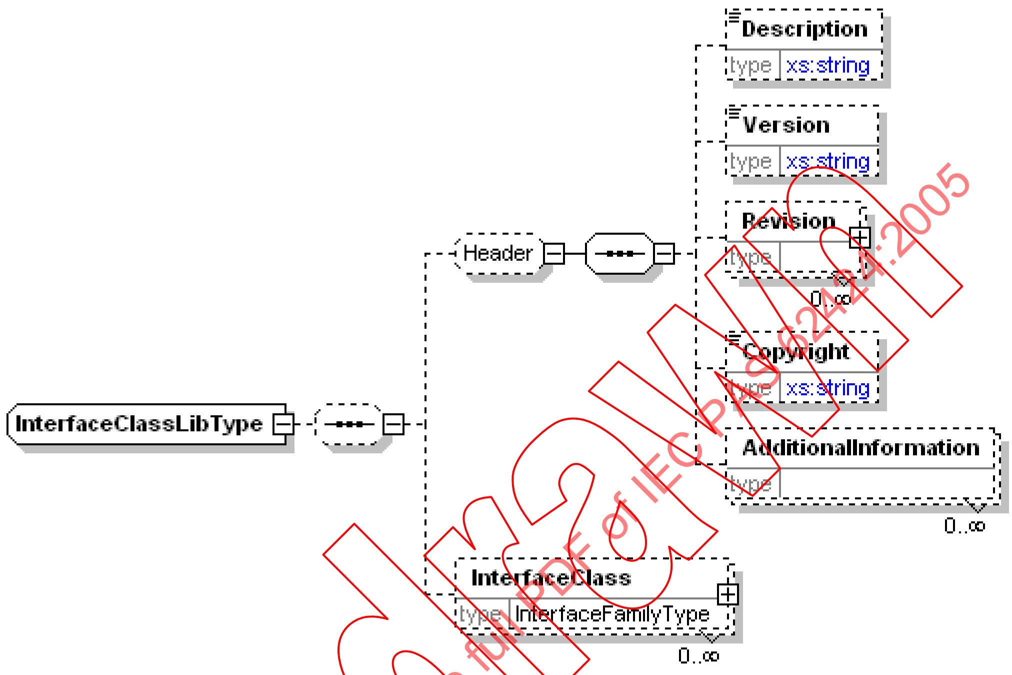
<table>
<tr><td>diagram</td><td colspan="6">
</td></tr>
<tr><td>type</td><td colspan="6"><b>AttributeValueRequirementType</b></td></tr>
<tr><td>children</td><td colspan="6"><b>OrdinalScaledType NominalScaledType UnknownType</b></td></tr>
<tr><td>attributes</td><td>Name
constraintName
PropertyString
changeMode</td><td>Type
xs:string
propertystring
changeMode</td><td>Use
required
optional
optional</td><td>Default

state</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>source</td><td colspan="6">&lt;xs:element name="Constraint" type="AttributeValueRequirementType" minOccurs="0" maxOccurs="unbounded"/&gt;</td></tr>
</table>

A.7 ComplexType InterfaceClassLibType

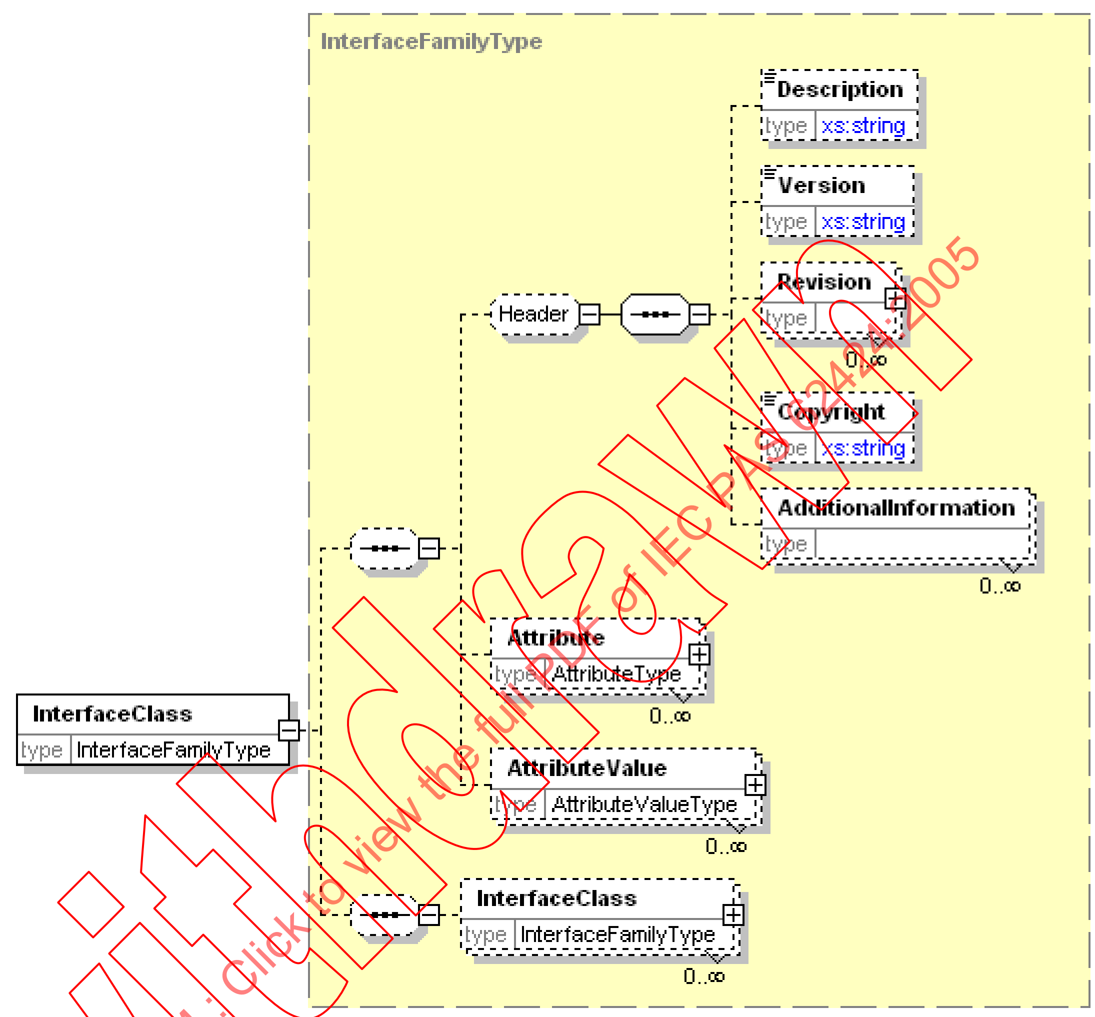
A.7.1 General

The datatype “InterfaceClassLibType” specifies the structure of interface libraries. It may contain organizational information, stored in the element “Header” and class definitions of interfaces. The datatype shall contain the field “GlobalInterfaceLibName”.

diagram	 InterfaceClassLibType Header Description type xs:string Version type xs:string Revision type 0..∞ Copyright type xs:string AdditionalInformation type 0..∞ InterfaceClass type InterfaceFamilyType 0..∞					
children	**Description Version Revision Copyright AdditionalInformation InterfaceClass**					
used by	element **CAEXFile/InterfaceClassLib**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	GlobalInterfaceLibName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:complexType name="InterfaceClassLibType"> <xs:sequence minOccurs="0"> <xs:group ref="Header" minOccurs="0"/> <xs:element name="InterfaceClass" type="InterfaceFamilyType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="GlobalInterfaceLibName" type="xs:string" use="required"/> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:complexType>					

A.7.2 Element InterfaceClassLibType/InterfaceClass

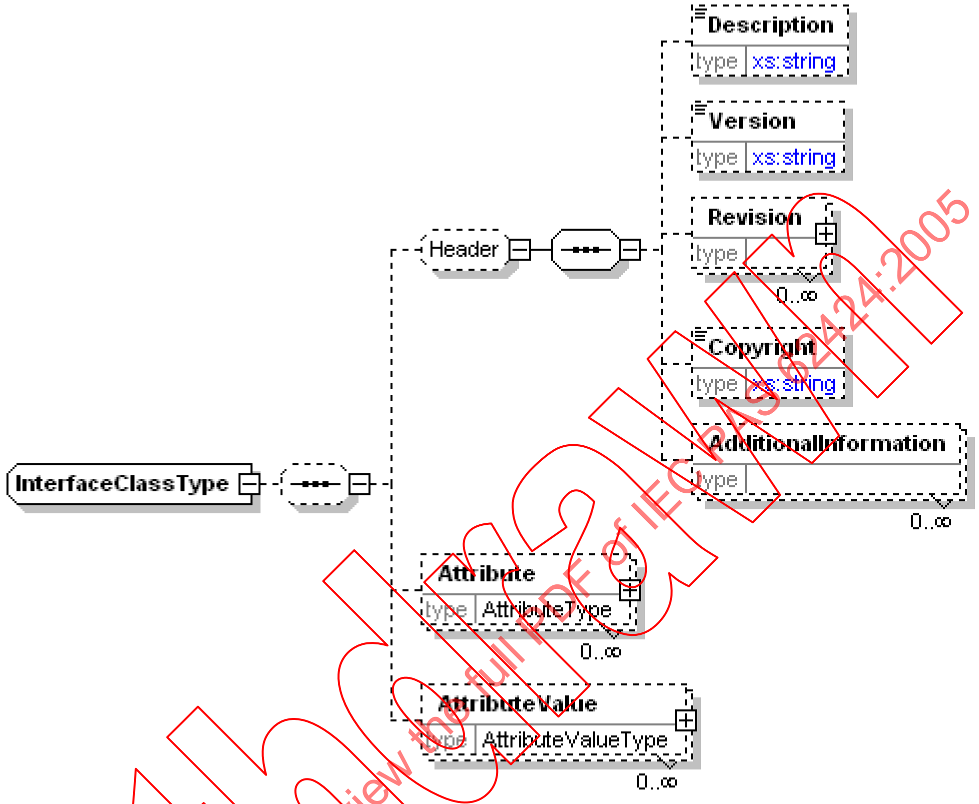
Interfaces are connection points of plant items. The definition of an interface may be stored in the element “InterfaceClass”. Each interface class shall contain a field “InterfaceClassName” which shall be uniqe in the structure of the interface library.

<table>
<tr><td>diagram</td><td></td></tr>
<tr><td>type</td><td><b>InterfaceFamilyType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue InterfaceClass</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>InterfaceClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>Optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>Optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="InterfaceClass" type="InterfaceFamilyType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.8 ComplexType InterfaceClassType

A.8.1 General

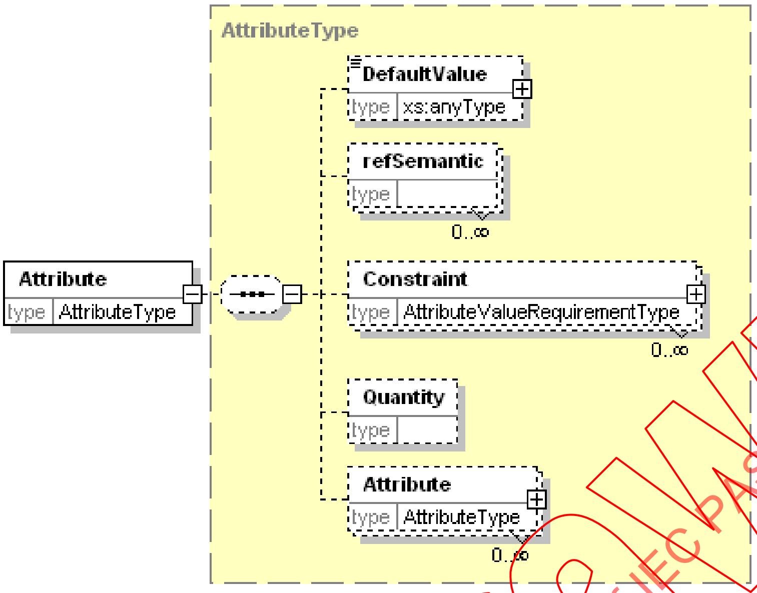
The datatype “InterfaceClassType” defines structures for the specification of interface classes. Each interface shall have an unique identifier stored in the field “InterfaceClassName”.

diagram	 Description type xs:string Version type xs:string Revision type 0..∞ Header Copyright type xs:string AdditionalInformation type 0..∞ InterfaceClassType Attribute type AttributeType 0..∞ AttributeValue type AttributeValueType 0..∞
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue**
used by	element **SystemUnitClassType/InternalElement/RoleRequirements/AdditionalExternalInterface** complexType **InterfaceFamilyType**
attributes	Name / Type / Use / Default / Fixed / Annotation InterfaceClassName / xs:string / required PropertyString / propertystring / optional changeMode / changeMode / optional / state

source	<xs:complexType name="InterfaceClassType">
	  <xs:sequence minOccurs="0">
	    <xs:group ref="Header" minOccurs="0"/>
	    <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
	    <xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/>
	  </xs:sequence>
	  <xs:attribute name="InterfaceClassName" type="xs:string" use="required"/>
	  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
	  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
	</xs:complexType>

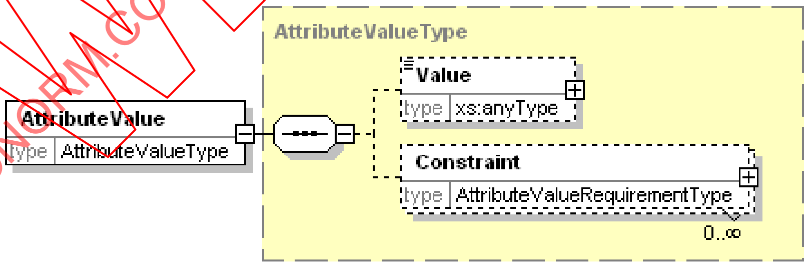
A.8.2 Element InterfaceClassType/Attribute

The element “Attribute” is of the datatype “AttributeType”. For a description of this datatype, see A.4.2.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.8.3 Element InterfaceClassType/AttributeValue

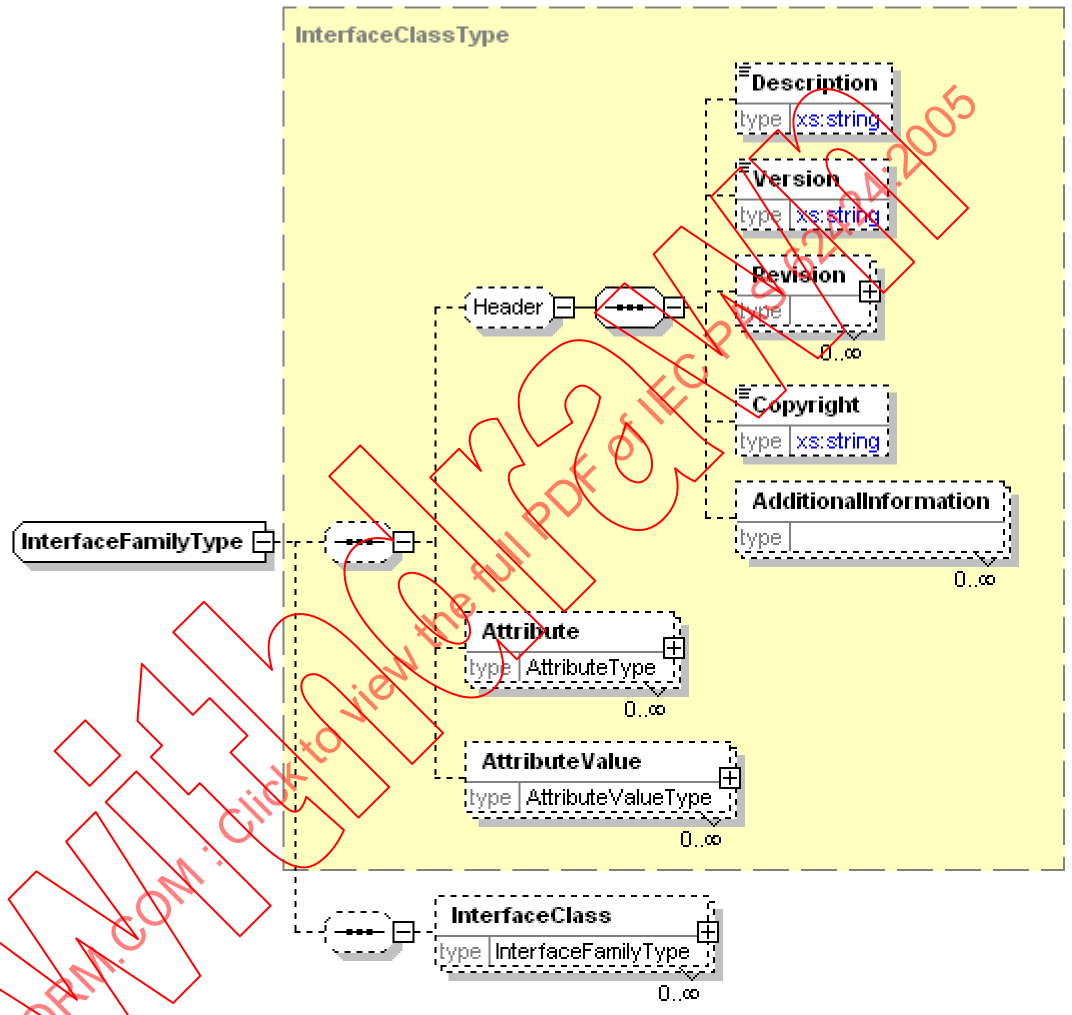
The element “AttributeValue” is of the datatype “AttributeValueType”. For a description of this datatype, see Clause A.6.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeValueType</b></td></tr>
<tr><td>children</td><td><b>Value Constraint</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>refAttribute</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.9 ComplexType InterfaceFamilyType

A.9.1 General

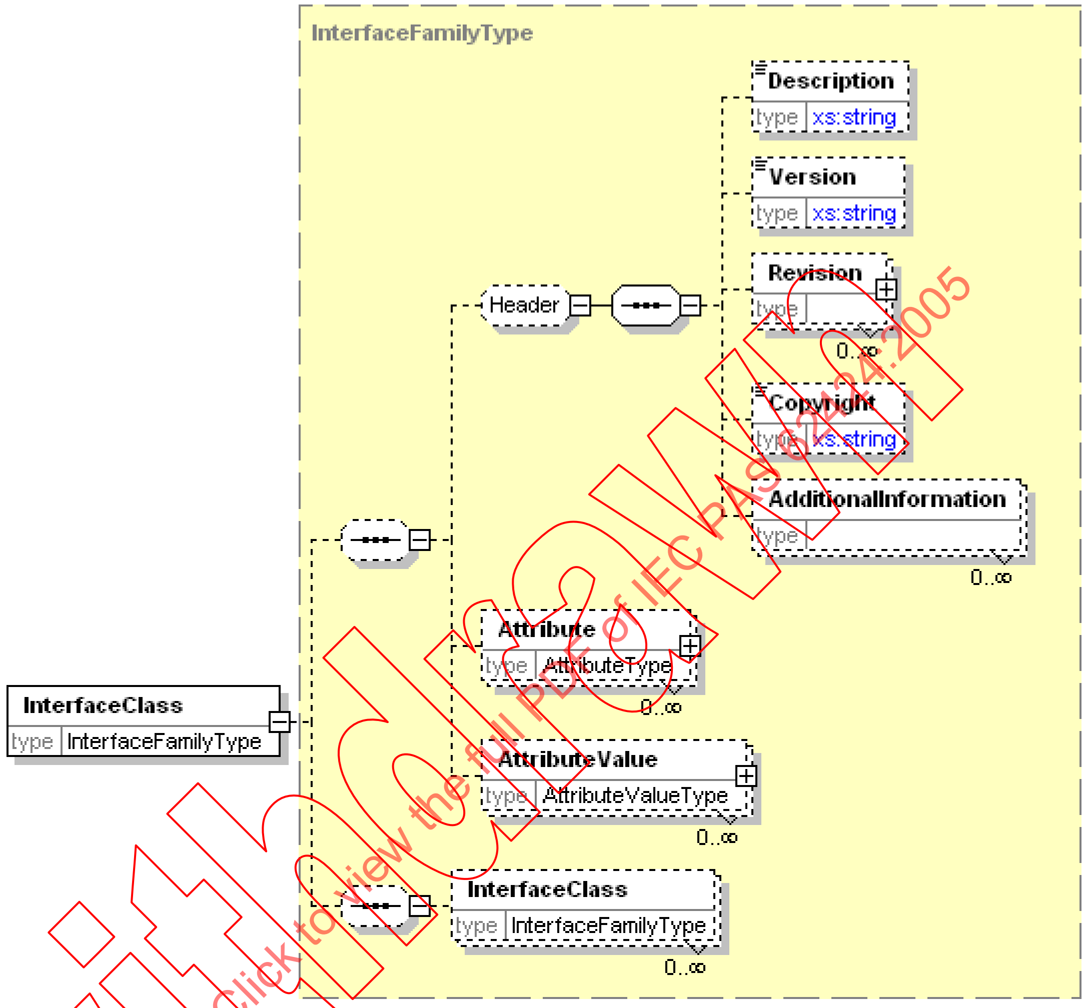
The datatype “InterfaceFamilyType” is an extension of the datatype “InterfaceClassType”. For a description of this datatype, see Clause A.8. The datatype “InterfaceFamilyType” allows the creation of child elements of interface classes. Child classes contain all characteristics (attribute elememts) of the base class but may be extended by new definitions of characteristics (attribute elements). Derived classes may restrict the value of predefined characteristics (attribute elements) by means of constraints.

diagram	 InterfaceClassType Description type xs:string Version type xs:string Revision type 0..∞ Header Copyright type xs:string AdditionalInformation type 0..∞ InterfaceFamilyType Attribute type AttributeType 0..∞ AttributeValue type AttributeValueType 0..∞ InterfaceClass type InterfaceFamilyType 0..∞
type	extension of **InterfaceClassType**
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue InterfaceClass**
used by	elements **InterfaceFamilyType/InterfaceClass InterfaceClassLibType/InterfaceClass**
attributes	Name Type Use Default Fixed Annotation InterfaceClassName xs:string required PropertyString propertystring optional changeMode changeMode optional state

source	<xs:complexType name="InterfaceFamilyType">
	  <xs:complexContent>
	    <xs:extension base="InterfaceClassType">
	      <xs:sequence minOccurs="0">
	        <xs:element name="InterfaceClass" type="InterfaceFamilyType" minOccurs="0" maxOccurs="unbounded"/>
	      </xs:sequence>
	    </xs:extension>
	  </xs:complexContent>
	</xs:complexType>

A.9.2 Element InterfaceFamilyType/InterfaceClass

The element “InterfaceClass” defines a class derived from base interface class. The element is of the datatype “InterfaceFamilyType”. For a description of this datatype, see Clause A.9.

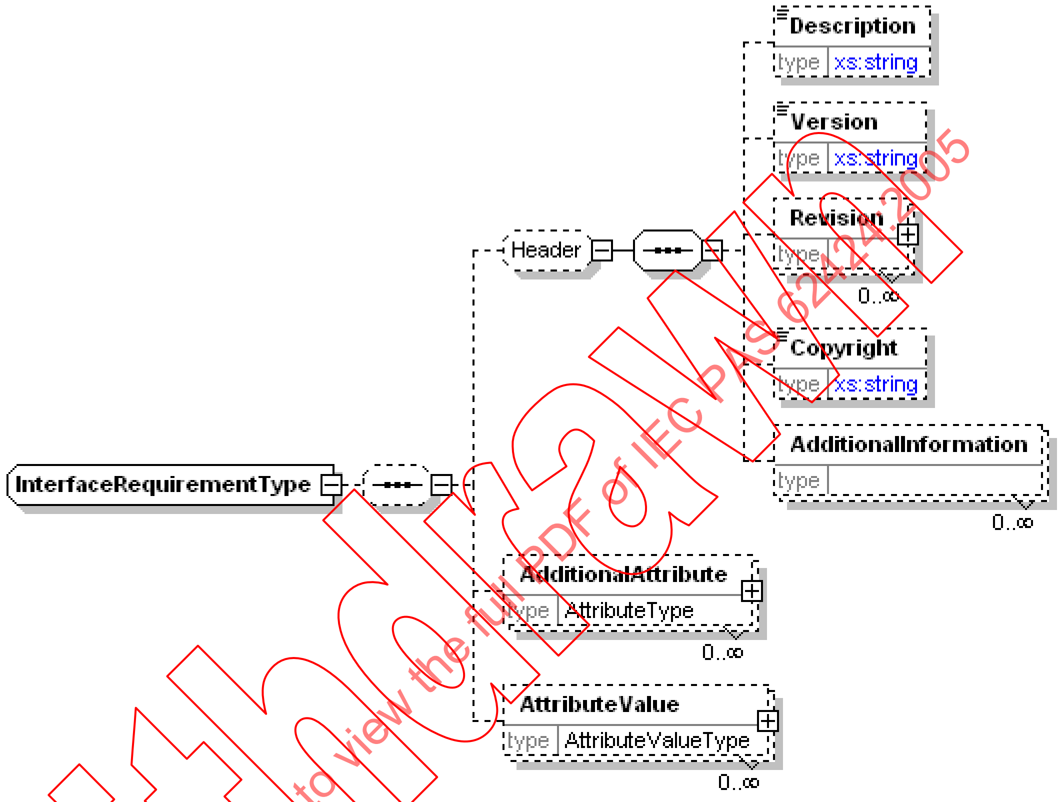
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>InterfaceFamilyType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue InterfaceClass</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>InterfaceClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="InterfaceClass" type="InterfaceFamilyType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.10 ComplexType InterfaceRequirementType

A.10.1 General

The datatype “InterfaceRequirementType” may specifiy the requirements of an interface. Existing interface class definitions may be enhanced by additional attributes or values of attributes to be set. If used, this datatype shall contain a field “LocalInterfaceName”. With the field “refInterfaceClass” an existing interface class definition may be referenced.

diagram



children	**Description Version Revision Copyright AdditionalInformation AdditionalAttribute AttributeValue**
used by	elements **RoleClassType/ExternalInterface SystemUnitClassType/ExternalInterface**

attributes

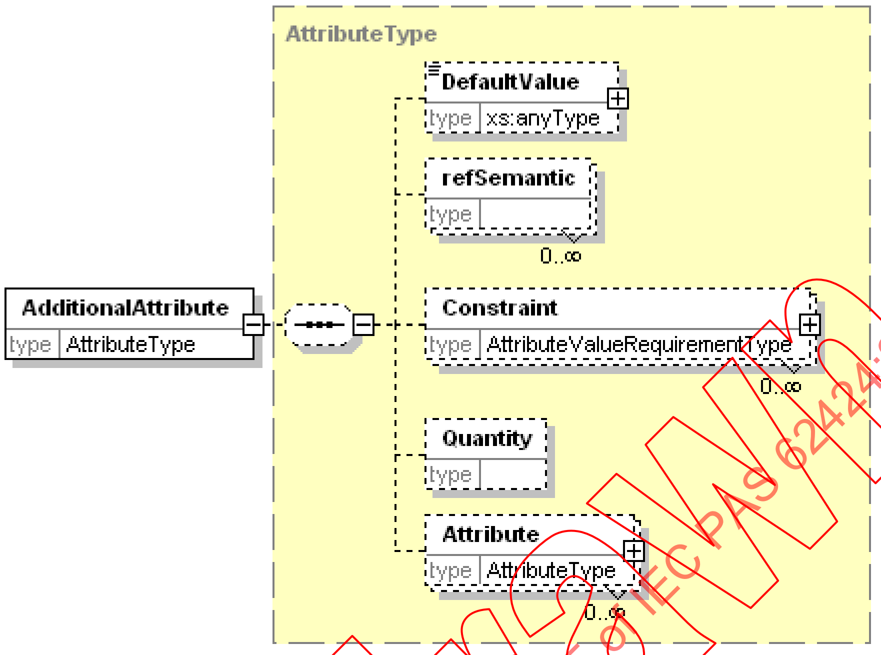
Name	Type	Use	Default	Fixed	Annotation
LocalInterfaceName	xs:string	required			
GlobalInterfaceClassLibName	xs:string	optional			
refInterfaceClass	reference	optional			
description	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:complexType name="InterfaceRequirementType">
 <xs:sequence minOccurs="0">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 <xs:attribute name="LocalInterfaceName" type="xs:string" use="required"/>
 <xs:attribute name="GlobalInterfaceClassLibName" type="xs:string" use="optional"/>
 <xs:attribute name="refInterfaceClass" type="reference" use="optional"/>
 <xs:attribute name="description" type="xs:string" use="optional"/>
 <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
 <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType>
```

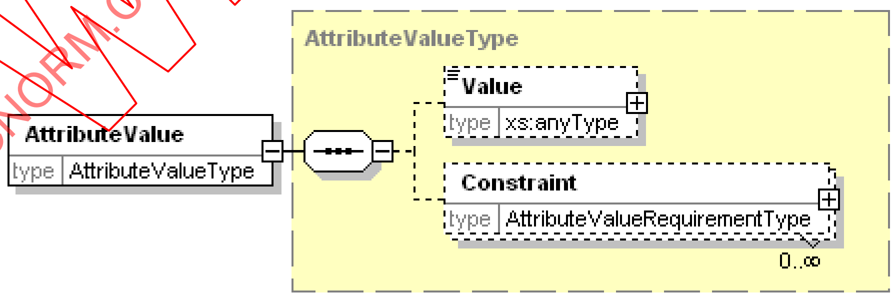
A.10.2 Element InterfaceRequirementType/AdditionalAttribute

The element “AdditionalAttribute”is of the datatype “AttributeType”. For a description of this datatype, see Clause A.4.2.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.10.3 Element InterfaceRequirementType/AttributeValue

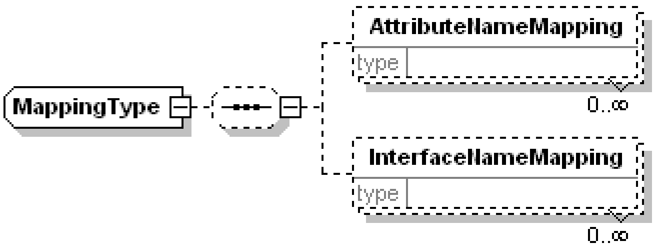
The element “AttributeValue” is of the datatype “AttributeValueType”. For a description of this datatype, see Clause A.6.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeValueType</b></td></tr>
<tr><td>children</td><td><b>Value Constraint</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>refAttribute</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.11 ComplexType MappingType

A.11.1 General

The datatype “MappingType” defines structures for the mapping of a general identifier to an user defined identifier.

diagram	
children	**AttributeNameMapping InterfaceNameMapping**
used by	elements **SystemUnitClassType/InternalElement/MappingObject** **SystemUnitClassType/SupportedRoleClass/MappingObject**

attributes

Name	Type	Use	Default	Fixed	Annotation
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:complexType name="MappingType">
 <xs:sequence minOccurs="0">
  <xs:element name="AttributeNameMapping" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:attribute name="SystemUnitAttributeName" type="xs:string" use="required"/>
    <xs:attribute name="RoleAttributeName" type="xs:string" use="required"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="InterfaceNameMapping" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:attribute name="SystemUnitInterfaceName" type="xs:string" use="required"/>
    <xs:attribute name="RoleInterfaceName" type="xs:string" use="required"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
 <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
 <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType>
```

A.11.2 Element MappingType/AttributeNameMapping

The element “AttributeNameMapping” maps the role attribute name to a system unit attribute name. Therefore, it contains the fields“SystemUnitAttributeName” and “RoleAttributeName”.

diagram	

attributes

Name	Type	Use	Default	Fixed	Annotation
SystemUnitAttributeName	xs:string	required			
RoleAttributeName	xs:string	required			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:element name="AttributeNameMapping" minOccurs="0" maxOccurs="unbounded">
 <xs:complexType>
  <xs:attribute name="SystemUnitAttributeName" type="xs:string" use="required"/>
  <xs:attribute name="RoleAttributeName" type="xs:string" use="required"/>
  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
 </xs:complexType>
</xs:element>
```

A.11.3 Element MappingType/InterfaceNameMapping

The element “InterfaceNameMapping” maps the role interface name to a system unit interface name. Therefore, it contains the fields “SystemUnitInterfaceName” and “RoleInterfaceName”.

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
SystemUnitInterfaceName	xs:string	required			
RoleInterfaceName	xs:string	required			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

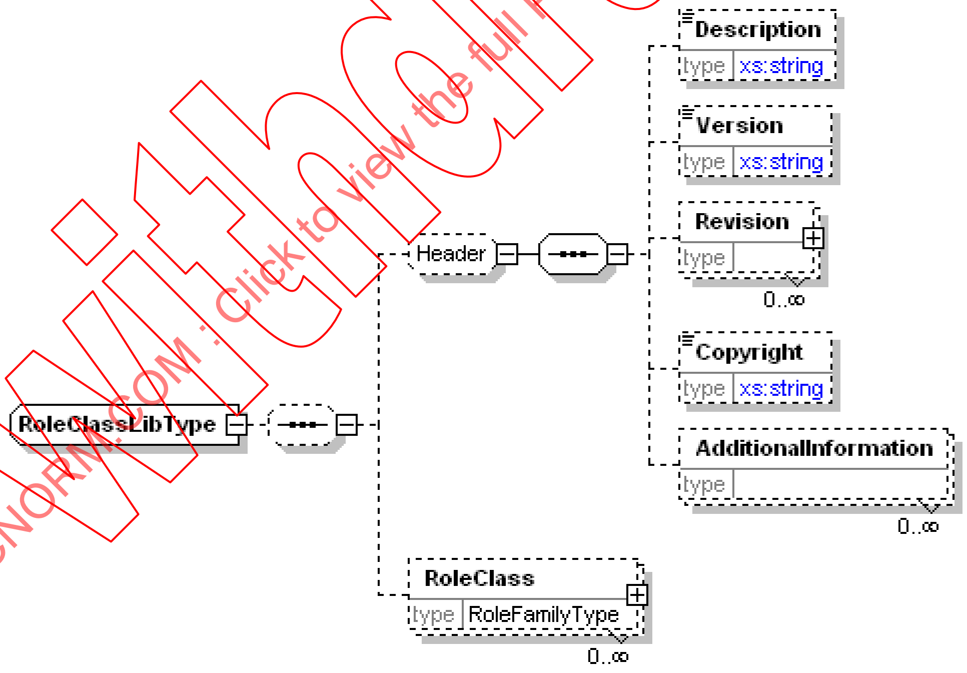
```
<xs:element name="InterfaceNameMapping" minOccurs="0" maxOccurs="unbounded">
 <xs:complexType>
  <xs:attribute name="SystemUnitInterfaceName" type="xs:string" use="required"/>
  <xs:attribute name="RoleInterfaceName" type="xs:string" use="required"/>
  <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
  <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
 </xs:complexType>
</xs:element>
```

A.12 ComplexType RoleClassLibType

A.12.1 General

The datatype “RoleClassLibType” specifies the structure of role libraries. Role libraries may contain multiple role class definitions. Each Role library shall have an unique library name stored in the field “GlobalRoleLibName”.

diagram



children **Description Version Revision Copyright AdditionalInformation RoleClass**

used by element **CAEXFile/RoleClassLib**

attributes

Name	Type	Use	Default	Fixed	Annotation
GlobalRoleLibName	xs:string	required			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

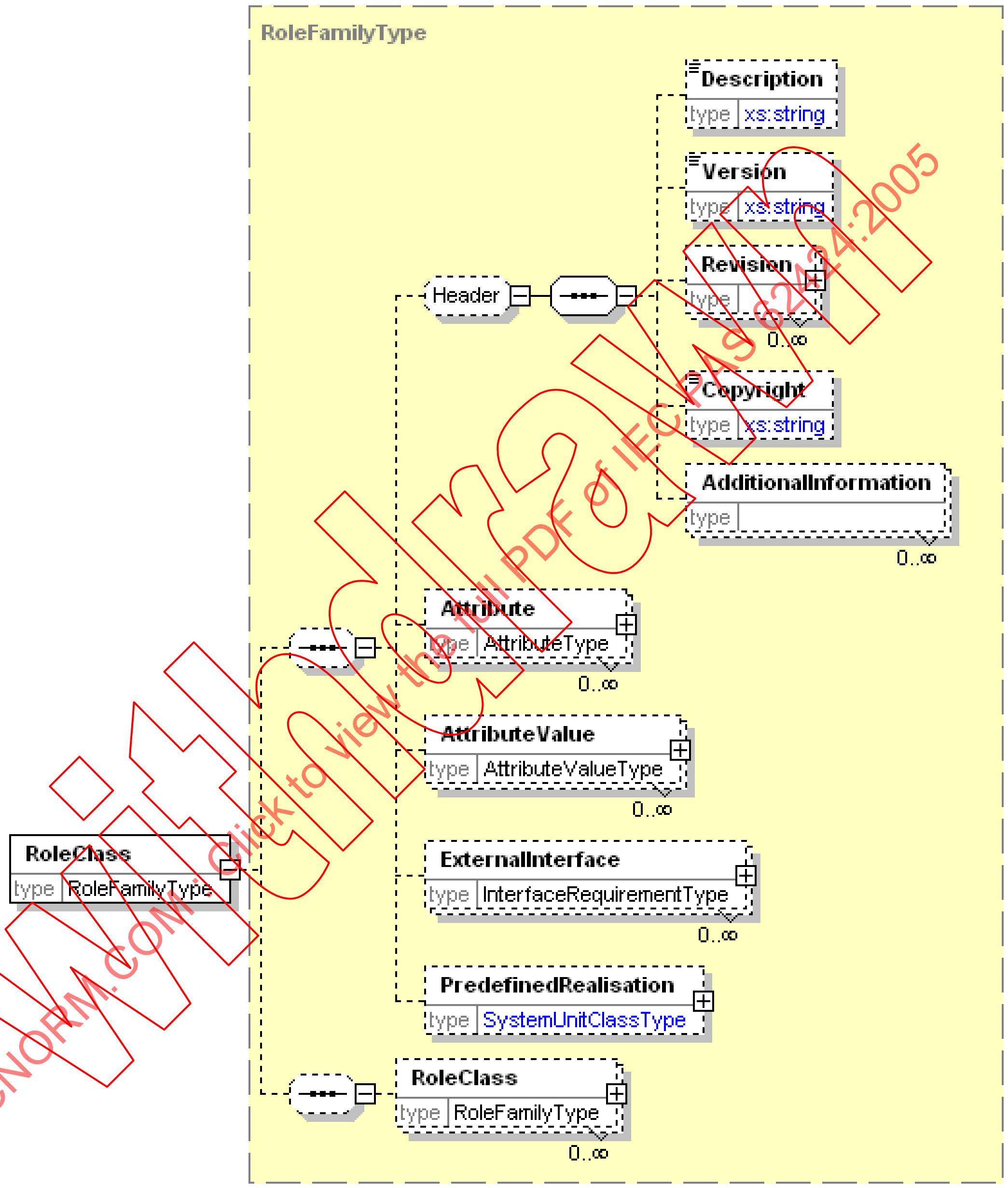
source

```
<xs:complexType name="RoleClassLibType">
 <xs:sequence minOccurs="0">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:element name="RoleClass" type="RoleFamilyType" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
```

	`<xs:attribute name="GlobalRoleLibName" type="xs:string" use="required"/>`
	`<xs:attribute name="PropertyString" type="propertystring" use="optional"/>`
	`<xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>`
	`</xs:complexType>`

A.12.2 Element RoleClassLibType/RoleClass

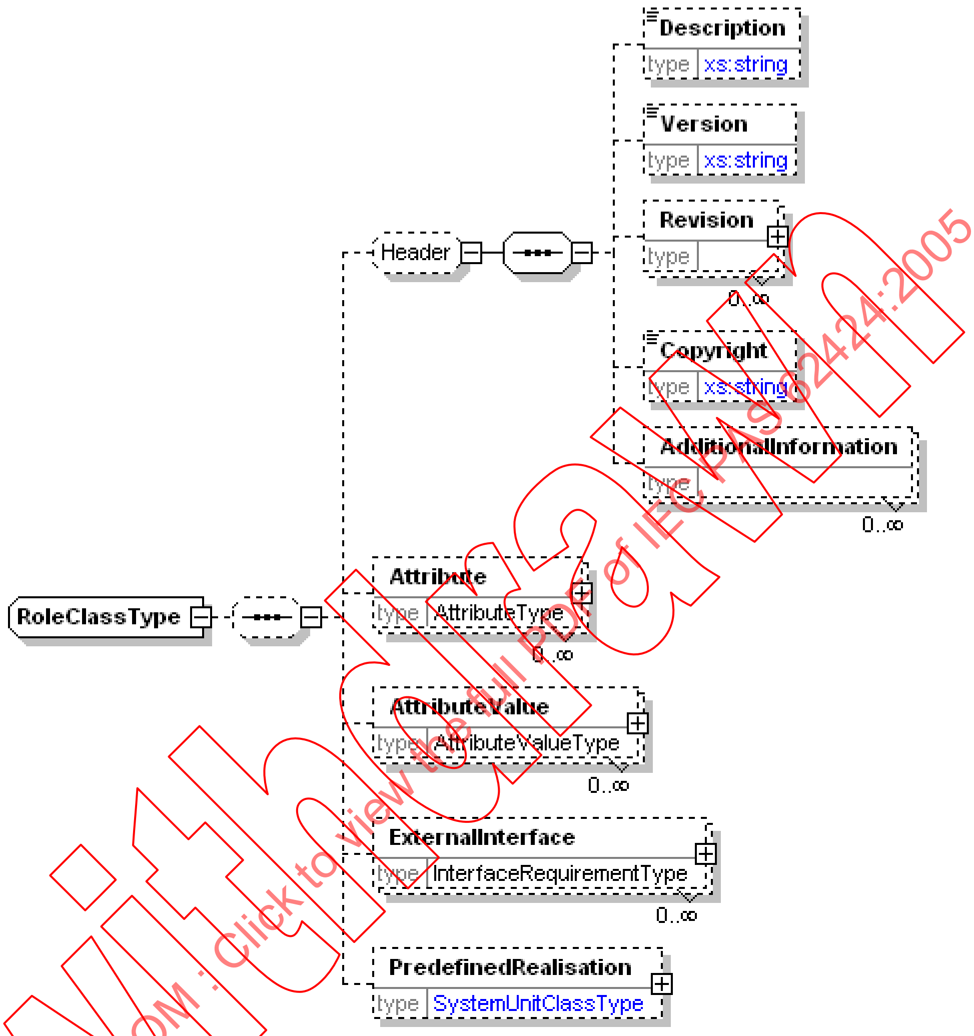
The specification of roles may be stored in the element “RoleClass”. Each role class shall contain a field “RoleClassName” which shall be uniqe in the structure of the role library.

diagram	 RoleFamilyType Description type xs:string Version type xs:string Revision type 0..∞ Copyright type xs:string AdditionalInformation type 0..∞ Header Attribute type AttributeType 0..∞ AttributeValue type AttributeValueType 0..∞ ExternalInterface type InterfaceRequirementType 0..∞ PredefinedRealisation type SystemUnitClassType RoleClass type RoleFamilyType 0..∞ RoleClass type RoleFamilyType
type	**RoleFamilyType**
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface PredefinedRealisation RoleClass**
attributes	Name Type Use Default Fixed Annotation RoleClassName xs:string required PropertyString propertystring optional changeMode changeMode optional state
source	`<xs:element name="RoleClass" type="RoleFamilyType" minOccurs="0" maxOccurs="unbounded"/>`

A.13 ComplexType RoleClassType

A.13.1 General

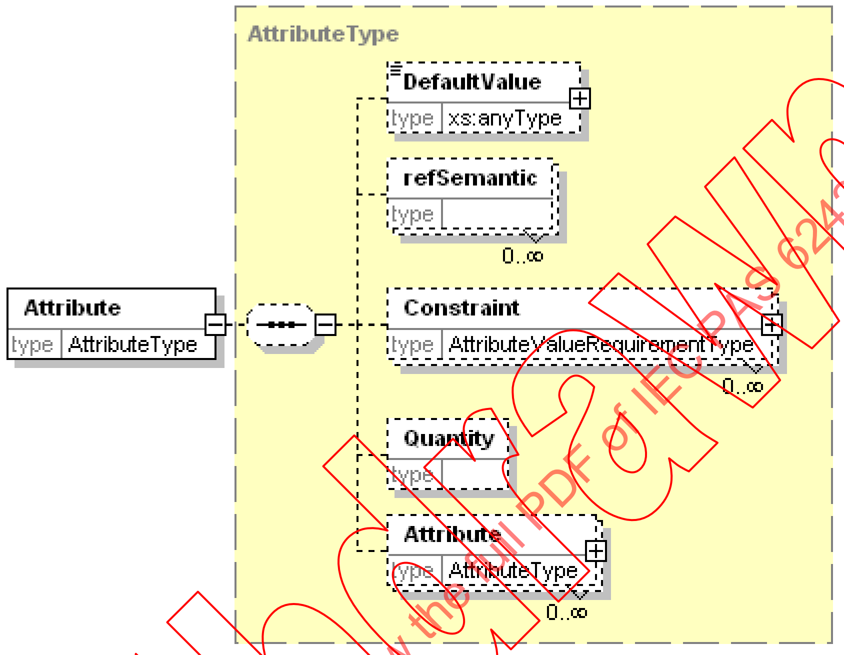
The datatype “RoleClassType” defines structures for the specification of role classes. Each role shall have an unique identifier stored in the field “RoleClassName”.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface PredefinedRealisation</b></td></tr>
<tr><td>used by</td><td>complexType <b>RoleFamilyType</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>RoleClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><pre>
<xs:complexType name="RoleClassType">
 <xs:sequence minOccurs="0">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="ExternalInterface" type="InterfaceRequirementType" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="PredefinedRealisation" minOccurs="0">
   <xs:complexType>
    <xs:complexContent>
     <xs:extension base="SystemUnitClassType">
      <xs:attribute name="GlobalSystemUnitClassLibName" type="xs:string" use="optional"/>
      <xs:attribute name="refBaseSystemUnitClass" type="xs:string" use="optional"/>
      <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
      <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
     </xs:extension>
    </xs:complexContent>
</pre></td></tr>
</table>

	</xs:complexType> </xs:element> </xs:sequence> <xs:attribute name="RoleClassName" type="xs:string" use="required"/> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:complexType>

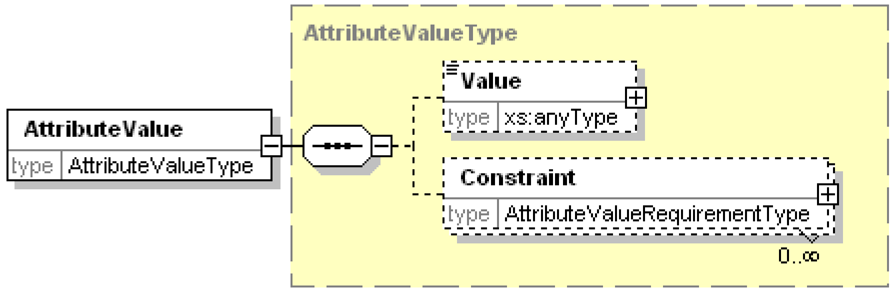
A.13.2 Element RoleClassType/Attribute

The element “Attribute” is of the datatype “AttributeType”. For a description of this datatype, see Clause A.4.2.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/&gt;</td></tr>
</table>

A.13.3 Element RoleClassType/AttributeValue

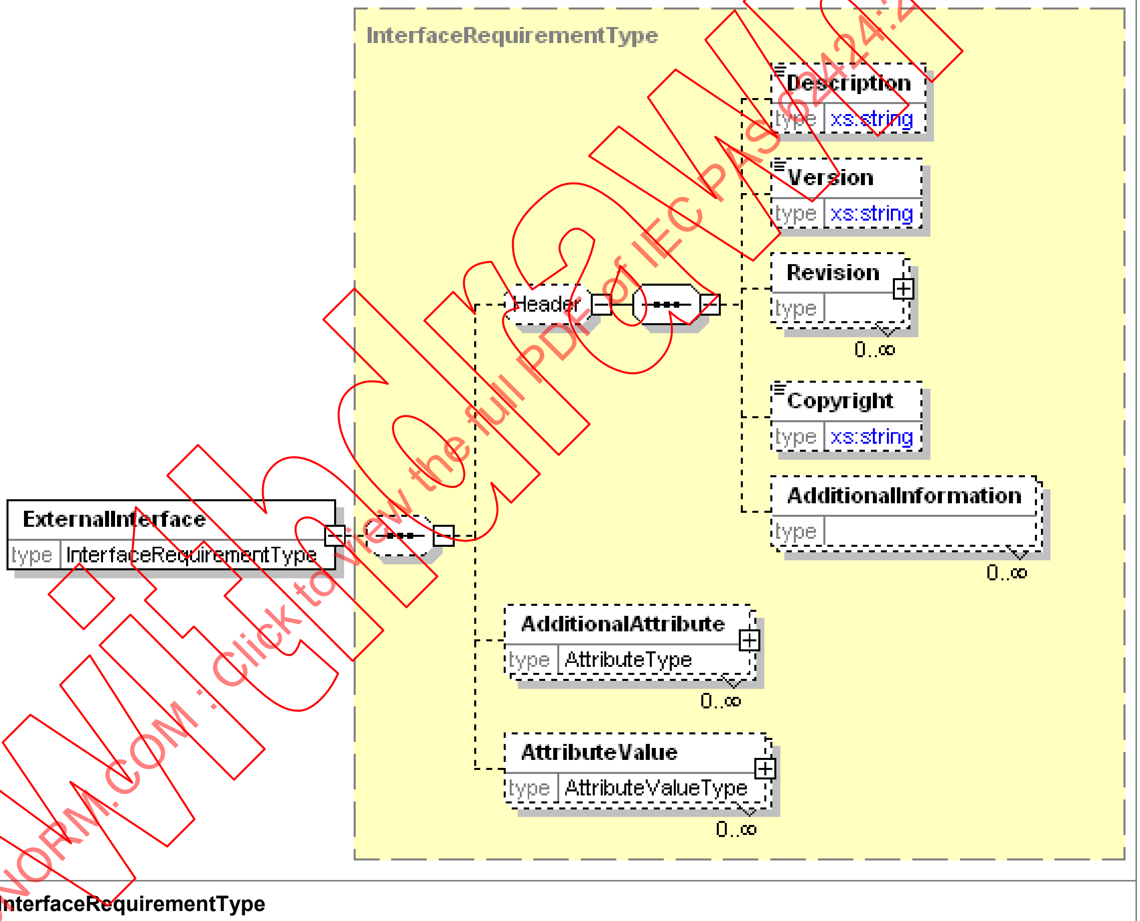
The element “AttributeValue” is of the datatype “AttributeValueType”. For a description of this datatype, see Clause A.6.

diagram	 AttributeValueType Value type xs:anyType AttributeValue type AttributeValueType Constraint type AttributeValueRequirementType 0..∞
type	**AttributeValueType**

<table>
<tr><td>children</td><td><b>Value Constraint</b></td></tr>
<tr><td>attributes</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>refAttribute</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>source</td><td><xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.13.4 Element RoleClassType/ExternalInterface

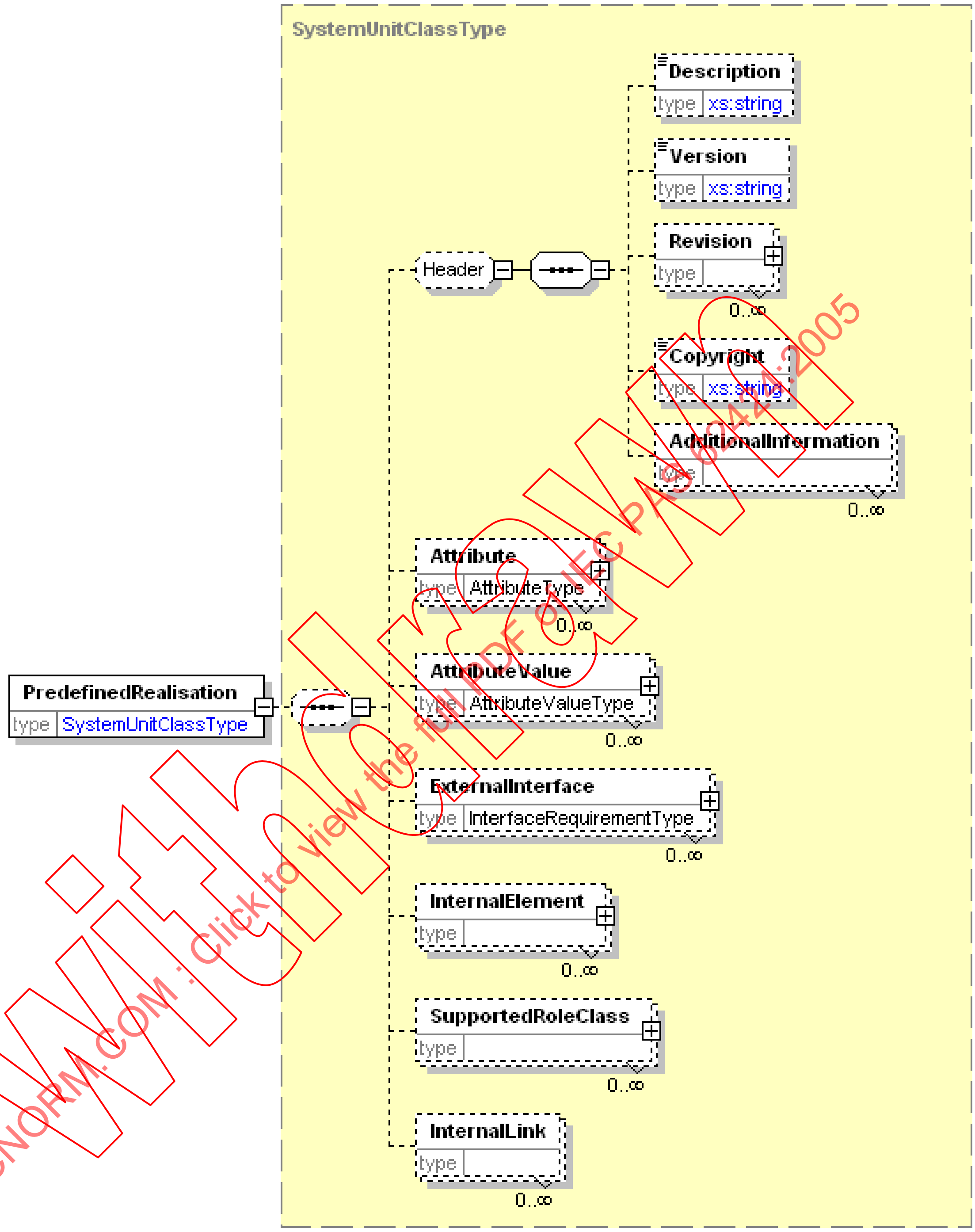
The element “ExternalInterface” may specify external interfaces of roles. The element is of the datatype “InterfaceRequirementType”. For a description of this datatype, see Clause A.10. The element shall have a field “LocalInterfaceName”. Each interface shall have an unique interface identifier. Existing interfaces may be referenced by the fields “refInterfaceClass” and “GlobalInterfaceClassLibName” when these interface classes are defined.

<table>
<tr><td>diagram</td><td>


</td></tr>
<tr><td>type</td><td><b>InterfaceRequirementType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation AdditionalAttribute AttributeValue</b></td></tr>
<tr><td>attributes</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>LocalInterfaceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>GlobalInterfaceClassLibName</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>refInterfaceClass</td><td>reference</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>source</td><td><xs:element name="ExternalInterface" type="InterfaceRequirementType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.13.5 Element RoleClassType/PredefinedRealisation

The element “PredefinedRealisation” may assign a certain class definition of a system unit to a role specification. Therefore, it contains the field “GlobalSystemUnitLibName” to reference the library where the class is defined. The field “refBaseSystemUnitClass” may reference to an existing system unit class. The referenced system unit class definition may be enhanced by a detailed specification of the datatype “SystemUnitClassType”. The element “PredefinedRealisation” is of the datatype “SystemUnitClassType”. For a description of this datatype, see Clause A.18.

diagram	 SystemUnitClassType Description type xs:string Version type xs:string Revision type 0..∞ Header Copyright type xs:string AdditionalInformation type 0..∞ Attribute type AttributeType 0..∞ PredefinedRealisation type SystemUnitClassType AttributeValue type AttributeValueType 0..∞ ExternalInterface type InterfaceRequirementType 0..∞ InternalElement type 0..∞ SupportedRoleClass type 0..∞ InternalLink type 0..∞
type	extension of **SystemUnitClassType**
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink**
attributes	Name / Type / Use / Default / Fixed / Annotation GlobalSystemUnitClassLibName / xs:string / optional refBaseSystemUnitClass / xs:string / optional PropertyString / propertystring / optional changeMode / changeMode / optional / state

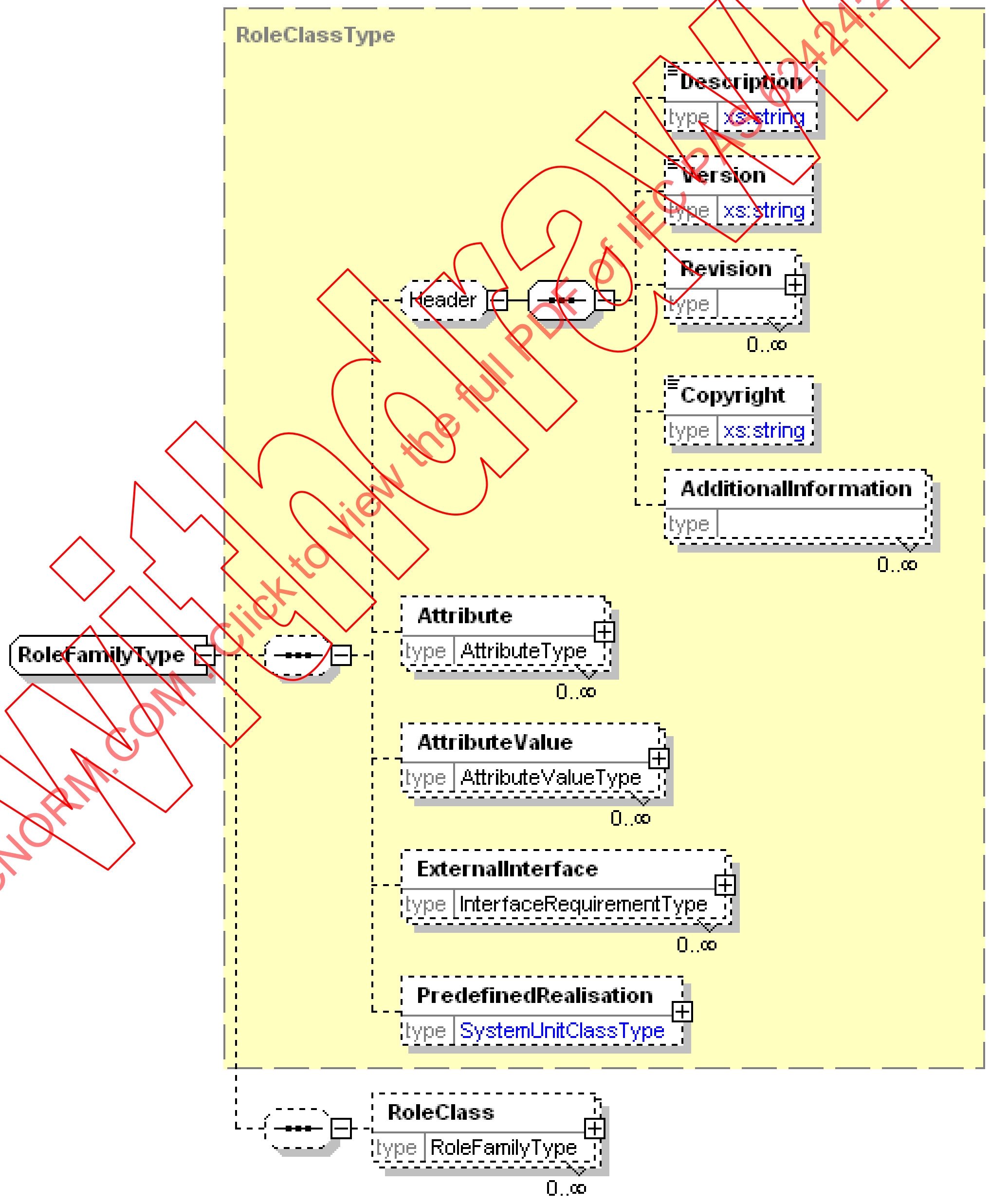
source	<xs:element name="PredefinedRealisation" minOccurs="0">
	  <xs:complexType>
	    <xs:complexContent>
	      <xs:extension base="SystemUnitClassType">
	        <xs:attribute name="GlobalSystemUnitClassLibName" type="xs:string" use="optional"/>

```
        <xs:attribute name="refBaseSystemUnitClass" type="xs:string" use="optional"/>
        <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
        <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

A.14 ComplexType RoleFamilyType

A.14.1 General

The datatype “RoleFamilyType” is an extension of the datatype “RoleClassType”. For a description of this datatype see A.13. The datatype “RoleFamilyType” allows the creation of child elements of role classes. Child classes contain all characteristics (attribute elements) of the base class but may be extended by new definitions of characteristics (attribute elements). Derived classes may restrict the value of predefined characteristics (attribute elements) by means of constraints.

diagram	
type	extension of **RoleClassType**
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface PredefinedRealisation RoleClass**

used by	elements	RoleFamilyType/RoleClass RoleClassLibType/RoleClass				
attributes	Name	Type	Use	Default	Fixed	Annotation
	RoleClassName	xs:string	required			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		

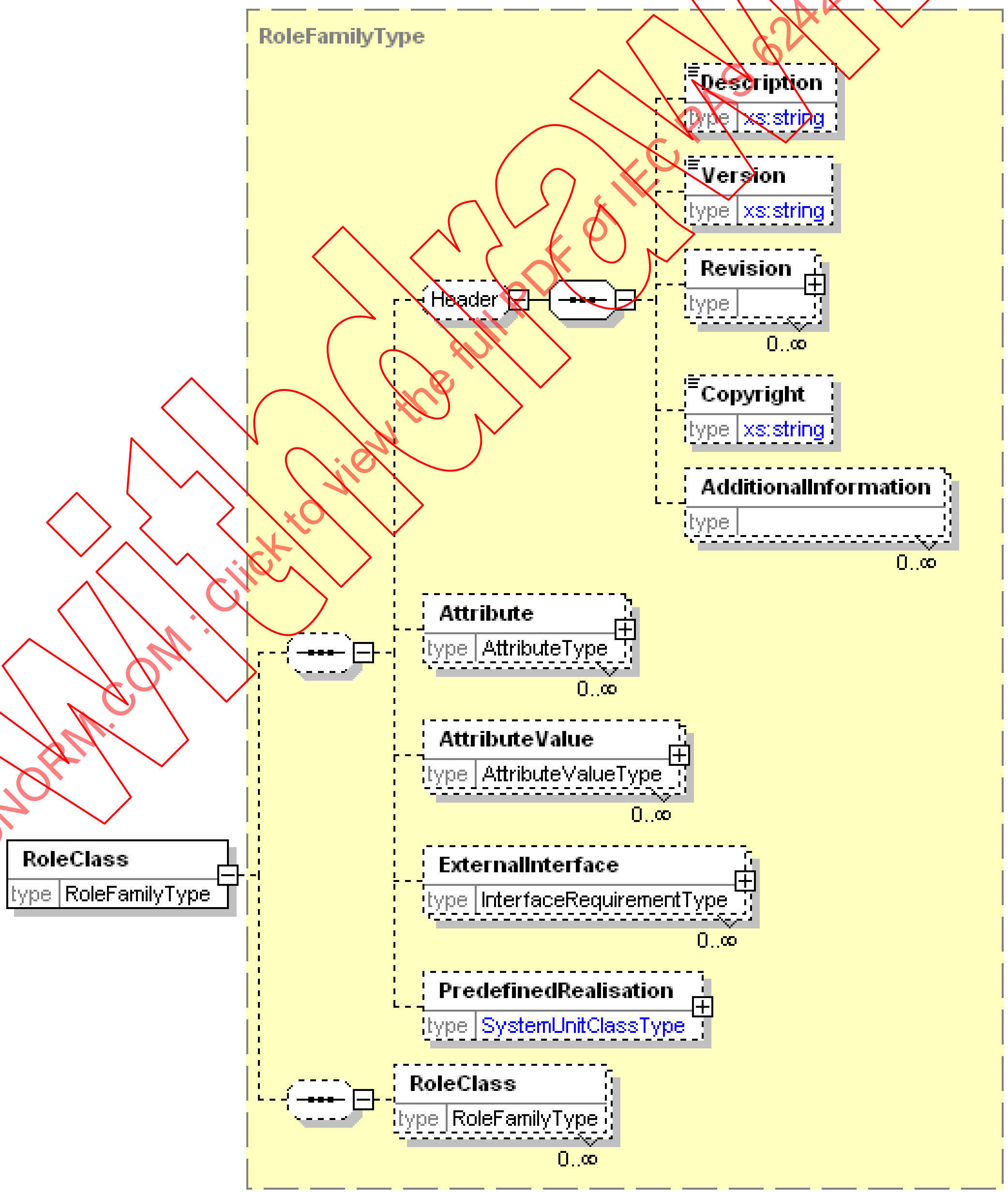
source

```
<xs:complexType name="RoleFamilyType">
 <xs:complexContent>
  <xs:extension base="RoleClassType">
   <xs:sequence minOccurs="0">
    <xs:element name="RoleClass" type="RoleFamilyType" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

A.14.2 Element RoleFamilyType/RoleClass

The element “RoleClass” defines a derived class of a base role class. The element is of the datatype “RoleFamilyType”. For a description of this datatype. see Clause A.14.

diagram



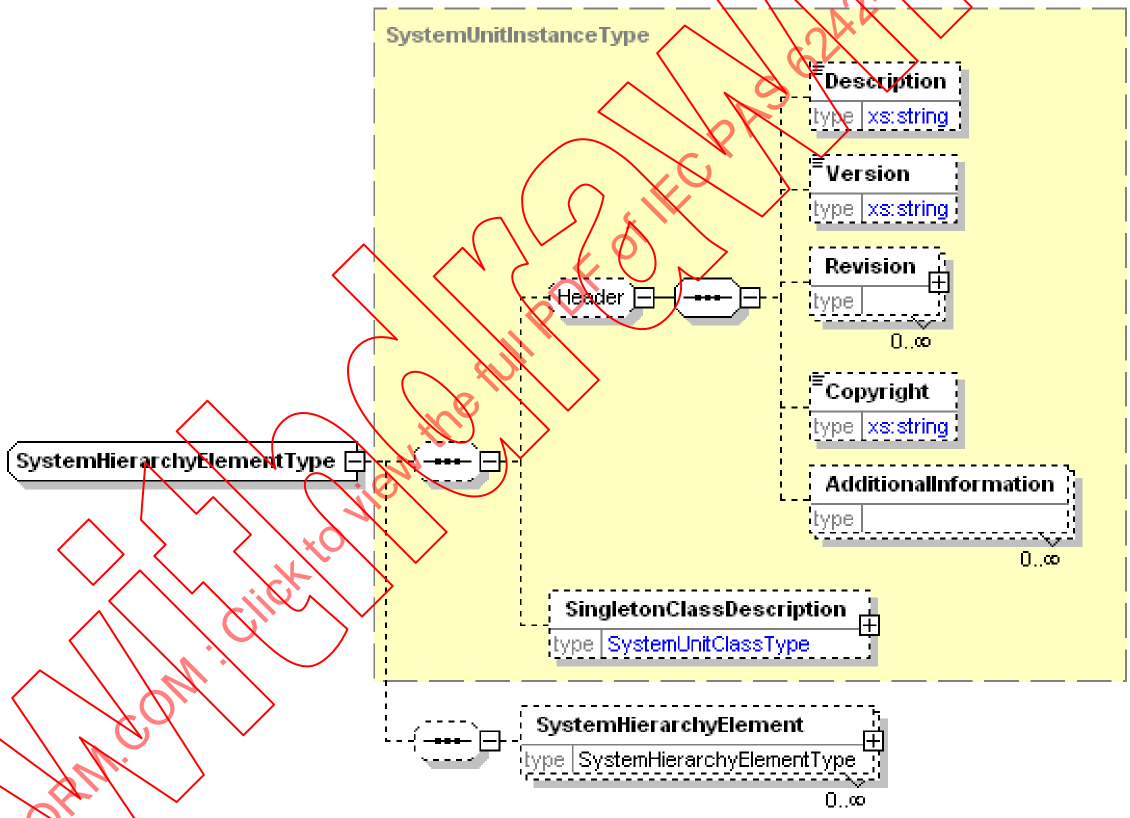
type	**RoleFamilyType**

<table>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface PredefinedRealisation RoleClass</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>RoleClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><pre>
<xs:element name="RoleClass" type="RoleFamilyType" minOccurs="0" maxOccurs="unbounded"/>
</pre></td></tr>
</table>

A.15 ComplexType SystemHierarchyElementType

A.15.1 General

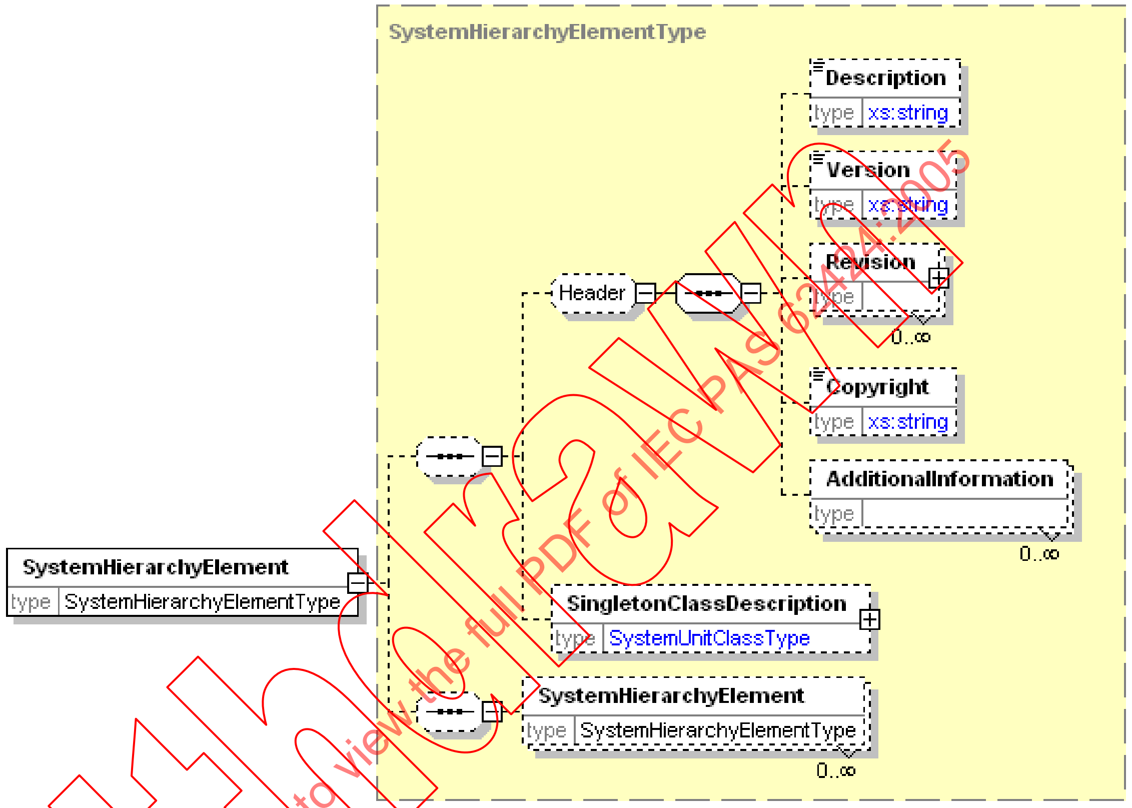
The datatype “SystemHierarchyElementType” allows the definition of hierarchical structures. This datatype is an extension of the datatype “SystemUnitInstanceType”. For a description of this datatype, see Clause A.20. The subelement “SystemHierarchyElement” allows a recursive definition of plant hierarchies. Each “SystemHierarchyElementType” shall have a field “SystemUnitInstanceName” to store the identifier of the hierarchy.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>extension of <b>SystemUnitInstanceType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SingletonClassDescription SystemHierarchyElement</b></td></tr>
<tr><td>used by</td><td><table>
<tr><td>elements</td><td><b>SystemHierarchyElementType/SystemHierarchyElement</b>
<b>SystemHierarchyLibType/SystemHierarchyElement</b></td></tr>
</table></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>SystemUnitInstanceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><pre>
<xs:complexType name="SystemHierarchyElementType">
  <xs:complexContent>
    <xs:extension base="SystemUnitInstanceType">
      <xs:sequence minOccurs="0">
        <xs:element name="SystemHierarchyElement" type="SystemHierarchyElementType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
</pre></td></tr>
</table>

	</xs:complexContent> </xs:complexType>

A.15.2 Element SystemHierarchyElementType/SystemHierarchyElement

The element “SystemHierarchyElement” is of the datatype “SystemHierarchyElementType”. For a description of this datatype, see Clause A.15.

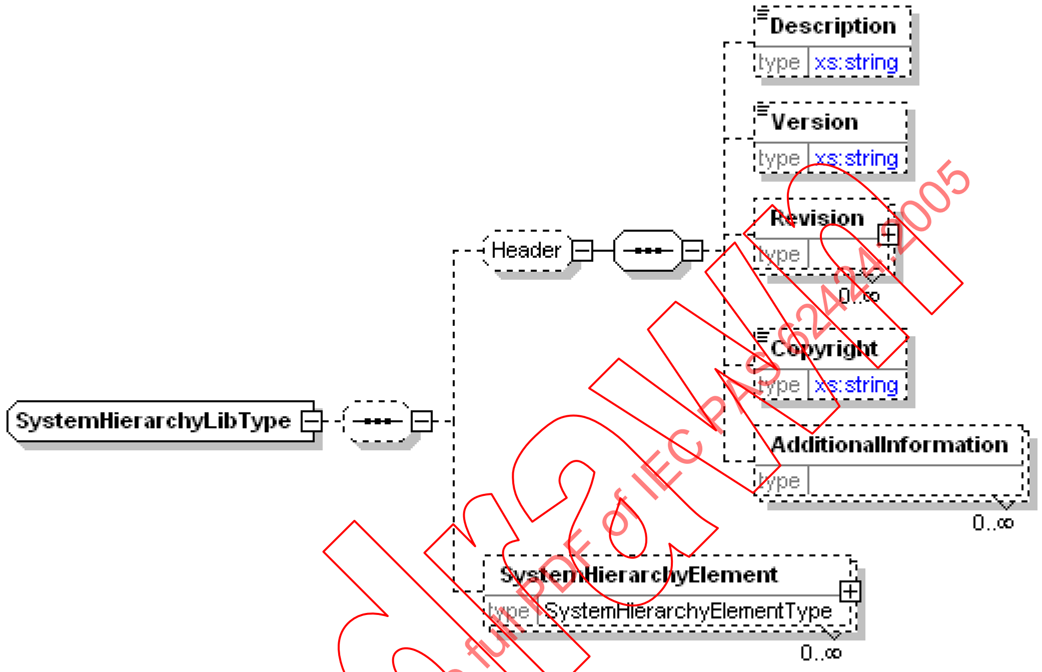
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>SystemHierarchyElementType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SingletonClassDescription SystemHierarchyElement</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>SystemUnitInstanceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="SystemHierarchyElement" type="SystemHierarchyElementType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.16 ComplexType SystemHierarchyLibType

A.16.1 General

The datatype “SystemHierarchyLibType” allows for the definition of plant hierarchies. The name of the base hierarchy shall be stored in the field “GlobalSystemHierarchyName”. Underlaying hierarchies may be defined by the subelement “SystemHierarchyElement”.

diagram



children: **Description Version Revision Copyright AdditionalInformation SystemHierarchyElement**

used by: element **CAEXFile/SystemHierarchy**

attributes

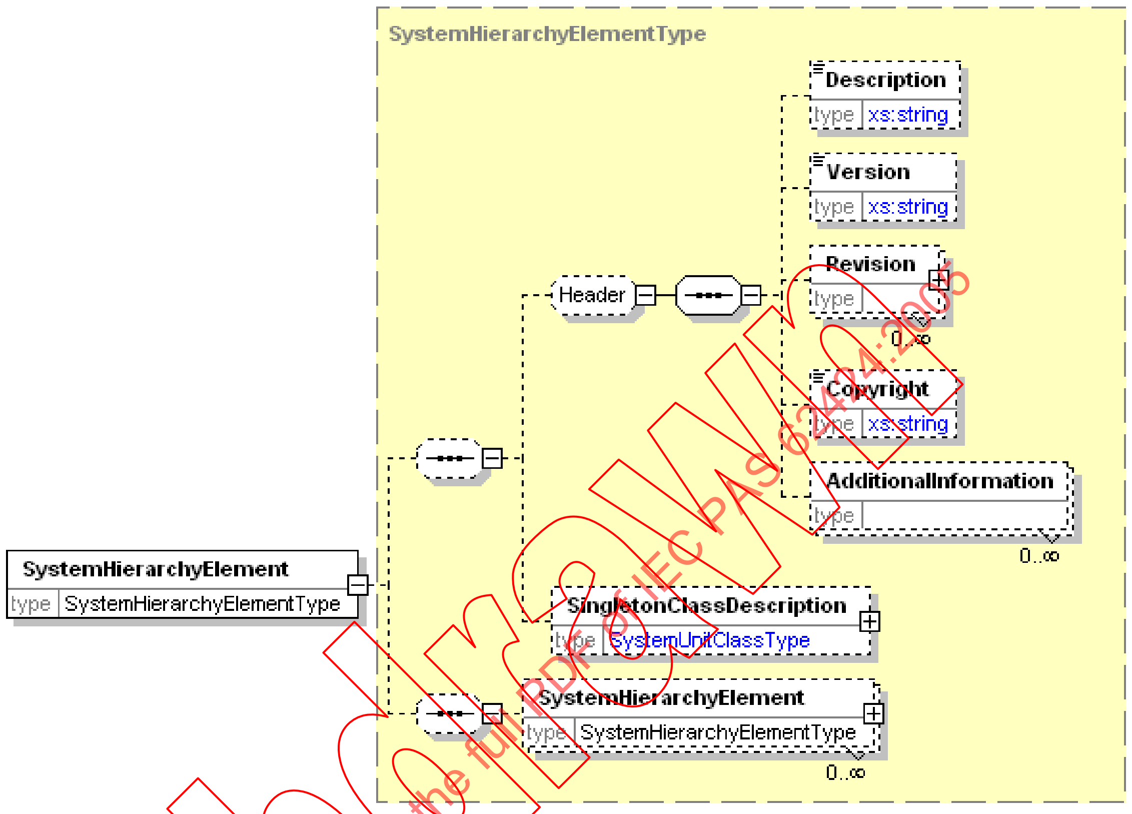
Name	Type	Use	Default	Fixed	Annotation
GlobalSystemHierarchyName	xs:string	required			
description	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:complexType name="SystemHierarchyLibType">
 <xs:sequence minOccurs="0">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:element name="SystemHierarchyElement" type="SystemHierarchyElementType" minOccurs="0"
maxOccurs="unbounded"/>
 </xs:sequence>
 <xs:attribute name="GlobalSystemHierarchyName" type="xs:string" use="required"/>
 <xs:attribute name="description" type="xs:string" use="optional"/>
 <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
 <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType>
```

A.16.2 Element SystemHierarchyLibType/SystemHierarchyElement

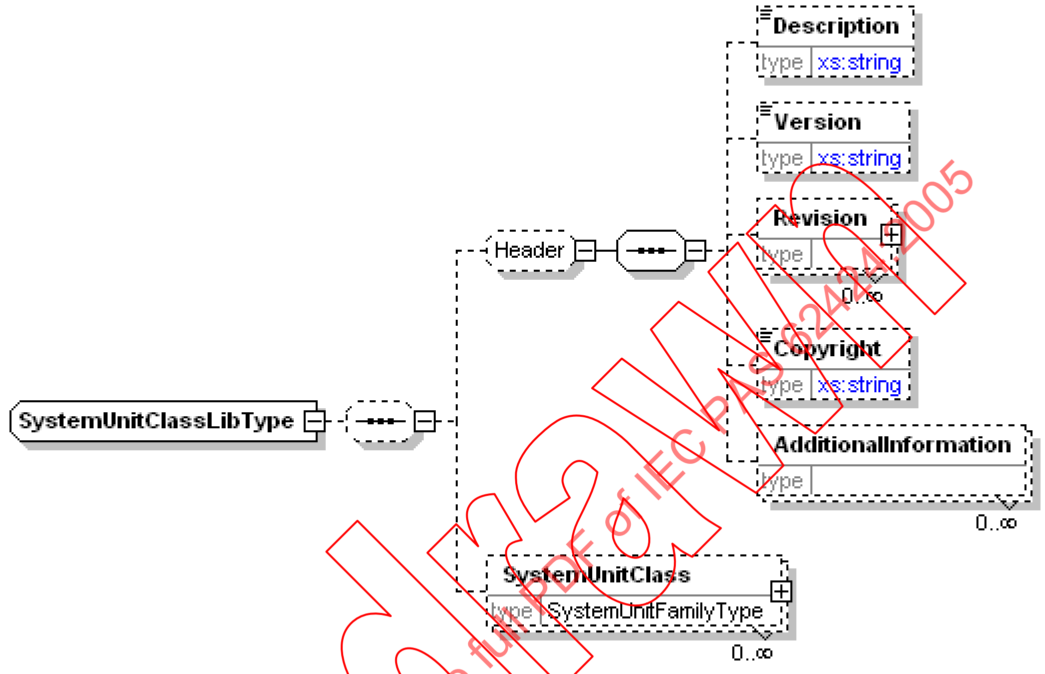
The element “SystemHierarchyElement” is of the datatype “SystemHierarchyElementType”. For a description of this datatype, see Clause A.15.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>SystemHierarchyElementType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SingletonClassDescription SystemHierarchyElement</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>SystemUnitInstanceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:element name="SystemHierarchyElement" type="SystemHierarchyElementType" minOccurs="0" maxOccurs="unbounded"/&gt;</td></tr>
</table>

A.17 ComplexType SystemUnitClassLibType

A.17.1 General

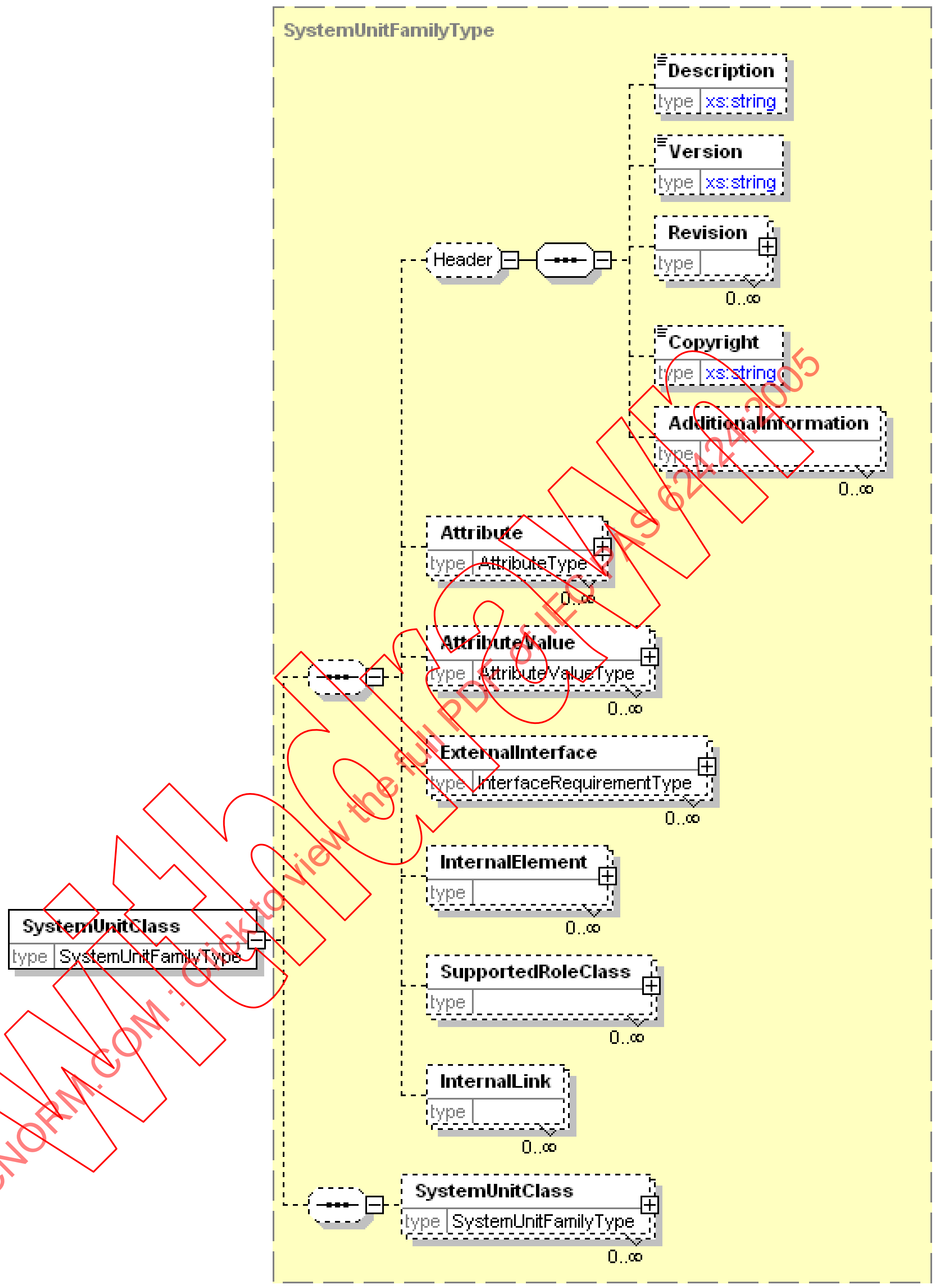
The datatype “SystemUnitClassLibType” specifies the structure of system unit libraries. System unit libraries may contain multiple system unit class definitions. Each system unit library shall have an unique library name stored in the field “GlobalSystemUnitLibName”.

<table>
<tr><td>diagram</td><td>
Description
type xs:string
Version
type xs:string
Revision
type
0..∞
Header
Copyright
type xs:string
AdditionalInformation
type
0..∞
SystemUnitClassLibType
SystemUnitClass
type SystemUnitFamilyType
0..∞

</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SystemUnitClass</b></td></tr>
<tr><td>used by</td><td>element <b>CAEXFile/SystemUnitClassLib</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>GlobalSystemUnitLibName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:complexType name="SystemUnitClassLibType">
 <xs:sequence minOccurs="0">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:element name="SystemUnitClass" type="SystemUnitFamilyType" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 <xs:attribute name="GlobalSystemUnitLibName" type="xs:string" use="required"/>
 <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
 <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
</xs:complexType&gt;</td></tr>
</table>

A.17.2 Element SystemUnitClassLibType/SystemUnitClass

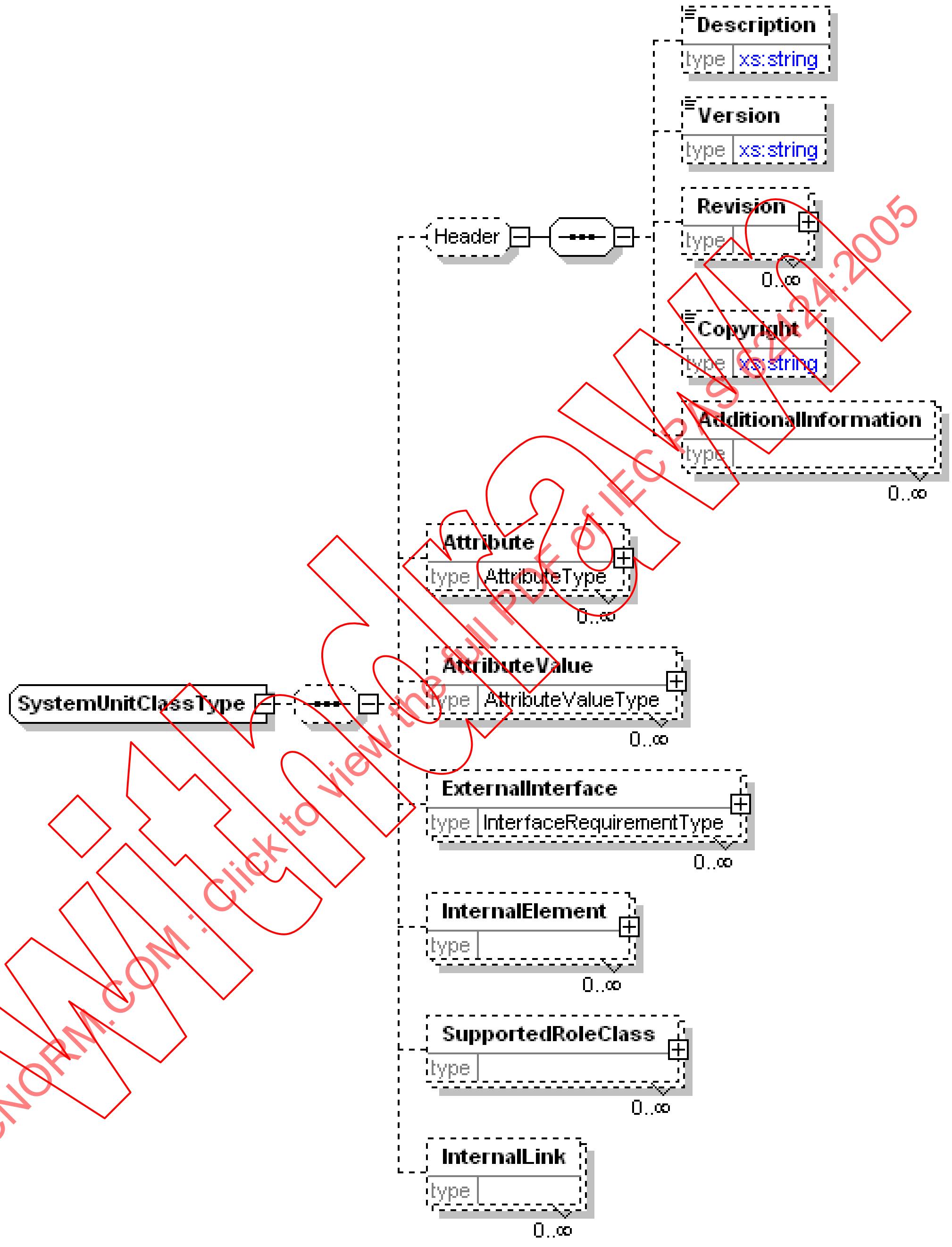
The specification of a system unit may be stored in the element “SystemUnitClass”. Each system unit class shall contain a field “SystemUnitClassName” which shall be unique in the structure of the system unit library.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>SystemUnitFamilyType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink SystemUnitClass</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>SystemUnitClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:element name="SystemUnitClass" type="SystemUnitFamilyType" minOccurs="0" maxOccurs="unbounded"/&gt;</td></tr>
</table>

A.18 ComplexType SystemUnitClassType

A.18.1 General

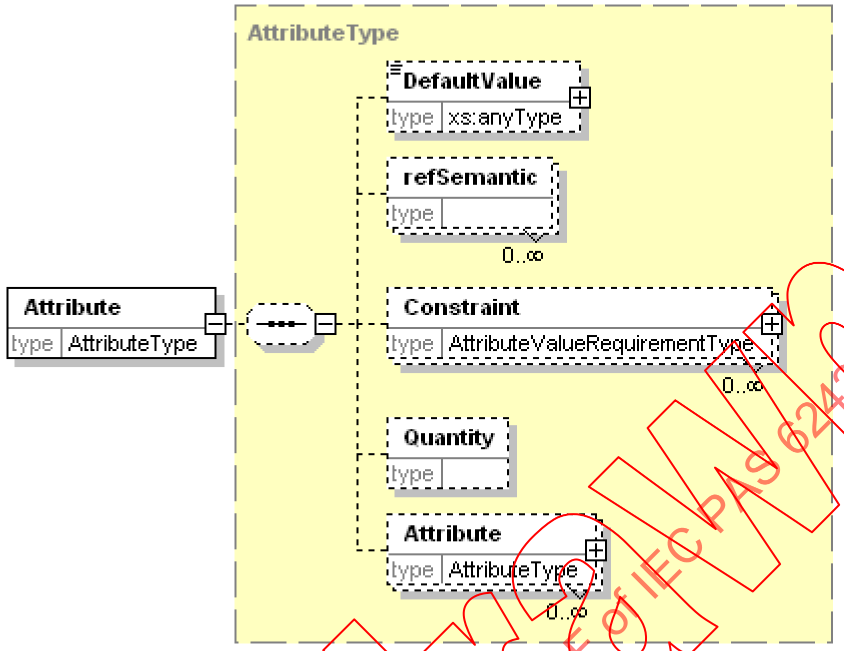
The datatype “SystemUnitClassType” defines structures for the specification of system unit classes. Each system unit requires a unique identifier stored in the field “RoleClassName”.

diagram	 Description type xs:string Version type xs:string Revision type 0..∞ Copyright type xs:string AdditionalInformation type 0..∞ Header Attribute type AttributeType 0..∞ AttributeValue type AttributeValueType 0..∞ SystemUnitClassType ExternalInterface type InterfaceRequirementType 0..∞ InternalElement type 0..∞ SupportedRoleClass type 0..∞ InternalLink type 0..∞
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink**
used by	elements **RoleClassType/PredefinedRealisation** **SystemUnitClassType/InternalElement/PredefinedRealisation** **SystemUnitInstanceType/SingletonClassDescription** complexType **SystemUnitFamilyType**
source	<xs:complexType name="SystemUnitClassType"> <xs:sequence minOccurs="0"> <xs:group ref="Header" minOccurs="0"/> <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="ExternalInterface" type="InterfaceRequirementType" minOccurs="0" maxOccurs="unbounded"/>

```
  <xs:element name="InternalElement" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:sequence minOccurs="0">
     <xs:element name="RoleRequirements" minOccurs="0">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="AdditionalAttributeValue" type="AttributeValueType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="AdditionalExternalInterface" type="InterfaceRequirementType" minOccurs="0"
maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="RoleClassLibName" type="xs:string" use="required"/>
       <xs:attribute name="refRoleClass" type="reference" use="required"/>
       <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
       <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
      </xs:complexType>
     </xs:element>
     <xs:element name="PredefinedRealisation" minOccurs="0">
      <xs:complexType>
       <xs:complexContent>
        <xs:extension base="SystemUnitClassType">
         <xs:attribute name="GlobalSystemUnitClassLibName" type="xs:string" use="optional"/>
         <xs:attribute name="refBaseSystemUnitClass" type="reference" use="optional"/>
         <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
         <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
        </xs:extension>
       </xs:complexContent>
      </xs:complexType>
     </xs:element>
     <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="localElementName" type="xs:string" use="required"/>
    <xs:attribute name="description" type="xs:string" use="optional"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="SupportedRoleClass" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:sequence minOccurs="0">
     <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="GlobalRoleLibName" type="xs:string" use="required"/>
    <xs:attribute name="refRoleClass" type="reference" use="required"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="InternalLink" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
    <xs:attribute name="localLinkName" type="xs:string" use="required"/>
    <xs:attribute name="refPartnerSideA" type="reference" use="optional"/>
    <xs:attribute name="refPartnerSideB" type="reference" use="optional"/>
    <xs:attribute name="description" type="xs:string" use="optional"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

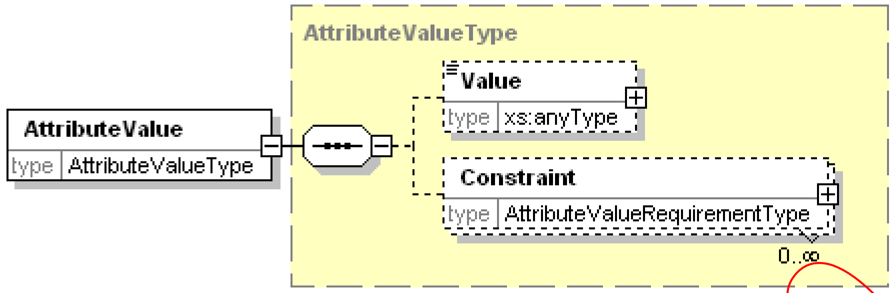
A.18.2 Element SystemUnitClassType/Attribute

The element “Attribute” is of the datatype “AttributeType”. For a description of this datatype, see Clause A.4.2.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

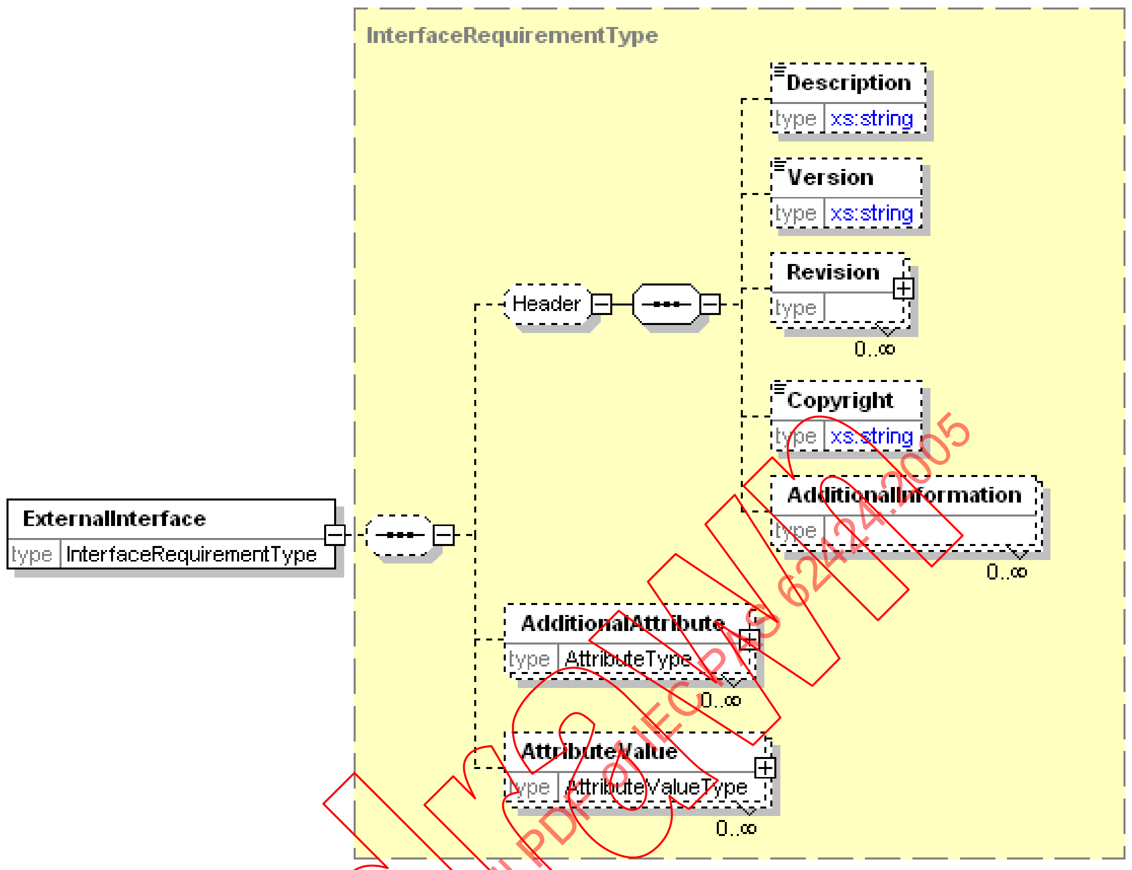
A.18.3 Element SystemUnitClassType/AttributeValue

The element “AttributeValue” is of the datatype “AttributeValueType”. For a description of this datatype, see Clause A.6.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeValueType</b></td></tr>
<tr><td>children</td><td><b>Value Constraint</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>refAttribute</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>&lt;xs:element name="AttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/&gt;</td></tr>
</table>

A.18.4 Element SystemUnitClassType/ExternalInterface

The element “ExternalInterface” may specify external interfaces of system units. The element is of the datatype “InterfaceRequirementType”. For a description of this datatype, see Clause A.10. The element shall have a field “LocalInterfaceName”. Each interface shall have an unique interface identifier. Existing interfaces may be referenced by the fields “refInterfaceClass” and “GlobalInterfaceClassLibName” when these interface classes are defined.

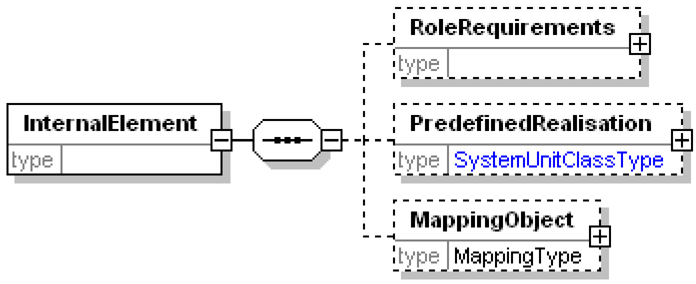
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>InterfaceRequirementType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation AdditionalAttribute AttributeValue</b></td></tr>
<tr><td>attributes</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr>
<tr><td>LocalInterfaceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>GlobalInterfaceClassLibName</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>refInterfaceClass</td><td>reference</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>source</td><td><xs:element name="ExternalInterface" type="InterfaceRequirementType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.18.5 Element SystemUnitClassType/InternalElement

The element “InternalElement” allows for the definition of the internal structure of a system unit. It shall have the attribute “LocalElementName” to store the identifier of the element. The description of the element may be stored in the attribute “description”.

The “InternalElement” may have a reference to a predefined, pre-existing unit, which may be realized by the element “PredefinedRealisation”. Requirements to the defined internal elements can be made by the definition of the element “RoleRequirements”. The child element “MappingObject” allows for the mapping of user defined names to predefined names in class definitions.

diagram



children **RoleRequirements PredefinedRealisation MappingObject**

attributes

Name	Type	Use	Default	Fixed	Annotation
localElementName	xs:string	required			
description	xs:string	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

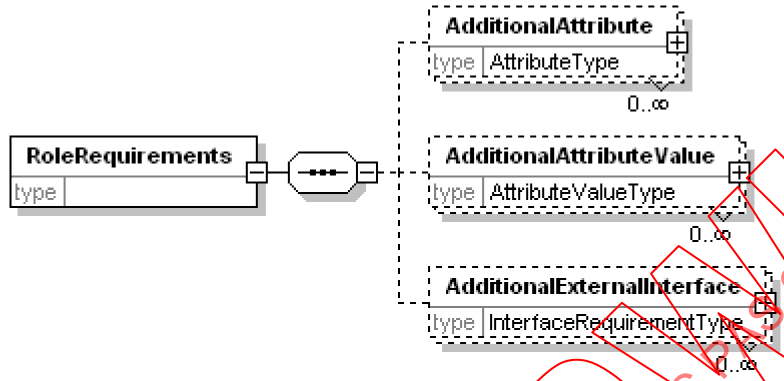
```
<xs:element name="InternalElement" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="RoleRequirements" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="AdditionalAttributeValue" type="AttributeValueType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="AdditionalExternalInterface" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="InterfaceClassType">
                    <xs:attribute name="LocalInterfaceName" type="xs:string" use="required"/>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="RoleClassLibName" type="xs:string" use="required"/>
          <xs:attribute name="refRoleClass" type="reference" use="required"/>
          <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
          <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="PredefinedRealisation" minOccurs="0">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="SystemUnitClassType">
              <xs:attribute name="GlobalSystemUnitClassLibName" type="xs:string" use="optional"/>
              <xs:attribute name="refBaseSystemUnitClass" type="reference" use="optional"/>
              <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
              <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="localElementName" type="xs:string" use="required"/>
    <xs:attribute name="description" type="xs:string" use="optional"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
  </xs:complexType>
</xs:element>
```

A.18.6 Element SystemUnitClassType/InternalElement/RoleRequirements

The element “RoleRequirements” allows for the definition of requirements to a predefined role class definition. To reference the defined role class, it shall have the attributes “RoleClassLibName” and “refRoleClass”.

The sub elements “AdditionalAttribute”, “AdditionalAttributeValue” and “AdditionaExternalInterface” allow for the enhancement of the referenced role class definition by new attribute definitions, with new value requirements or new definitions of required external interfaces.

diagram



children

AdditionalAttribute AdditionalAttributeValue AdditionalExternalInterface

attributes

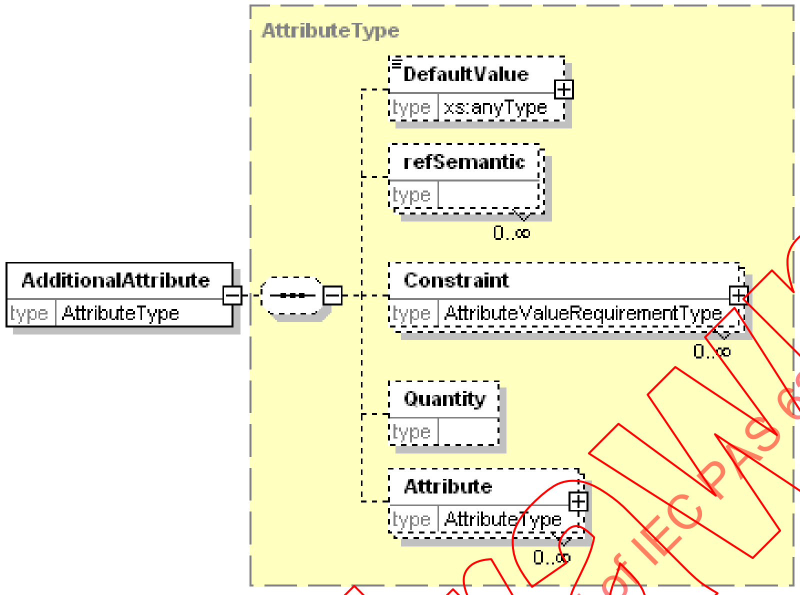
Name	Type	Use	Default	Fixed	Annotation
RoleClassLibName	xs:string	required			
refRoleClass	reference	required			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

source

```
<xs:element name="RoleRequirements" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="AdditionalAttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="AdditionalExternalInterface" type="InterfaceRequirementType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="RoleClassLibName" type="xs:string" use="required"/>
    <xs:attribute name="refRoleClass" type="reference" use="required"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
  </xs:complexType>
</xs:element>
```

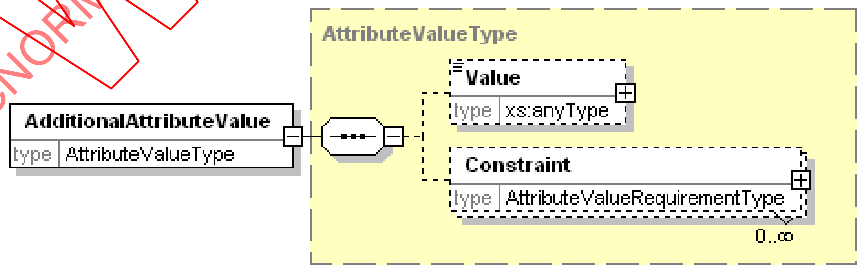
A.18.7 Element SystemUnitClassType/InternalElement/RoleRequirements/ AdditionalAttribute

The element “AdditionalAttribute” is of the datatype “AttributeType”. For a description of this datatype, see Clause A.4.2.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeType</b></td></tr>
<tr><td>children</td><td><b>DefaultValue refSemantic Constraint Quantity Attribute</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>LocalAttributeName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>attributeDataType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AdditionalAttribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

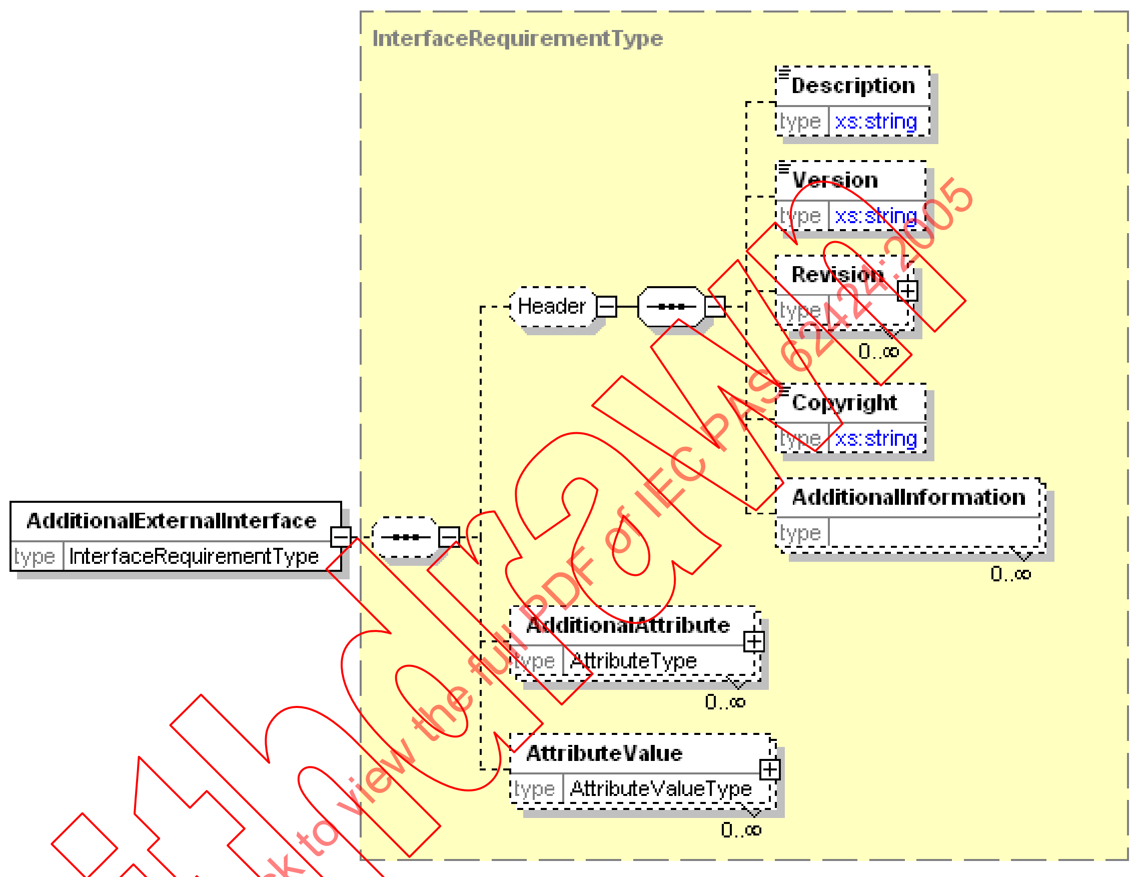
A.18.8 Element SystemUnitClassType/InternalElement/RoleRequirements/ AdditionalAttributeValue

The element “AttributeValue” is of the datatype “AttributeValueType”. For a description of this datatype, see Clause A.6.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>AttributeValueType</b></td></tr>
<tr><td>children</td><td><b>Value Constraint</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>refAttribute</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>dimension</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AdditionalAttributeValue" type="AttributeValueType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.18.9 Element SystemUnitClassType/InternalElement/RoleRequirements/ AdditionalExternalInterface

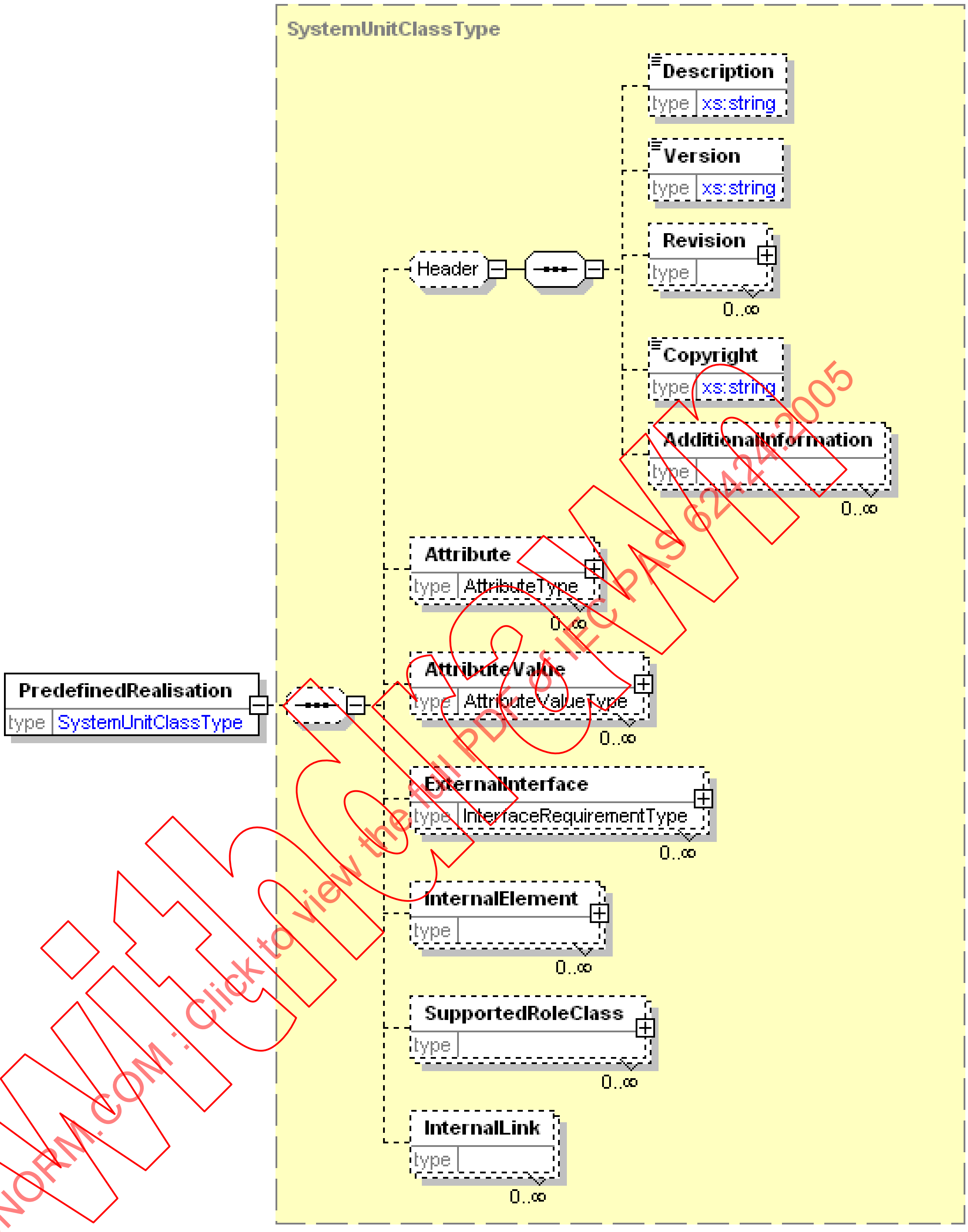
The element “AdditionalExternalInterface” is of the datatype “InterfaceRequirementType”. For a description of this datatype, see Clause A.10. Besides the attributes of the latter datatype, an element “additionalExternalInterface” shall have the attribute “LocalInterfaceName” to identify the specified new additional external interface.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>InterfaceRequirementType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation AdditionalAttribute AttributeValue</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>LocalInterfaceName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>GlobalInterfaceClassLibName</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>refInterfaceClass</td><td>reference</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>description</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="AdditionalExternalInterface" type="InterfaceRequirementType" minOccurs="0" maxOccurs="unbounded"/></td></tr>
</table>

A.18.10 Element SystemUnitClassType/InternalElement/PredefinedRealisation

The element “PredefinedRealisation” is of the datatype “SystemUnitClassType”. For a description of this datatype, see Clause A.18 The Element allows for the definition as well as for the enhancement of complex units and their assignation, for example, to internal elements (see Clause A.18.5).

diagram



type: extension of **SystemUnitClassType**

children: **Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink**

attributes

Name	Type	Use	Default	Fixed	Annotation
GlobalSystemUnitClassLibName	xs:string	optional			
refBaseSystemUnitClass	reference	optional			
PropertyString	propertystring	optional			
changeMode	changeMode	optional	state		

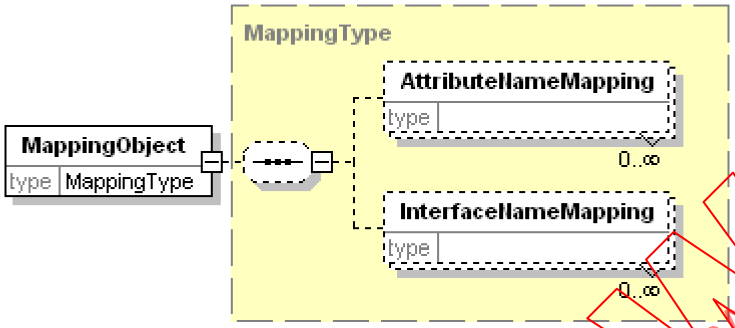
source

```
<xs:element name="PredefinedRealisation" minOccurs="0">
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="SystemUnitClassType">
    <xs:attribute name="GlobalSystemUnitClassLibName" type="xs:string" use="optional"/>
    <xs:attribute name="refBaseSystemUnitClass" type="reference" use="optional"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
```

	</xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

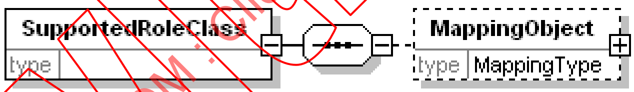
A.18.11 Element SystemUnitClassType/InternalElement/MappingObject

The element “MappingObject” is of the datatype “MappingType”. For a description of this datatype see A.11.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td><b>MappingType</b></td></tr>
<tr><td>children</td><td><b>AttributeNameMapping InterfaceNameMapping</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="MappingObject" type="MappingType" minOccurs="0"/></td></tr>
</table>

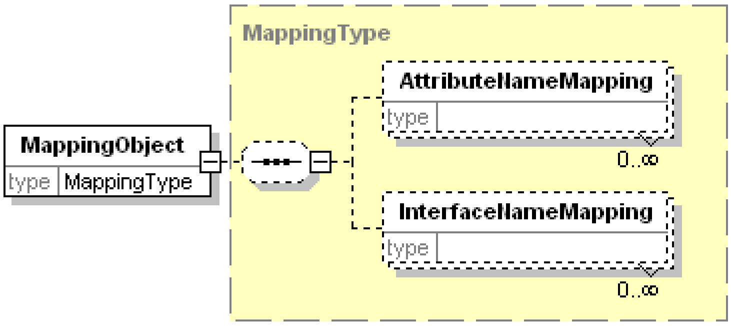
A.18.12 Element SystemUnitClassType/SupportedRoleClass

The element “SupportedRole” allows the assignment of predefined roles to the surrounding system unit class definition. To identify a role definition in a role library the element shall have the attributes “GlobalRoleLibName” and “refRoleClass”.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>children</td><td><b>MappingObject</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>GlobalRoleLibName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>refRoleClass</td><td>reference</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td><xs:element name="SupportedRoleClass" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="GlobalRoleLibName" type="xs:string" use="required"/>
    <xs:attribute name="refRoleClass" type="reference" use="required"/>
    <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
    <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
  </xs:complexType>
</xs:element></td></tr>
</table>

A.18.13 Element SystemUnitClassType/SupportedRoleClass/MappingObject

The element “MappingObject” is of the datatype “MappingType”. For a description of this datatype, see Clause A.11.

diagram						
type	**MappingType**					
children	**AttributeNameMapping InterfaceNameMapping**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="MappingObject" type="MappingType" minOccurs="0"/>					

A.18.14 Element SystemUnitClassType/InternalLink

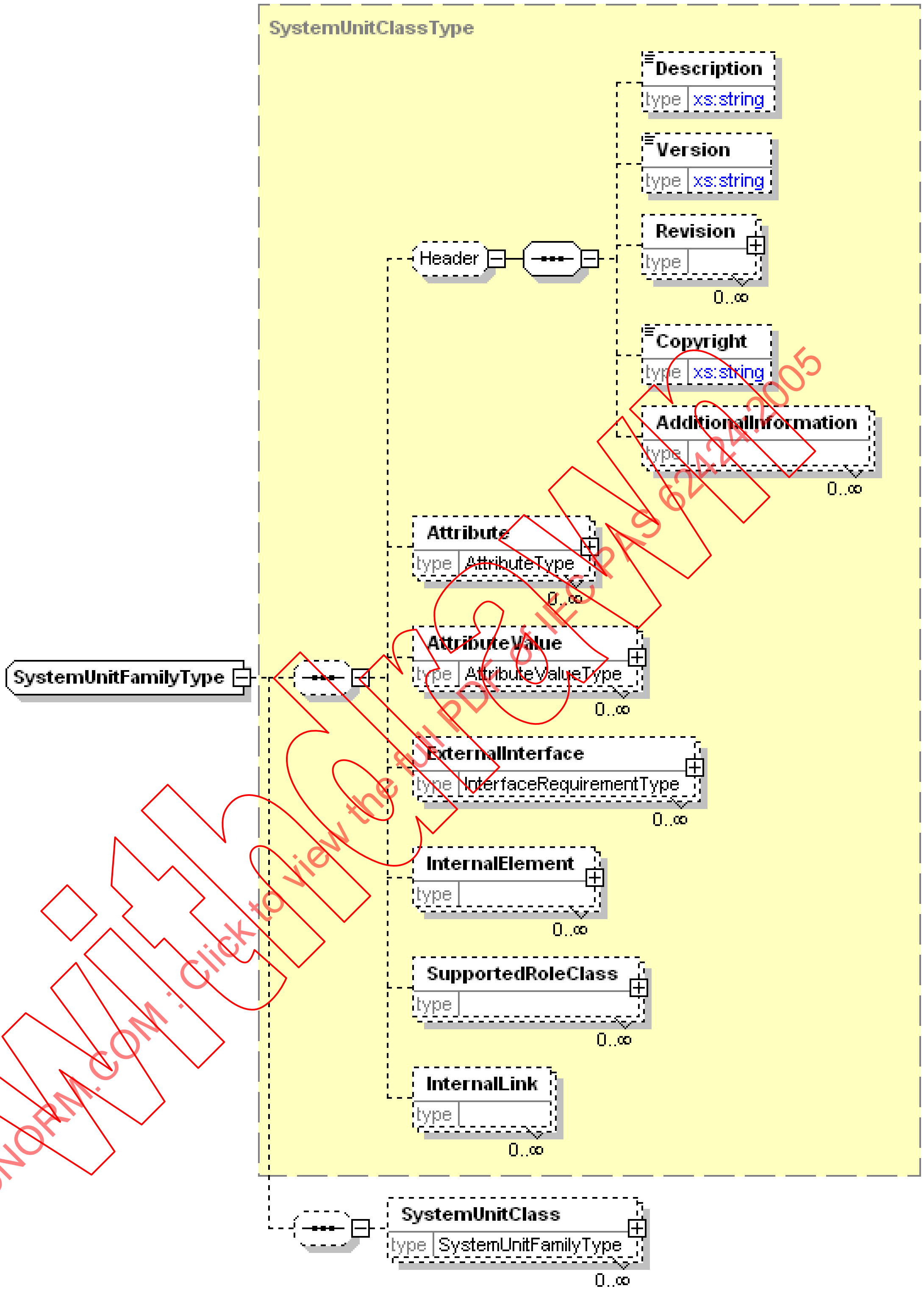
The element “InternalLink” allows the specification of the relationships of defined internal elements. To identify the relationship, it shall have a attribute “LocalLinkName”. To reference the partner elements, it has the two optional attributes: “refPartnerSideA” and “refPartnerSideB”.

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	localLinkName	xs:string	required			
	refPartnerSideA	reference	optional			
	refPartnerSideB	reference	optional			
	description	xs:string	optional			
	PropertyString	propertystring	optional			
	changeMode	changeMode	optional	state		
source	<xs:element name="InternalLink" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:attribute name="localLinkName" type="xs:string" use="required"/> <xs:attribute name="refPartnerSideA" type="reference" use="optional"/> <xs:attribute name="refPartnerSideB" type="reference" use="optional"/> <xs:attribute name="description" type="xs:string" use="optional"/> <xs:attribute name="PropertyString" type="propertystring" use="optional"/> <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/> </xs:complexType> </xs:element>					

A.19 ComplexType SystemUnitFamilyType

A.19.1 General

The datatype “SystemUnitFamilyType” is an extension of the datatype “SystemUnitClassType”. For a description of this datatype, see Clause A.18. The datatype “SystemUnitFamilyType” allows for the creation of child elements of system unit classes. Child classes contain all characteristics (attribute elements) of the base class but may be extended by new definitions of characteristics (attribute elements).

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>type</td><td>extension of <b>SystemUnitClassType</b></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink SystemUnitClass</b></td></tr>
<tr><td>used by</td><td>elements <b>SystemUnitFamilyType/SystemUnitClass SystemUnitClassLibType/SystemUnitClass</b></td></tr>
<tr><td>attributes</td><td><table>
<tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Fixed</th><th>Annotation</th></tr>
<tr><td>SystemUnitClassName</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>PropertyString</td><td>propertystring</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>changeMode</td><td>changeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>source</td><td>

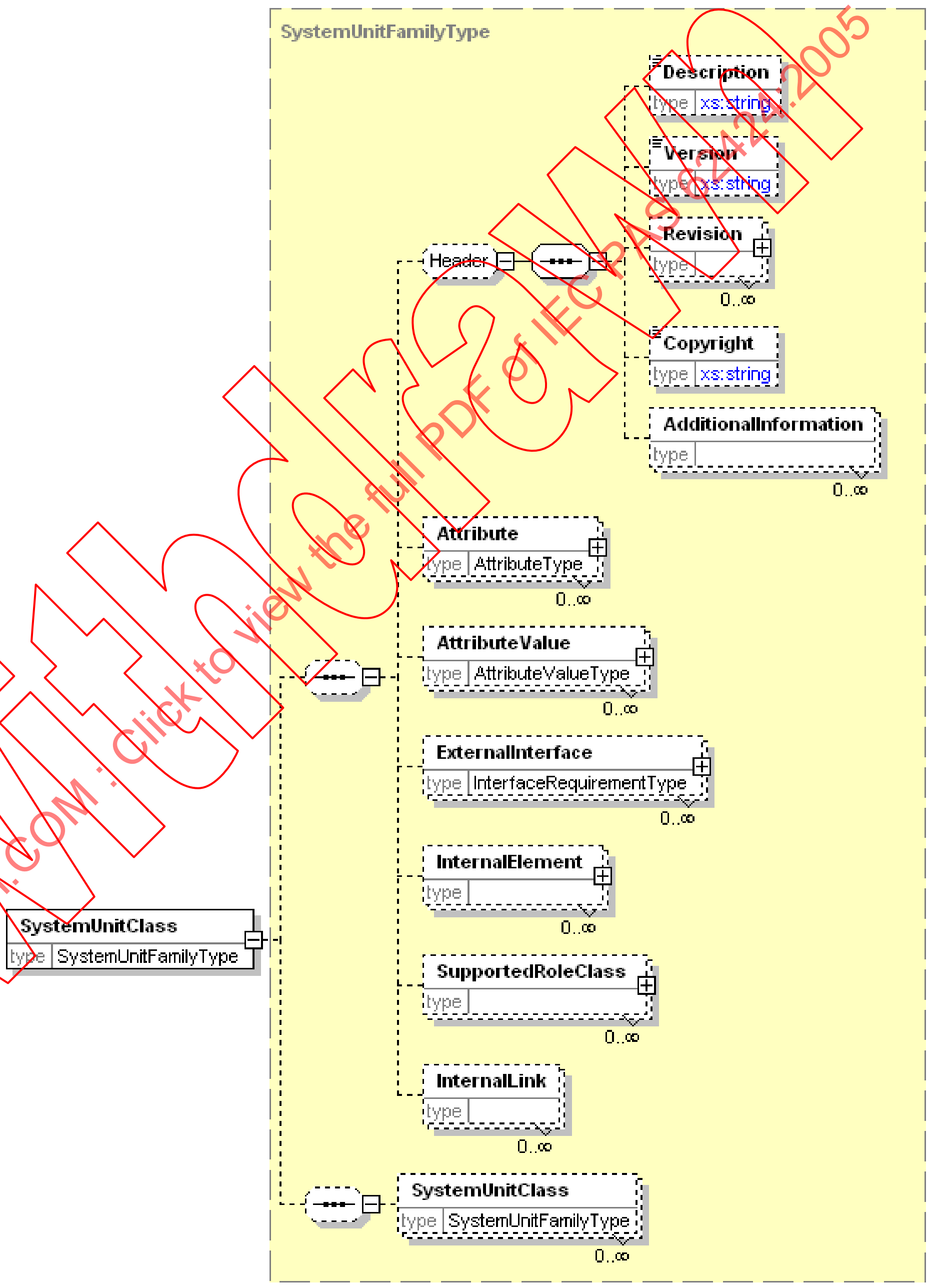
```
<xs:complexType name="SystemUnitFamilyType">
 <xs:complexContent>
```

</td></tr>
</table>

```
    <xs:extension base="SystemUnitClassType">
      <xs:sequence minOccurs="0">
        <xs:element name="SystemUnitClass" type="SystemUnitFamilyType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="SystemUnitClassName" type="xs:string" use="required"/>
      <xs:attribute name="PropertyString" type="propertystring" use="optional"/>
      <xs:attribute name="changeMode" type="changeMode" use="optional" default="state"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A.19.2 Element SystemUnitFamilyType/SystemUnitClass

The element is of the datatype "SystemUnitFamilyType". For a description of this datatype, see Clause A.19.

diagram	
type	**SystemUnitFamilyType**
children	**Description Version Revision Copyright AdditionalInformation Attribute AttributeValue ExternalInterface InternalElement SupportedRoleClass InternalLink SystemUnitClass**
attributes	Name Type Use Default Fixed Annotation