BS EN 62424:2016

BSI Standards Publication

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

National foreword

This British Standard is the UK implementation of EN 62424:2016. It is identical to IEC 62424:2016. It supersedes BS EN 62424:2009, which will be withdrawn on 19 August 2019.

The UK participation in its preparation was entrusted to Technical Committee GEL/65, Measurement and control.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2017

ISBN 978 0 580 82343 5
ICS 25.040.40; 35.240.50

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2017.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62424

November 2016

ICS 25.040.40; 35.240.50

Supersedes EN 62424:2009

English Version

Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (IEC 62424:2016)

Représentation de l'ingénierie de commande de processus - Demandes sous forme de diagrammes P&I et échange de données entre outils P&ID et outils PCE-CAE (IEC 62424:2016)

Darstellung von Aufgaben der Prozessleittechnik - Fließbilder und Datenaustausch zwischen EDV-Werkzeugen zur Fließbilderstellung und CAE-Systemen (IEC 62424:2016)

This European Standard was approved by CENELEC on 2016-08-19. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

European foreword

The text of document 65/544/CDV, future edition 2 of IEC 62424, prepared by IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62424:2016.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2017-05-19
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2019-08-19

This document supersedes EN 62424:2009.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 62424:2016 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 60848	NOTE	Harmonized as EN 60848.
IEC 61512-1	NOTE	Harmonized as EN 61512-1.
IEC 61987-1	NOTE	Harmonized as EN 61987-1.
ISO 10628-1	NOTE	Harmonized as EN ISO 10628-1.
ISO 10628-2	NOTE	Harmonized as EN ISO 10628-2.
ISO 13628-6	NOTE	Harmonized as EN ISO 13628-6.
ISO 13703	NOTE	Harmonized as EN ISO 13703.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC 61511-1	-	Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and application programming requirements	EN 61511-1	-
IEC 81346-1	2009	Industrial systems, installations and equipment and industrial products - Structuring principles and reference designations - Part 1: Basic rules	EN 81346-1	2009
ISO 13849-1	-	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design	EN ISO 13849-1	-
W3C XML 1.0	2004	Extensible Markup Language (XML) 1.0	-	-

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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
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.

International Standard IEC 62424 has been prepared by IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2008. This edition constitutes a technical revision.

This second edition is a compatible extension of the first edition. The main changes and extensions are detailed in Annex E and are summarized below:

a) updated definitions and new definitions;
b) identification replaced with reference designation;
c) updated PCE categories and process functions;
d) CAEX version 3.0, introduction of:
   - native multiple role support;

- nested interfaces;
- life cycle meta information;
- a separate Attribute library;
- updated examples;

e) updated electronic data model of the PCE request:

- new normative attribute library for basic PCE request attributes;
- new informative extended attribute library for further PCE request attributes;
- new informative electronic data model for the PCE request.

The text of this standard is based on the following documents:

CDV	Report on voting
65/544/CDV	65/560B/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

Efficient process engineering requires highly sophisticated tools for the different needs of the involved work processes and departments. These engineering tools are normally specialized in Process Design (PD), in Process Control Engineering (PCE), etc. Therefore, a working interoperability is essential to optimize the engineering process in total. Thus, the definition of a harmonized interface and data management is a core task to ensure a smooth workflow during the whole project and to guarantee data consistency in the different tools.

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

However, a definition for uniform semantics is still necessary. CAEX (Computer Aided Engineering eXchange) as it is 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 reference designation system and a unique description of the processing requirement is the key for a unique identification. For detailed information about representation of PCE loops in P&IDs 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 considered tools include private and common information. Both are stored at different places and in different divisions that are working on them. Hereby, the intermediate database CAEX only stores common information. In a wider approach, the intermediate database should 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 done via a neutral intermediate CAEX database, not directly from database to database. The intermediate CAEX database should be a file (for file based data exchange) or a stream (for network based data exchange). The term “CAEX database” within this standard has to be understood in this way, it does not denominate a database product as for example SQL.

Annex C of this standard contains the full XML schema of the CAEX Model. It is attached to this publication in XSD format.

NOTE Buyers of this publication can copy it for their own purposes only in the required amount.

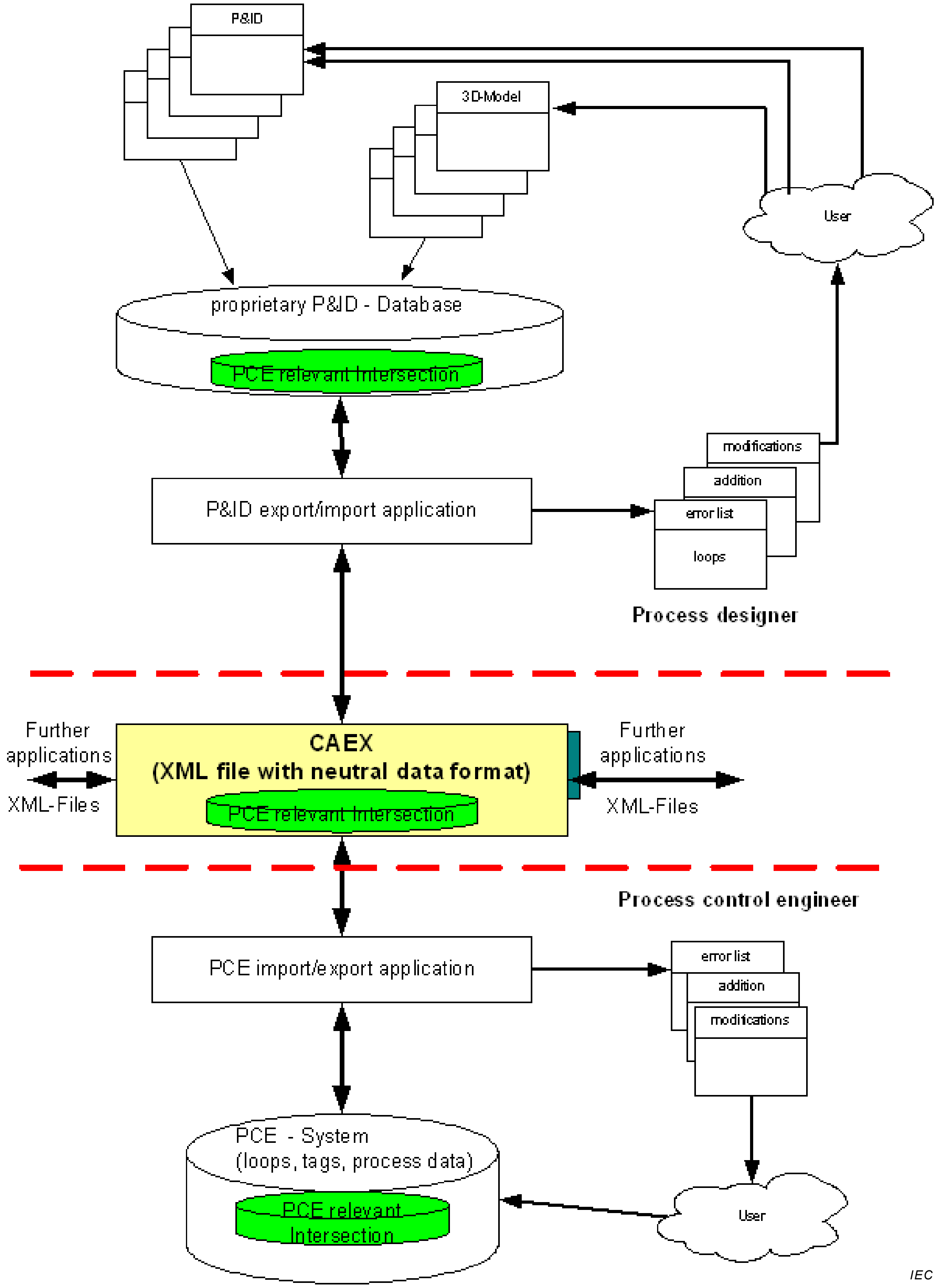

P&ID
3D-Model
User
proprietary P&ID - Database
PCE relevant Intersection
P&ID export/import application
modifications
addition
error list
loops
Process designer
Further applications
XML-Files
CAEX
(XML file with neutral data format)
PCE relevant Intersection
Further applications
XML-Files
Process control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

Figure 1 – Information flow between P&ID and PCE tool

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

1 Scope

This International Standard specifies how process control engineering requests are represented in a P&ID for automatic transferring data between P&ID and PCE tool and to avoid misinterpretation of graphical P&ID symbols for PCE.

It also defines the exchange of process control engineering request relevant data between a process control engineering tool and a P&ID 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&IDs will be 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 (e.g. vessels, valves, columns, etc.), their implementation and rules for the reference designation system are not in the scope of this standard. These rules are independent from this standard.

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

2 Normative references

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

IEC 61511-1, *Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and application programming requirements*

IEC 81346-1:2009, *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules*

ISO 13849-1, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

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 purposes of this document, the following terms and definitions apply.

3.1
actuator
functional unit that generates the manipulated variable, required to drive the final controlling element, from the output variable of the controlling element

EXAMPLE A practical example of an actuator acting directly on the final controlling element is a pneumatic control valve.

Note 1 to entry: If the final controlling element is mechanically actuated, it is controlled via an actuating drive. The actuator drives the actuating drive in this case.

[SOURCE: IEC 60050-351:2013, 351-49-07]

3.2
actuating drive
physical unit used for driving mechanically actuated final controlling elements

Note 1 to entry: Examples of actuating drives are electric, hydraulic or pneumatic actuating drives, diaphragm systems or piston actuators.

Note 2 to entry: No actuating drive is required for a final controlling element if the manipulated variable at the controller output is capable of directly influencing the mass flow or energy flow, i.e. without any mechanical intermediate variable (quantity).

[SOURCE: IEC 60050-351:2013, 351-56-16]

3.3
adjusted nominal pipe size
size of the related pipe for the process connection of the PCE request in case of pipe diameters size reduction based on process requirements

3.4
bubble
oval symbol used to denote the PCE category and processing function of a PCE request and to uniquely identify a PCE request

Note 1 to entry: On the basis of ISA 5.1:2009, Clause 3.

3.5
closed-loop control
process whereby one variable quantity, namely the controlled variable is continuously or sequentially measured, compared with another variable quantity, namely the reference variable, and influenced in such a manner as to adjust to the reference variable

Note 1 to entry: Characteristic for closed-loop control is the closed action in which the controlled variable continuously influences itself in the action path of the closed loop.

[SOURCE: IEC 60500-351:2013, 351-47-01]

3.6
control narrative
verbal description of a functional control scheme

3.7
design pressure
maximum pressure for which the system or component was designed for continuous usage

[SOURCE: ISO 13628-6:2006, 3.4]

3.8
design temperature
maximum temperature for which the system or component was designed for continuous usage

3.9
equipment ID
unique identifier of equipment

3.10
equipment/pipe flag
unique identifier of equipment/pipe type

3.11
final controlling element
functional unit forming part of the controlled system and arranged at its input, driven by the manipulated variable and manipulating the mass flow or energy flow

Note 1 to entry: If the final controlling element is mechanically actuated, an additional actuator (positioner) is used in some cases.

Note 2 to entry: The output variable of the final controlling equipment is usually not free from feedback. The interface between the actuator and the final controlling element should therefore be selected in such a way that the manipulated variable is not affected by feedback from the final controlling element.

[SOURCE: IEC 60050-351:2013, 351-49-08, modified – The figures are deleted.]

3.12
final controlling equipment
functional unit that consists of an actuator and a final controlling element

[SOURCE: IEC 60050-351:2013, 351-49-09, modified – The figure is deleted.]

3.13
function chart
graphic description tool with symbolic representation of sequential control systems

Note 1 to entry: The symbolic representation of steps, commands, transitions and directed links is based on input and output Boolean variables and also on internal state variables and binary delay elements.

Note 2 to entry: The elements, rules and basic structures for function charts are given in IEC 60848.

[SOURCE: IEC 60050-351:2013, 351-53-08, modified – The word "tool" has been added in the definition and Note 2 has been slightly shortened.]

3.14
heat tracing
heating system for pipe to prevent freezing of process requirements or to keep process conditions

3.15
heat tracing type
type of heating system for pipe

EXAMPLE Steam or electrical heating system.

3.16
heat tracing temperature set point
set point for the controller of a heat tracing

3.17
insulation type
description of the insulation type used

EXAMPLE Sound insulation.

3.18
insulation thickness
thickness of insulation added to the outer diameter of the pipe size

3.19
intermediate database
intermediate data storage system between source and target tool

3.20
material balance point
balance point of the process calculation

3.21
medium code
abbreviation and identifier for the fluid running through a process pipe

3.22
medium code description
description of the fluid running through a process pipe

3.23
neutral database
vendor independent data storage system

3.24
open-loop control
process whereby one or more variable quantities as input variables influence other variable quantities as output variables in accordance with the proper laws of the system

Note 1 to entry: Characteristic for open-loop control is the open action path or in case of a closed action path the fact that the output variables being influenced by the input variables are not continuously or sequentially influencing themselves and not by the same input variables.

[SOURCE: IEC 60050-351:2013, 351-47-02]

3.25
PCE category
letter that designates the kind of process control engineering request

Note 1 to entry: Unlike other standards, this standard uses the term "PCE category" instead of "measured variable" (e.g. temperature measurement) for the first digit of the PCE request. The PCE category as defined in this standard allows to unambiguously identify the kind of PCE request, without the need to specify a second letter as modifier for final controlling equipments. Based on this, only one letter for sensing element and final controlling equipment identification of PCE request is necessary.

3.26
PCE control function
function in a PCE control

Note 1 to entry: According to 4.2.7 of IEC 61512-1:1997.

3.27
PCE loop
collection of PCE requests and PCE control functions depicting their functional coherence

3.28
PCE request
requirement for process control equipment

Note 1 to entry: Each PCE request is graphically represented by a bubble which collects all information on the functional requirements.

3.29
pipe diameter size
nominal size of the related pipe for the process connection of the PCE request

3.30
pipe ID
unique identifier of pipe

EXAMPLE Isometrics number.

3.31
pipe specification
abbreviation and identifier for the specification of piping equipment

Note 1 to entry: Defines the size, material, design, pressure and temperature for all elements of a pipe.

3.32
process control equipment
entirety of devices and programs and, in a broader sense, all instructions and programs used for the task of controlling equipment having a process control function

Note 1 to entry: Control equipment also comprises the process control station and instructions include operating manuals.

Note 2 to entry: Providing a process with control equipment is denoted as process automation.

[SOURCE: IEC 60050-351:2013, 351-56-24, modified.]

3.33
process control function
function to work on process variable quantities, which is composed of basic functions of process control, specific to particular functional units of the plant

Note 1 to entry: In addition to process control functions associated with specific control levels, there can also be process control functions that link input and output variables across several control levels. 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 sensing element via the controller to the final controlling element. Another process control function connects the operator with the indicators for the process variables. In view of the diversity of definitions of process control functions, standardization is not appropriate at this time.

[SOURCE: IEC 60050-351:2013, 351-55-16]

3.34
processing function
function in a process

Note 1 to entry: A processing function serves a control module according to 3.10 and 5.2.2.4 of IEC 61512-1:1997.

3.35
proprietary database
vendor specific data storage system, with syntax and/or semantic not complying to any standard

3.36
PU-vendor
Package Unit vendor
supplier of a process unit in a plant

3.37
reference designation
identifier of a specific object formed with respect to the system of which the object is a constituent, based on one or more aspects of that system

Note 1 to entry: Terms "object", "aspect" and "system" are also defined in IEC 81346-1:2009, respectively at 3.1, 3.3 and 3.2.

[SOURCE: IEC 81346-1:2009, 3.11].

3.38
schema
XML based description of rules such that when an XML document conforms to those rules it is considered "valid" according to that schema

Note 1 to entry: On the basis of Extensible Mark-up Language (XML) 1.0 (Third Edition), W3C Recommendation, Clause 2.

3.39
sensing element
functional unit that senses the effect of a measured variable at its input and places a corresponding measurement signal at its output

EXAMPLE Thermocouple, foil strain gauge, pH electrode.

Note 1 to entry: The corresponding physical unit is named sensor or detecting device.

[SOURCE: IEC 351:2013, 351-56-26, modified.]

3.40
source database
data storage system of the source tool

3.41
target database
data storage system of the target tool

3.42
typical identification
abbreviation and identifier for a graphical diagram in a database, a group of signals or grouped PCE requests

4 Abbreviations

Table 1 shows abbreviations used in this standard.

Table 1 – Abbrevations

CAE	Computer Aided Engineering
CAEX	Computer Aided Engineering eXchange
CCR	Central Control Room
GMP	Good Manufacturing Practice
PCE	Process Control Engineering
PCS	Process Control System
P&ID	Piping and Instrumentation Diagram
PD	Process design
PL	Performance level according to ISO 13849-1
PU	Package Unit
SIL	Safety Integrity Level according to IEC 61511-1
SIS	Safety Instrumented System according to IEC 61511-1
XML	Extensible Markup Language

5 Conformity

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

To claim conformity to this standard 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.

The goal of the import/export application is to provide for 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 reconciliate 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 (e.g. 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 discovered data inconsistencies shall be reported by the import application. The generation of the report shall be configurable. The import/export application has to assure that the intersection of the different databases contains 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 have to be consistent. This requires a consistency check before exporting the data. This procedure has to 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 fulfill the following requirements:

Data export from source database to neutral database shall comprise the following activities.

a) Check P&ID and PCE database for at least:
   1) duplicate PCE requests or loop designations;
   2) mandatory fields being filled in;
   3) correct use of numbering system of the PCE requests.

Inconsistent data shall not be exported.

b) Generate PCE relevant information;
c) Check for changed information in comparison with previously stored data in the neutral database;
d) Renaming of PCE request shall be supported by the export functionality;
e) Perform data export from proprietary into neutral database:
   1) For example, if the PCE request has been changed, the old PCE request within the neutral database has to be deleted and the new one shall be exported from the proprietary database into the neutral database. The old PCE request information shall be stored in a backup storage system;

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

f) Generate reports after each data exchange:

   For example, new PCE requests list, missing PCE requests list, changed PCE requests list, deleted PCE requests list, problems and error list.

Data import from neutral database into the target database shall comprise the following activities.

g) Generate PCE relevant information from neutral database;

h) Check for changed information by comparing the neutral database with the target database;

i) Perform data import from the neutral into the proprietary database;

j) Renaming of PCE request shall be supported by the import functionality;

k) Generate reports after each data exchange:

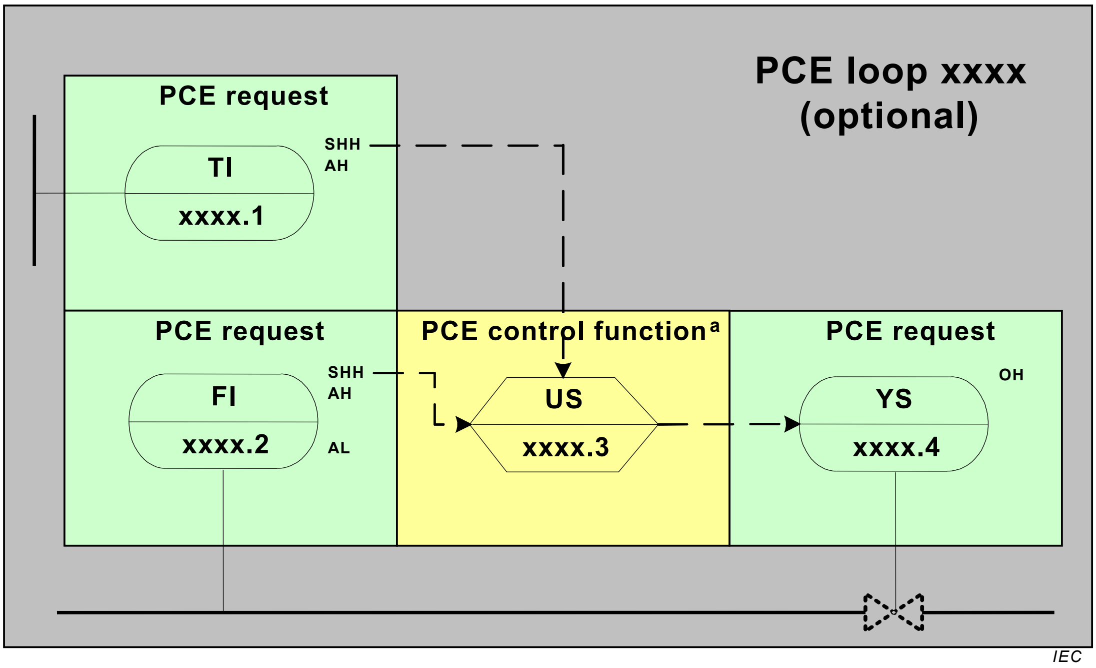
   1) For example, error lists;

   2) inconsistencies due to imported data shall be detected by the target application during the import process and are not considered within this standard.

6 Representation of PCE requests in a 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. Consequentially, 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 reference designation. In order to meet the requirements of data handling, the same reference designation shall not be used for different PCE requests. Functional coherence should be depicted by collecting the individual PCE requests in a PCE loop. A PCE loop does not have a graphical representation. Depending on the engineering strategy, a PCE loop thereby consists of at least one, but may also combine several PCE requests. If PCE loops are used, these shall be represented in the reference designation of all concerned PCE requests. 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 – Organization of PCE requests

6.2 Objectives and principles

This 6.2 defines how to represent the process control engineering functionality in P&IDs. Technical details of the used equipment 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 the following items are set out in the standard:

a) the PCE categories and functions;

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

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

d) the graphical representation of signals in a P&ID.

In addition a reference designation system used for PCE requests in a P&ID shall be specified (see 6.3.5).

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

6.3 Requirements for the reference designation 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 ten data fields outside the bubble are defined to hold all the information of a PCE request (see Figure 3). For detailed information see 6.3.3 to 6.3.9.

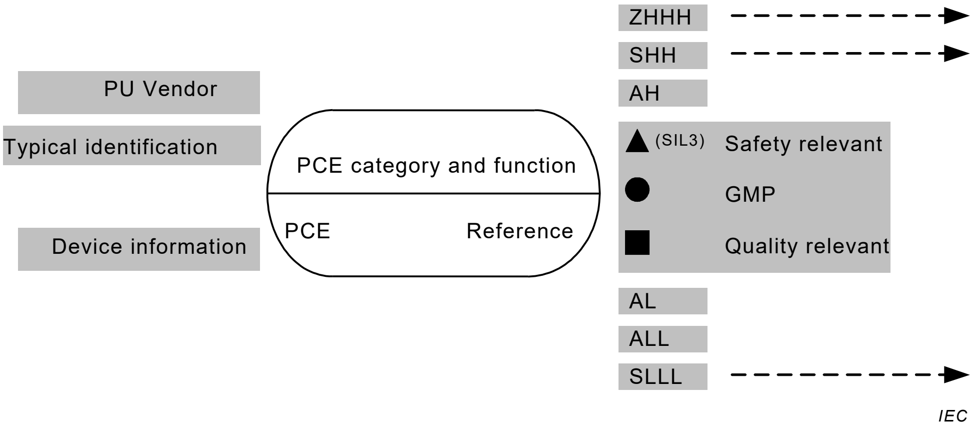


Figure 3 – General representation of a PCE-Request in a P&ID

As stated before, only the PCE functionality shall be depicted on the 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-sensing 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.

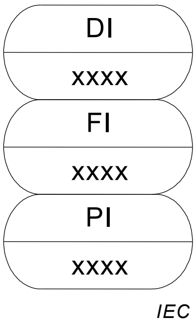


Figure 4 – Multi-sensing element

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. Source of information flow shall be a bubble of a PCE control function or request or switching action consisting of the six fields on the right side of the bubble. The sink of 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. Multi-sensing elements 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 a local interface, a local control panel and a central control room. The location does not reflect any realization in systems.

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

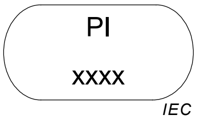


Figure 5 – Local interface

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

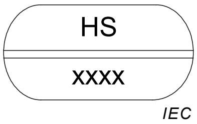


Figure 6 – Manually operated switch in local control panel

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

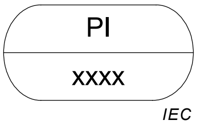


Figure 7 – Pressure indication in central control room by a central control system

It is recommended to simplify the presentation of PCE for combined local interface and indication in central control room by only one bubble as shown in Figure 7. For this case it is required to have a note in the definitions of a P&ID.

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 one PCE category and should have one PCE processing function. For the definition of the processing function see 6.3.4.3.

6.3.4.2 PCE categories

The first letter represents the PCE category and shall be selected from Table 2 if the measuring or initiating variable is listed in that table. If this is not the case, a new category may be defined. A unique definition is recommended to provide for an automatic transfer to the specific equipment specification for the process control engineer. In case of amendments of categories of Table 2, a coding via letter X as described in table footnote [b] may be used.

Modifying letters for the PCE category shall not be used to avoid misinterpretation in the course of automatic transfer to the specific equipment specification for the process control engineer.

Table 2 – PCE categories

Letter	PCE category
A	Analysis
B	Optical measurement, e.g. flame detection
C	[a]
D	Density
E	Voltage
F	Flow
G	Distance, length, position
H	Hand or manual and manually initiated operation
I	Current
J	Power
K	Time based function
L	Level
M	Moisture or humidity
N	Actuation setting electrical (all type of electrical consumer, e.g. motor, heater) [c]
O	[a]
P	Pressure
Q	Quantity or counter
R	Radiation
S	Speed or frequency (including acceleration)
T	Temperature
U	Used for PCE control function (see 6.3.10)
V	Vibration, mechanical analysis, torque
W	Weight, mass, force
X	[b]
Y	Actuation setting non electrical like hydraulic or pneumatic (switching, varying, restricting, e.g. valve-operated) [c]
Z	[a]

[a] The definition of this letter should be defined by users.

[b] 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.

[c] The use of N for motor driven or heater final controlling equipments and Y for hydraulic or pneumatic valve driven final controlling equipments is based on different PCE activities and specific maintenance requirements for both types of final controlling equipments. Moreover, in the light of increased maintenance requirements in the plant, immediate identification for transferring of data and relevant attributes of the final controlling equipment to asset management systems is necessary.

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 (all kind of control scheme, e.g. split-range, PID controller or ON-OFF controller – typically used for closed-loop control)
D	Difference
E	Shall not be used
F	Ratio
G	Shall not be used
H	High limit, on, opened
I	Indication of analogue values
J	Shall not be used
K	Time rate of change e.g. for acceleration or calculating a derivation
L	Low limit, off, closed
M	Shall not be used
N	Shall not be used
O	Local or PCS status indication of binary signals
P	Point (test) connection
Q	Integrating, quantity or counting
R	Recorded value
S	Binary control function or switching function (not safety relevant)
T	Shall not be used
U	Shall not be used
V	Shall not be used
W	Shall not be used
X	[b]
Y	Computing function
Z	Binary control function or switching function (safety relevant) [a]

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

[b] 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 B, Q, I and R apply to the preceding processing function and can be used several times in one processing function, e.g. FIQI means the indication of a flow and its counting.

The PCE processing functions A, H, L, O, S and Z shall only be used outside the bubble. In this case the PCE category may stand as a single value in the upper part of the bubble. In addition, a detailed definition for the signal information (see 6.3.2) to be transferred automatically to the control equipment specification for the process control engineer is given by this way.

In cases of a PCE request with PCE processing functions A, H, L, O, S and Z only a PCE request can be without a PCE processing functions, e.g. see Figure 22 for the PCE category H.

NOTE The letters O, A, S, and Z are used on a specific level, e.g. AH for High-Alarm or SHH for High-High-Switch point. On every level it is possible to combine the letters OASZ, e.g. OSL for binary status indication and switching at low level.

For the PCE categories N and Y the processing function S and Z remain inside the bubble due to the pure unique binary on/off function and no dedicated signal line deviated from the processing functions S and Z for a final controlling equipment (see Table 5).

The combination of processing functions shall be used in the sequence given in Table 4. The table hierarchy shall be from left to right and per column top down.

Table 4 – Sequence combinations

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

6.3.4.4 PCE processing functions for final controlling equipments

The PCE processing functions shall be used for final controlling equipments in the same way as for sensing elements. Some examples are shown in Table 5.

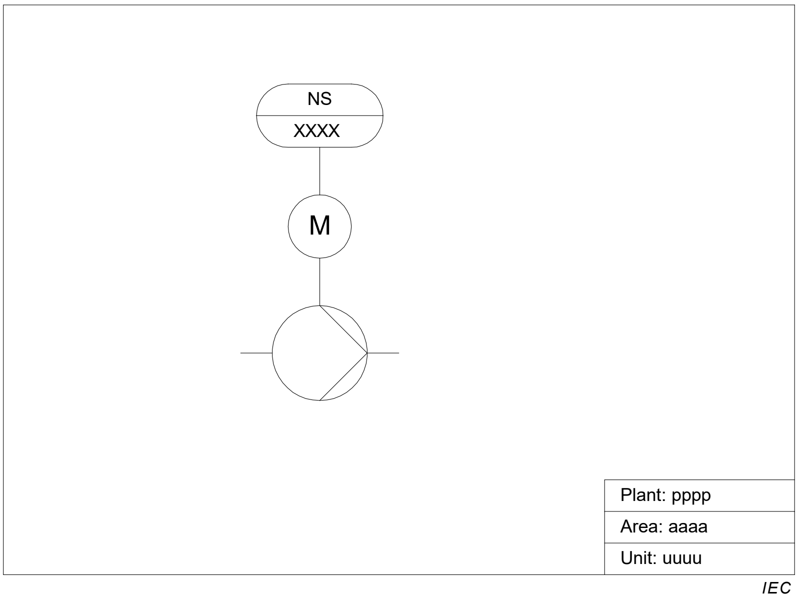
Table 5 – PCE processing functions for final controlling equipments

Letter	Processing function
YS	Non electrical actuating drive with open-loop-control function e.g. On/off valve
YC	Non electrical actuating drive with closed-loop-control function e.g. Control valve
YCS	Non electrical actuating drive with closed-loop-control function and open-loop-control (open/close) function e.g. Control valve with on/off function
YZ	Non electrical actuating drive with open-loop-control function (safety related e.g. On/off valve (safety relevant)
YIC	Non electrical actuating drive with closed-loop-control function and position indication e.g. Control valve with position indication
NS	Electrical actuating drive with open-loop-control function e.g. On/off motor
NC	Electrical actuating drive with closed-loop-control function e.g. Control motor

Graphical representation of the equipment valve including further functional details in the equipment symbols in accordance to ISO 10628-1 and ISO 10628-2 in the P&ID cannot be used in the CAEX model. Such details should be placed in the PID database.

6.3.5 PCE request reference designation system

A reference designation system (e.g. IEC 81346-1) shall be used in order to identify the PCE request unambiguously. This reference designation shall be independent of the PCE processing function of the PCE request and depicted in the lower part of the bubble. Preceding reference designation levels (e.g. site, plant, unit, area) may be omitted in the bubble if the request within the context of the P&ID is unique (see Figure 8). If PCE requests are combined in a PCE loop, their reference designation shall have separated levels for the loop and the request.



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

Figure 8 – Example of PCE request reference designation

6.3.6 PU-vendor and typical identification

If applicable, PU vendor information should be given above the horizontal line, but outside the bubble on its upper left side, as shown in Figure 9. If this field is not used for PU vendor information, it may be used to show other project specific information.

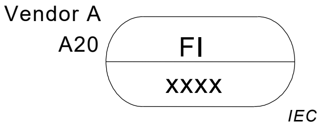


Figure 9 – Example of flow measurement with indication in the CCR delivered by vendor A specified by typical identification A20

To support automatic loop, request and tag generation with PCE-CAE tool, requests, especially motor requests should be indicated by a typical identification on the upper left side, outside the bubble. These typical identification are fixed by the project team and are used to determine the composition of the PCE request, e.g. how the motor drive should be switched (with start-stop only, with start-stop and running indication, with current measurement, etc.) or a combination of measuring systems.

6.3.7 Device information

If, because of the PCE category, additional device information is needed (e.g. orifice for flow measurement), this should be indicated in the lower zone outside the bubble on the left side (see Figure 10).

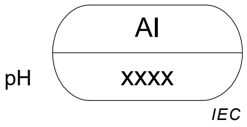


Figure 10 – Example of pH-measurement with indication in the CCR

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 (e.g. H, HH, HHH) it shall be possible to combine alarming and switching function, e.g. 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.

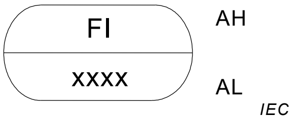


Figure 11 – Example of flow measurement with indication in the CCR and high and low alarm

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

It shall be unambiguous and shall be connected to the controlling function or the final controlling equipment starting with the SH, SHH, SHHH, SL, SLL or SLLL symbols as shown in Figure 12.

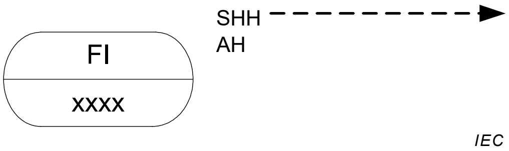


Figure 12 – Flow measurement with indication in the CCR and high alarm and a high-high switching function

The combination of Figure 11 and Figure 12 with additional safety relevant switches may be used as shown in Figure 13.

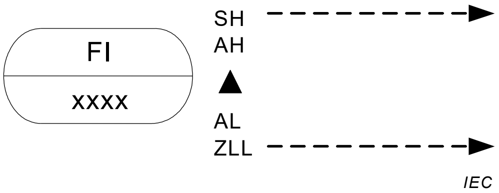


Figure 13 – Flow measurement with indication in the CCR and a high-high switch limit, a high alarm, a low alarm and a low-low switch limit for a safety function

To identify the level of an alarm or switching level the letters H, HH, HHH, L, LL and LLL should be used. The sequence from upper level to lower level is HHH – HH – H – L – LL – LLL.

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 sensing elements or final controlling equipments and a square for the indication of a quality relevant PCE request. A triangle should be used for a safety function (categorized by SIL or PL) (see Figure 14).

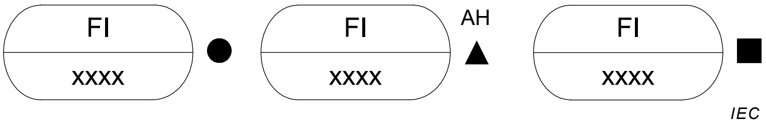


Figure 14 – GMP relevant, safety relevant and quality relevant flow measurement with indication in the CCR

These symbols shall be placed as closely as possible to the bubbles on the right side. Crossing of signal lines connected in the center is acceptable.

6.3.10 PCE control functions

PCE control functions essentially contain the functional relationship between sensing elements and final controlling equipments. These control functions are the "building stones", the elements of the entire process functionality. Mostly they are technically achieved via control system configuration. Safety related control functions are usually implemented by SIS configurations (logic solver) according to IEC 61511-1.

In simple configurations, e.g. one sensing element and one final controlling equipment, where the relation is unambiguously represented in the P&ID, the PCE control function should be omitted.

The symbol for the PCE control function is the hexagon. This hexagon, see Figure 15, symbolizes the control functionality which has one or more sensing elements as inputs, and one or more final controlling equipments as outputs.

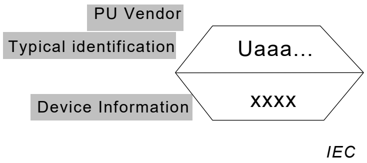


Figure 15 – Control function

The hexagon symbol shall be connected with signal lines (see 6.3.2.) to the various bubbles that represent the relevant PCE requests (see Annex B). The arrows indicate the direction of the information (sensing element to PCE control function and PCE control function to final controlling equipment).

If applicable, PU vendor information shall be given above the horizontal line, but outside the hexagon on its upper left side. If this field is not used for PU vendor information, it may be used to show other project specific information.

To support automatic loop, request and tag generation with PCE-CAE tool, especially functional logic design request, should be indicated especially functional logic design should be indicated by a typical identification on the upper left side, outside the hexagon.

In case of a safety relevant control function, UZ…., required SIL or PL shall be indicated in the lower zone outside the hexagon on the left side as shown in Figure 16. Other relevant information, e.g. 2oo3 configuration should be added as appropriate. For non-safety control functions this field should be used for additional relevant information.

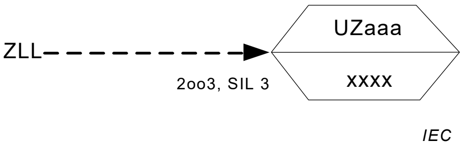


Figure 16 – Safety relevant control function

The PCE control functions shall be identified separately. The PCE control function shall be identified unambiguously within the reference designation system used. This reference designation shall be independent of the PCE processing function of the PCE control function and depicted in the lower part of the hexagon. Preceding reference designation levels (e.g. site, plant, unit, area) may be omitted in the hexagon if the control function within the context of the P&ID is unique (see 6.3.5). If control functions are integrated in a PCE loop, their reference designation 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 reference designation.

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

NOTE Compared to a PCE request, the processing functions are normally not unique in a PCE control function. Therefore the processing function A, S and Z remain in the hexagon as placeholder and higher level explanation of the complete function.

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 sensing element and one final controlling equipment which is safety relevant, this means that at least one sensing element and one final controlling equipment 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&IDs include a variety of information relevant for process control engineering purposes. Clause 6 defines how basic information concerning PCE requests and their process relevant functionality shall be represented in a P&ID. The given specification concerns primarily the graphical notation, but of course, this establishes structural and semantic fixings too. In this Clause 7, these structural and semantic fixings will be mapped to a semiformal 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 modeled data between the P&ID system and the PCE systems.

7.2 Meaning of P&ID elements

P&IDs show a plant (or a part of it) in its function as a physical framework. 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&IDs, representing PCE requests in accordance with this standard show functional requirements (roles) and not the assembly of equipment. A shown pump symbolizes not the equipment "pump" but the requirement: 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.

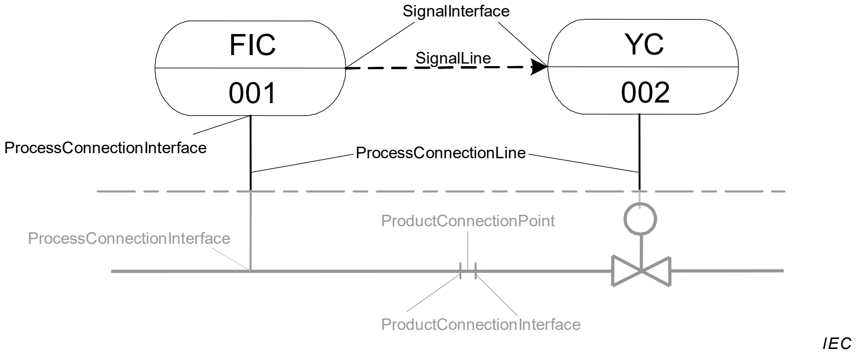


Figure 17 – P&ID elements and associations (PCE relevant items are shown in dark lines)

P&IDs show graphically the functional relationship between the elements. In the example given in Figure 17 four main classes of relationships are shown.

Note that graphical representations of the equipment including further functional details in the equipment symbols in accordance with ISO 10628-1 and ISO 10628-2 used in the P&ID cannot be used in the CAEX model. Such details will be placed in a database.

a) Signal connections

Signal connections are notated as declared in Clause 6 by a dashed line, the so-called “SignalLine”. The SignalLine only symbolizes the functional influence between PCE requests and not electrical wiring.

b) Process connections

Process connections are notated as declared in Clause 6 by a plain line, the so-called “ProcessConnectionLine”. The ProcessConnectionLine symbolizes the information flow from the control world to the physical process or vice-versa. The ProcessConnectionLine only symbolizes the functional coupling between a PCE request and the material balance point but not the actual layout in the plant.

c) Product connections

Product connections symbolize the coupling of two pieces of equipment with the possibility of material transfer between them (such as pipe-pipe or pipe-vessel). The properties of this kind of association are not the subject of this standard.

d) Mechanical connections

Mechanical connections symbolize the mechanical coupling within actuation elements (drive-valve, motor-pump). The properties of this kind of connections are not the subject of this standard.

7.3 PCE relevant information of P&ID tools

Besides general structural and functional information, P&ID tools handle a variety of information which is of direct interest to the PCE.

a) Control relevant information

PCE requests, process connections, signal lines with all their attributes and interfaces as described in Clause 6 comprise 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. That information is usually important for the process control engineering tools too. Clause 8 gives a minimum list of relevant additional parameters.

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

7.4.1 General object model of a plant hierarchy

The P&ID 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 also a data exchange format which supports an open information flow from the P&ID tools to the PCE tools and vice versa.

Figure 18 shows a general data model of a plant hierarchy and highlights the PCE data model for PCE relevant information as described in Clause 6.

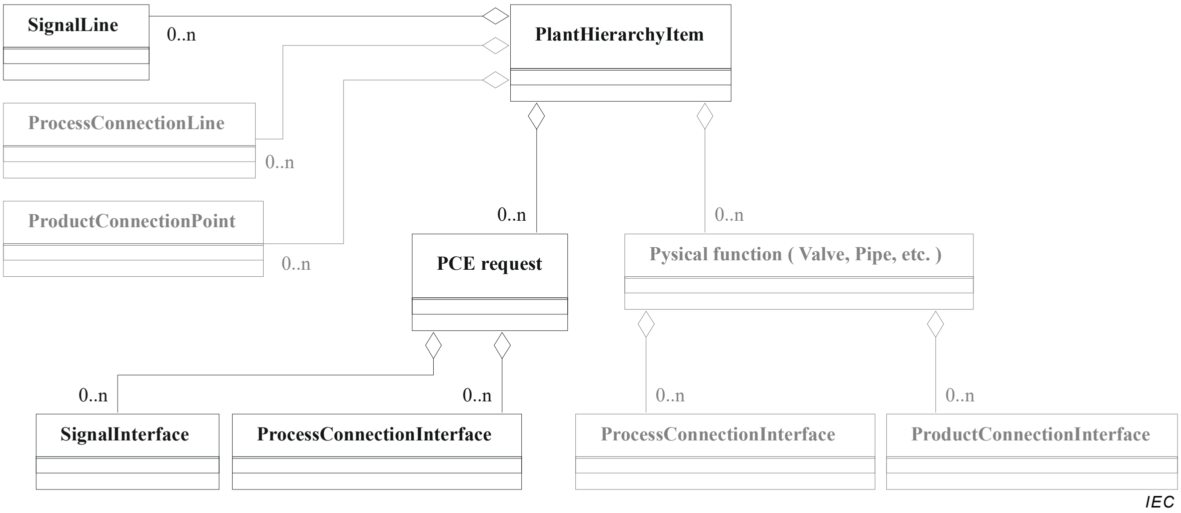


Figure 18 – Process data model (PCE relevant items are shown in dark lines)

The specifications in Clause 6 provide that

- a plant hierarchy item is a logical element collecting PCE request, SignalLines, Physical functions, ProcessConnectionLines and ProductConnectionPoints; the shaded objects shown in Figure 18 are not in the scope of this standard; plant hierarchy items may contain other nested plant hierarchy items (this allows creating a hierarchical breakdown structure of the plant);
- each PCE request contains 0...n ProcessConnectionInterfaces and 0...n SignalInterfaces;
- each plant hierarchy item, PCE request, SignalLine, ProcessConnectionInterface and SignalInterface shall have a set of attributes;
- each PCE request is part of one and only one plant hierarchy item; and
- control functions shall be handled in the same way as PCE requests but do not include ProcessConnectionInterfaces.

7.4.2 General object model of a PCE request

Figure 19 depicts the data model of a PCE request. A PCE request shall consist of 1...n interfaces and a set of attributes which may be extended by additional attributes and additional interfaces. Furthermore, common types of interfaces are presented.

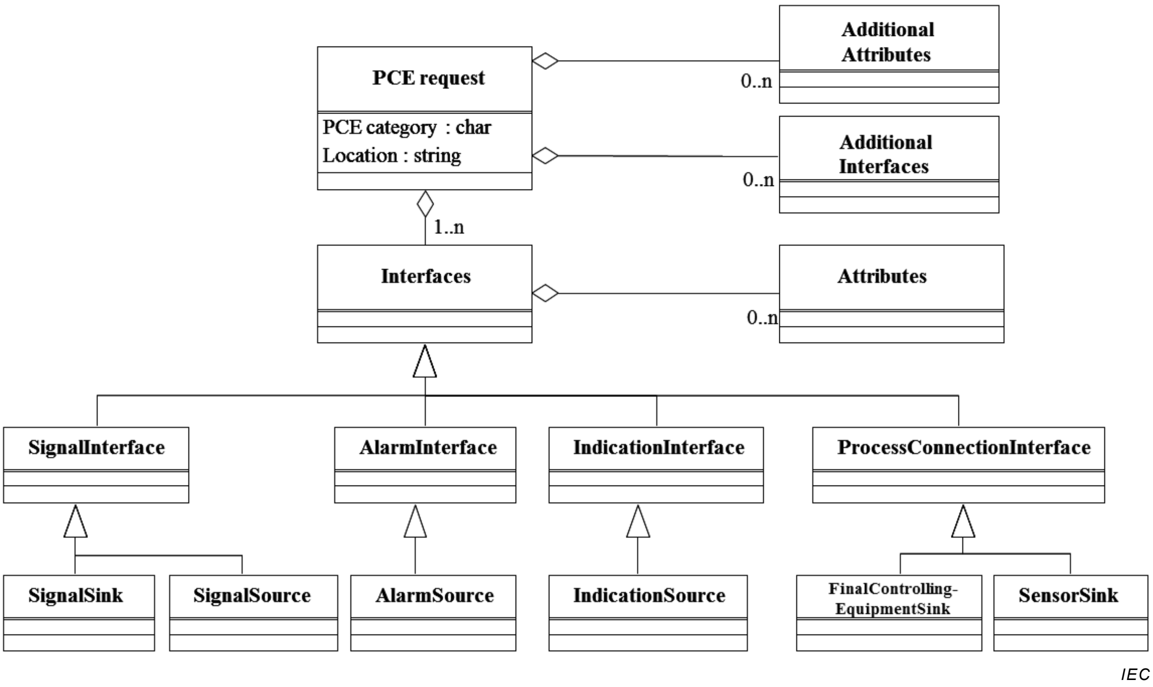


Figure 19 – PCE request data model

Each concrete PCE request possesses at least either a SignalInterface or a Process-ConnectionInterface with respect to the signal output of its processing function. A PCE request without any interface makes no sense.

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

- PCE category (see Table 2);
- PCE reference designation;
- Location (local, local control panel, central control system).

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

- PU vendor (string);
- typical identification (string);
- device information (string);
- processing function (string) (see Table 3);
- GMP relevant (Boolean);
- safety relevant (Boolean);
- quality relevant (Boolean).

The CAEX data model of these attributes is defined in 7.5.3. Additional PCE relevant attributes are defined in Clauses 8 and D.1.

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

7.5 Modeling PCE relevant information using the CAEX system description language

7.5.1 Overview

The electronic data representation and exchange of plant information including PCE requests is supported by the data exchange format CAEX defined in this standard. The CAEX system description language provides an XML schema that supports an exchange of CAE data by means of an XML file. The syntax of CAEX and the semantic definitions of this standard allow for the exchange of instance data (plant data), type data (class data) and complete libraries as well. Furthermore it comprises means to support the change management process.

The data format CAEX is generally able to model and to exchange arbitrary object model information with a standardized syntax, but it does not provide a general semantic standard. Instead, it is designed to model semantics. Hence, CAEX is conceptually able to model and to exchange e.g. a complete P&ID including graphics and proprietary semantics, but this standard focus on PCE relevant information only and standardizes the semantics of a PCE request and PCE relevant attributes for an electronic data exchange between P&ID tools and PCE-CAE tools.

The XML schema of CAEX and the concepts behind are specified and explained in Annex A. The schema file is given in Annex C. Examples are given in Annex D.

7.5.2 Basic CAEX mappings

The exchange of plant information by means of CAEX requires the modeling of the described engineering data in a CAEX data model. This comprises the modeling of templates as follows.

a) A CAEX data model of PCE request related attribute types: Basis of a template modeling is a predefined attribute type library. A normative definition of a CAEX attribute type library is defined in 7.5.3. An informative definition of additional attributes is provided in Clause D.1. Examples about the application of both libraries are provided in Clause D.4.

b) A CAEX data model of required signal classes: A template for common interfaces shall be predefined as CAEX InterfaceClasses, e.g. SignalSource, SignalSink, FinalControlllingEquipmentSource, AlarmSource, SensorSink and IndicationSource. An informative example for a CAEX interface class library is given in Figure D.3.

c) A CAEX data model of a PCE request role class: An example of a CAEX role class is provided in Clause D.3. A template for a PCE request and a SignalLine shall be predefined as each one CAEX RoleClass, e.g. PCERequest and SignalLine. These predefined role classes utilize the standard attributes from the attribute type library a) and interface classes b) required for the data exchange. An informative example for a CAEX role class PCERequest is given in Figure D.5.

In addition to the described templates, the exchange of plant information requires the modeling of concrete and individual object instances according to Figure 18. This comprises the following.

d) A CAEX data model of a concrete plant hierarchy, which contains individual instances of physical functions or PCE requests with their interfaces and attributes and relations. The plant hierarchy shall be represented by a CAEX InstanceHierarchy.

e) A CAEX description of concrete elements of the plant hierarchy. Those plant hierarchy items, e.g. the physical function of a pipe, shall be represented by a CAEX InternalElement with an optional association to a role class e.g. “Pipe”. CAEX InternalElements may contain further InternalElements as nested objects. This allows for defining the desired breakdown structure. The detailed data model of the InternalElements may be user defined and is not in the scope of this standard.

f) A CAEX description of concrete PCE requests: A PCE request shall be represented in CAEX as InternalElement within the plant hierarchy associating a PCE request role class. The CAEX tag “Name” of the InternalElement shall represent the name of the PCE request. The associated PCE request role class references standard CAEX attributes (a) defined in 7.5.3. The concrete requirements for the PCE request (attribute values and the required interfaces) shall be stored in the RoleRequirements of the InternalElement. If applicable, additional attributes and interfaces, which are not predefined in the role class, shall be added here too.

NOTE In a later engineering phase, the same InternalElement can additionally be assigned to a corresponding SystemUnitClass which describes the concrete technical implementation of the PCE request. This is not in the scope of this standard. See A.2.10 for related CAEX concept details.

g) A CAEX data model of concrete signal lines which is supported in two ways:

   – Either a signal line between two PCE requests of the same plant hierarchy item is described with CAEX by means of an InternalLink of the belonging PCE request which directly links the corresponding interfaces of the two PCE requests. InternalLinks do not support properties, therefore they can only represent simple relations. An example for those signal lines is given in Clause D.4.

   – Or the signal line is represented as an own CAEX object. If the SignalLine is considered as an object for itself with its own properties, this shall be represented as a CAEX InternalElement with an associated RoleClass “SignalLine”. A signal line implements two external interfaces which shall be named as “SideA” and “SideB”. The connection between two PCE requests is modeled by means of each, an InternalElement for both PCE requests and, and another InternalElement for the SignalLine. Furthermore, two InternalLinks have to be defined: One InternalLink connects the source PCE request interface with the “SideA” interface of the signal line, and a second InternalLink connects the signal line interface “SideB” with the sink interface of the second PCE request.

   A signal line between two plant hierarchy items of the same level shall be described in CAEX in the same way as signal lines between two PCE requests, linking the corresponding interfaces of the two plant hierarchy items. An example for those signal lines is given in Figure 22.

h) A CAEX data model of concrete interfaces: PCE requests associated to the PCE request role class inherit the predefined interfaces of this role class. Additionally required interfaces shall be additionally implemented by means of the CAEX element ExternalInterface within the corresponding InternalElement.

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

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

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

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

i) CAEX description of concrete process connections

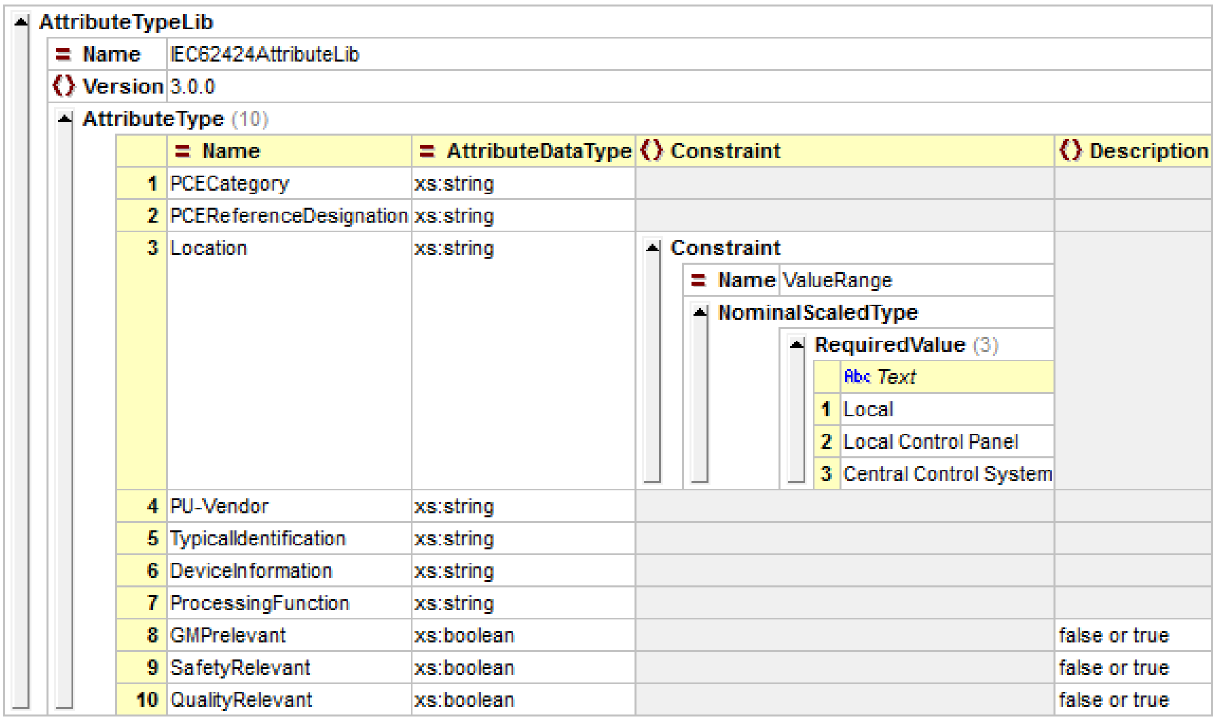
   Process connections are outside the scope of PCE and are not mapped to the CAEX model within this standard. All additional information given by the P&ID tool with respect to a process connection shall be mapped to attributes of the corresponding ProcessConnectionInterface. Each end of a process connection at a PCE request implements an additional ProcessConnectionInterface within this PCE request.

7.5.3 Standard CAEX library of PCE request related attributes

This 7.5.3 defines a standard CAEX representation of PCE request related attributes according to 7.4.2 in form of a standard CAEX attribute type library. Figure 20 and Figure 21 show this library which is named as IEC62424AttributeLib. This standard attribute type library defines the CAEX data model of the attributes covering syntax and semantics. Regarding this library, the following provisions apply.

- Attributes of PCE requests shall reference corresponding attributes of the AttributeTypeLib “IEC62424AttributeLib”.

NOTE Examples for the application of this library are provided in Clause D.4.



AttributeTypeLib

Name	IEC62424AttributeLib
Version	3.0.0

AttributeType (10)

	Name	AttributeDataType	Constraint	Description
1	PCECategory	xs:string		
2	PCEReferenceDesignation	xs:string		
3	Location	xs:string	Constraint Name ValueRange NominalScaledType RequiredValue (3) Text 1 Local 2 Local Control Panel 3 Central Control System	
4	PU-Vendor	xs:string		
5	TypicalIdentification	xs:string		
6	DeviceInformation	xs:string		
7	ProcessingFunction	xs:string		
8	GMPrelevant	xs:boolean		false or true
9	SafetyRelevant	xs:boolean		false or true
10	QualityRelevant	xs:boolean		false or true

IEC

Figure 20 – CAEX data model of major PCE request related attributes

```
<AttributeTypeLib Name="IEC62424AttributeLib">
  <Version>3.0.0</Version>
  <AttributeType Name="PCECategory" AttributeDataType="xs:string"/>
  <AttributeType Name="PCEReferenceDesignation" AttributeDataType="xs:string"/>
  <AttributeType Name="Location" AttributeDataType="xs:string">
    <Constraint Name="ValueRange">
      <NominalScaledType>
        <RequiredValue>Local</RequiredValue>
        <RequiredValue>Local Control Panel</RequiredValue>
        <RequiredValue>Central Control System</RequiredValue>
      </NominalScaledType>
    </Constraint>
  </AttributeType>
  <AttributeType Name="PU-Vendor" AttributeDataType="xs:string"/>
  <AttributeType Name="TypicalIdentification" AttributeDataType="xs:string"/>
  <AttributeType Name="DeviceInformation" AttributeDataType="xs:string"/>
  <AttributeType Name="ProcessingFunction" AttributeDataType="xs:string"/>
  <AttributeType Name="GMPrelevant" AttributeDataType="xs:boolean">
    <Description>false or true</Description>
  </AttributeType>
  <AttributeType Name="SafetyRelevant" AttributeDataType="xs:boolean">
    <Description>false or true</Description>
  </AttributeType>
  <AttributeType Name="QualityRelevant" AttributeDataType="xs:boolean">
    <Description>false or true</Description>
  </AttributeType>
</AttributeTypeLib>
```

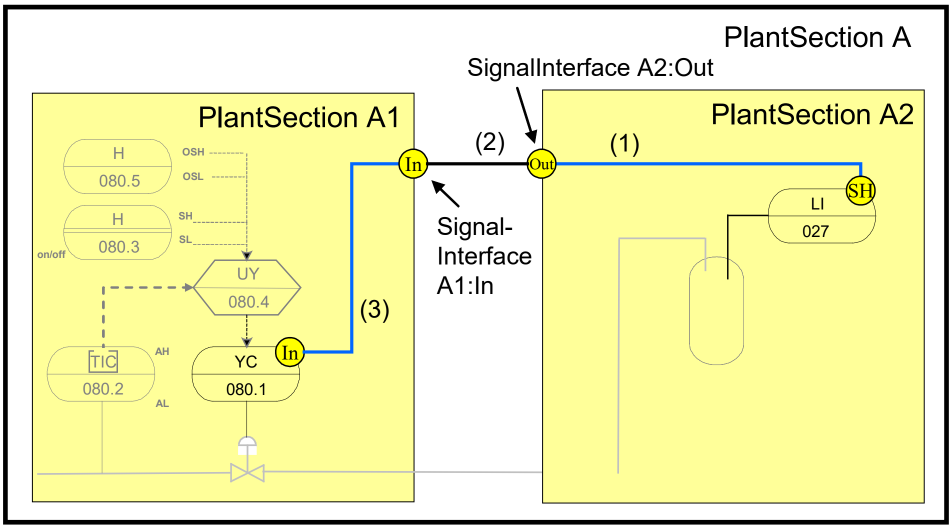
IEC

Figure 21 – XML code of the attribute type library

7.5.4 Mapping of indirect links between PCE requests of different plant sections

This 7.5.4 defines how to model indirect links between PCE requests of different plant sections. If a signal interface of a PCE request represents an external interface of the corresponding plant section, the internal signal interfaces of the considered PCE request shall be mapped to the external interfaces of the corresponding plant section. The mapping between a PCE request interface and an external interface of the corresponding plant section is defined by means of an additional InternalLink stored in the corresponding plant section.

The described mapping and a corresponding use case is illustrated in Figure 22 which gives an example in which a SignalLine couples a PCE request of PlantSection A1 with a PCE request of PlantSection A2. In this case, the plant sections themselves get each external signal interfaces “In” and “Out” respectively.



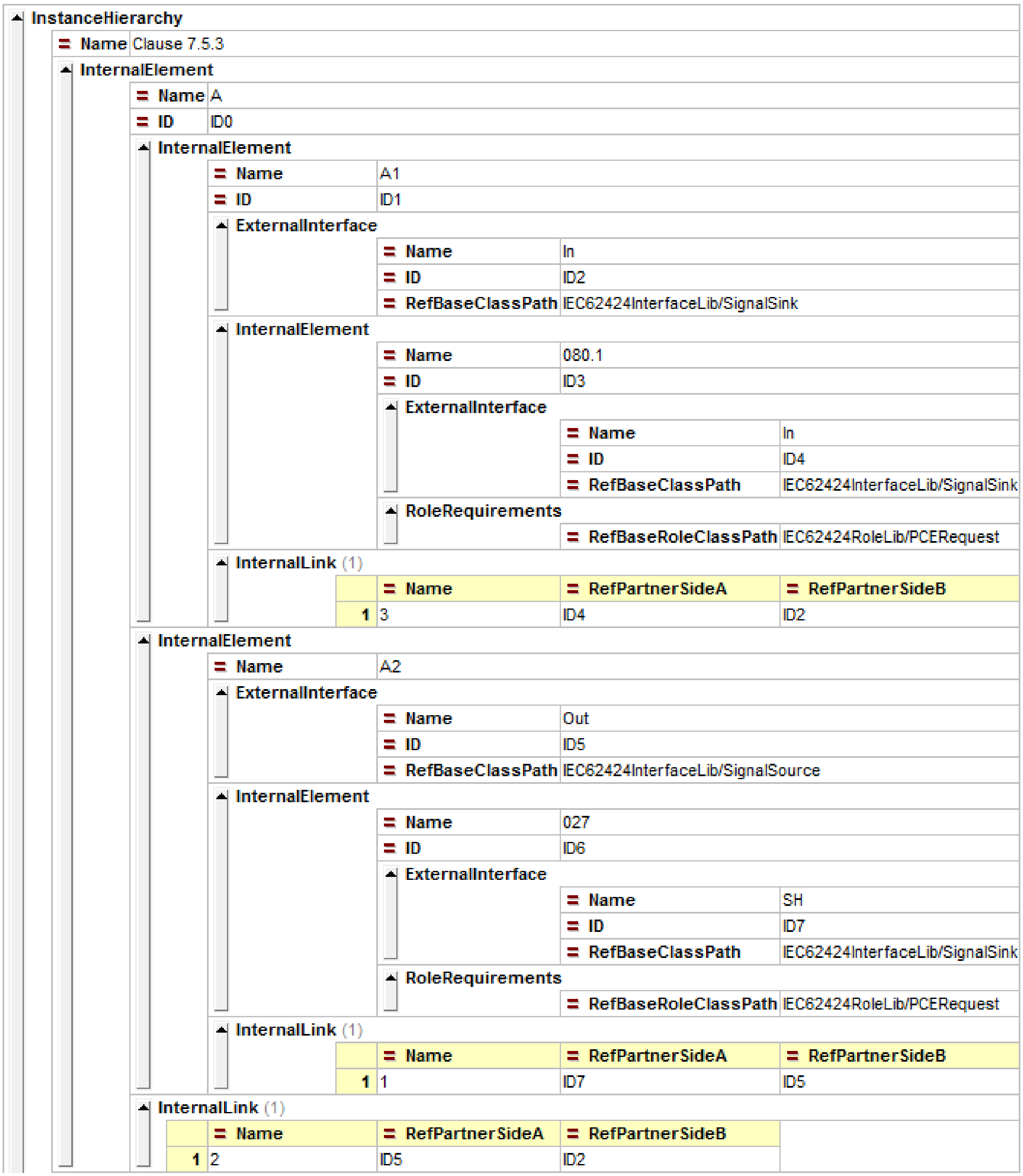
IEC

Figure 22 – Example of two plant sections and a signal connection via external interfaces

The SignalLine in this case is described in CAEX by means of three links:

1) link which is part of plant section A2, connecting A2/027/SH with A2/Out;
2) link which is part of the superior plant section A, connecting A2/Out with A1/In;
3) link which is part of plant section A1, connecting A1/In with A1/080.1/In.

A corresponding CAEX model is shown in Figure 23 which demonstrates how the signal line parts are defined separately in the InternalElements A, A1 and A2. Note that this simplified CAEX representation only models involved PCE requests.



```
InstanceHierarchy
  = Name  Clause 7.5.3
  InternalElement
    = Name  A
    = ID    ID0
    InternalElement
      = Name  A1
      = ID    ID1
      ExternalInterface
        = Name              In
        = ID                ID2
        = RefBaseClassPath  IEC62424InterfaceLib/SignalSink
      InternalElement
        = Name  080.1
        = ID    ID3
        ExternalInterface
          = Name              In
          = ID                ID4
          = RefBaseClassPath  IEC62424InterfaceLib/SignalSink
        RoleRequirements
          = RefBaseRoleClassPath  IEC62424RoleLib/PCERequest
      InternalLink (1)
           = Name  = RefPartnerSideA  = RefPartnerSideB
        1  3       ID4                ID2
    InternalElement
      = Name  A2
      ExternalInterface
        = Name              Out
        = ID                ID5
        = RefBaseClassPath  IEC62424InterfaceLib/SignalSource
      InternalElement
        = Name  027
        = ID    ID6
        ExternalInterface
          = Name              SH
          = ID                ID7
          = RefBaseClassPath  IEC62424InterfaceLib/SignalSink
        RoleRequirements
          = RefBaseRoleClassPath  IEC62424RoleLib/PCERequest
      InternalLink (1)
           = Name  = RefPartnerSideA  = RefPartnerSideB
        1  1       ID7                ID5
    InternalLink (1)
         = Name  = RefPartnerSideA  = RefPartnerSideB
      1  2       ID5                ID2
```

IEC

Figure 23 – Simplified CAEX model of indirect links between PCE requests across different plant hierarchy items

The full XML text for this example is shown in Figure 24. For better readability, mandatory attributes of the PCE request are not present.

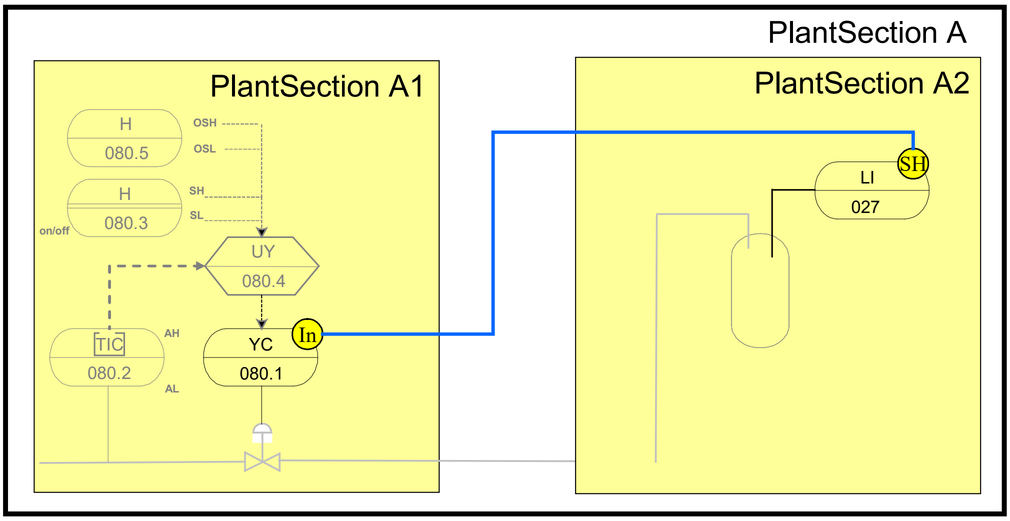
```
<InstanceHierarchy Name="Clause 7.5.3">
  <InternalElement Name="A" ID="ID0">
    <InternalElement Name="A1" ID="ID1">
      <ExternalInterface Name="In" ID="ID2" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
      <InternalElement Name="080.1" ID="ID3">
          <ExternalInterface Name="In" ID="ID4" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest"/>
      </InternalElement>
      <InternalLink Name="3" RefPartnerSideA="ID4" RefPartnerSideB="ID2"/>
    </InternalElement>
    <InternalElement Name="A2">
      <ExternalInterface Name="Out" ID="ID5" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
      <InternalElement Name="027" ID="ID6">
        <ExternalInterface Name="SH" ID="ID7" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest"/>
      </InternalElement>
      <InternalLink Name="1" RefPartnerSideA="ID7" RefPartnerSideB="ID5"/>
    </InternalElement>
    <InternalLink Name="2" RefPartnerSideA="ID5" RefPartnerSideB="ID2"/>
  </InternalElement>
  <InternalElement Name="A" ID="ID1">
    <InternalElement Name="A1" ID="ID2">
      <InternalElement Name="080.1" ID="ID3">
        <ExternalInterface Name="In" ID="ID4"/>
      </InternalElement>
    </InternalElement>
    <InternalElement Name="A2" ID="ID5">
      <InternalElement Name="027" ID="ID6">
        <ExternalInterface Name="SH" ID="ID7"/>
      </InternalElement>
    </InternalElement>
    <InternalLink Name="1" RefPartnerSideA="ID7" RefPartnerSideB="ID4"/>
  </InternalElement>
</InstanceHierarchy>
```

IEC

Figure 24 – Simplified CAEX model of indirect links between PCE requests across different plant hierarchy items

7.5.5 CAEX description of direct links between PCE requests of different plant sections

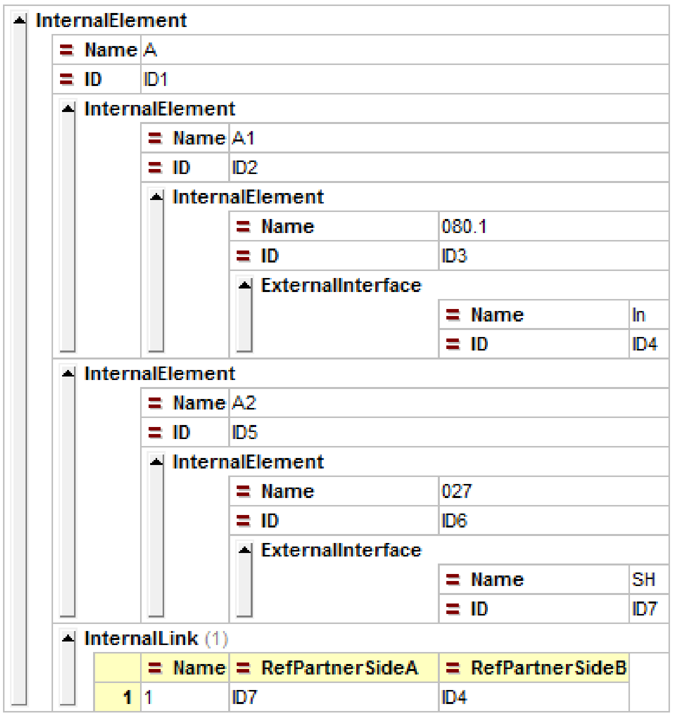
This 7.5.5 defines how to model direct links between PCE requests of different sections of a plant. If a signal interface of a PCE request is not represented by an external interface of the corresponding plant hierarchy item, a link to other PCE request interface of other plant hierarchy items shall be described in CAEX by a CAEX InternalLink that references both PCE request interfaces directly by means of their IDs (see Figure 25). The link is a part of a higher level plant hierarchy item.



IEC

Figure 25 – Example of two plant sections and a direct connection

A corresponding CAEX model is shown in Figure 26 which demonstrates how the signal line is defined as part of the InternalElement A (“PlantSection A”). Note that this simplified CAEX representation only models involved PCE requests.



- InternalElement
  - Name: A
  - ID: ID1
  - InternalElement
    - Name: A1
    - ID: ID2
    - InternalElement
      - Name: 080.1
      - ID: ID3
      - ExternalInterface
        - Name: In
        - ID: ID4
  - InternalElement
    - Name: A2
    - ID: ID5
    - InternalElement
      - Name: 027
      - ID: ID6
      - ExternalInterface
        - Name: SH
        - ID: ID7
  - InternalLink (1)

	Name	RefPartnerSideA	RefPartnerSideB
1	1	ID7	ID4

IEC

Figure 26 – Simplified CAEX model of direct links between PCE requests across different sections of a plant

The full XML text for this example is shown in Figure 27.

```
<InternalElement Name="A" ID="ID1">
  <InternalElement Name="A1" ID="ID2">
    <InternalElement Name="080.1" ID="ID3">
      <ExternalInterface Name="In" ID="ID4"/>
    </InternalElement>
  </InternalElement>
  <InternalElement Name="A2" ID="ID5">
    <InternalElement Name="027" ID="ID6">
      <ExternalInterface Name="SH" ID="ID7"/>
    </InternalElement>
  </InternalElement>
  <InternalLink Name="1" RefPartnerSideA="ID7" RefPartnerSideB="ID4"/>
</InternalElement>
```

IEC

Figure 27 – XML code of the simplified CAEX model

7.5.6 PCE loops

PCE loops are identified by a reference designation system. PCE loops will not be mapped to CAEX structural elements. The target tool has to know the special meaning of the reference designation system to be able to identify PCE loops.

8 Additional PCE attributes

The objective of this Clause 8 is to give a minimum set of typical attributes which are usually stored in P&ID systems and are relevant in the PCE environments. If applicable, these attributes shall be exchanged using the syntax as 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.

A CAEX based electronic data model of these attributes is provided in Clause D.1.

Table 6 – P&ID attributes relevant in PCE environment

Attributes	Syntax	CAEX mapping
Medium code	MediumCode	RoleClass/Attribute (see A.3.24 and D.1)
Medium code description	MediumCodeDescription	RoleClass/Attribute (see A.3.24 and D.1)
Material balance point	MaterialBalancePoint	RoleClass/Attribute (see A.3.24 and D.1)
Pressure rating	PressureRating	RoleClass/Attribute (see A.3.24 and D.1)
Design temperature	DesignTemperature	RoleClass/Attribute (see A.3.24 and D.1)
Design pressure	DesignPressure	RoleClass/Attribute (see A.3.24 and D.1)
Pipe specification	PipeSpecification	RoleClass/Attribute (see A.3.24 and D.1)
Pipe diameter size	PipeDiameterSize	RoleClass/Attribute (see A.3.24 and D.1)
Adjusted nominal pipe size	AdjustedNominalPipeSize	RoleClass/Attribute (see A.3.24 and D.1)
Heat tracing	HeatTracing	RoleClass/Attribute (see A.3.24 and D.1)
Heat tracing type	HeatTracingType	RoleClass/Attribute (see A.3.24 and D.1)
Heat tracing temperature set point	HeatTracingTemperatureSetPoint	RoleClass/Attribute (see A.3.24 and D.1)
Equipment/pipe flag	EquipmentPipeFlag	RoleClass/Attribute (see A.3.24 and D.1)
Equipment ID	EquipmentID	RoleClass/Attribute (see A.3.24 and D.1)
Pipe ID	PipeID	RoleClass/Attribute (see A.3.24 and D.1)
Insulation type	InsulationType	RoleClass/Attribute (see A.3.24 and D.1)
Insulation thickness	InsulationThickness	RoleClass/Attribute (see A.3.24 and D.1)

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	Syntax	CAEX mapping
Internal unique ID	InternalUniqueID	RoleClass/Attribute (see A.3.24 and D.1)
Short description	ShortDescription	RoleClass/Attribute (see A.3.24 and D.1)

Annex A
(normative)

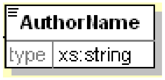
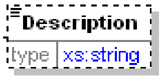
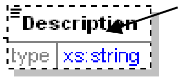
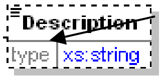
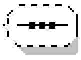
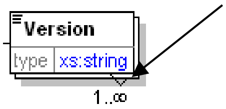
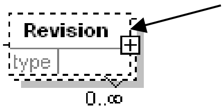
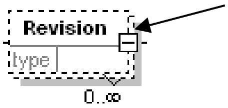
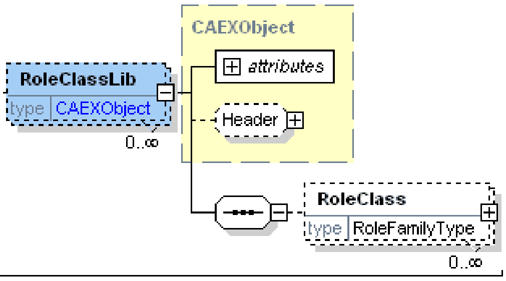
CAEX – Data model for machine information exchange

A.1 CAEX and its diagram conventions

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

The schema diagrams use the following conventions in order to illustrate the structure of the CAEX schema elements, the types of the elements, the attributes, the rules for optional elements and the repetitions (see Table A.1).

Table A.1 – XML notation conventions

Diagram element	Description	Example
Rectangle with solid border	Indicates a mandatory XML element	
Rectangle with dashed border	Indicates an optional XML element that may be implemented	
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 to the namespace of W3C (xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema). The targetNamespace of the CAEX types is http://www.dke.de/CAEX	 Used Namespace
Sequence	Indicates that the following elements shall be in the defined order	
Range	Indicates the allowed number of occurrences. For example 1 to infinite	 Range of the element
Plus sign	Indicates that this XML-element contains other elements. The containing elements are hidden	 Contains sub-elements
Minus sign	Indicates that all containing XML-elements are shown	 All contained sub-elements are shown
Greying background with dashed rectangle	Indicates that the shown elements are composed in a defined datatype. The name of the dataype is shown at the top of the dashed rectangle with dashed lines	 CAEXObject attributes RoleClassLib type CAEXObject 0..∞ Header RoleClass type RoleFamilyType 0..∞

A.2 General CAEX concepts

A.2.1 General CAEX terms

This A.2.1 describes all CAEX terms (see Table A.2).

Table A.2 – CAEX data types and elements *(1 of 3)*

Data types and elements	Detailed description
AdditionalInformation	Optional auxiliary field that may contain any additional information about a CAEX object. It shall be used in the substructure of the header.
Alias	Describes the alias name of an external CAEX file to enable referencing elements of the external CAEX file.
Attribute	Characterizes properties of a SystemUnitClass, RoleClass, InterfaceClass, InternalElement or RoleRequirements.
AttributeDataType	Describes the data type of the attribute using XML notation.
AttributeFamilyType	Defines base structures for attribute type definitions.
AttributeNameMapping	Allows the definition of the mapping between attributes of a related role class or its interfaces and attributes of the hosting system unit.
AttributeType	Defines base structures for attribute definitions.
AttributeTypeLib	Container element for a hierarchy of Attribute type definitions. CAEX supports multiple attribute type libraries.
AttributeValueRequirementType	Defines base structures for definition of value requirements of an attribute.
CAEXBasicObject	CAEX basic object that comprises a basic set of attributes and header information which exist for all CAEX elements.
CAEXFile	Root element of the CAEX schema.
CAEXObject	CAEX basis object derived from CAEXBasicObject, augmented by Name (required) and ID (optional).
ChangeMode	Optionally describes the change state of a CAEX object. If used, the ChangeMode shall have the following value range: state, create, delete and change. This information should be used for further change management applications.
Constraint	Element to restrict the range of validity of a defined attribute.
Copyright	Organizational information about copyright.
DefaultValue	A predefined default value for an attribute.
Description	Textual description for CAEX objects.
ExternalInterface	Description of an external interface of a RoleClass, SystemUnitClass or InternalElement.
ExternalReference	Container element for the alias definition of external CAEX files.
FileName	Describes the name of the CAEX file.
Header	Defines a group of organizational information, like description, version, revision, copyright, etc.
ID	Optional attribute that describes a unique identifier of a CAEX object.
InstanceHierarchy	Root element for a system hierarchy of object instances.
InterfaceClass	Class definition for interfaces.

Table A.2 *(2 of 3)*

Data types and elements	Detailed description
InterfaceClassLib	Container element for a hierarchy of InterfaceClass definitions. It shall contain any interface class definitions. CAEX supports multiple interface libraries.
InterfaceClassType	Shall be used for InterfaceClass definition, provides base structures for an interface class definition.
InterfaceFamilyType	Defines base structures for a hierarchical InterfaceClass tree. The hierarchical structure of an interface library has organizational character only.
InterfaceIDMapping	Allows the definition of the mapping between interfaces of a related role class and interfaces of the hosting system unit.
InternalElement	Shall be used in order to define nested objects inside of a SystemUnitClass or another InternalElement. Allows description of the internal structure of a CAEX object.
InternalElementType	Type for the definition of nested objects inside of a SystemUnitClass.
InternalLink	Shall be used in order to define the relationships CAEX ExternalInterfaces.
LastWritingDateTime	Date and time of the creation of the CAEX document.
MappingObject	Host element for AttributeNameMapping and InterfaceIDMapping.
MappingType	Base element for the MappingObject.
Name	Describes the name of the CAEX object.
NominalScaledType	Element to define constraints of nominal scaled attribute values.
OrdinalScaledType	Element to define constraints of ordinal scaled attribute values.
OriginName	Name of the origin of the CAEX document, e.g. the source engineering tool or an exporter software.
OriginID	Unique identifier of the origin of the CAEX document, e.g. a unique identifier of a source engineering tool or an exporter software. The ID shall not change even if the origin gets renamed.
OriginVendor	The vendor of the data source of the CAEX document.
OriginVendorURL	The vendor's URL of the data source of the CAEX document.
OriginVersion	Version of the origin of the CAEX document, e.g. the version of the source engineering tool or the exporter software.
OriginRelease	Release information of the origin of the CAEX document, e.g. the version of the source engineering tool or the exporter software.
OriginProjectTitle	The title of the corresponding source project.
OriginProjectID	A unique identifier of the corresponding source project.
Path	Describes the path of the external CAEX file. Absolute and relative paths are allowed.
RefAttributeType	Refences an attribute type in the attribute library.
RefBaseClassPath	Stores the reference of a class to its base class. References contain the full path to the referred class object.
RefBaseSystemUnitPath	Stores the reference of an InternalElement to a class or instance definition. References contain the full path information.
RefSemantic	A reference to a definition of a defined attribute, e.g. to an attribute in a standardized library, this allows the semantic definition of the attribute.

Table A.2 *(3 of 3)*

Data types and elements	Detailed description
RequiredMaxValue	Element to define a maximum value of an attribute.
RequiredMinValue	Element to define a minimum value of an attribute.
RequiredValue (NominalScaledType)	Element to define a required value of an attribute. It may be defined multiple times in order to define a discrete value range of the attribute.
RequiredValue (OrdinalScaledType)	Element to define a required value of an attribute.
Requirements	Defines informative 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.
RoleClassFamilyType	Defines base structures for a hierarchical RoleClass tree. The hierarchical structure of a role library has organizational character only.
RoleClassLib	Container element for a hierarchy of RoleClass definitions. It shall contain all RoleClass definitions. CAEX supports multiple role libraries.
RoleClassType	Shall be used for RoleClass definition, provides base structures for a role class definition.
RoleRequirements	Describes role requirements of an InternalElement. It allows the definition of a reference to a RoleClass and the specification of role requirements like required attributes and required interfaces.
SchemaVersion	Describes the version of the schema. Each CAEX document shall specify which CAEX version it requires. The version number of a CAEX document shall fit to the version number specified in the CAEX schema file.
SourceDocumentInformationType	Defines a structure to model information about the data source of the present CAEX document.
SourceObjID	Attribute representing the ID of the source object in the source data model.
SourceObjectInformation	Organizational information about the source of the corresponding CAEX object.
SuperiorStandardVersion	Describes the version of a superior standard. The version string is defined in the superior standard.
SupportedRoleClass	Allows the association of the corresponding SystemUnitClass to a RoleClass. This describes, which role the SystemUnitClass can play. A SystemUnitClass may reference multiple roles.
SystemUnitClass	Shall be used for SystemUnitClass definition, provides definition of a class of a SystemUnitClass type.
SystemUnitClassLib	Container element for a hierarchy of SystemUnitClass definitions. It shall contain all SystemunitClass definitions. CAEX supports multiple SystemUnitClass libraries.
SystemUnitClassType	Defines base structures for a SystemUnit class definition.
SystemUnitFamilyType	Defines base structures for a hierarchical SystemUnitClass tree. The hierarchical structure of a SystemUnit library has organizational character only.
Unit	Describes the unit of a variable.
UnknownType	Element to define constraints for attribute values of an unknown scale type.
Value	Element to describe the value of an attribute.
Version	Organizational information about the state of the version.

A.2.2 General CAEX concept description

A.2.2.1 CAEX basic concept

The general goal of CAEX is the vendor independent storage of hierarchical object information. Object-oriented concepts as encapsulation, classes, class libraries, instances, instance hierarchies, inheritance, relations, attributes, attribute types, attribute type libraries and interfaces are explicitly supported. XML provides means for modeling classes and instances.

A CAEX class or AttributeType represents a re-usable data model (template) of a real physical or logical item and is modeled either as a SystemUnitClass, a RoleClass, an Interface Class or an AttributeType.

a) **SystemUnitClasses** describe physical or logical plant objects or units including their technical realization and internal architecture. They consist of attributes, interfaces, nested internal elements and relations between the internal elements. The internal elements may contain further nested elements – this allows for the description of predefined structures with multiple hierarchy levels. The concept of internal elements allows describing the internal architecture of a plant object.

   SystemUnitClasses are collected in libraries of the type **SystemUnitClassLib**: This CAEX element allows collecting an arbitrary number of objects of the type SystemUnitClassType within a library. CAEX supports the definition of multiple SystemUnitClass libraries. SystemUnitClasses can be arranged within the library as a tree in order to depict the user's library breakdown structure. A SystemUnitClass can further be inherited from another SystemUnitClass by means of a reference. SystemUnitClassLibs can for example be used to store product catalogues.

b) **RoleClasses** also describe physical or logical plant objects, but, compared to SystemUnitClasses, they are an abstraction of a concrete technical realization. RoleClasses contain attributes and interfaces, but do not describe the concrete internal implementation of the object. A RoleClass is used in order to define the meaning (semantics of) and the requirements for a plant object.

   RoleClassLib: This CAEX element allows for collecting an arbitrary number of objects of the type RoleClassType within a library. CAEX supports the definition of multiple RoleClass libraries. RoleClasses can be arranged within the library as a tree in order to depict the user's library breakdown structure. A RoleClass can further be inherited from another RoleClass by means of a reference.

c) **InterfaceClasses** describe types of interfaces. InterfaceClasses comprise a set of specific attributes and are used in order to specify interfaces for RoleClasses, SystemUnitClasses and InternalElements. Interfaces are required in order to define relations between objects.

   InterfaceClassLib: This CAEX element allows collecting an arbitrary number of objects of the type InterfaceClassType within a library. CAEX supports definition of multiple InterfaceClass libraries. InterfaceClasses can be arranged within the library as a tree in order to depict the user's library breakdown structure. An InterfaceClass can further be inherited from another InterfaceClass by means of a reference.

d) **AttributeTypes** describe types of attributes. Attributes are required in order to define properties of objects comprising their attribute names, values etc.

   AttributeTypeLib: This CAEX element allows for collecting an arbitrary number of attribute types within a library. CAEX supports the definition of multiple AttributeType libraries. AttributeTypes can be arranged within the library as a tree in order to depict the user's breakdown structure. An AttributeType can further be inherited from another AttributeType by means of a reference.

SystemUnitClasses, RoleClasses, InterfaceClasses and AttributeTypes finally serve for predefining useful templates to be instantiated and re-used. The CAEX element **InstanceHierarchy** allows the storage of those object instances. A CAEX object instance is a concrete data object that corresponds to a certain real physical or logical item. A CAEX instance is modeled either as an InternalElement or an ExternalInterface. The term "instance" describes an individual object with individual properties. Each class can be instantiated multiple times, e.g. the object instances “c1”, “c2” and c3” may be instantiated from a single class “c”.

The CAEX InstanceHierarchy consists of an arbitrary number of internal elements which are recursively nested – this allows for describing arbitrary object hierarchies. The parent-child relation between instances is explained in A.2.8.2. CAEX supports multiple instance hierarchies.

The InstanceHierarchy can be used in one of the following ways.

a) **Working without classes**: Instances are defined in the instance hierarchy in the form of nested InternalElements as an object tree. For each single object, all required attributes, interfaces and links etc. are defined on instance level. This workflow supports data storage without classes at all. This might be of interest for example if existing libraries are not the objective of the data exchange.
b) **Working with classes only:** The desired plant hierarchy is defined by a single InternalElement in the InstanceHierarchy. This InternalElement references a complex SystemUnitClass which comprises the complete system description including the plant topology, units, components, attributes etc. This workflow is of interest if the plant or unit structure to be stored in CAEX is a standard solution and is intended to be used several times.
c) **Mixed workflow:** This is the typical workflow for practical use. Typical components are defined as SystemUnitClasses; sub-structures of the SystemUnitClass are defined by the aggregation of objects as InternalElements. Attributes may be predefined, default attribute values may be set. The InstanceHierarchy is being used for the plant topology definition. In the next step, each defined internal hierarchical element can be associated with a role class in order to define the requirements to this object. Finally, it can be associated to a SystemUnitClass that describes the technical implementation of the object.

For a detailed CAEX data definition for classes and instances, see A.3.6 to A.3.13 and A.3.20.

A.2.2.2 General CAEX document provisions

Regarding the general application of CAEX, the following provisions apply.

- CAEX is defined as XML schema in this standard. The corresponding XML schema file shall be named “CAEX_ClassModel_V.3.0.xsd”.
- Each CAEX document which claims conformity to this standard shall be well formed against the CAEX schema file. The CAEX schema definition (xsd file) allows the automatic verification of whether the CAEX document conforms to the CAEX schema file.
- In addition to the conformity regarding the CAEX schema, a CAEX document shall conform to additional normative provisions which are defined in this standard.
- A CAEX document shall not be considered as a database, CAEX is a static data exchange format. The validity of stored data is the responsibility of the source tool or the corresponding exporter/importer tool. CAEX does not provide software functionality, no semantic checks, no data consistency checks and no data plausibility checks.

A.2.2.3 Superior standard version

A CAEX document may follow superior standards with additional rules or normative provisions beyond this part of the IEC 62424. In this case, the following normative provisions apply.

- Information about superior standards shall be stored in the CAEX element “SuperiorStandardVersion”. The identification string of the superior standard is defined in the superior standard.
- The CAEX element “SuperiorStandardVersion” may be used multiple times if multiple superior standards shall be applied.

For a detailed CAEX data definition of the CAEX superior standard version, see A.3.3.

A.2.2.4 CAEX document version

A CAEX document shall be compatible to one CAEX XML schema version. In order to avoid version conflicts, the following normative provisions apply.

- Each CAEX document shall set the mandatory CAEX attribute “SchemaVersion” to a version string. This value corresponds to the version of the CAEX schema file. The version according of the present standard shall be “3.0”.
- Every referenced external CAEX document shall follow the same schema versions specified in the CAEX version specification of the referencing CAEX document.
- Mixing of external CAEX documents with different CAEX schema versions is explicitly forbidden.

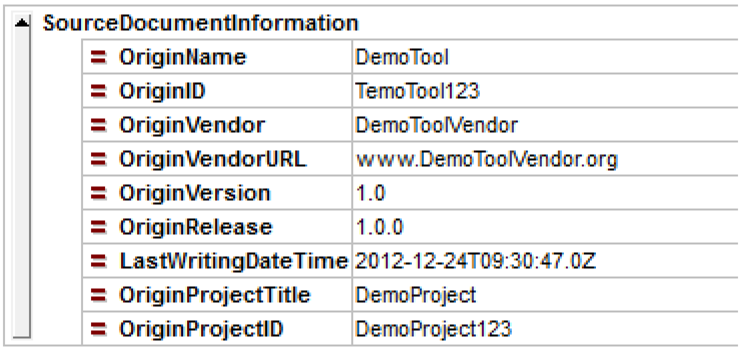
For a detailed CAEX data definition of the CAEX document version, see A.3.2.

A.2.2.5 Storing information about the source of a CAEX document

CAEX is able to store both standardized as well as proprietary or user defined objects and attributes with a non-standardized semantic. In order to interpret that information, it is needed to model information about the origin or the CAEX document. The origin may be a source engineering tool or a standard. The following provisions apply.

- Each CAEX document shall provide information about the origin of the CAEX document.
- In a data exchange tool chain, all participating tools shall add their origin information in the CAEX document. In the result, a CAEX document may contain information about multiple source tools of a data exchange tool chain.
- A tool may remove the origin information of other tools. This can hinder the iterative data exchange with the other tools: hence the removal of origin information of other tools is not recommended.
- The origin information shall be stored by means of the CAEX element “SourceDocument-Information” of the root object of the CAEX document. This information contains the following aspects:
  - A suitable name of the origin shall be stored in the CAEX attribute “OriginName”.
  - A unique ID of the origin shall be stored in the CAEX attribute “OriginID”. This ID should not change over the lifetime of the origin.
  - Optionally, the vendor of the origin shall be stored in the CAEX attribute “OriginVendor”.
  - Optionally, the vendors URL of the origin shall be stored in the CAEX attribute “OriginVendorURL”.
  - The version of the origin shall be stored in the CAEX attribute “OriginVersion”.
  - Optionally, the release information of the origin shall be stored in the CAEX attribute “OriginRelease”.
  - The data and time of the CAEX document creation shall be stored in the CAEX attribute “LastWritingDateTime”.
  - Optionally, an origin project title shall be stored in the CAEX attribute “OriginProjectTitle”.
  - Optionally, an origin project identifier shall be stored in the CAEX attribute “OriginProjectID”. This ID should not change over time.
- The values of the origin information shall be embedded by the tool that creates the CAEX document and shall be of type xs:string.

Figure A.1 illustrates an example of the source related information.



SourceDocumentInformation

OriginName	DemoTool
OriginID	TemoTool123
OriginVendor	DemoToolVendor
OriginVendorURL	www.DemoToolVendor.org
OriginVersion	1.0
OriginRelease	1.0.0
LastWritingDateTime	2012-12-24T09:30:47.0Z
OriginProjectTitle	DemoProject
OriginProjectID	DemoProject123

IEC

Figure A.1 – XML text of the CAEX source document information

A detailed CAEX data definition including required data types of the CAEX SourceDocument-Information is defined in A.3.4 and A.3.26.

A.2.2.6 CAEX object identification

CAEX follows the object-oriented paradigm. All engineering information is modelled as object or belongs to an object. But, in a heterogeneous tool landscape, different engineering tools use different concepts for the identification of objects, e.g. a unique name, a unique identifier or a unique path. Some tools allow changes of the identifiers over the life time, others do not. IEC 62424 enables the data exchange between different engineering tools with such individual object identification concepts. Owing to the described characteristics, this standard neutralizes this variety and defines one mandatory object identification concept.

Regarding the object identification, the following provisions apply.

- CAEX classes or types (RoleClasses, InterfaceClasses, SystemUnitClasses and AttributeTypes), attributes, libraries and CAEX InstanceHierarchies shall be identified by their CAEX tag “Name”. Their names shall be unique across their siblings or across all child elements of the same CAEX parent element over their life time. This shall assure that referencing a library, a class, a type or an attribute by its path delivers a unique result.
- Referencing of classes shall be done via full paths using the corresponding path separators according to A.2.9.
- All CAEX instances (InternalElements and ExternalInterfaces) shall be identified by their CAEX tag “ID”. Once created, the identifier of the same InternalElement or ExternalInterface shall not change over the life time of the corresponding object. To achieve this, it is recommended that the identifier should be a Universal Unique Identifier (UUID) according to ISO/IEC 9834-8 or should follow a suited unique naming convention that provides uniqueness over time.
- A possible implementation of the UUID is the Global Unique Identifier (GUID). Alternatives to a UUID are allowed if the uniqueness is ensured, e.g. a unique path string.

  NOTE 1 In this standard, UUIDs are presented in a short form like “GUID1” or “ID1” etc. in order to improve the readability of the present document.
- For object instances, the CAEX tag “Name” is a display name; it has informative character only and may change over time or with the tool. Consequently, InternalElements or ExternalInterfaces may have the same name.
- Every reference to an object instance (either InternalElement or ExternalInterface) shall use the “ID” value of the object instance.

  NOTE 1 E.g. the InternalLink uses the identifier of the corresponding interfaces.

  NOTE 2 Examples for object identification are given in Figure A.3 and Figure A.17.

A detailed CAEX data definition for the object identifier is given in A.3.20.

A.2.2.7 Storage of version information

CAEX allows transporting static version information for each object. For this, all CAEX objects are directly or indirectly derived from the CAEX base type “CAEXBasicObject” which defines a subset of optional and generic version information. These properties are useful for iterative data exchange with repeated export and import.

A detailed CAEX data definition is given in A.3.15 and A.3.19.

The data type definition is characterized by the following properties:

- **ChangeMode**: This optional attribute is intended to give information about the change state of an object compared to a previous data exchange. Valid values of ChangeMode are defined in CAEX, they are “state”, “create”, “delete” and “change” (see A.3.29). The value “state” shall be used for objects that have not changed since previous data exchange. The value “create” shall be used for new objects that have been created. The value “delete” shall be used if an object is to be deleted. The object is therefore not physically removed out of the CAEX file but marked as to be deleted. The value “change” shall be used if the object has changed. The ChangeMode is only valid for the item itself. If for example an attribute has changed its value, only the value is marked with the ChangeMode value “change”, neither the attribute nor the host object of the attribute;
- **Description, Version, Revision, Copyright**: These attributes and elements allow storage of version information for each object;
- **AdditionalInformation:** This attribute allows storage of arbitrary additional information of any type;
- **SourceObjectInformation, OriginID and SourceObjID:** These CAEX items allow storing organizational information about the origin of each CAEX object.

Regarding the application of the present version related elements and attributes, the following normative provisions apply in addition to the CAEX schema.

- Every CAEX library shall define its version number utilizing the CAEX element “Version”. The syntax and semantic of the value of the version number is not defined in this standard.
- If required, CAEX classes shall define their version number utilizing the CAEX element “Version”. The syntax and semantic of the version number of classes within a library is not defined in this standard.
- Libraries and instance hierarchies with same names are forbidden to be stored in the same CAEX file.

  NOTE 1 This ensures the uniqueness of library names within a CAEX file.
- The creator of a CAEX document shall ensure that only version compatible classes and external documents are referenced.
- A new version of a class shall be modelled as a new class with another name. Within the new class, the full path of the old version of the class shall be stored in the CAEX tag “OldVersion” of the CAEX element “Revision”.

  NOTE 2 This provision supports tracking of changes across different versions of a class.

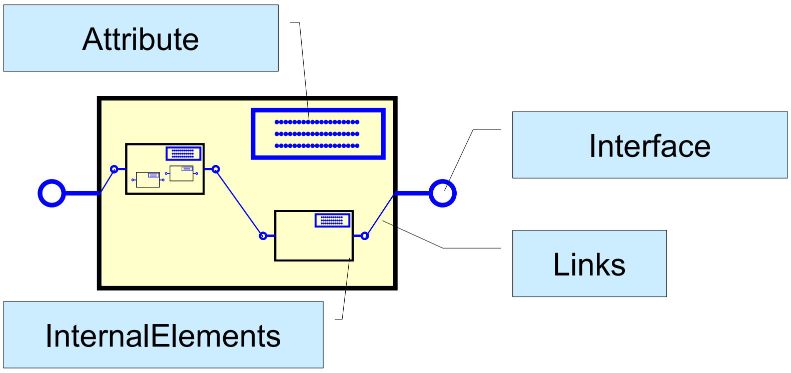
A.2.3 Data definition of SystemUnitClass

A.2.3.1 Architecture of a SystemUnitClass

A SystemUnitClass is being identified by the following properties (see Figure A.2).

- **Attribute:** Allows the specification of object attributes;
- **ExternalInterface:** Allows the specification of object interfaces;
- **InternalElement:** Allows the specification of nested internal objects;

- **SupportedRoleClass**: Allows specification of supported RoleClasses;
- **InternalLink:** Allows specification of relations between interfaces.



IEC

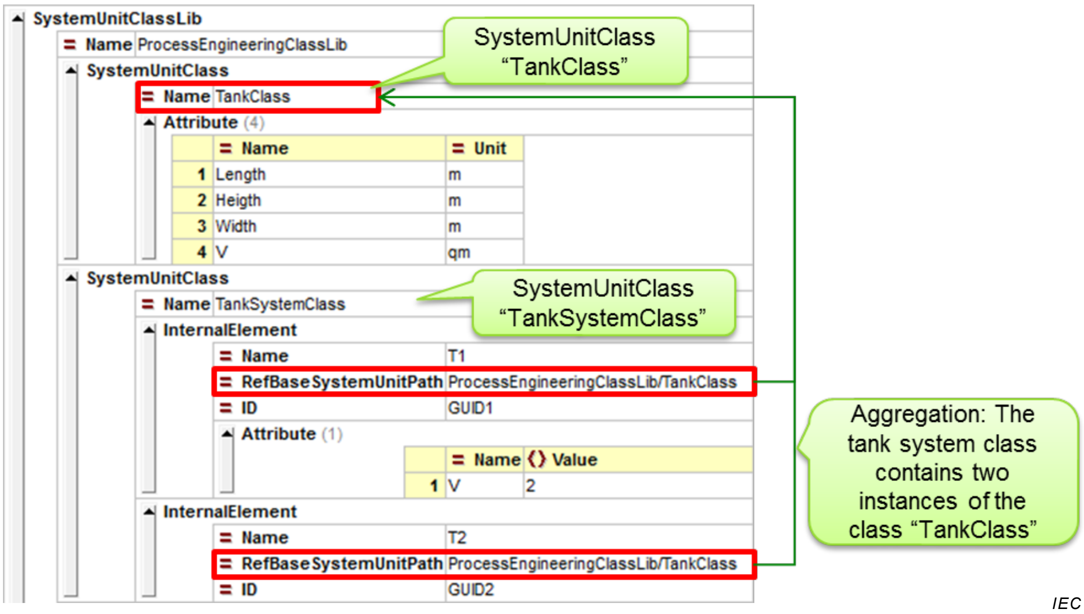
Figure A.2 – CAEX architecture of a SystemUnitClass

The general concept of SystemUnitClasses is described in A.2.2. For a detailed CAEX data definition, see A.3.12, A.3.13 and A.3.27.

A.2.3.2 Example

The following example demonstrates the concepts of SystemUnitClasses. Figure A.3 presents the SystemUnitClassLib "ProcessEngineeringClassLib" which contains 2 classes.

- The class "TankClass" presents the architecture of a simple SystemUnitClass with user defined attributes.
- The class "TankSystemClass" aggregates two objects "T1" and "T2" that are based on the "TankClass". Both objects inherit the attributes of the "TankClass". "T1" specifies the value of the inherited attribute "V". The use of attributes is furthermore described in A.2.4.



```
SystemUnitClassLib
  = Name ProcessEngineeringClassLib
  SystemUnitClass
      = Name TankClass
      Attribute (4)
          = Name     = Unit
        1 Length     m
        2 Heigth     m
        3 Width      m
        4 V          qm
  SystemUnitClass
      = Name TankSystemClass
      InternalElement
          = Name                 T1
          = RefBaseSystemUnitPath ProcessEngineeringClassLib/TankClass
          = ID                   GUID1
          Attribute (1)
                = Name {} Value
              1 V       2
      InternalElement
          = Name                 T2
          = RefBaseSystemUnitPath ProcessEngineeringClassLib/TankClass
          = ID                   GUID2
```

SystemUnitClass "TankClass"

SystemUnitClass "TankSystemClass"

Aggregation: The tank system class contains two instances of the class "TankClass"

IEC

Figure A.3 – Example of a SystemUnitClassLib

In Figure A.4, the full XML text is shown for this example.

```
<SystemUnitClassLib Name="ProcessEngineeringClassLib">
  <SystemUnitClass Name="TankClass">
    <Attribute Name="Length" Unit="m"/>
    <Attribute Name="Heigth" Unit="m"/>
    <Attribute Name="Width" Unit="m"/>
    <Attribute Name="V" Unit="qm"/>
  </SystemUnitClass>
  <SystemUnitClass Name="TankSystemClass">
    <InternalElement Name="T1" RefBaseSystemUnitPath="ProcessEngineeringClassLib/TankClass" ID="GUID1">
      <Attribute Name="V">
        <Value>2</Value>
      </Attribute>
    </InternalElement>
    <InternalElement Name="T2" RefBaseSystemUnitPath="ProcessEngineeringClassLib/TankClass ID="GUID2"/>
  </SystemUnitClass>
</SystemUnitClassLib>
```

IEC

Figure A.4 – XML code of the example of a SystemUnitClassLib

A.2.4 Definition of attributes

A.2.4.1 Architecture of an attribute

Attributes specify properties of an object and individual values. Beside the attribute name, CAEX defines the following information.

- **Value**: This element allows the definition of the property value, e.g. “3.5”. The decimal separators shall be selected according to the AttributeDataType definition, e.g. “xs:float” requires a “.” as decimal separator.
- **Unit**: This element defines the unit of the attribute, e.g. “m”.
- **AttributeDataType**: This element defines the data type of the attribute. If this optional attribute is not defined, the data type is assumed to be “xs:string”, whereas “xs” represents for example the used XML namespace "http://www.w3.org/2001/XMLSchema". If the attribute is defined, the value shall use the standard XML data types, e.g. “xs:boolean”, “xs:integer”, “xs:float”, etc. An overview gives http://www.w3.org/TR/xmlschema-2/#built-in-datatypes. Corresponding to the data type, the values of an attribute shall conform to the XML rules, e.g. “xs:boolean” expects the values “true” or “false”, whereas “TRUE” or “FALSE” does not conform.
- **RefAttributeType:** This element stores a path reference to an attribute type defined in a AttributeTypeLib. If the referenced attribute type bases on a XML data type, the AttributeDataType shall provide this base type of the referenced attribute. If the referenced attribute type does not base on an XML standard base type, the Attribute-DataType may remain empty or not present. An example for this is given in Figure A.7.
- **DefaultValue**: This element allows for the definition of the initial value of the attribute. It may be overwritten by the value definition.
- **Constraints:** This element allows for the definition of constraints. CAEX supports two constraint types: OrdinalScaledType and NominalScaledType. OrdinalScaledType allows for the definition of the “required value”, the “max value” and the “min value”. NominalScaledType allows for the definition of a discrete value range, e.g. the allowed value range of an attribute “safe” might have the value range “yes” and “no”.
- **RefSemantic:** This element allows for the definition of a semantic reference to a normative or informal dictionary, e.g. SI units, IEC 61987-1, a web site, etc.
- **Attribute:** This element allows for the definition of attributes which may contain further attributes. This enables the description of hierarchical attribute structures.

The following properties related to attribute information are normative.

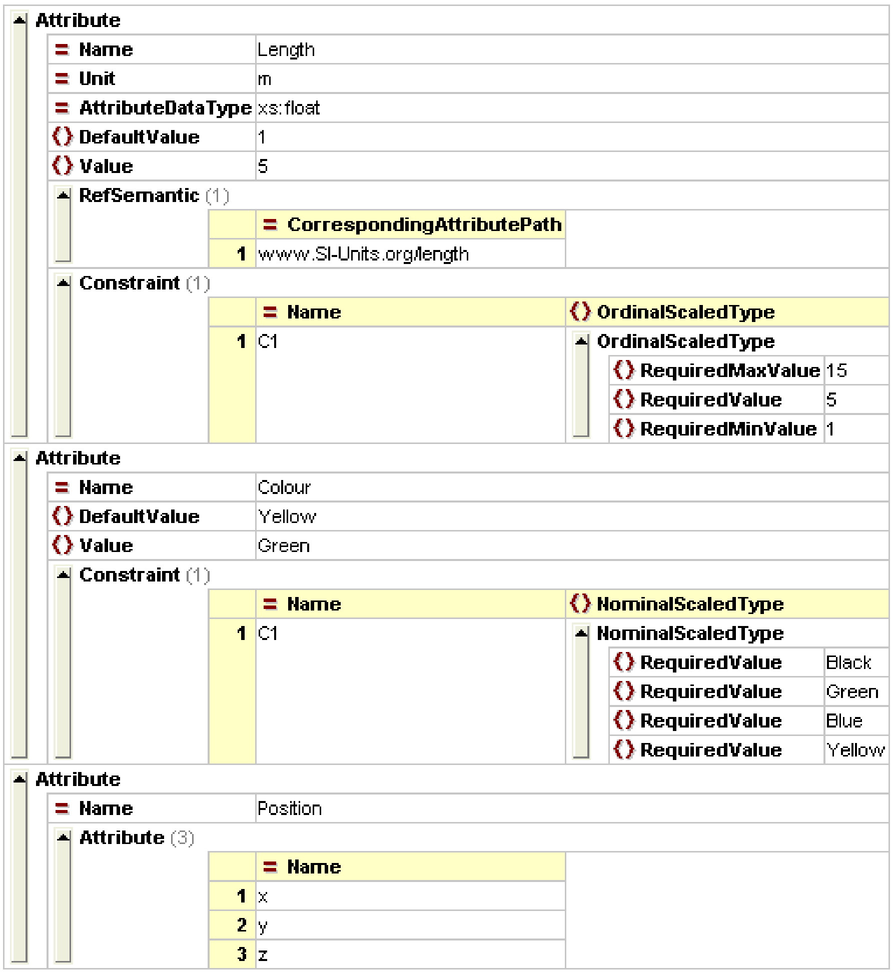
- The semantic of an attribute is in general user defined. The syntax and semantic of PCE request related attributes are defined in 7.5.2 and 7.5.3 of this standard. Other standards may provide further semantic definitions and are not in the scope of this standard.
- In case the value of a CAEX attribute is not present, but a DefaultValue is present, then the DefaultValue shall be used instead of the Value.
- The correctness of attributes is not in the scope of CAEX, and it does not provide consistency checks of constraints and attribute values; this is a task of a source or target tool.

For a detailed CAEX data definition, see A.3.18.

A.2.4.2 Example

Figure A.5 presents 3 Attributes with different properties.

- The attribute “Length” explains the concept of RefSemantic and OrdinalScaledType constraints. The value of this attribute shall be between 1 and 15, the required value is 5.
- The attribute “Colour” explains the concept of DefaultValue and NominalScaledType constraints. The DefaultValue is “Yellow”, which is overwritten by the value definition “Green”. The NominalScaledType constraints define the allowed discrete value range.
- The attribute “Position” explains the concept of nested attributes by means of the sub attributes “x”, “y”, “z”.



Attribute
= Name Length
= Unit m
= AttributeDataType xs:float
{} DefaultValue 1
{} Value 5
RefSemantic (1)

	= CorrespondingAttributePath
1	www.SI-Units.org/length

Constraint (1)

	= Name	{} OrdinalScaledType
1	C1	OrdinalScaledType {} RequiredMaxValue 15 {} RequiredValue 5 {} RequiredMinValue 1

Attribute
= Name Colour
{} DefaultValue Yellow
{} Value Green
Constraint (1)

	= Name	{} NominalScaledType
1	C1	NominalScaledType {} RequiredValue Black {} RequiredValue Green {} RequiredValue Blue {} RequiredValue Yellow

Attribute
= Name Position
Attribute (3)

	= Name
1	x
2	y
3	z

IEC

Figure A.5 – Examples of attributes

The full XML text for this example is shown in Figure A.6.

```
<Attribute Name="Length" Unit="m" AttributeDataType="xs:float">
    <DefaultValue>1</DefaultValue>
    <Value>2</Value>
    <RefSemantic CorrespondingAttributePath="www.SI-Units.org/length"/>
    <Constraint Name="C1">
        <OrdinalScaledType>
            <RequiredMaxValue>15</RequiredMaxValue>
            <RequiredValue>5</RequiredValue>
            <RequiredMinValue>1</RequiredMinValue>
        </OrdinalScaledType>
    </Constraint>
</Attribute>
<Attribute Name="Colour">
    <DefaultValue>Yellow</DefaultValue>
    <Value>Green</Value>
    <Constraint Name="C1">
        <NominalScaledType>
            <RequiredValue>Black</RequiredValue>
            <RequiredValue>Green</RequiredValue>
            <RequiredValue>Blue</RequiredValue>
            <RequiredValue>Yellow</RequiredValue>
        </NominalScaledType>
    </Constraint>
</Attribute>
<Attribute Name="Position">
    <Attribute Name="x"/>
    <Attribute Name="y"/>
    <Attribute Name="z"/>
</Attribute>
```

IEC

Figure A.6 – XML code of the example

A.2.5 Data definition of an AttributeType

A.2.5.1 Architecture of an AttributeType

CAEX allows for the definition of attribute type libraries containing user defined or standardized attribute types which are subject of re-use. Attribute type libraries can be used in order to define complex attributes or they may be used in order to define a predefined set of attributes with a well defined syntax and semantic. Attribute types are characterized by the following CAEX elements.

- **AttributeType:** Attributes allow for the specification of an attribute type. The type has the same architecture as a general CAEX attribute (see A.2.4.1).
- **Attribute:** Attributes allow for the specification of attribute structures.

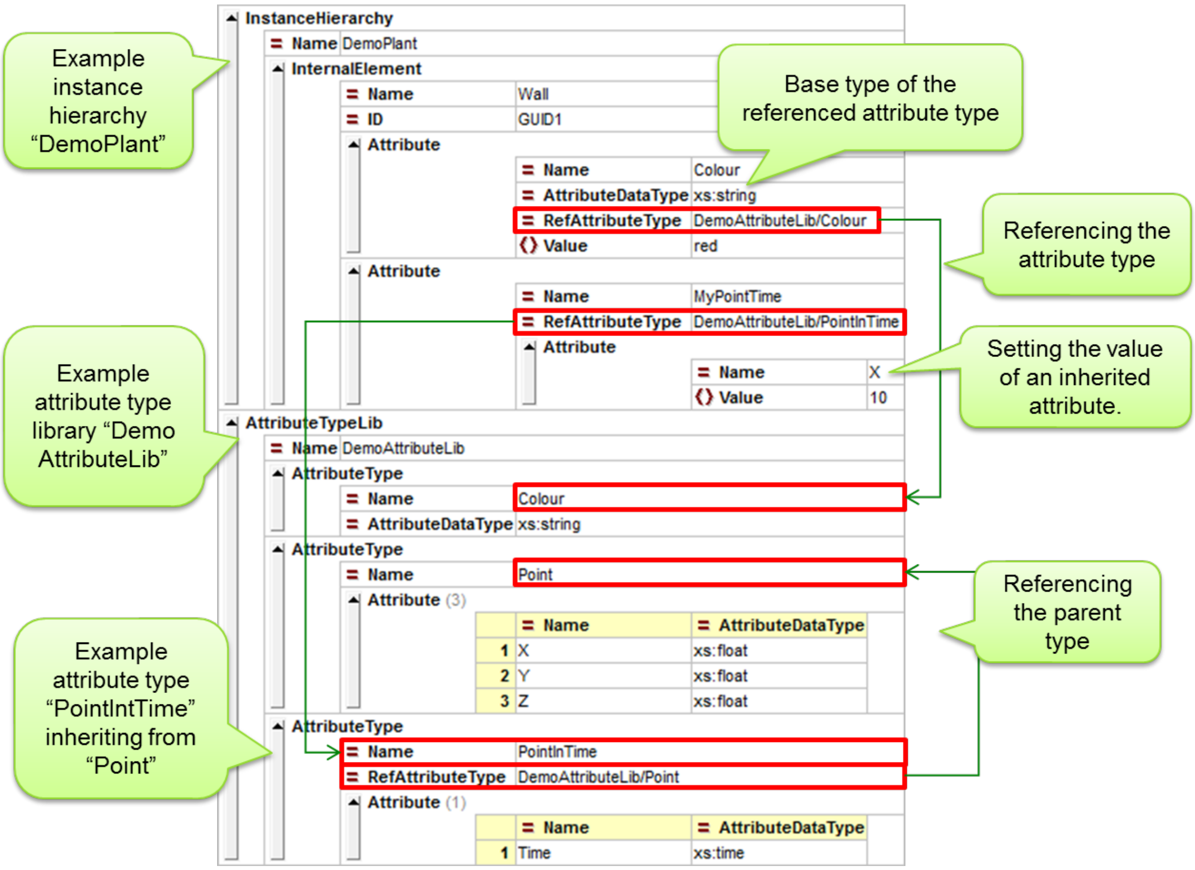
The following CAEX properties related to attribute types are normative in addition to the CAEX schema.

- Attribute types may contain child attributes. The concept of child-attributes allows for describing user defined attribute structures.
- Attribute types may contain child attribute types. The concept of child-attribute-types allows for describing a user defined hierarchy of attribute types, the hierarchy itself has no semantic. The hierarchy may be used to depict the users' library structure.
- Inheritance relations between attribute types are defined by means of a reference to a parent attribute type.

A detailed CAEX data definition is provided in A.3.14 and A.3.18.

A.2.5.2 Example

Figure A.7 presents an example of attribute type library and illustrates the application in an example instance hierarchy.

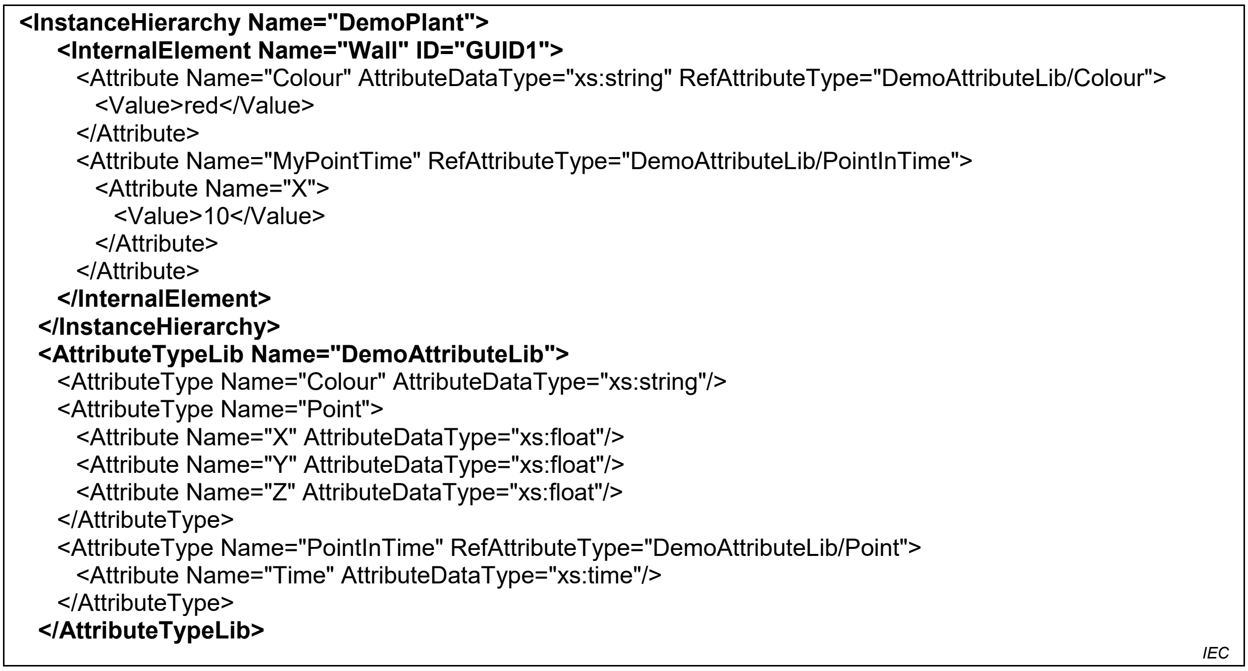


```
InstanceHierarchy
  Name DemoPlant
  InternalElement
    Name Wall
    ID GUID1
    Attribute
      Name Colour
      AttributeDataType xs:string
      RefAttributeType DemoAttributeLib/Colour
      Value red
    Attribute
      Name MyPointTime
      RefAttributeType DemoAttributeLib/PointInTime
      Attribute
        Name X
        Value 10
AttributeTypeLib
  Name DemoAttributeLib
  AttributeType
    Name Colour
    AttributeDataType xs:string
  AttributeType
    Name Point
    Attribute (3)
          Name  AttributeDataType
        1 X     xs:float
        2 Y     xs:float
        3 Z     xs:float
  AttributeType
    Name PointInTime
    RefAttributeType DemoAttributeLib/Point
    Attribute (1)
          Name  AttributeDataType
        1 Time  xs:time
```

Example instance hierarchy "DemoPlant"
Base type of the referenced attribute type
Referencing the attribute type
Setting the value of an inherited attribute.
Example attribute type library "Demo AttributeLib"
Referencing the parent type
Example attribute type "PointIntTime" inheriting from "Point"
IEC

Figure A.7 – Example of an AttributeTypeLib and its application in an instance hierarchy

The XML code of the AttributeTypeLib example is shown in Figure A.8.



```
<InstanceHierarchy Name="DemoPlant">
    <InternalElement Name="Wall" ID="GUID1">
      <Attribute Name="Colour" AttributeDataType="xs:string" RefAttributeType="DemoAttributeLib/Colour">
        <Value>red</Value>
      </Attribute>
      <Attribute Name="MyPointTime" RefAttributeType="DemoAttributeLib/PointInTime">
        <Attribute Name="X">
          <Value>10</Value>
        </Attribute>
      </Attribute>
    </InternalElement>
  </InstanceHierarchy>
  <AttributeTypeLib Name="DemoAttributeLib">
    <AttributeType Name="Colour" AttributeDataType="xs:string"/>
    <AttributeType Name="Point">
      <Attribute Name="X" AttributeDataType="xs:float"/>
      <Attribute Name="Y" AttributeDataType="xs:float"/>
      <Attribute Name="Z" AttributeDataType="xs:float"/>
    </AttributeType>
    <AttributeType Name="PointInTime" RefAttributeType="DemoAttributeLib/Point">
      <Attribute Name="Time" AttributeDataType="xs:time"/>
    </AttributeType>
  </AttributeTypeLib>
```

Figure A.8 – XML code of the AttributeTypeLib example

A.2.6 Data definition of InterfaceClass

A.2.6.1 Architecture of an InterfaceClass

CAEX allows for the definition of interfaces by means of InterfaceClasses. Interfaces can be characterized by the following CAEX elements.

- **Attribute:** Attributes allow the specification of object attributes.
- **ExternalInterface:** Allows the specification of nested interfaces. The concept of nested interfaces supports modeling of complex interfaces.

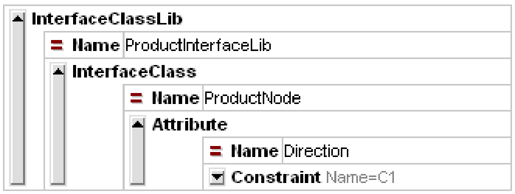
The following CAEX properties related to properties are normative in addition to the CAEX schema.

- Interfaces do not have a direction property. If an interface direction is required, this shall be added as user defined CAEX attribute of the interface.
- The concept of child-interface classes in interface libraries allows for describing a user defined hierarchy of interfaces, the hierarchy itself has no semantics. The hierarchy may be used in order to depict the user's library structure.
- Inheritance relations are defined by means of a reference to the parent interface class. See A.2.8.4 for more information about inheritance.
- Required external interfaces shall be defined by means of the CAEX element "ExternalInterface". Aggregation shall be done either via referencing an existing interface class or by the definition of all required interface property directly. Aggregated interfaces can be extended, additional attributes may be defined, and inherited attributes may be specified and nested interfaces may be added.

For a detailed CAEX data definition, see A.3.8, A.3.9 and A.3.21.

A.2.6.2 Examples

Figure A.9 presents an InterfaceClassLib with the InterfaceClass "ProductNode". Typical further use cases for interface classes are "SignalNode", "DigitalIn", "DigitalOut", etc.



```
InterfaceClassLib
    Name ProductInterfaceLib
    InterfaceClass
        Name ProductNode
        Attribute
            Name Direction
            Constraint Name=C1
```

Figure A.9 – Example of an InterfaceClassLib

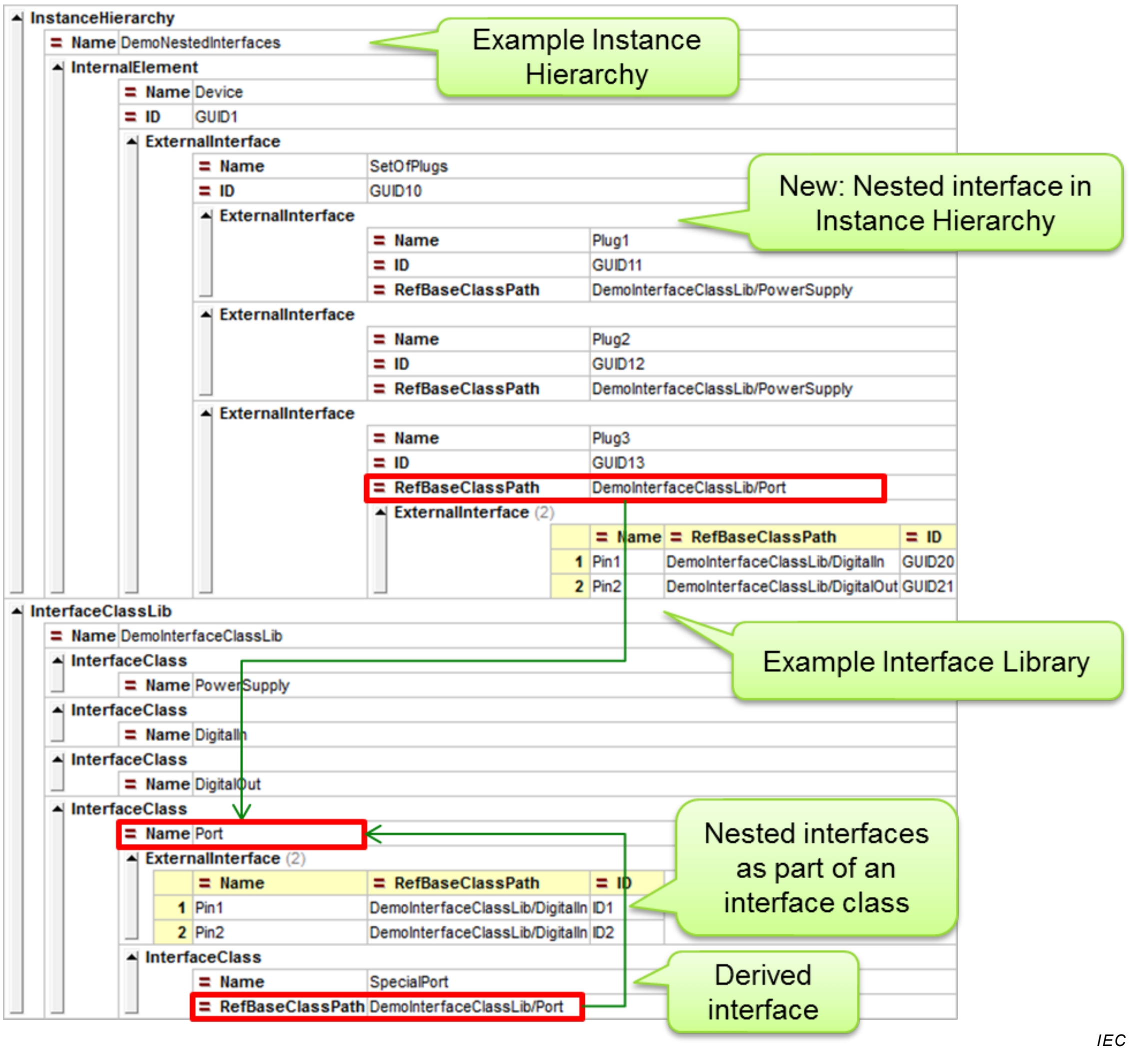
The full XML text for this example is shown in Figure A.10.

```
<InterfaceClassLib Name="ProductInterfaceLib">
    <InterfaceClass Name="ProductNode">
        <Attribute Name="Direction">
            <Constraint Name="C1">
                <NominalScaledType>
                    <RequiredValue>In</RequiredValue>
                    <RequiredValue>Out</RequiredValue>
                    <RequiredValue>Undirected</RequiredValue>
                </NominalScaledType>
            </Constraint>
        </Attribute>
    </InterfaceClass>
</InterfaceClassLib>
```

IEC

Figure A.10 – XML code of the example of an InterfaceClassLib

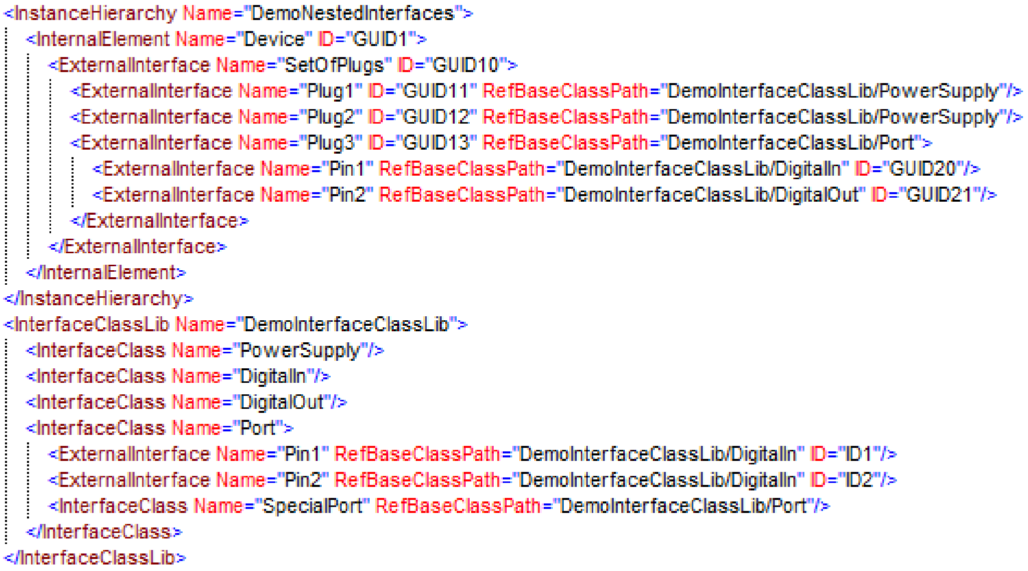
Figure A.11 and Figure A.12 present a second example illustrating the usage of nested interfaces.



```
InstanceHierarchy
  = Name DemoNestedInterfaces
  InternalElement
      = Name Device
      = ID   GUID1
      ExternalInterface
          = Name   SetOfPlugs
          = ID     GUID10
          ExternalInterface
              = Name              Plug1
              = ID                GUID11
              = RefBaseClassPath  DemoInterfaceClassLib/PowerSupply
          ExternalInterface
              = Name              Plug2
              = ID                GUID12
              = RefBaseClassPath  DemoInterfaceClassLib/PowerSupply
          ExternalInterface
              = Name              Plug3
              = ID                GUID13
              = RefBaseClassPath  DemoInterfaceClassLib/Port
              ExternalInterface (2)
                     = Name  = RefBaseClassPath                    = ID
                  1  Pin1    DemoInterfaceClassLib/DigitalIn       GUID20
                  2  Pin2    DemoInterfaceClassLib/DigitalOut      GUID21
InterfaceClassLib
  = Name DemoInterfaceClassLib
  InterfaceClass
      = Name PowerSupply
  InterfaceClass
      = Name DigitalIn
  InterfaceClass
      = Name DigitalOut
  InterfaceClass
      = Name Port
      ExternalInterface (2)
             = Name  = RefBaseClassPath                 = ID
          1  Pin1    DemoInterfaceClassLib/DigitalIn    ID1
          2  Pin2    DemoInterfaceClassLib/DigitalIn    ID2
      InterfaceClass
          = Name              SpecialPort
          = RefBaseClassPath  DemoInterfaceClassLib/Port
```

Example Instance Hierarchy
New: Nested interface in Instance Hierarchy
Example Interface Library
Nested interfaces as part of an interface class
Derived interface
IEC

Figure A.11 – Second example of an InterfaceClassLib and the usage of nested interfaces



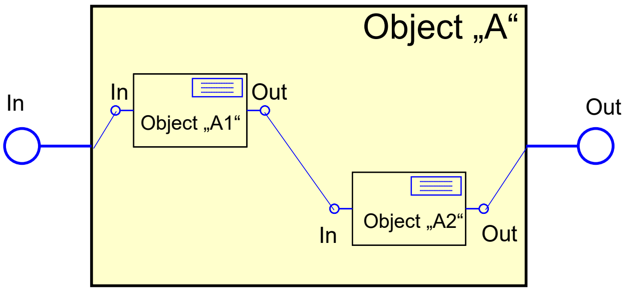
```
<InstanceHierarchy Name="DemoNestedInterfaces">
  <InternalElement Name="Device" ID="GUID1">
    <ExternalInterface Name="SetOfPlugs" ID="GUID10">
      <ExternalInterface Name="Plug1" ID="GUID11" RefBaseClassPath="DemoInterfaceClassLib/PowerSupply"/>
      <ExternalInterface Name="Plug2" ID="GUID12" RefBaseClassPath="DemoInterfaceClassLib/PowerSupply"/>
      <ExternalInterface Name="Plug3" ID="GUID13" RefBaseClassPath="DemoInterfaceClassLib/Port">
        <ExternalInterface Name="Pin1" RefBaseClassPath="DemoInterfaceClassLib/DigitalIn" ID="GUID20"/>
        <ExternalInterface Name="Pin2" RefBaseClassPath="DemoInterfaceClassLib/DigitalOut" ID="GUID21"/>
      </ExternalInterface>
    </ExternalInterface>
  </InternalElement>
</InstanceHierarchy>
<InterfaceClassLib Name="DemoInterfaceClassLib">
  <InterfaceClass Name="PowerSupply"/>
  <InterfaceClass Name="DigitalIn"/>
  <InterfaceClass Name="DigitalOut"/>
  <InterfaceClass Name="Port">
    <ExternalInterface Name="Pin1" RefBaseClassPath="DemoInterfaceClassLib/DigitalIn" ID="ID1"/>
    <ExternalInterface Name="Pin2" RefBaseClassPath="DemoInterfaceClassLib/DigitalIn" ID="ID2"/>
    <InterfaceClass Name="SpecialPort" RefBaseClassPath="DemoInterfaceClassLib/Port"/>
  </InterfaceClass>
</InterfaceClassLib>
```

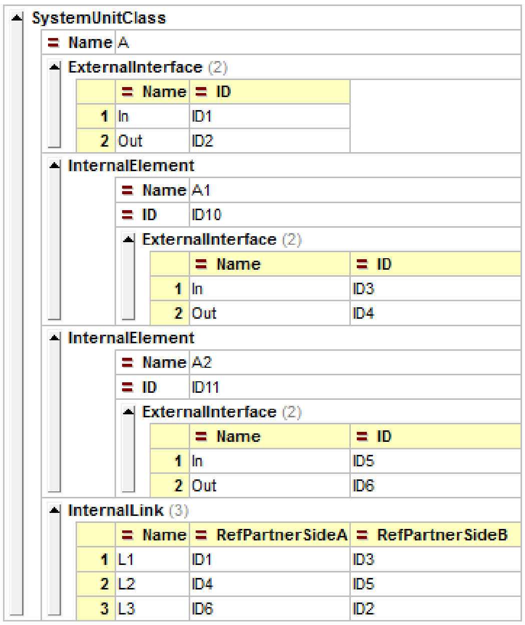
IEC

Figure A.12 – XML code of the second example

A.2.6.3 Usage of interfaces and linking of interfaces

Interfaces describe connection points of objects. Links between object interfaces shall be modeled by means of the CAEX element "InternalLink" which is part of the CAEX SystemUnit definition.





SystemUnitClass

= Name A

ExternalInterface (2)

	= Name	= ID
1	In	ID1
2	Out	ID2

InternalElement

= Name A1

= ID ID10

ExternalInterface (2)

	= Name	= ID
1	In	ID3
2	Out	ID4

InternalElement

= Name A2

= ID ID11

ExternalInterface (2)

	= Name	= ID
1	In	ID5
2	Out	ID6

InternalLink (3)

	= Name	= RefPartnerSideA	= RefPartnerSideB
1	L1	ID1	ID3
2	L2	ID4	ID5
3	L3	ID6	ID2

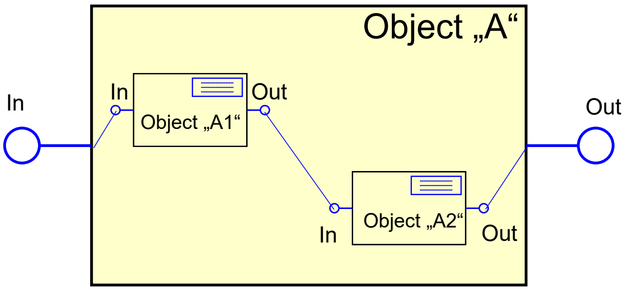
IEC

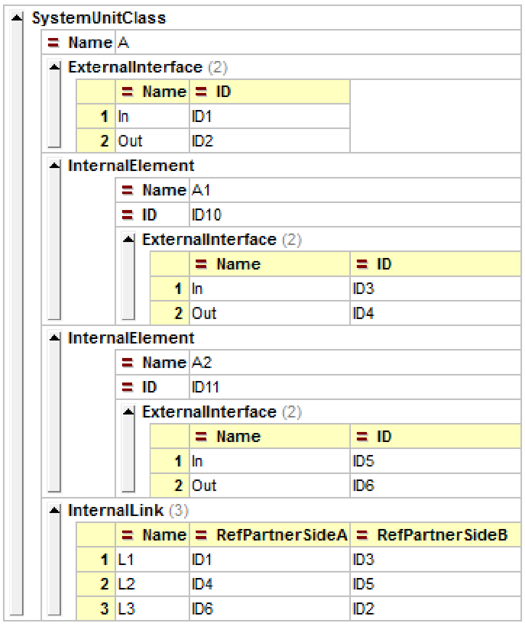
Figure A.13 describes as an example a SystemUnit "A" that provides the interfaces "In" and "Out". Furthermore, it contains two aggregated internal objects "A1" and "A2" with each two interfaces "In" and "Out". The links between the internal objects as well as the inner interfaces and the external interfaces of "A" are described as an example in CAEX as shown below. For a detailed CAEX data definition, see the SystemUnitClass definition in A.3.13.

The following CAEX properties related to links are normative in addition to the CAEX schema.

- A CAEX link shall link a pair of corresponding interfaces by means of their IDs.
- CAEX links do not have a direction.

- Links across different hierarchy levels are allowed.
- CAEX links do not have a data type. If required, data types shall be assigned to the corresponding interfaces individually, but CAEX does not provide this explicitly.
- CAEX does not provide consistency checks for links. Invalid links have to be identified by the source or the target tool.
- If at least one side of a link is unknown or undefined, the corresponding attribute “Ref-PartnerSideA” or “RefPartnerSideB” shall be an empty string.





```
SystemUnitClass
  = Name  A
  ExternalInterface (2)
        = Name  = ID
     1    In      ID1
     2    Out     ID2
  InternalElement
     = Name  A1
     = ID    ID10
     ExternalInterface (2)
           = Name  = ID
        1    In      ID3
        2    Out     ID4
  InternalElement
     = Name  A2
     = ID    ID11
     ExternalInterface (2)
           = Name  = ID
        1    In      ID5
        2    Out     ID6
  InternalLink (3)
        = Name  = RefPartnerSideA  = RefPartnerSideB
     1    L1      ID1                ID3
     2    L2      ID4                ID5
     3    L3      ID6                ID2
```

IEC

Figure A.13 – Usage of Links

The full XML text is shown in Figure A.14.

```
<SystemUnitClass Name="A">
        <ExternalInterface Name="In" ID="ID1"/>
        <ExternalInterface Name="Out" ID="ID2"/>
        <InternalElement Name="A1" ID="ID10">
                <ExternalInterface Name="In" ID="ID3"/>
                <ExternalInterface Name="Out" ID="ID4"/>
        </InternalElement>
        <InternalElement Name="A2" ID="ID11">
                <ExternalInterface Name="In" ID="ID5"/>
                <ExternalInterface Name="Out" ID="ID6"/>
        </InternalElement>
        <InternalLink Name="L1" RefPartnerSideA="ID1" RefPartnerSideB="ID3"/>
        <InternalLink Name="L2" RefPartnerSideA="ID4" RefPartnerSideB="ID5"/>
        <InternalLink Name="L3" RefPartnerSideA="ID6" RefPartnerSideB="ID2"/>
</SystemUnitClass>
```

IEC

Figure A.14 – XML code for the usage of links

A.2.7 Data definition of RoleClass

A.2.7.1 Architecture of a RoleClass

CAEX allows for the definition of roles by means of RoleClasses. Roles are characterized by CAEX attributes and ExternalInterfaces.

- **Attribute:** Attributes allow for the specification of role attributes.
- **ExternalInterface:** Allows for the specification of role interfaces.

Regarding role classes, the following provisions apply.

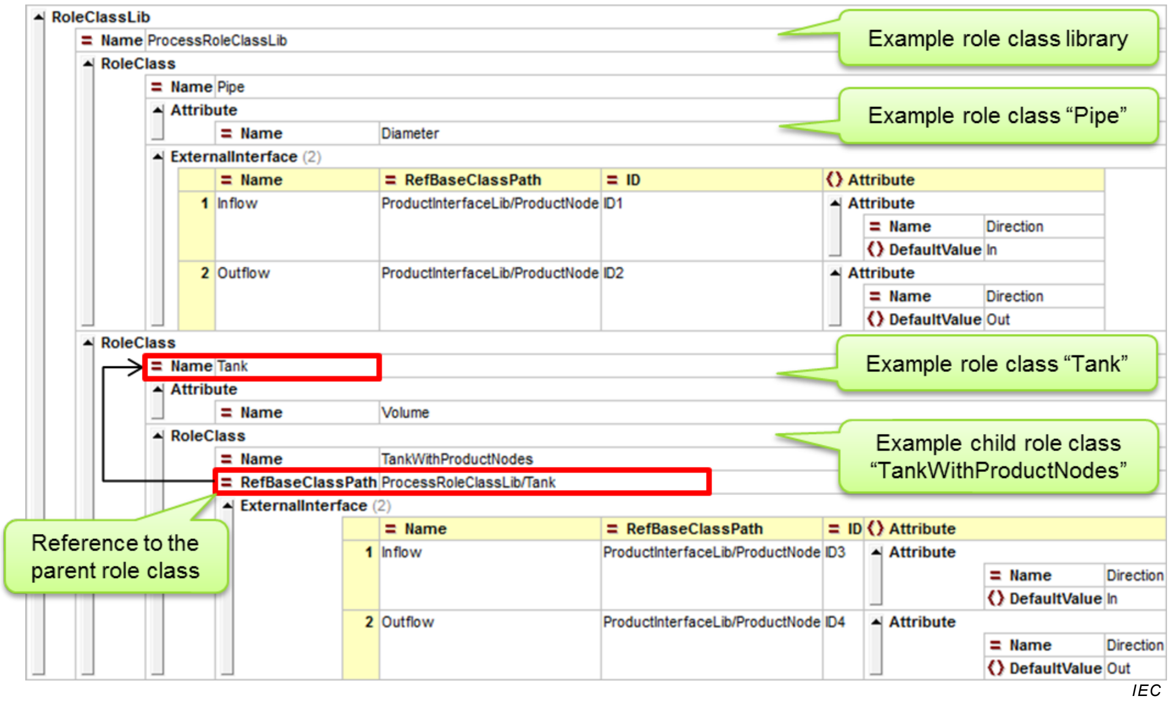
- RoleClasses do not contain nested roles.
- The concept of child-roles allows for describing a hierarchy of roles, the hierarchy itself has no semantics.
- Inheritance relations are defined by means of a reference to a parent role class.

For a detailed CAEX data definition, see A.3.10, A.3.11 and A.3.24.

A.2.7.2 Example

Figure A.15 presents a RoleClassLib “ProcessRoleClassLib” with two role classes: “Pipe” and “Tank”.

- The role “Pipe” comprises one attribute “Diameter” without closer specification of its Unit or DefaultValue. Additionally, it comprises two interfaces of the type “Product Node”. This base class provides the attribute “Direction” – the value is set to “In” or “Out” respectively.
- The role “Tank” additionally demonstrates the concept of creating role hierarchies and role class inheritance. The role “Tank” simply specifies only one attribute. The role “TankWithProductNodes” is placed as a child of the role class “Tank”. This parent-child-relation has no semantics but allows for the definition of arbitrary library hierarchies. Additionally, the child role “TankWithProductNodes” references the role “Tank” as base class. This defines an inheritance relation: this role class inherits all attributes and interfaces from “Tank”.



```
RoleClassLib
  Name ProcessRoleClassLib
  RoleClass
    Name Pipe
    Attribute
      Name       Diameter
    ExternalInterface (2)
         Name      RefBaseClassPath                 ID    Attribute
       1 Inflow    ProductInterfaceLib/ProductNode  ID1   Attribute
                                                            Name          Direction
                                                            DefaultValue  In
       2 Outflow   ProductInterfaceLib/ProductNode  ID2   Attribute
                                                            Name          Direction
                                                            DefaultValue  Out
  RoleClass
    Name Tank
    Attribute
      Name       Volume
    RoleClass
      Name              TankWithProductNodes
      RefBaseClassPath  ProcessRoleClassLib/Tank
      ExternalInterface (2)
           Name      RefBaseClassPath                 ID    Attribute
         1 Inflow    ProductInterfaceLib/ProductNode  ID3   Attribute
                                                              Name          Direction
                                                              DefaultValue  In
         2 Outflow   ProductInterfaceLib/ProductNode  ID4   Attribute
                                                              Name          Direction
                                                              DefaultValue  Out
```

Example role class library

Example role class “Pipe”

Example role class “Tank”

Example child role class “TankWithProductNodes”

Reference to the parent role class

IEC

Figure A.15 – Example of a RoleClassLib

A.2.8 Modelling of relations

A.2.8.1 Overview

Modelling objects makes it necessary to define mechanisms to set these objects in relation to each other. Additional mechanisms are needed to link these objects with external stored data.

A relation expresses an association between two or more objects. This dependency may be of any nature including physical and logical dependencies. CAEX supports the following relations:

- **parent-child relations (see A.2.8.2 and A.2.8.3)**
  – parent-child relations between CAEX InternalElements
  – parent-child relations between CAEX classes
- **inheritance relations (see A.2.8.4)**
  – inheritance relations between SystemUnitClasses
  – inheritance relations between RoleClasses
  – inheritance relations between InterfaceClasses
  – inheritance relations between AttributeTypes
- **class-instance relations (see A.2.8.5)**
  – relations between a SystemUnitClass and an InternalElement
  – relations between a RoleClass and an InternalElement
  – relations between an InterfaceClass and an ExternalInterface
  – relations between an AttributeType and Attribute
- **instance-instance relations (see A.2.8.6 and A.2.8.7)**
  – relations between CAEX ExternalInterface
  – relations between CAEX InternalElements

Figure A.16 presents the mentioned relation types supported by CAEX by means of an example.

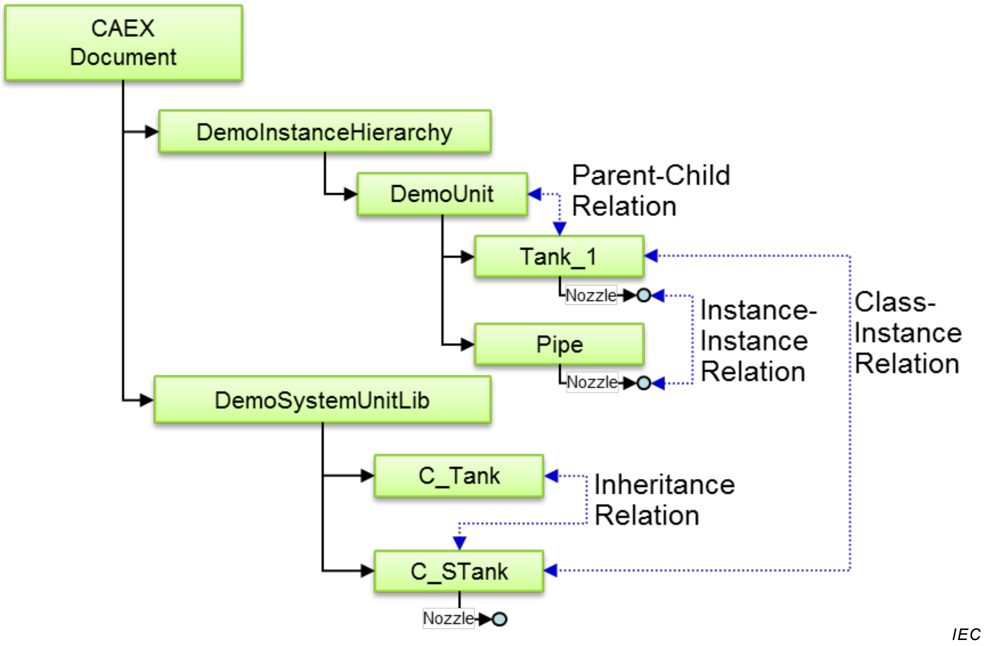
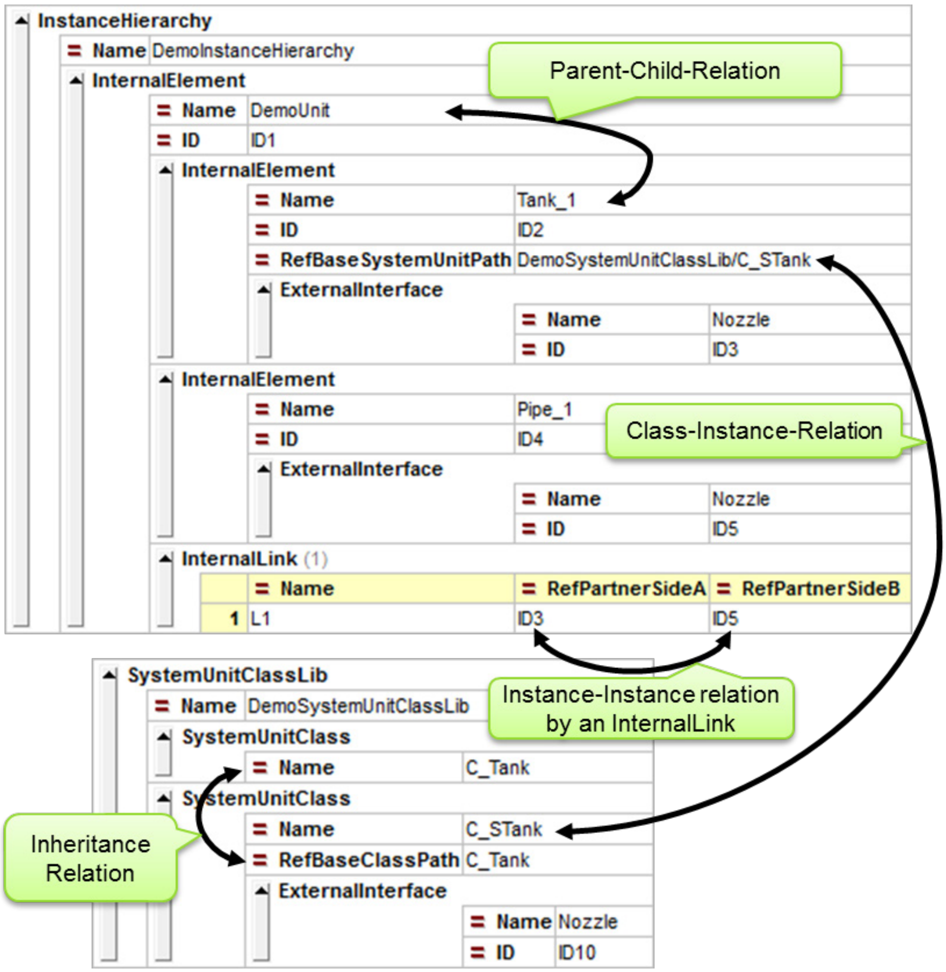


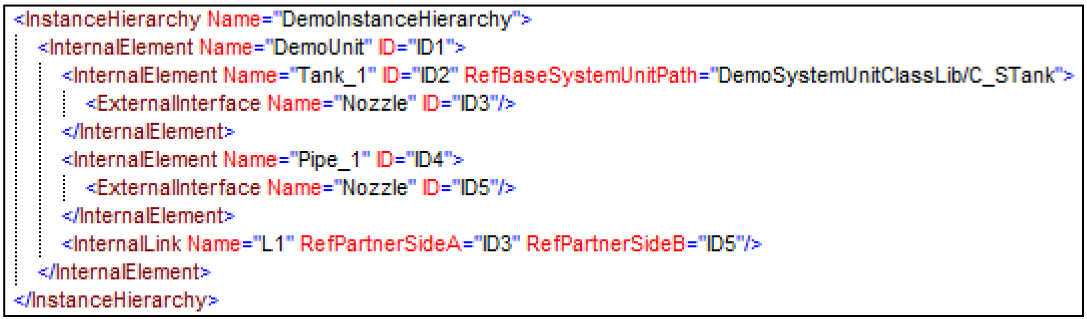
Figure A.16 – Relations in CAEX

Figure A.17 to Figure A.19 illustrate the CAEX model corresponding to the example by means of a table view and the corresponding XML code.



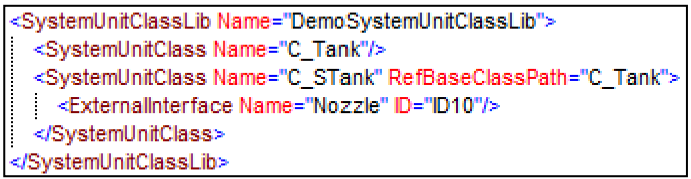
```
InstanceHierarchy
    Name  DemoInstanceHierarchy
    InternalElement
        Name  DemoUnit
        ID  ID1
        InternalElement
            Name  Tank_1
            ID  ID2
            RefBaseSystemUnitPath  DemoSystemUnitClassLib/C_STank
            ExternalInterface
                Name  Nozzle
                ID  ID3
        InternalElement
            Name  Pipe_1
            ID  ID4
            ExternalInterface
                Name  Nozzle
                ID  ID5
        InternalLink (1)
               Name   RefPartnerSideA   RefPartnerSideB
            1  L1     ID3               ID5

SystemUnitClassLib
    Name  DemoSystemUnitClassLib
    SystemUnitClass
        Name  C_Tank
    SystemUnitClass
        Name  C_STank
        RefBaseClassPath  C_Tank
        ExternalInterface
            Name  Nozzle
            ID  ID10
```

Figure A.17 – XML description of the relations example



```
<InstanceHierarchy Name="DemoInstanceHierarchy">
  <InternalElement Name="DemoUnit" ID="ID1">
    <InternalElement Name="Tank_1" ID="ID2" RefBaseSystemUnitPath="DemoSystemUnitClassLib/C_STank">
      <ExternalInterface Name="Nozzle" ID="ID3"/>
    </InternalElement>
    <InternalElement Name="Pipe_1" ID="ID4">
      <ExternalInterface Name="Nozzle" ID="ID5"/>
    </InternalElement>
    <InternalLink Name="L1" RefPartnerSideA="ID3" RefPartnerSideB="ID5"/>
  </InternalElement>
</InstanceHierarchy>
```

Figure A.18 – XML text of the InstanceHierarchy of the relations example



```
<SystemUnitClassLib Name="DemoSystemUnitClassLib">
  <SystemUnitClass Name="C_Tank"/>
  <SystemUnitClass Name="C_STank" RefBaseClassPath="C_Tank">
    <ExternalInterface Name="Nozzle" ID="ID10"/>
  </SystemUnitClass>
</SystemUnitClassLib>
```

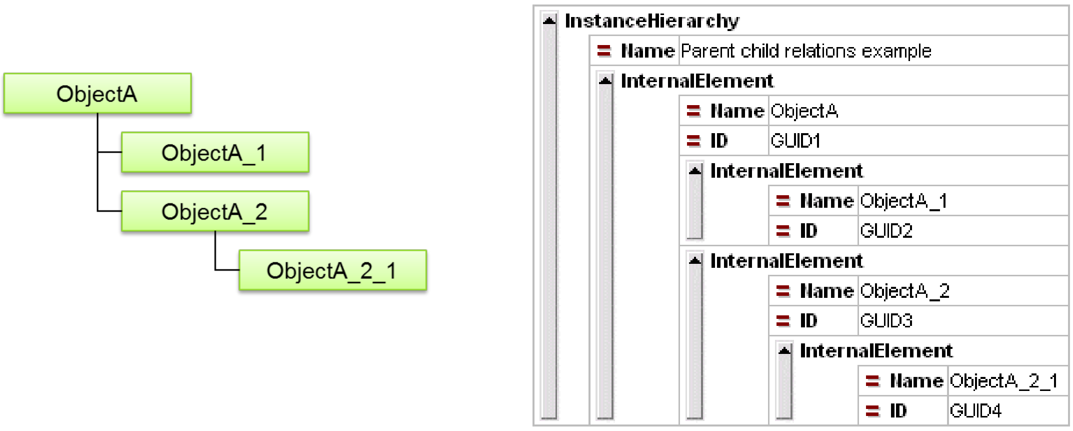
Figure A.19 – XML text of the SystemUnitClassLib of the relations example

A.2.8.2 Parent-child-relations between CAEX object instances

Parent-child relations between object instances are used to represent hierarchical object structures. Regarding parent-child relations between CAEX objects, the following provisions apply.

- A plant hierarchy is stored as a tree of CAEX object instances within a CAEX InstanceHierarchy element. The CAEX element InstanceHierarchy consists of an arbitrary number of internal elements which are nested.
- Crossed hierarchies (object networks) are explicitely supported and modelled according to A.2.8.7.

Figure A.20 gives an example of a simple object hierarchy and its CAEX data model.



InstanceHierarchy
Name: Parent child relations example
InternalElement
Name: ObjectA
ID: GUID1
InternalElement
Name: ObjectA_1
ID: GUID2
InternalElement
Name: ObjectA_2
ID: GUID3
InternalElement
Name: ObjectA_2_1
ID: GUID4

```
<InstanceHierarchy Name="Parent child relations example">
  <InternalElement Name="ObjectA" ID="GUID1">
      <InternalElement Name="ObjectA_1" ID="GUID2"/>
      <InternalElement Name="ObjectA_2" ID="GUID3">
          <InternalElement Name="ObjectA_2_1" ID="GUID4"/>
      </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

IEC

Figure A.20 – Example of a parent-child-relation between CAEX InternalElements

Based on this technique, industrial plant hierarchies can be modeled as shown in Figure A.21.

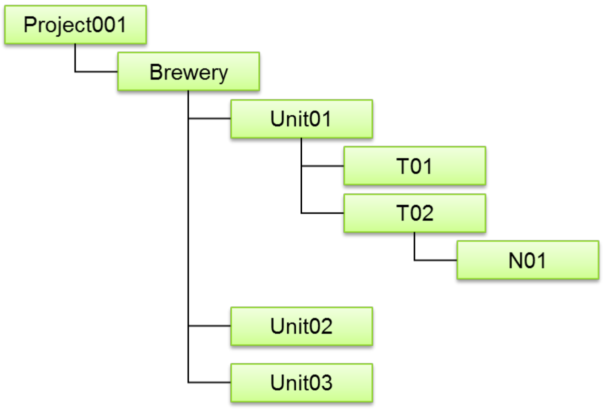


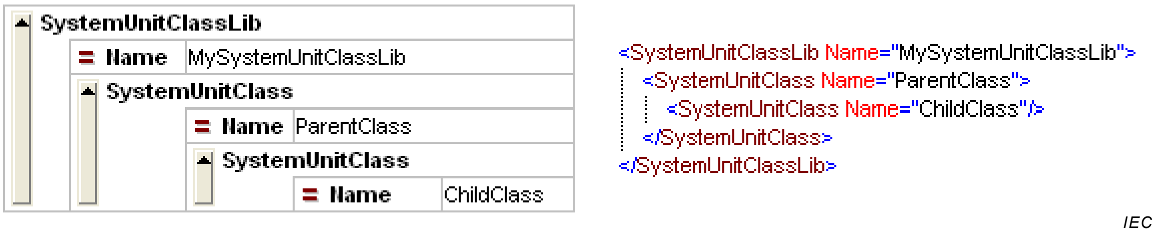
Figure A.21 – Example for a hierarchical plant structure

A.2.8.3 Parent-child relations between CAEX classes

Regarding parent-child relations between CAEX classes/types, the following provisions apply.

- A parent-child-relation between CAEX classes/types shall describe their hierarchical neighbourhood only. This allows definition of any user-defined hierarchical structure.
- This relation has not further semantics.

NOTE A parent child relation does not imply an inheritance relation.

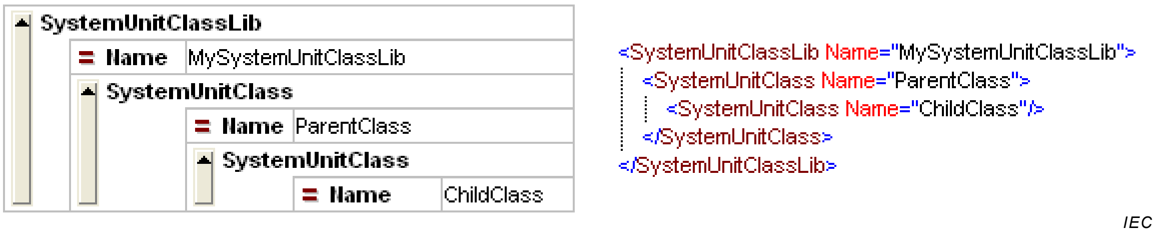


SystemUnitClassLib
Name MySystemUnitClassLib
SystemUnitClass
Name ParentClass
SystemUnitClass
Name ChildClass

```
<SystemUnitClassLib Name="MySystemUnitClassLib">
  <SystemUnitClass Name="ParentClass">
    <SystemUnitClass Name="ChildClass"/>
  </SystemUnitClass>
</SystemUnitClassLib>
```

IEC

Figure A.22 gives an example of a parent-child relation between the classes “ParentClass” and “ChildClass”. The “ChildClass” has no inheritance relation to its parent.



SystemUnitClassLib
Name MySystemUnitClassLib
SystemUnitClass
Name ParentClass
SystemUnitClass
Name ChildClass

<SystemUnitClassLib Name="MySystemUnitClassLib">
  <SystemUnitClass Name="ParentClass">
    <SystemUnitClass Name="ChildClass"/>
  </SystemUnitClass>
</SystemUnitClassLib>

IEC

Figure A.22 – Example of a parent-child relation between classes

A.2.8.4 Inheritance relations between CAEX classes

CAEX supports inheritance between two classes. The inheritance relation is defined in CAEX by means of a reference concept. Each CAEX class owns an attribute “RefBaseClassPath” which allows for the specification of the path of the corresponding parent class. The inheritance concept is identical for InterfaceClasses, RoleClasses, SystemUnitClasses and AttributeTypes. Regarding inheritance relations, the following provisions apply.

- **Inheritance** is allowed among classes. A class can have an arbitrary number of child classes, but only one parent class. All changes in the class shall be automatically reflected by all child classes.
- **Inheritance** means that all available parent and grand parent attributes, interfaces, internal elements, mapping objects or further content shall be automatically present in the child objects.
- **Inherited classes** can be extended on class level with new attributes, interfaces, etc.
- **Storage of inherited data:** Inherited data is valid for the child data and may, but not shall, be copied to the child physically in the XML document. Redefinition and storage of already inherited data is possible and useful in order to overwrite or extend inherited information. If data is copied physically from a parent class to a child and changed on the parent class later on, the copied child data shall be updated.
- **Overwriting of inherited data:** Overwriting of inherited properties is possible by the redefinition of the corresponding data again in the child object with updated values. As long as given attribute constraints are defined in the parent class, the overwritten data shall fulfill these requirements.

- **Deleting inherited data:** Deleting of inherited properties is possible by the redefinition of the corresponding data again in the child object with the ChangeMode attribute set to "deleted".
- **Inheritance** is supported in a linear way. A child class may inherit from one parent class and may be a parent class of other classes at the same time. CAEX allows for the definition of parents, children and grandchildren in this way with arbitrary deepness. The grandchild thus inherits from parents and grandparents etc. CAEX only supports inheritance from one parent.
- If inheritance is required, the parent class shall be specified using the CAEX tag "RefBaseClassPath" comprising the full path of the class. The referenced class shall be valid and present.
- If the desired parent class is placed one hierarchy level above the child class, the parent class can be specified by storing the name of the parent class in the CAEX tag "RefBaseClassPath" without providing the full path.

  NOTE Figure A.16 and Figure A.17 provide an example of the parent class "C_Tank" and the derived class "C_STank". In addition to this example, the CAEX tag "RefBaseClassPath" can either be "DemoSystem-UnitClassLib/C_Tank" as well as "C_Tank" since the parent class is one hierarchy level above the class "C_STank".
- A SystemUnitClass shall only inherit from a SystemUnitClass; an InterfaceClass shall only inherit from an InterfaceClass; a RoleClass shall only inherit from a RoleClass and an AttributeType shall only inherit from an AttributeType. Cross inheritance shall not be allowed.
- Inheritance is optional. If inheritance is not required, the reference attribute "RefBaseClassPath" shall be empty or shall not be present at all.
- A class shall not inherit from itself or from a derivative of itself.
- CAEX does not provide consistency checks of valid inheritance relations or of the valid existence of the reference item.

A.2.8.5 Class-instance relations

Instances are characterized by a unique identifier and a parameter set. Regarding class-instance relations the following provisions apply.

- A CAEX InternalElement or a CAEX ExternalInterface may be a singleton without a relation to any class.
- If a CAEX **InternalElement** has a class-instance-relation to a SystemUnitClass, it shall be created as a copy of this SystemUnitClass including the internal architecture of the class and all inherited information at the present time. The copied source class shall be indicated in the CAEX tag "RefBaseSystemUnitPath" of the instance for further usage. This tag shall comprise the full path and name of the source class. Only one SystemUnitClass can be referenced. Changes in the class are not automatically reflected in the corresponding object instance. Furthermore, the object instance can be transported without the class information, it self-contains whole belonging information.

  NOTE 1 A class serves as a template in this way.

  NOTE 2 If the source-class of an instance changes, this does not entail a change of the instance. An automatic reflection or update of instance data according to a changed source class is a tool functionality and out of the scope of IEC 62424. The present path to the source class supports this functionality.
- If a CAEX **ExternalInterface** has a class-instance-relation to an InterfaceClass, it shall be created as a copy of this InterfaceClass including the internal architecture of the class and all inherited information at the present time. The copied source class shall be indicated in the CAEX tag "RefBaseClassPath" of the ExternalInterface for further usage. This tag shall comprise the full path and name of the source class. Only one InterfaceClass can be referenced.
- The relation between a CAEX InternalElement and a **RoleClass** shall be indicated by the attribute "RefBaseRoleClassPath" of the belonging CAEX element RoleRequirement. All RoleClass specifications shall be copied to the corresponding CAEX object. If an attribute of a role class has no value, it may be removed from the instance data if not required.

- The relation between a CAEX Attribute and a CAEX **AttributeType** shall be indicated by the CAEX tag “RefAttributeType”. All type specifications shall be copied to the corresponding CAEX Attribute. If an attribute of the attribute type specification has no value, it may be removed from the CAEX attribute if required.
- During the process of copying class data into an instance, all objects in the instance having an ID shall receive a new unique ID. The class is not changed. All references using the old IDs shall be updated accordingly across the whole CAEX document.
- The extension or reduction of instance data compared to the source class is allowed.

  NOTE 3 The source class is intended to be a suitable starting point for the instance model.

Figure A.16 and Figure A.17 give an example of a class-instance relation between the object “Tank_1” and a user-defined SystemUnitClass “C_STank”.

A.2.8.6 Instance-instance relations between two CAEX ExternalInterfaces

Instance-instance relations are relations between two interfaces of arbitrary CAEX InternalElements. Regarding Instance-Instance relations, the following provisions apply.

- Instance-instance relations shall be stored according A.2.6.3 by means of the CAEX InternalLink functionality.
- CAEX InternalLinks should be stored at the CAEX InternalElement or SystemUnitClass which is the lowest common parent of the corresponding connected CAEX objects.

Figure A.16 and Figure A.17 illustrate this by means of the InternalLink “L1”.

A.2.8.7 Instance-instance relations between CAEX InternalElements

CAEX supports modelling of multiple hierarchies at the same time. Since different hierarchical structures may depict the same data in different ways, it may be the case that a single instance needs to be part of multiple hierarchies. CAEX supports this by means of a “mirror concept”.

Figure A.23 explains this concept by means of two example structures “Location Hierarchy” and “Resource Hierarchy” and a corresponding library SystemUnitClassLib “DemoLib”. The InternalElement “Tank1” is an instance of Class “C_Tank”. This object has a second representation “Tank1*” which is positioned in a second hierarchy. Whereas master object “Tank1” references its class “C_Tank”, “Tank1*” references the CAEX InternalElement “Tank1”. Hence, the object “Tank1” acts as “master object” whereas “Tank1*” acts as “mirror object”. In the result, a single CAEX object is present at two positions.

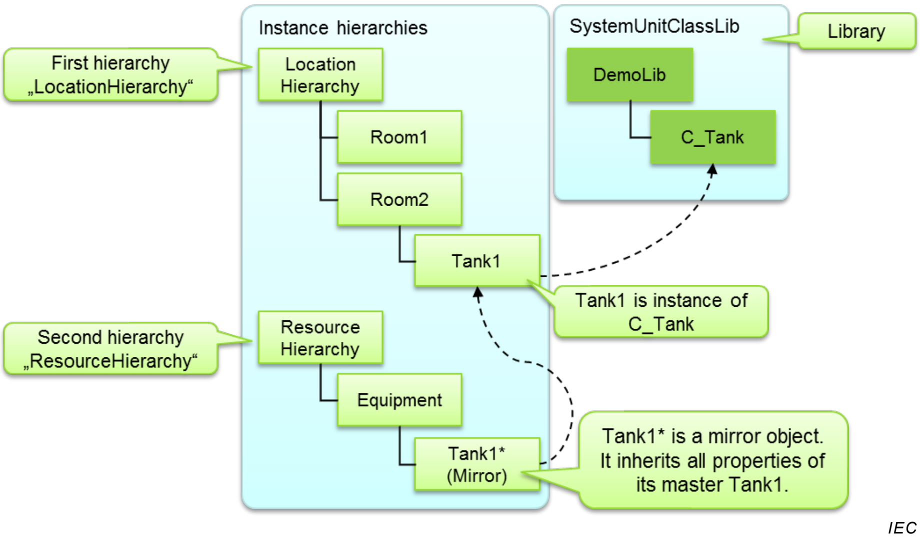

Instance hierarchies
First hierarchy „LocationHierarchy“
Location Hierarchy
Room1
Room2
Tank1
Second hierarchy „ResourceHierarchy“
Resource Hierarchy
Equipment
Tank1* (Mirror)
SystemUnitClassLib
Library
DemoLib
C_Tank
Tank1 is instance of C_Tank
Tank1* is a mirror object. It inherits all properties of its master Tank1.
IEC

Figure A.23 – Multiple crossed structures

The mirror concept can be applied in the same way to CAEX ExternalInterfaces and CAEX Attributes. Figure A.24 illustrates this by means of the attributes “Price” of the objects “Tank” or “Pump”. The same prices are modelled in another hierarchy at the object “Prices” which references the corresponding master attributes. Furthermore, this figure illustrates how mirror objects can be restructured in an alternative structure. But, by definition, all mirror objects shall always form leaves in an object tree.

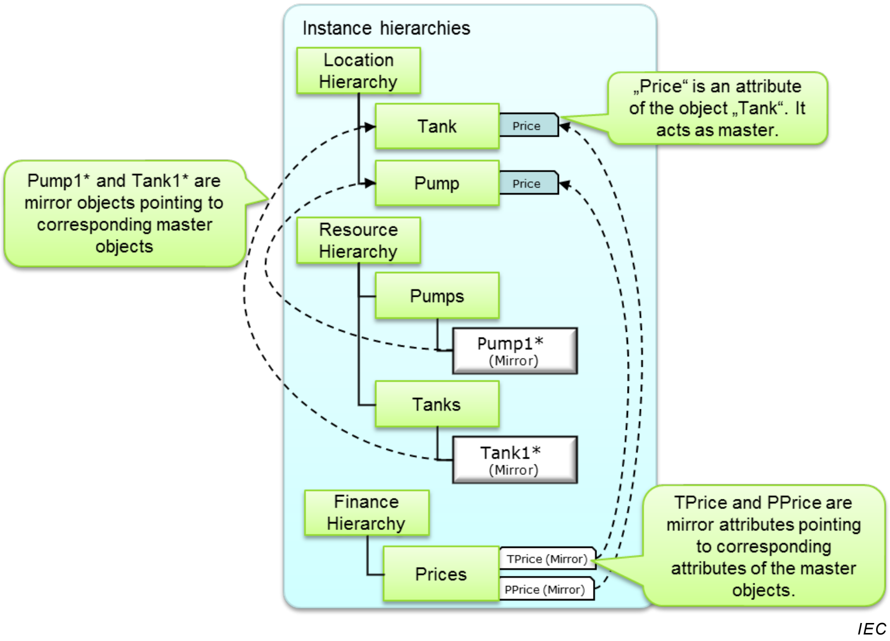

Instance hierarchies
Location Hierarchy
Tank
Price
Pump
Price
„Price“ is an attribute of the object „Tank“. It acts as master.
Pump1* and Tank1* are mirror objects pointing to corresponding master objects
Resource Hierarchy
Pumps
Pump1* (Mirror)
Tanks
Tank1* (Mirror)
Finance Hierarchy
Prices
TPrice (Mirror)
PPrice (Mirror)
TPrice and PPrice are mirror attributes pointing to corresponding attributes of the master objects.
IEC

Figure A.24 – Example for mirror attributes and restructured mirror objects

Regarding the mirror concept, the following normative provisions apply.

- If more than one representation of an CAEX InternalElement, CAEX ExternalInterface or CAEX Attribute is required, each of them shall be modelled as corresponding CAEX InternalElement, CAEX ExternalInterface or CAEX Attribute at the required position.
- One of them shall act as the “master object”. This master object holds all required information as header information, attributes, interfaces and internal elements and may have an instance-class relation to a CAEX class or type as defined in A.2.8.5.

- The other objects act as “mirror objects” and shall reference the “master object”. A “mirror object” acts as a pointer to the “master object”. For this, CAEX InternalElements shall store the ID of the master object in the CAEX tag “RefBaseSystemUnitPath”, CAEX ExternalInterfaces shall store the ID of the master object in the CAEX tag “RefBaseClassPath”, CAEX Attributes shall store the ID of the master attributes parent instance followed by the separator string “/” and the path of the attribute in the CAEX tag “RefAttributeType”.
- A mirror object shall not reference any class or type.
- A master object shall not have a back reference to one of its mirror objects.
- If required, back references have to be handled by a software tool.
- Mirror objects shall not have children and shall not store object related information except of the reference to the master object or the ChangeMode. Changes and modification at a mirror object shall be exlusively modeled at the master object.
- The mirror object may have another name than the master object and may have its own header information.
- A mirror CAEX InternalElement or ExternalInterface shall have an own unique ID.

  NOTE 1 A mirror object is considered to be identical to the master object. The individual ID supports distinguishing the mirror representations from the master.
- If a master object is deleted, all corresponding mirror objects shall be also deleted in order to avoid inconsistencies.

  NOTE 2 This is a tool functionality out of the scope of this part of the standard.

  NOTE 3 It is possible to replace one of the mirror objects by the master object and delete the old master object.
- If a mirror object is deleted, the master object shall not be affected.
- CAEX InternalLinks shall interconnect master objects only.
- Master objects and belonging mirror objects shall be positioned within one or several CAEX InstanceHierarchies, within one SystemUnitClass, or within one InterfaceClass. Master objects and belonging mirror objects shall not be positioned across class borders.

  NOTE 4 Consequently, Role Classes do not contain mirror objects.

A.2.9 Usage of paths

A.2.9.1 Separator definitions

Paths are the basis for referencing classes or attribute types. Paths require the definition of separators between different path elements. CAEX distinguishes between 2 separator types: Alias separator and object separator.

- Alias separator (used after alias): “@”
- Object separator (used between object hierarchies): “/”

The following CAEX properties related to paths are normative in addition to the CAEX schema.

- A full path to a class or attribute type shall comprise
  - if available, the alias name followed by the alias separator “@”,
  - the name of the library followed by the object separator “/”,
  - the names of all parent elements separated by the object separator “/”,
  - the name of the class or type separated by the objet separator “/”.
- A full path to an attribute of a class or type shall comprise
  - the full path to the belonging object,
  - in case of a nested attribute, the names of all parent attributes separated by the object separator “/”,

  - the name of the attribute, separated by the objet separator “/”.
- A full path to an object instance shall comprise the ID of the object instance.
- A full path to an attribute of an object instance shall comprise
  - the ID of the object instance followed by the objet separator “/”,
  - in case of a nested attribute, the names of all parent attributes separated by the object separator “/” ,
  - the name of the attribute, separated by the object separator “/”.
- A short path is allowed if the referenced class or attribute is positioned in the next upper hierarchy level of the referencing item. It shall comprise the name of the class or attribute type or the attribute only.
- If defined separators are potentially a valid part of object names, the following syntax shall be used: all path elements shall be separated by square brackets “[“ <name> “]”. This allows for using the original names and the defined separators at the same time.
- If the conflict case arises that the described brackets are part of object names, the brackets in the object names shall be escaped by means of common XML escape-sequences.
- It is allowed to use brackets also without any occurrence of conflicts.
- CAEX does not check the validity of a path, neither the use of the normative separators nor the existence of the referenced item. The conformity with this standard requires the correct use of paths and the defined separators.

A.2.9.2 Examples

Examples of paths.

- Full path to a class in a library: “ProcessEngineeringClassLib/Tank”
- Full path to a class in a library using brackets: [DemoLib]/[Tank/@01]
- Full path to a class using alias definitions: “ExternalLibAlias@ClassLib/PipeClass”
- Full path to a class attribute: “ProcessEngineeringClassLib/Tank/height”
- Full path to an attribute type: “MyAliasTypeLib/BaseAttributes/SpecialStringAttribute”
- Short path to a class in the next upper level of the library hierarchy: “C_Tank”
- Short path to a role attribute in the Mapping object: “Speed”
- Full path to an interface attribute in the MappingObject: ID1/Speed

A.2.10 CAEX role concept

A.2.10.1 Usage of the role concept

The main goal of the CAEX role concept is the separation of abstract role information and the definition of concrete implementation information. Figure A.25 explains the role concept by means of an InternalElement “B1” which is stored in an arbitrary position of the plant structure. For a detailed CAEX data definition, see A.3.11 and A.3.23.

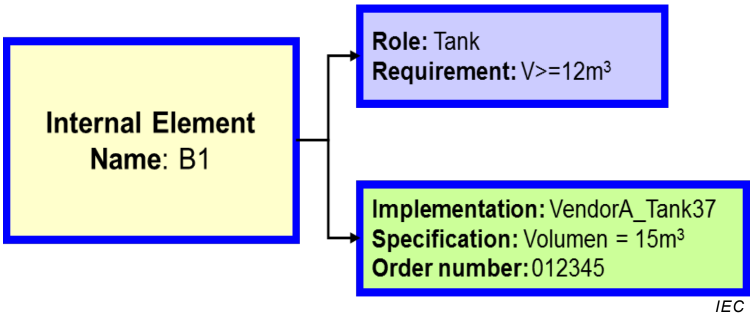


Figure A.25 – CAEX role concept

Use case 1: B1 is described by its name only. B1 has no further meaning or semantics, and it is just a placeholder for future use.

Figure A.26 depicts the corresponding CAEX data model.

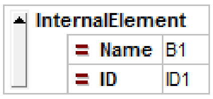
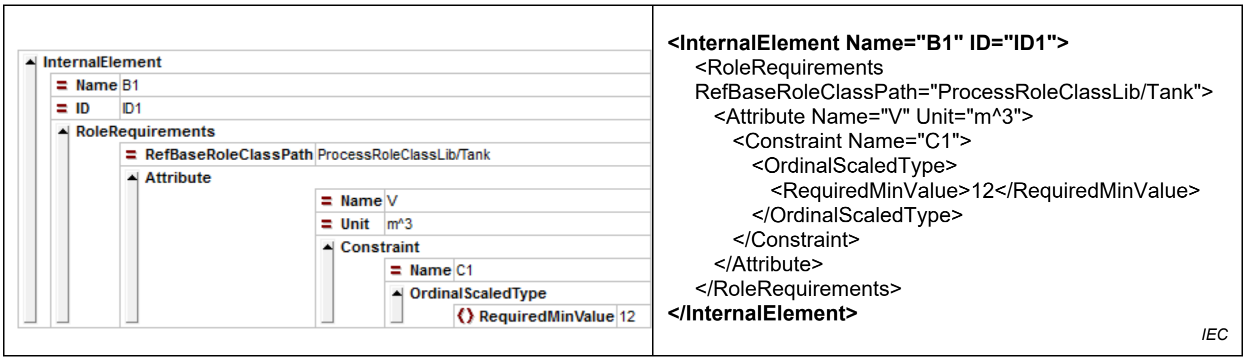
 InternalElement Name B1 ID ID1	<InternalElement Name="B1" ID="ID1"/> IEC

Figure A.26 – CAEX data definition for use case 1

Use case 2: During the iterative engineering, a suited role class is selected which describes the role that “B1” has to play. This gives B1 a meaning/semantics. The role class provides predefined attributes and interfaces that are required. If no suited role class is defined, all role requirements can be defined here. In the given example, B1 is assigned to a role “Tank”, and the required attribute “V” is set to “>= 12 m^3”. Working with roles allows for abstracting from technical implementations. Figure A.27 depicts the corresponding CAEX data model.



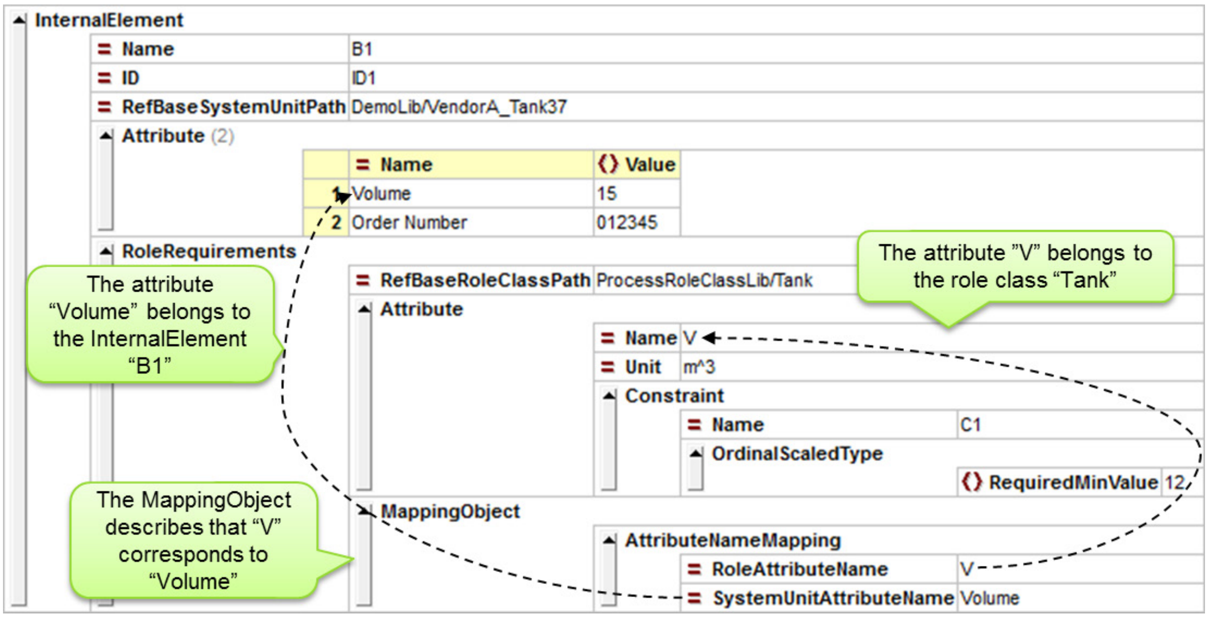
```
InternalElement
  Name  B1
  ID    ID1
  RoleRequirements
    RefBaseRoleClassPath  ProcessRoleClassLib/Tank
    Attribute
      Name  V
      Unit  m^3
      Constraint
        Name  C1
        OrdinalScaledType
          RequiredMinValue  12
```

```
<InternalElement Name="B1" ID="ID1">
  <RoleRequirements
  RefBaseRoleClassPath="ProcessRoleClassLib/Tank">
    <Attribute Name="V" Unit="m^3">
      <Constraint Name="C1">
        <OrdinalScaledType>
          <RequiredMinValue>12</RequiredMinValue>
        </OrdinalScaledType>
      </Constraint>
    </Attribute>
  </RoleRequirements>
</InternalElement>
```

IEC

Figure A.27 – CAEX data definition for use case 2

Use case 3: In a later engineering phase, the concrete technical implementation is of interest. Based on the requirement definitions, a suited technical realization has to be selected in the form of a SystemUnitClass. In the given example, a reference to "VendorA_Tank37" is set. This class fulfills the requirements. Figure A.28 depicts the corresponding CAEX data structure. It becomes visible that the attributes which are defined in the role requirements do not need to match with the attribute names that come from the corresponding SystemUnitClass. For this purpose, CAEX supports a MappingObject which allows for mapping the corresponding attribute names of the role and the SystemUnitClass. The same is valid for interfaces. For more information about mappings, see A.2.11.



```
InternalElement
  Name                     B1
  ID                       ID1
  RefBaseSystemUnitPath    DemoLib/VendorA_Tank37
  Attribute (2)
        Name           Value
      1 Volume         15
      2 Order Number   012345
  RoleRequirements
    RefBaseRoleClassPath  ProcessRoleClassLib/Tank
    Attribute
      Name  V
      Unit  m^3
      Constraint
        Name  C1
        OrdinalScaledType
          RequiredMinValue  12
    MappingObject
      AttributeNameMapping
        RoleAttributeName        V
        SystemUnitAttributeName  Volume
```

The attribute "Volume" belongs to the InternalElement "B1"

The attribute "V" belongs to the role class "Tank"

The MappingObject describes that "V" corresponds to "Volume"

IEC

Figure A.28 – CAEX data definition for use case 3

The full XML text for this example is shown in Figure A.29.

```
<InternalElement Name="B1" ID="ID1" RefBaseSystemUnitPath="DemoLib/VendorA_Tank37">
  <Attribute Name="Volume">
    <Value>15</Value>
  </Attribute>
  <Attribute Name="Order Number">
    <Value>012345</Value>
  </Attribute>
  <RoleRequirements RefBaseRoleClassPath="ProcessRoleClassLib/Tank">
    <Attribute Name="V" Unit="m^3">
      <Constraint Name="C1">
        <OrdinalScaledType>
          <RequiredMinValue>12</RequiredMinValue>
        </OrdinalScaledType>
      </Constraint>
    </Attribute>
    <MappingObject>
      <AttributeNameMapping RoleAttributeName="V" SystemUnitAttributeName="Volume"/>
    </MappingObject>
  </RoleRequirements>
</InternalElement>
```

IEC

Figure A.29 – XML code for use case 3

Regarding the role concept, the following provisions apply.

- An InternalElement shall reference maximum one SystemUnitClass at the same time but may reference multiple RoleClasses at the same time (see A.2.10.2).
- The use of RoleClasses or RoleRequirements is not required. All project data may be stored without using the role concept. This concept supports the iterative engineering process in a flexible way but is not mandatory.
- The RoleRequirements definition at an InternalElement is valid for the individual InternalElement. It may be extended or reduced by further CAEX attributes or CAEX ExternalInterfaces, even when they are not defined in the referenced RoleClass. This supports extending the requirements for the related InternalElement.
- Above the specifications in the RoleRequirements, the related InternalElement may have additional specifications (Attributes, Interfaces), which are not defined at the related RoleClass. This allows defining implementation specific details of the individual Internal-Element.
- The specification of the InternalElement may violate the specification of the RoleRequirements or SupportedRole Class. AML explicitly supports the storage of inconsistent or incomplete engineering data. The validity of the data is a matter of the tools, CAEX only depicts their current data.
- CAEX does not provide consistency checks regarding the role concept, the valid mapping of attributes or interfaces, or the fulfillment of the requirements.

A.2.10.2 Multiple role support

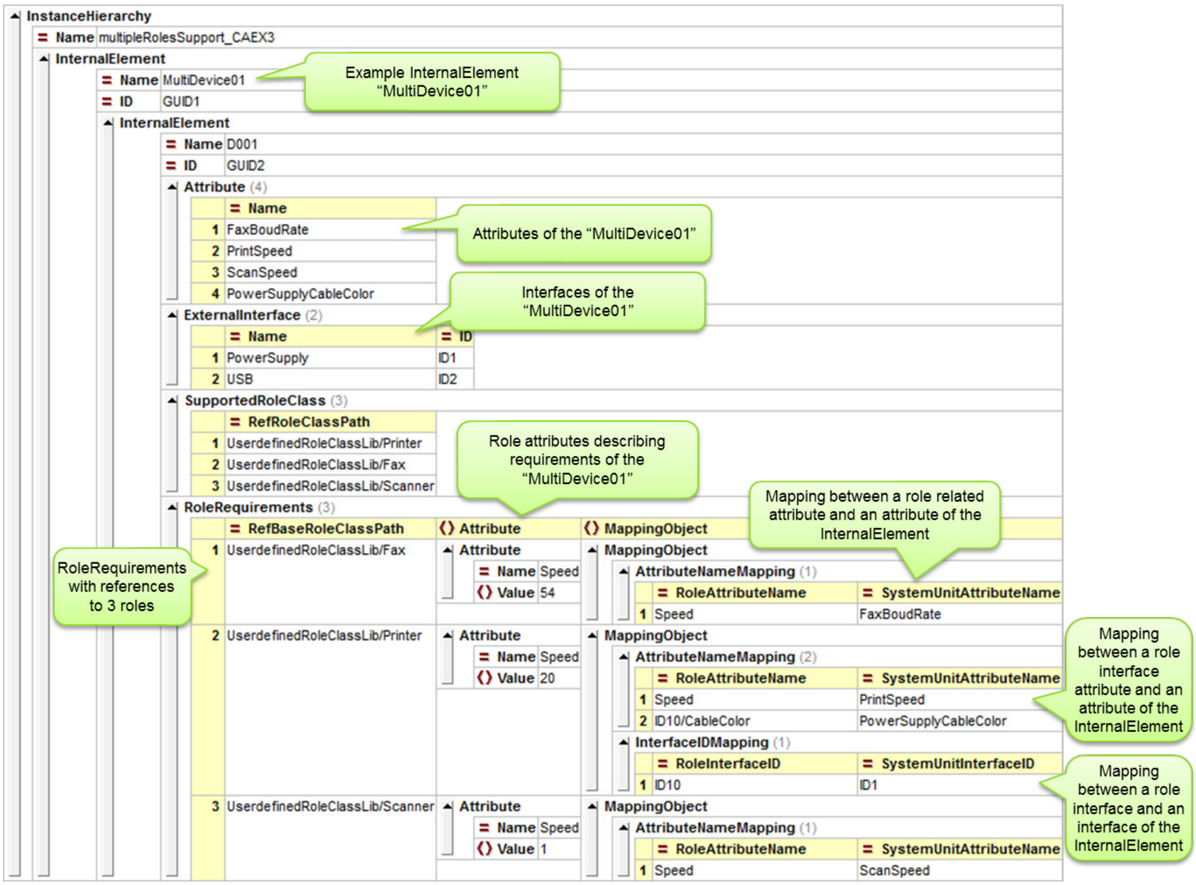
Industrial devices may fulfill more than one role at the same time. For this, CAEX provides support for referencing multiple roles. Multiple roles are of interest, if an object can have multiple functionalities.

An example is a multi functional device that is a scanner, a printer or a fax device at the same time. Figure A.30 models the object “MultiDevice01” with three attributes “FaxBoudRate”, “PrintSpeed” and “FaxSpeed”, and two interfaces “PowerSupply” and “USB”.

Since this object can play three roles at the same time, the InternalElement “MultiDevice01” models three separate CAEX RoleRequirements referencing the roles “Printer”, “Fax” and “Scanner” individually. The requirements of the three different roles are modelled separately and are illustrated by the individual role attributes speed of the printer, fax and scanner and the present role interfaces.

For each CAEX RoleRequirement, an optional CAEX MappingObject allows to define which attribute or interface of the corresponding role is associated to which attribute or interface or the related InternalElement. Given role attribute names in the MappingObject are relative to the referenced role class, hence each RoleRequirement forms its own context.

Figure A.31 presents the corresponding XML code of the example.



```
InstanceHierarchy
  = Name  multipleRolesSupport_CAEX3
  InternalElement
      = Name  MultiDevice01
      = ID    GUID1
      InternalElement
          = Name  D001
          = ID    GUID2
          Attribute (4)
                = Name
              1 FaxBoudRate
              2 PrintSpeed
              3 ScanSpeed
              4 PowerSupplyCableColor
          ExternalInterface (2)
                = Name            = ID
              1 PowerSupply       ID1
              2 USB               ID2
          SupportedRoleClass (3)
                = RefRoleClassPath
              1 UserdefinedRoleClassLib/Printer
              2 UserdefinedRoleClassLib/Fax
              3 UserdefinedRoleClassLib/Scanner
          RoleRequirements (3)
                = RefBaseRoleClassPath              () Attribute          () MappingObject
              1 UserdefinedRoleClassLib/Fax         Attribute             MappingObject
                                                      = Name  Speed         AttributeNameMapping (1)
                                                      () Value 54              = RoleAttributeName   = SystemUnitAttributeName
                                                                             1 Speed                 FaxBoudRate
              2 UserdefinedRoleClassLib/Printer     Attribute             MappingObject
                                                      = Name  Speed         AttributeNameMapping (2)
                                                      () Value 20              = RoleAttributeName   = SystemUnitAttributeName
                                                                             1 Speed                 PrintSpeed
                                                                             2 ID10/CableColor       PowerSupplyCableColor
                                                                            InterfaceIDMapping (1)
                                                                               = RoleInterfaceID     = SystemUnitInterfaceID
                                                                             1 ID10                  ID1
              3 UserdefinedRoleClassLib/Scanner     Attribute             MappingObject
                                                      = Name  Speed         AttributeNameMapping (1)
                                                      () Value 1               = RoleAttributeName   = SystemUnitAttributeName
                                                                             1 Speed                 ScanSpeed
```

Example InternalElement “MultiDevice01”

Attributes of the “MultiDevice01”

Interfaces of the “MultiDevice01”

Role attributes describing requirements of the “MultiDevice01”

Mapping between a role related attribute and an attribute of the InternalElement

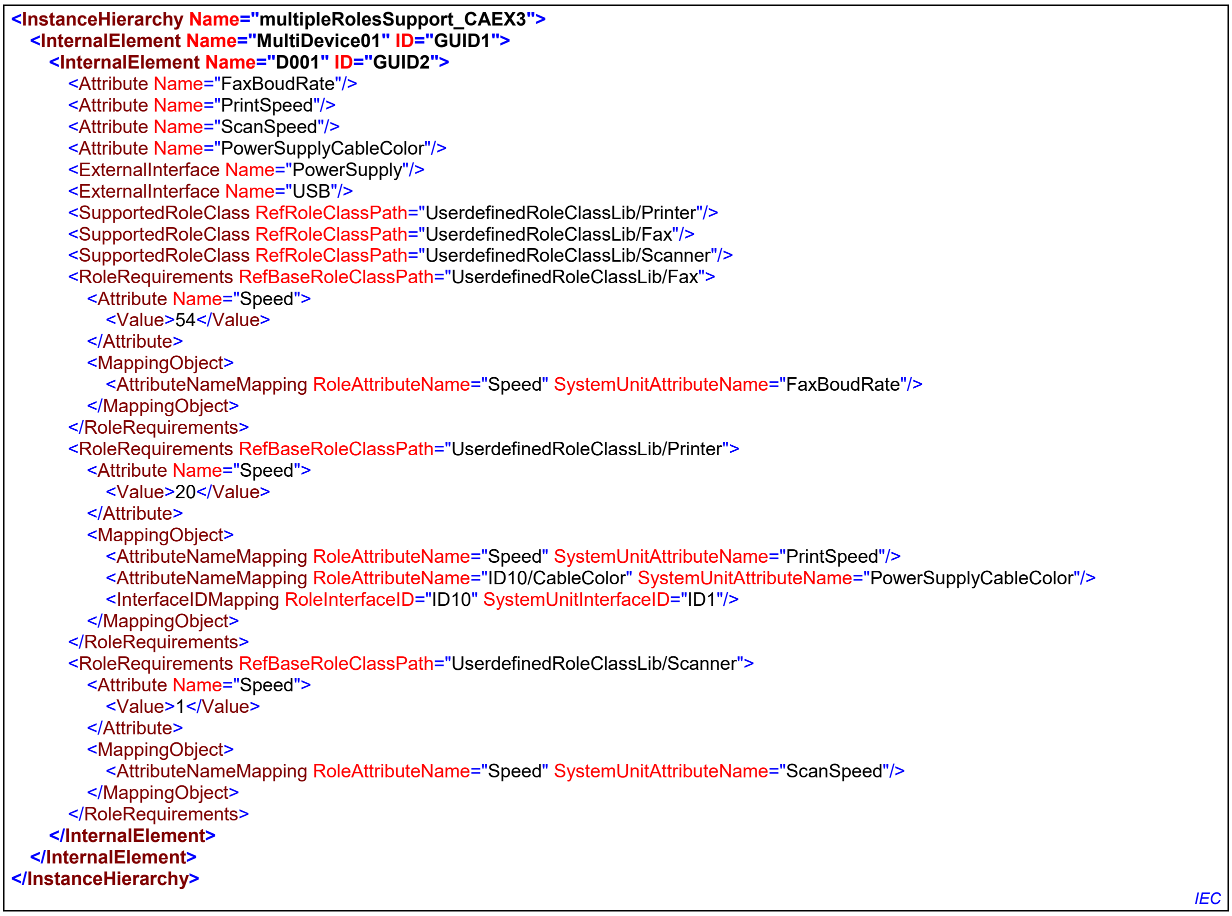
RoleRequirements with references to 3 roles

Mapping between a role interface attribute and an attribute of the InternalElement

Mapping between a role interface and an interface of the InternalElement

IEC

Figure A.30 – Multiple role support



```
<InstanceHierarchy Name="multipleRolesSupport_CAEX3">
  <InternalElement Name="MultiDevice01" ID="GUID1">
    <InternalElement Name="D001" ID="GUID2">
      <Attribute Name="FaxBoudRate"/>
      <Attribute Name="PrintSpeed"/>
      <Attribute Name="ScanSpeed"/>
      <Attribute Name="PowerSupplyCableColor"/>
      <ExternalInterface Name="PowerSupply"/>
      <ExternalInterface Name="USB"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Printer"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Fax"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Scanner"/>
      <RoleRequirements RefBaseRoleClassPath="UserdefinedRoleClassLib/Fax">
        <Attribute Name="Speed">
          <Value>54</Value>
        </Attribute>
        <MappingObject>
          <AttributeNameMapping RoleAttributeName="Speed" SystemUnitAttributeName="FaxBoudRate"/>
        </MappingObject>
      </RoleRequirements>
      <RoleRequirements RefBaseRoleClassPath="UserdefinedRoleClassLib/Printer">
        <Attribute Name="Speed">
          <Value>20</Value>
        </Attribute>
        <MappingObject>
          <AttributeNameMapping RoleAttributeName="Speed" SystemUnitAttributeName="PrintSpeed"/>
          <AttributeNameMapping RoleAttributeName="ID10/CableColor" SystemUnitAttributeName="PowerSupplyCableColor"/>
          <InterfaceIDMapping RoleInterfaceID="ID10" SystemUnitInterfaceID="ID1"/>
        </MappingObject>
      </RoleRequirements>
      <RoleRequirements RefBaseRoleClassPath="UserdefinedRoleClassLib/Scanner">
        <Attribute Name="Speed">
          <Value>1</Value>
        </Attribute>
        <MappingObject>
          <AttributeNameMapping RoleAttributeName="Speed" SystemUnitAttributeName="ScanSpeed"/>
        </MappingObject>
      </RoleRequirements>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

Figure A.31 – XML code of the multiple role support example

A.2.10.3 Usage of the SupportedRoleClass

The CAEX element SupportedRoleClass is a sub-element of the SystemUnitClass. For every SystemUnitClass it allows to define which RoleClasses it supports. This concept enables a computer aided selection of suited SystemUnitClasses for a certain RoleClass.

Regarding supported role classes, the following provisions apply.

- A SystemUnitClass can support an arbitrary number of RoleClasses.
- Children or parents of the supported RoleClass are not automatically supported because they may be incompatible to the SystemUnitClass. If children of a RoleClass are also supported by a SystemUnitClass, they shall be added into the SupportedRoleClass definition.
- For each supported RoleClass, a mapping object can be defined that allows for the definition of the mapping between corresponding attribute names and interfaces. For more information about mappings, see A.2.11.
- CAEX does not provide checks about validity of the supported RoleClasses, neither their existence nor their validity. This shall be part of the CAEX import/export tool or the source/target engineering tool.

Figure A.30 illustrates the application of multiple supported role classes. A detailed CAEX data definition of the SupportedRoleClass is given in A.3.13 and A.3.27.

A.2.11 Use of the CAEX MappingObject

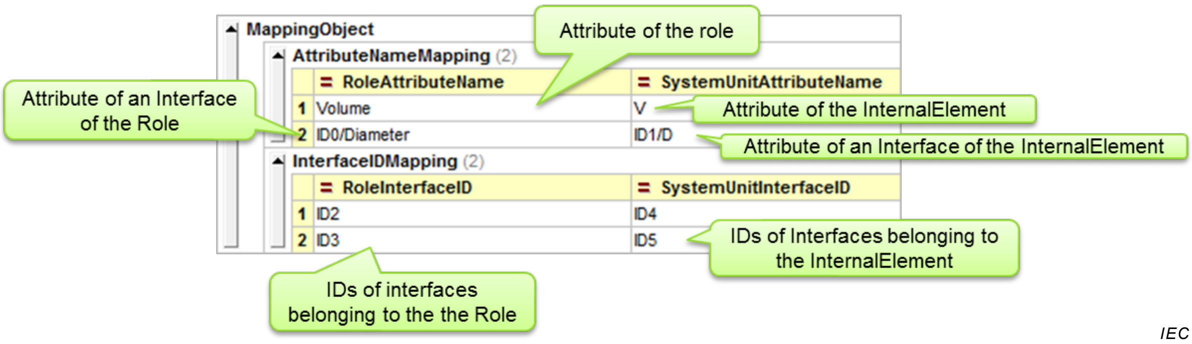
The CAEX MappingObject supports the CAEX role concept. Both, RoleClasses and SystemUnitClasses allow for the definition of attributes and interfaces. If an internal element is associated with a RoleClass, their attribute names may not necessarily be the same. The

MappingObject allows for mapping them to each other. For a detailed CAEX data definition, see A.3.23.

Regarding the MappingObject, the following provisions apply.

- If attributes or interfaces of the RoleRequirement definition needs to be associated to corresponding attributes or interfaces of the InternalElement, a CAEX MappingObject shall be added to the InternalElement. Regarding attributes, the mapping is already implicitly defined if the related attribute names (or paths in case of nested attributes) are identical: for this case no MappingObject is required.
- For each role attribute name mapping, a CAEX element AttributeNameMapping shall be added to the MappingObject. The “RoleAttributeName” shall provide the name (or path in case of a nested attribute) of a role attribute, the “SystemUnitAttributeName” shall provide the corresponding attribute name (or path) of the InternalElement.
- For each interface attribute name mapping, an element AttributeNameMapping shall be added to the MappingObject. The “RoleAttributeName” shall provide the path to the attribute comprising the interfaces ID and the name (or path) of the attribute separated by “/”. The “SystemUnitAttributeName” shall provide the InternalElements interface ID and the corresponding attribute name separated by “/”.
- For each interface mapping, an element InterfaceIDMapping shall be added to the MappingObject. The “RoleInterfaceID” shall provide the ID of the role-interface, the “SystemUnitInterfaceID” shall provide the ID of the corresponding interface of the InternalElement.
- In case an InternalElement attribute does neither have a Value nor a Default Value, but is mapped to a related Attribute of a RoleRequirement or SupportedRoleClass, then these Value or DefaultValue shall not be used instead, they only reflect requirements.
- Multiple roles may have attributes with the same name but with different semantics. The mapping to related attributes of the parent InternalElement shall be defined using a CAEX MappingObject. Mapping of multiple role attributes of different meaning to the same attribute of the related InternalElement is strictly forbidden.

Figure A.32 gives an example for different mapping types. The example RoleClass defines an attribute “Volume”, an interface “Input” and an interface attribute “Diameter”. The Internal-Element defines the attribute “V”, an interface “In” and an interface attribute “D”. The MappingObject defines that “V” is related to “Volume” and “ID0/Diameter” is related to “ID1/D”. Furthermore, it defines that the role interfaces “ID2” and “ID3” correspond to the interfaces “ID4” and “ID5” of the related InternalElement.



MappingObject

AttributeNameMapping (2)

	= RoleAttributeName	= SystemUnitAttributeName
1	Volume	V
2	ID0/Diameter	ID1/D

InterfaceIDMapping (2)

	= RoleInterfaceID	= SystemUnitInterfaceID
1	ID2	ID4
2	ID3	ID5

Attribute of the role
Attribute of an Interface of the Role
Attribute of the InternalElement
Attribute of an Interface of the InternalElement
IDs of Interfaces belonging to the InternalElement
IDs of interfaces belonging to the the Role

IEC

Figure A.32 – CAEX data definition of a MappingObject

The full XML text for this example is shown in Figure A.33.

```
<MappingObject>
  <AttributeNameMapping RoleAttributeName="Volume" SystemUnitAttributeName="V"/>
  <AttributeNameMapping RoleAttributeName="ID0/Diameter" SystemUnitAttributeName="ID1/D"/>
  <InterfaceIDMapping RoleInterfaceID="ID2" SystemUnitInterfaceID="ID4"/>
  <InterfaceIDMapping RoleInterfaceID="ID3" SystemUnitInterfaceID="ID5"/>
</MappingObject>
```

IEC

Figure A.33 – XML code for the data definition of a MappingObject

A.2.12 References to external CAEX files

A.2.12.1 General provisions

CAEX explicitly supports accessing external CAEX files by means of the CAEX element “ExternalReference”. For a detailed CAEX data definition, see A.3.5.

Regarding ExternalReferences, the following provisions apply.

- Each ExternalReference shall reference to another CAEX document of the same schema version.
- Each ExternalReference shall provide a valid URI to the external CAEX document and an Alias which shall be unique within the CAEX document. No other information shall be stored.
- The referenced external CAEX documents shall be valid and accessible.
- The alias may be used for referencing classes or instances. In this case, the reference tag shall begin with the alias name, followed by the alias separator “@”, followed by the path to the referenced class or the ID of the referenced InternalElement or ExternalInterface.
- CAEX InternalLinks or mirror objects are allowed to reference mirror objects that are stored in another file. In this case, the external file(s) shall be referenced as ExternalReference.
- CAEX documents can be splitted across multiple files. Across all files, multiple occurrences of ExternalReferences to the same file(s) are allowed. Circular reference between CAEX files are allowed.

NOTE This means that a CAEX file may reference another CAEX file, but a split file of the same document may reference the same CAEX file again.

A.2.12.2 Example

Figure A.34 gives an example of a CAEX file that requires access to 3 other files. The files “CAEXFile01”, “CAEXFile02” and “CAEXFile03” may contain different libraries which shall be referenced in the main file “CurrentCAEXFile”.

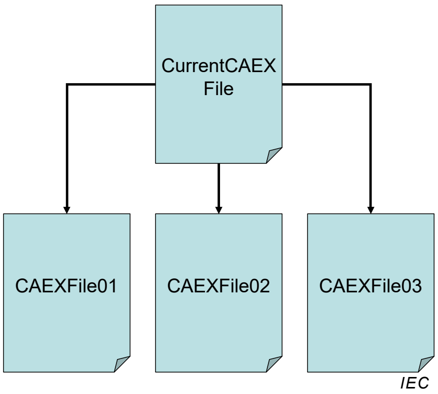
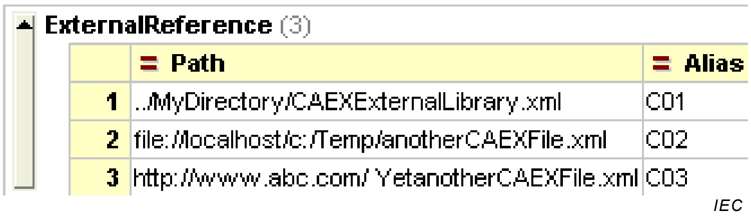


Figure A.34 – Distribution of data in several CAEX files

The described example shall be defined in CAEX by the definition of external references which comprise the URI or the relative path of the external CAEX files and an alias name that allows for internal access to this external CAEX file. Alias names have to be unique and do not contain names of CAEX objects, and only the CAEX document itself shall be referenced by its path. The referencing of external CAEX files is illustrated in Figure A.35.



ExternalReference (3)

	Path	Alias
1	../MyDirectory/CAEXExternalLibrary.xml	C01
2	file://localhost/c:/Temp/anotherCAEXFile.xml	C02
3	http://www.abc.com/ YetanotherCAEXFile.xml	C03

IEC

Figure A.35 – Referencing of external CAEX files

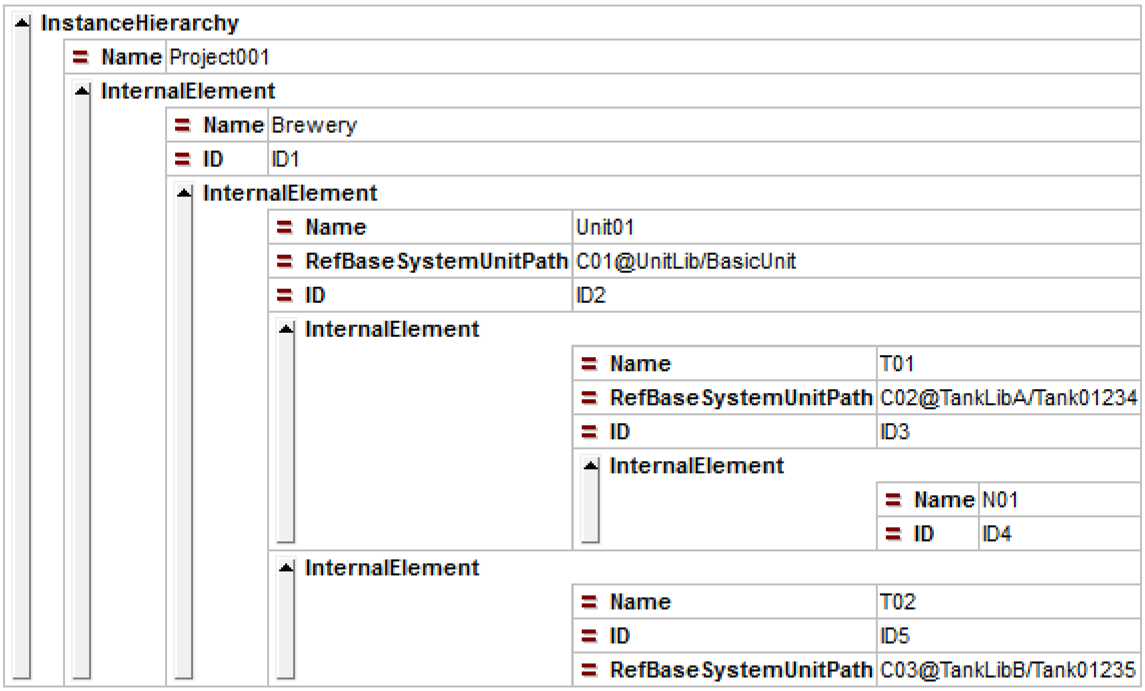
The full XML text of this example is shown Figure A.36.

```
<ExternalReference Path=".../MyDirectory/CAEXExternalLibrary.xml" Alias="C01"/>
<ExternalReference Path="file://localhost/c:/Temp/anotherCAEXFile.xml" Alias="C02"/>
<ExternalReference Path="http://www.abc.com/ YetanotherCAEXFile.xml" Alias="C03"/>
```

IEC

Figure A.36 – XML code for referencing of external CAEX files

Figure A.37 gives an example about how to use the defined references to external CAEX files. The reference to the external file is described by means of the alias name. This name is separated by the alias separator "@" and is followed by the full path to the corresponding class.

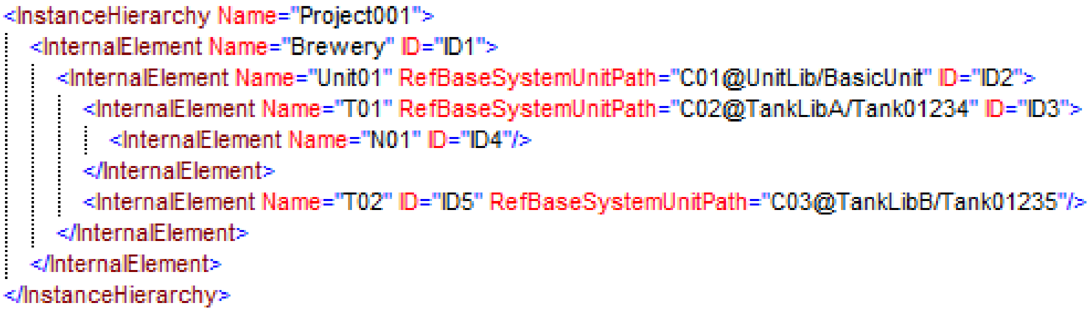


```
InstanceHierarchy
    Name  Project001
    InternalElement
        Name  Brewery
        ID    ID1
        InternalElement
            Name                   Unit01
            RefBaseSystemUnitPath  C01@UnitLib/BasicUnit
            ID                     ID2
            InternalElement
                Name                   T01
                RefBaseSystemUnitPath  C02@TankLibA/Tank01234
                ID                     ID3
                InternalElement
                    Name  N01
                    ID    ID4
            InternalElement
                Name                   T02
                ID                     ID5
                RefBaseSystemUnitPath  C03@TankLibB/Tank01235
```

IEC

Figure A.37 – Example of how to use alias names

The full XML text for this example is shown in Figure A.38.



```
<InstanceHierarchy Name="Project001">
  <InternalElement Name="Brewery" ID="ID1">
    <InternalElement Name="Unit01" RefBaseSystemUnitPath="C01@UnitLib/BasicUnit" ID="ID2">
      <InternalElement Name="T01" RefBaseSystemUnitPath="C02@TankLibA/Tank01234" ID="ID3">
        <InternalElement Name="N01" ID="ID4"/>
      </InternalElement>
      <InternalElement Name="T02" ID="ID5" RefBaseSystemUnitPath="C03@TankLibB/Tank01235"/>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

IEC

Figure A.38 – XML code for the alias example

A.3 CAEX schema definition

A.3.1 General

The CAEX model is stored in the XML schema file “CAEX_ClassModel_V.3.0.xsd”, and consists of abstract XML elements and attributes for the specification of all plant items. Elements may have sub-elements and attributes.

CAEX itself has an object-oriented architecture and comprises the following type definitions:

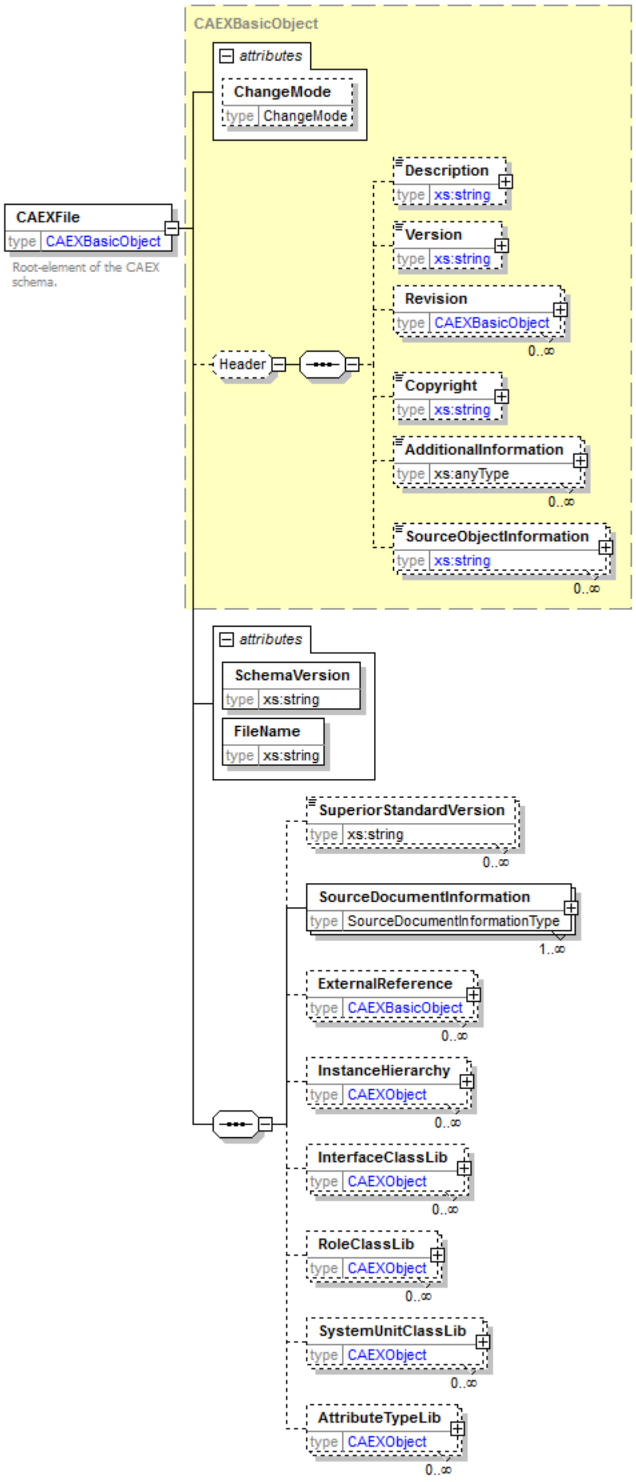
schema location:	CAEX_ClassModel_V.3.0.xsd
attribute form default:	**unqualified**
element form default:	**qualified**
targetNamespace:	**http://www.dke.de/CAEX**

Elements	Groups	Complex types	Simple types
CAEXFile	Header	AttributeFamilyType	ChangeMode

		AttributeType	
		AttributeValueRequirementType	
		CAEXBasicObject	
		CAEXObject	
		InterfaceClassType	
		InterfaceFamilyType	
		InternalElementType	
		MappingType	
		RoleClassType	
		RoleFamilyType	
		SourceDocumentInformationType	
		SystemUnitClassType	
		SystemUnitFamilyType	

A.3.2 Element CAEXFile

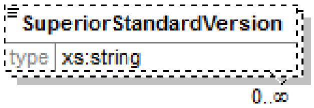
- The element “CAEXFile” describes the root element of the data exchange format. The attribute “SchemaVersion” shall store the CAEX version required, see A.2.2.4.
- The attribute “FileName” shall be used and stores the name of the transferred file.
- The main sub-elements of CAEX comprise libraries and instance hierarchies as well as reference definitions for external CAEX files. See A.2.2 for details.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXBasicObject</b></td></tr>
<tr><td>properties</td><td>content complex</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation SuperiorStandardVersion SourceDocumentInformation ExternalReference InstanceHierarchy InterfaceClassLib RoleClassLib SystemUnitClassLib AttributeTypeLib</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>SchemaVersion</td><td><b>xs:string</b></td><td>required</td><td></td><td>3.0</td><td></td></tr>
<tr><td>FileName</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.3 CAEXFile/SuperiorStandardVersion

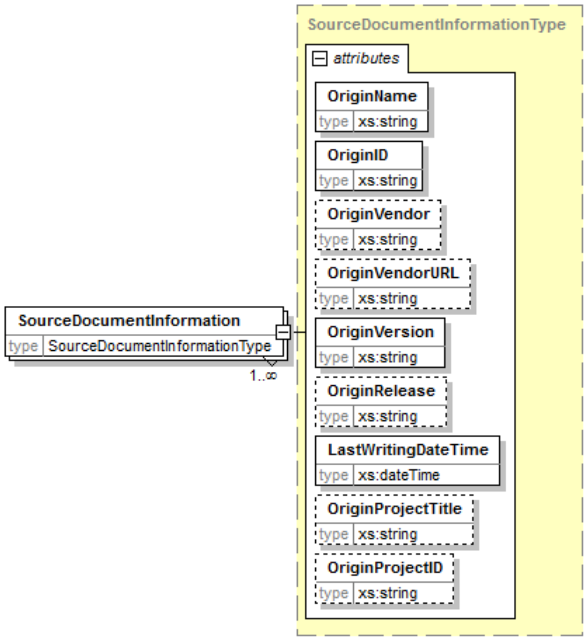
The CAEX element describes the version of a superior standard. The version string is defined in the superior standard.

The application and normative provisions are provided in A.2.2.3.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><b>xs:string</b></td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>simple</td></tr>
</table></td></tr>
</table>

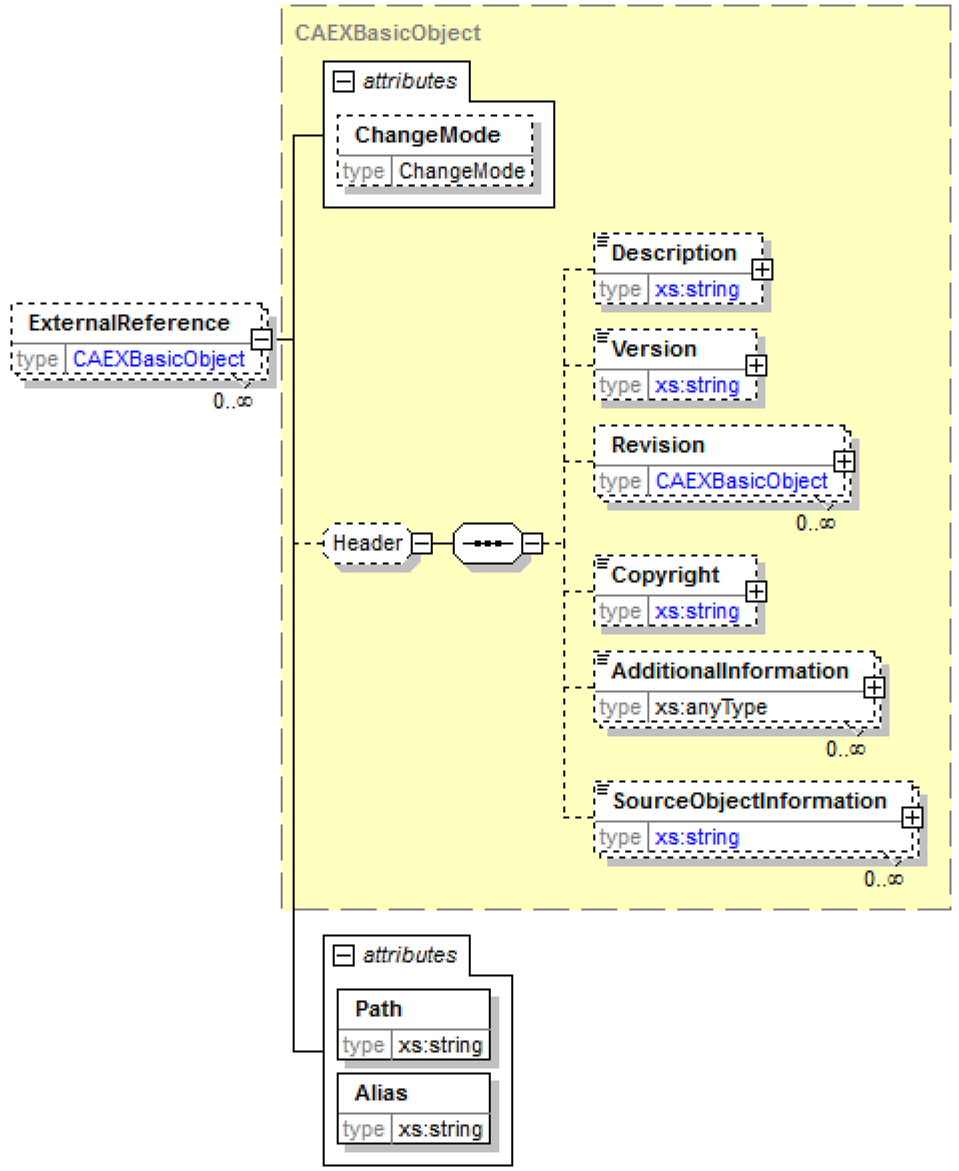
A.3.4 CAEXFile/ SourceDocumentInformation

This CAEX element provides information about the source(s) of the CAEX document. The application and normative provisions are defined in A.2.2.5, the type is described in A.3.26.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><u>SourceDocumentInformationType</u></td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>1</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></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>OriginName</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>OriginID</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>OriginVendor</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>OriginVendorURL</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>OriginVersion</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>OriginRelease</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>LastWritingDateTime</td><td><b>xs:dateTime</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>OriginProjectTitle</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>OriginProjectID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.5 CAEXFile/ExternalReference

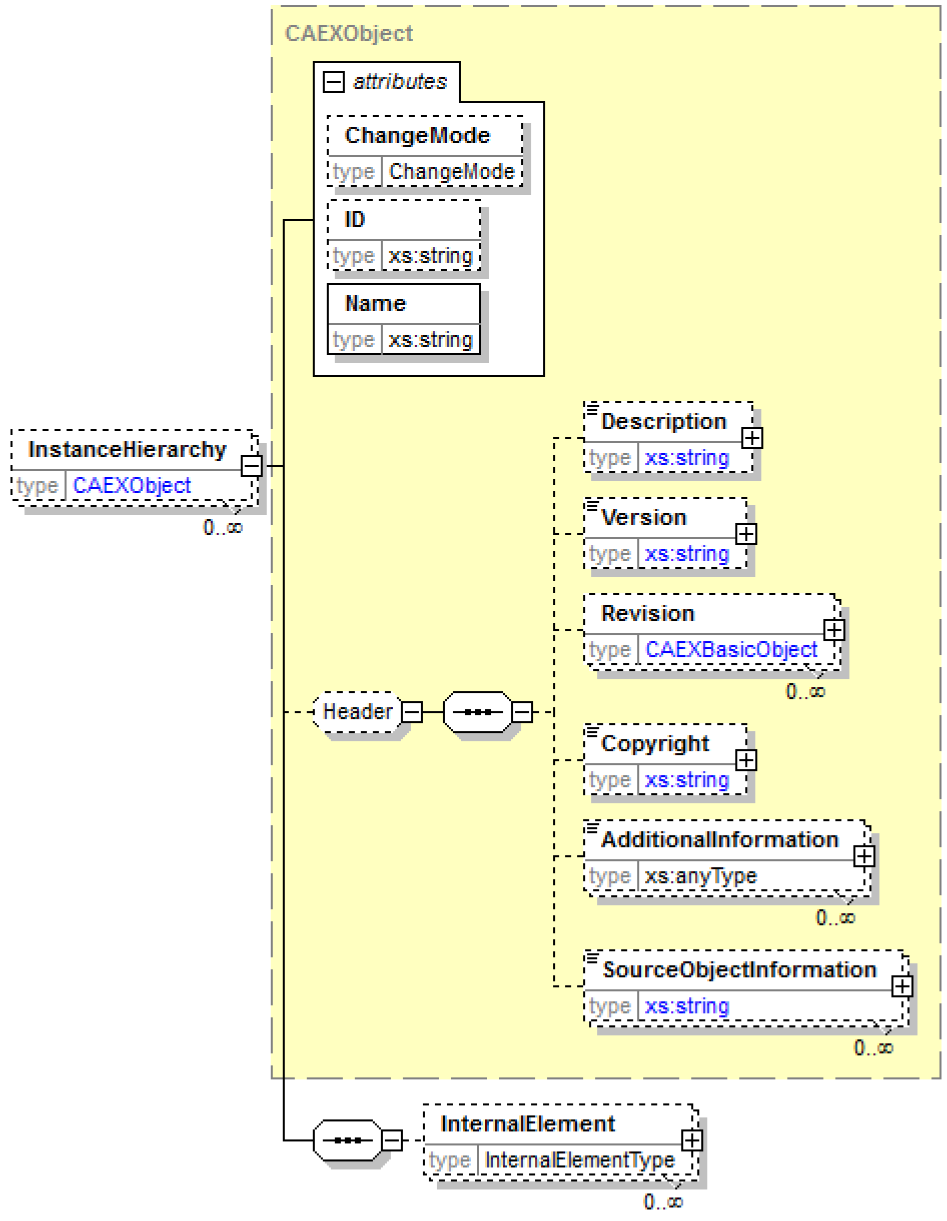
This CAEX element allows for the definition of references to external CAEX files. See A.2.12 for details and examples.

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


</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXBasicObject</b></td></tr>
<tr><td>properties</td><td>
<table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table>
</td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>Path</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Alias</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

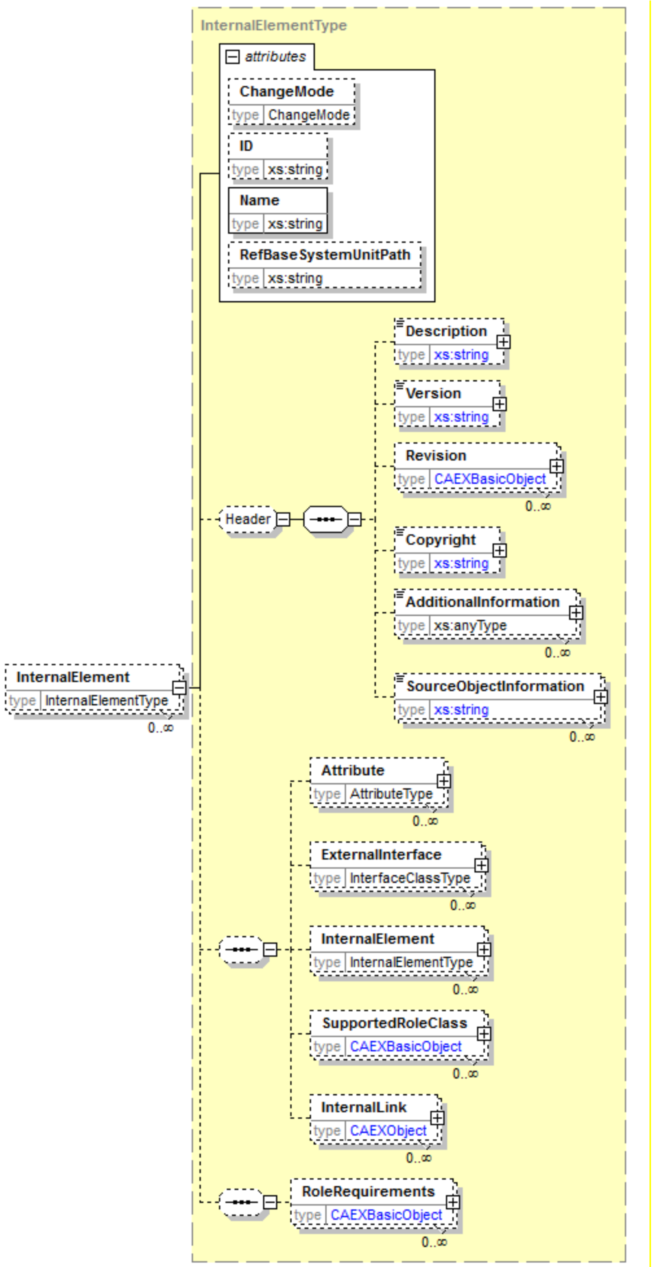
A.3.6 CAEXFile/InstanceHierarchy

The CAEX element “InstanceHierarchy” allows for the storage of hierarchical object information. CAEX supports storage of multiple instance hierarchies in the same CAEX file. See A.2.2.1 and A.2.8.2 for details and examples.

diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string InstanceHierarchy type CAEXObject 0..∞ Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ InternalElement type InternalElementType 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **InternalElement**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state ID **xs:string** optional Name **xs:string** required

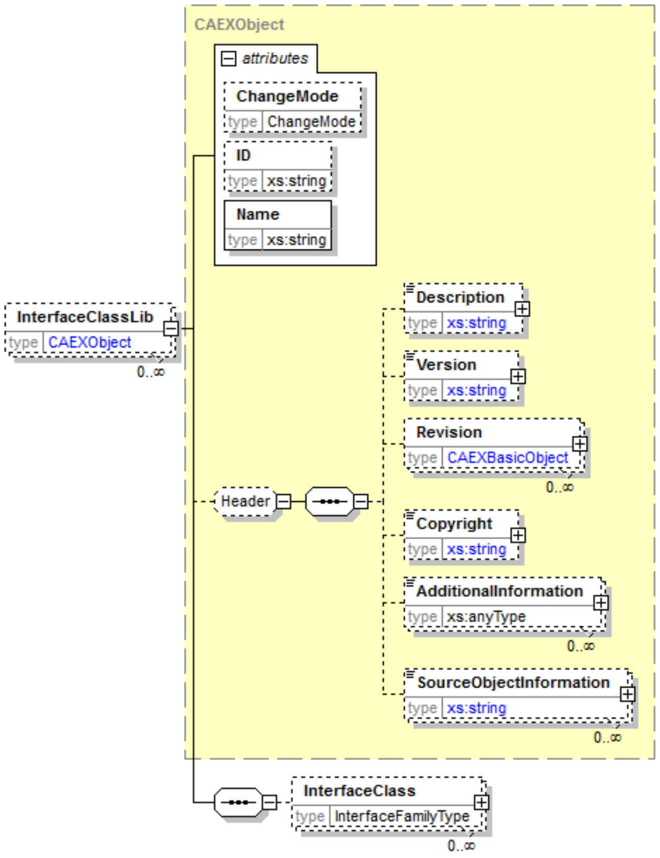
A.3.7 CAEXFile/InstanceHierarchy/InternalElement

The CAEX element “InternalElement” allows for the storage of nested object information. See A.2.2.1, A.2.8.2 and A.3.23 for details and examples.

diagram	 InternalElementType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseSystemUnitPath type xs:string Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ InternalElement type InternalElementType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InternalElement type InternalElementType 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ InternalLink type CAEXObject 0..∞ RoleRequirements type CAEXBasicObject 0..∞
namespace	http://www.dke.de/CAEX
type	**InternalElementType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface** **InternalElement** **SupportedRoleClass** **InternalLink** **RoleRequirements** **MappingObject**
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / **ChangeMode** / optional / state ID / **xs:string** / optional Name / **xs:string** / required RefBaseSystemUnitPath / **xs:string** / optional

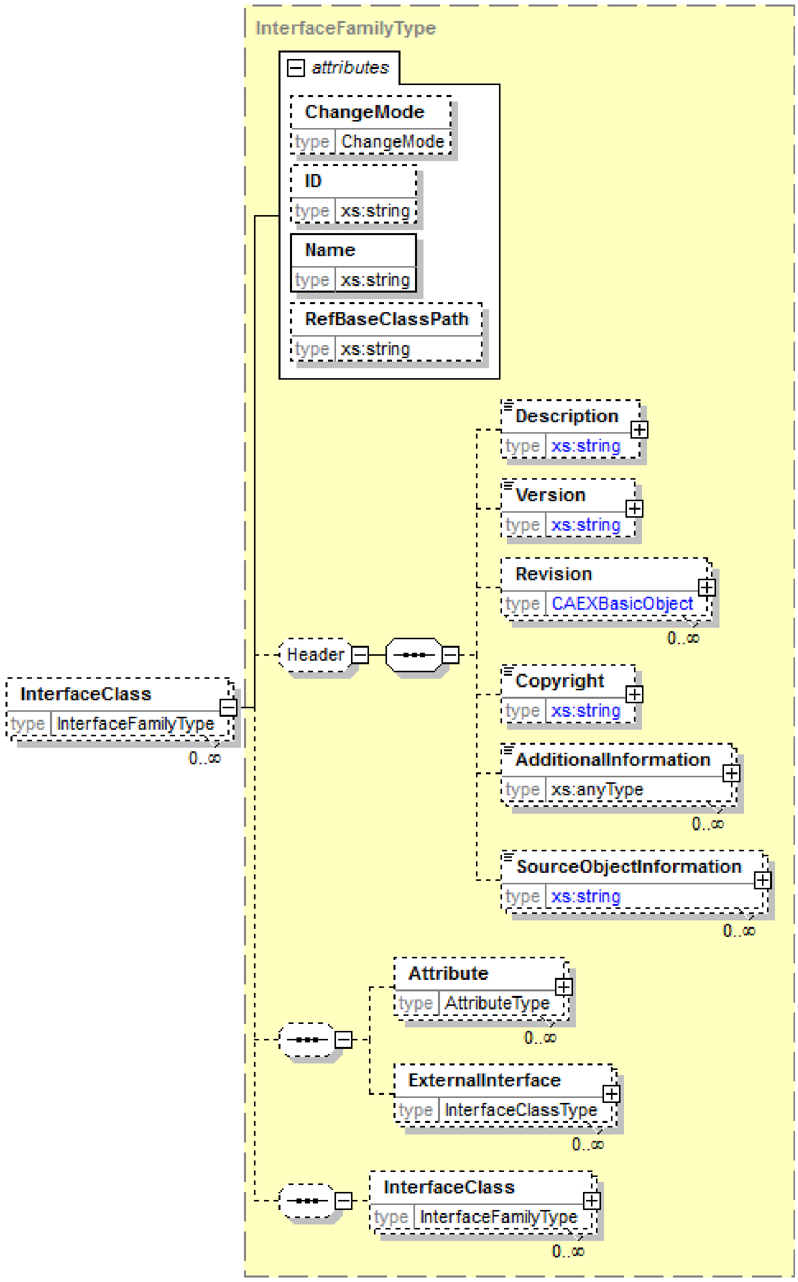
A.3.8 CAEXFile/InterfaceClassLib

The CAEX element “InterfaceClassLib” allows for collecting InterfaceClasses within libraries. See A.2.6 for details and examples.

diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string InterfaceClassLib type CAEXObject 0..∞ Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ InterfaceClass type InterfaceFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **InterfaceClass**
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / **ChangeMode** / optional / state ID / **xs:string** / optional Name / **xs:string** / required

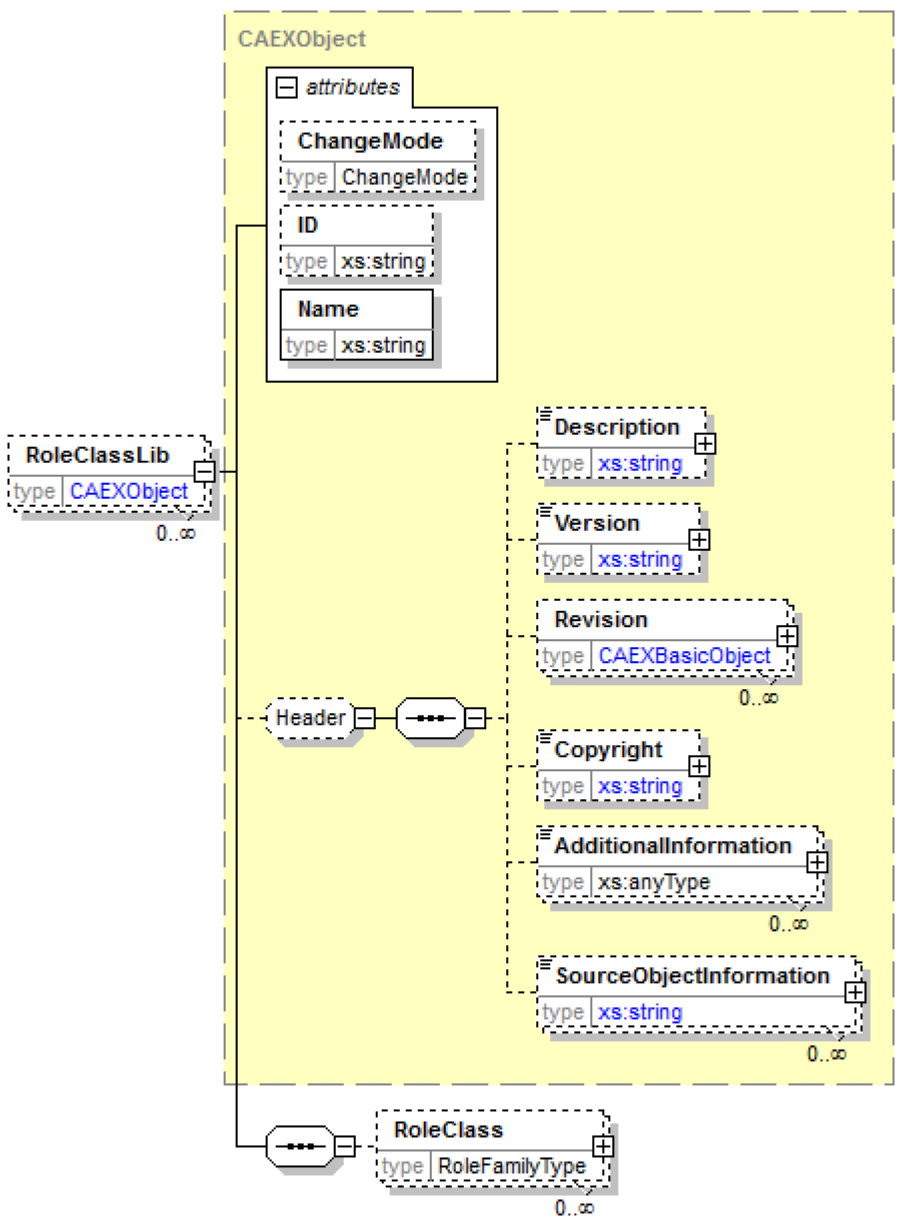
A.3.9 CAEXFile/InterfaceClass

The CAEX element “InterfaceClass” allows for the storage of interface class definitions. See A.2.6 for details and examples.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>InterfaceFamilyType</td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InterfaceClass</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>ChangeMode</td><td>ChangeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseClassPath</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

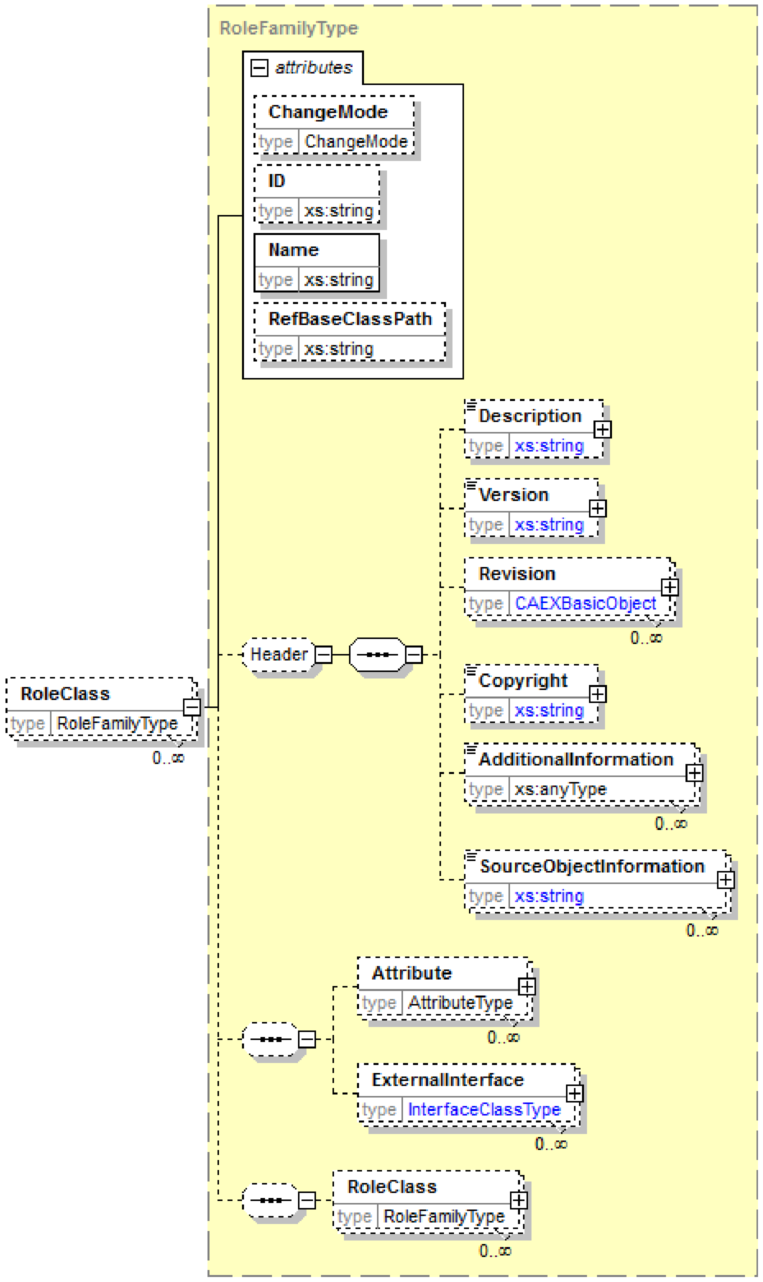
A.3.10 CAEXFile/RoleClassLib

The CAEX element “RoleClassLib” allows for collecting RoleClasses within libraries. See A.2.7 for details and examples.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXObject</b></td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</td></tr></table></td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</b> <b>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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr><tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr><tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr></table></td></tr>
</table>

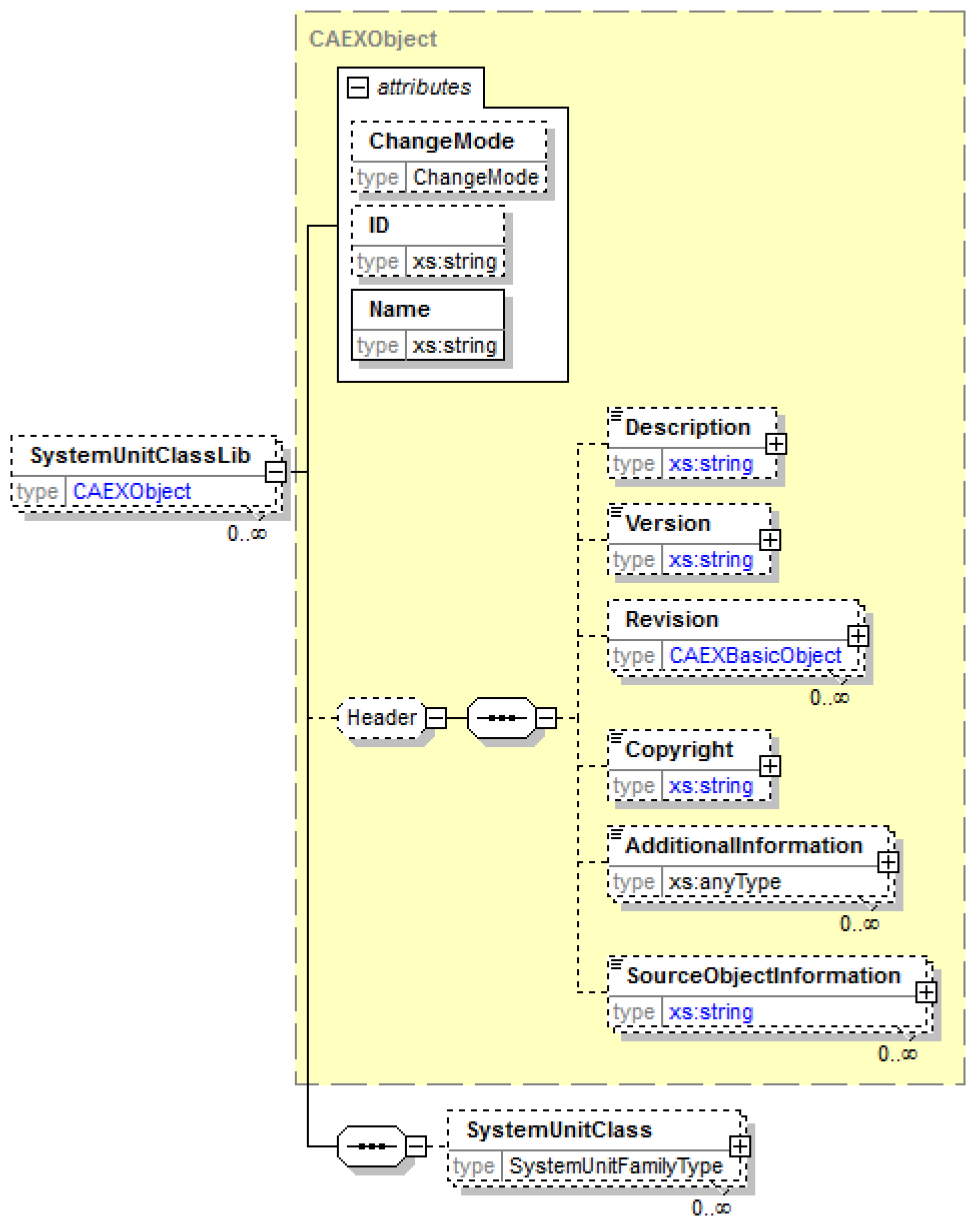
A.3.11 CAEXFile/RoleClass

The CAEX element “RoleClass” allows for the storage of role class definitions. See A.2.7 for details and examples.

diagram	 RoleFamilyType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ RoleClass type RoleFamilyType 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ RoleClass type RoleFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	**RoleFamilyType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface RoleClass**
attributes	Name | Type | Use | Default | Fixed | Annotation ChangeMode | **ChangeMode** | optional | state ID | **xs:string** | optional Name | **xs:string** | required RefBaseClassPath | **xs:string** | optional

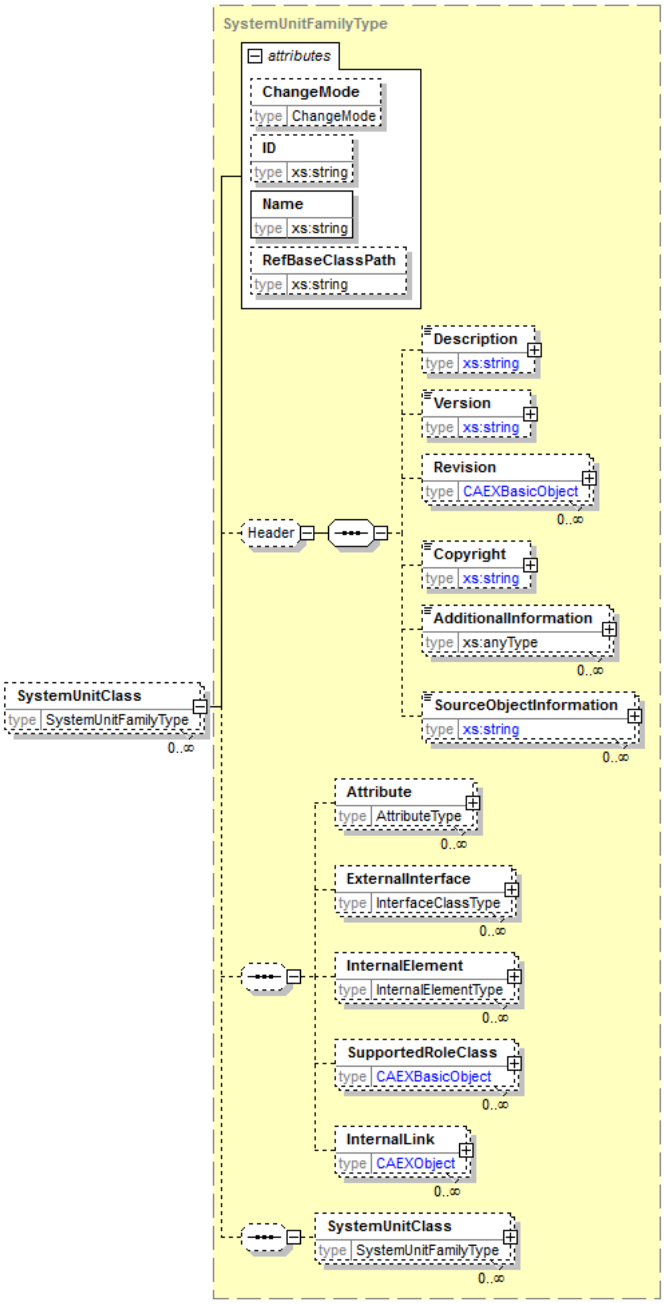
A.3.12 CAEXFile/SystemUnitClassLib

The CAEX element “SystemUnitClassLib” allows for collecting SystemUnitClasses within libraries. See A.2.3 for details and examples.

diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string SystemUnitClassLib type CAEXObject 0..∞ Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ SystemUnitClass type SystemUnitFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **SystemUnitClass**
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / **ChangeMode** / optional / state ID / **xs:string** / optional Name / **xs:string** / required

A.3.13 CAEXFile/SystemUnitClass

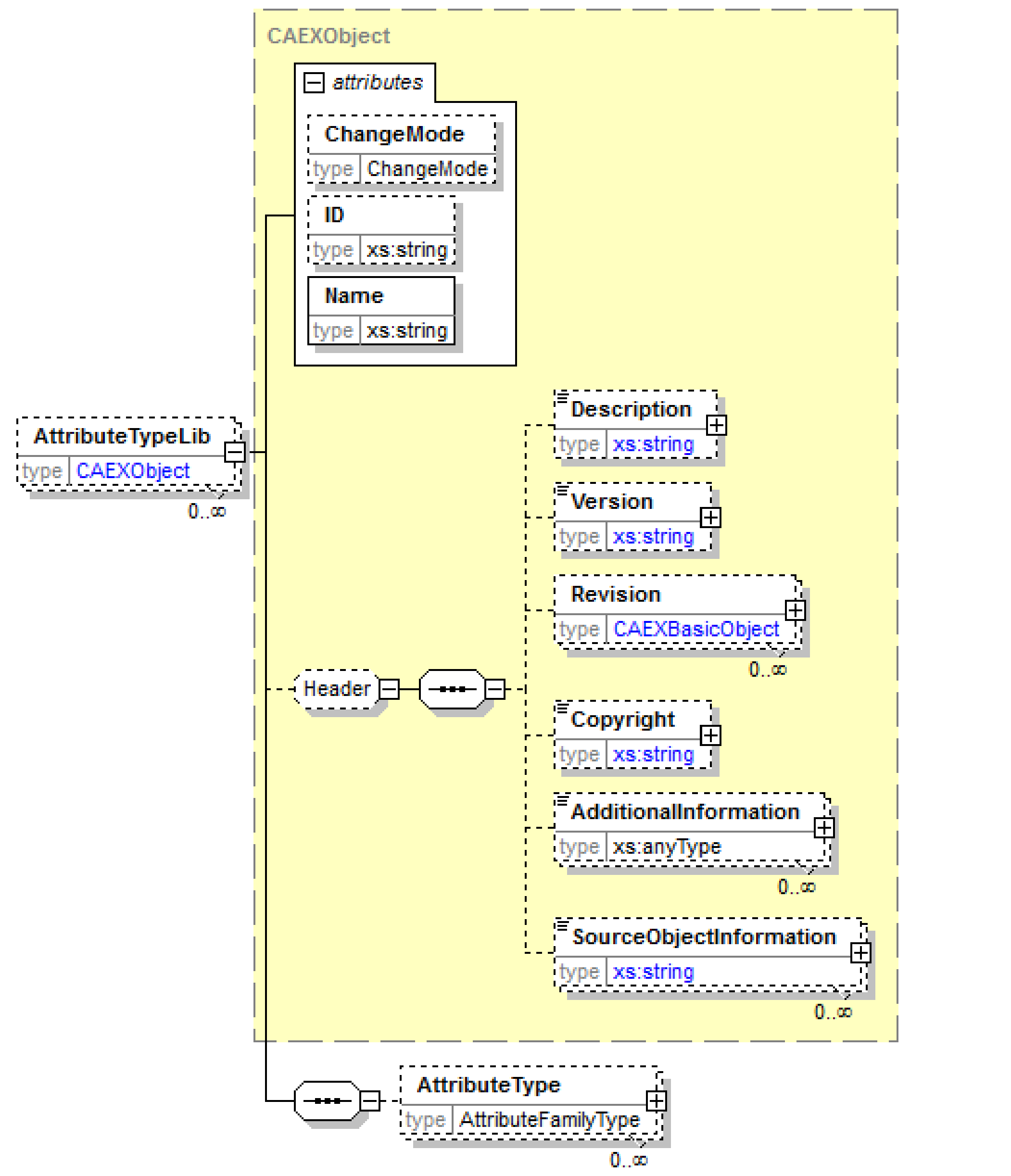
The CAEX element “SystemUnitClass” allows for the storage of system unit class definitions. See A.3.27 and A.2.3 for details and examples.

diagram	 SystemUnitFamilyType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SystemUnitClass type SystemUnitFamilyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InternalElement type InternalElementType 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ InternalLink type CAEXObject 0..∞ SystemUnitClass type SystemUnitFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	SystemUnitFamilyType
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InternalElement SupportedRoleClass InternalLink SystemUnitClass
Attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / ChangeMode / optional / state ID / xs:string / optional Name / xs:string / required RefBaseClassPath / xs:string / optional

A.3.14 CAEXFile/AttributeTypeLib

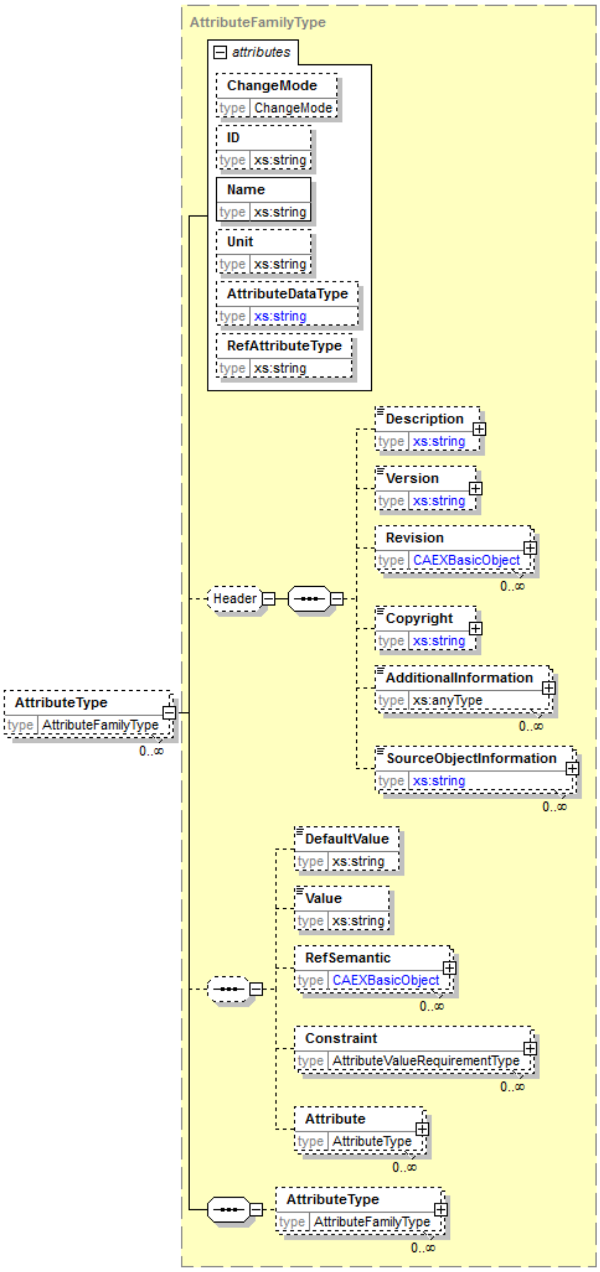
A.3.14.1 General

This CAEX element is a container element for a hierarchy of attribute type definitions. CAEX supports multiple attribute type libraries. See A.2.5 for details and examples.

diagram	
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **AttributeType**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state ID **xs:string** optional Name **xs:string** required

A.3.14.2 Element CAEXFile/AttributeTypeLib/AttributeType

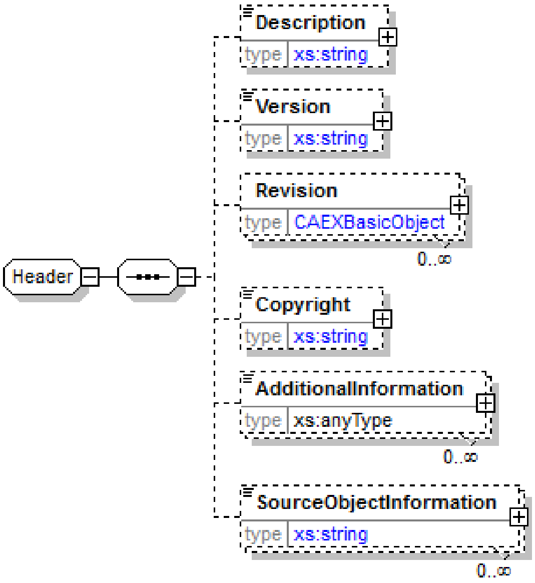
This CAEX element is a container element for a hierarchy of attribute type definitions. CAEX supports multiple attribute type libraries. See A.2.4 and A.2.5 for details and examples.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><b>AttributeFamilyType</b></td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute AttributeType</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td><b>derived by: xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.15 Group Header

A.3.15.1 General

The CAEX Group Header defines version information that is optionally available for each CAEX object. The Header is a part of the CAEX basis object “CAEXBasicObject” which is the root base class for every CAEX element. See A.2.2.7 for details.

diagram	 Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞
namespace	http://www.dke.de/CAEX
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation**
used by	complexType **CAEXBasicObject**

A.3.15.2 Element Header/Description

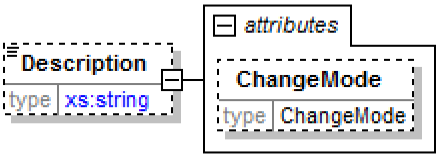
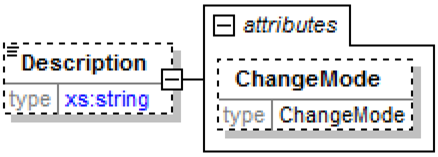
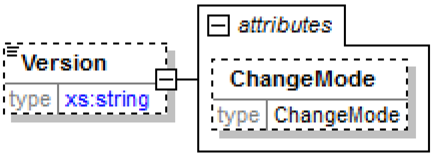
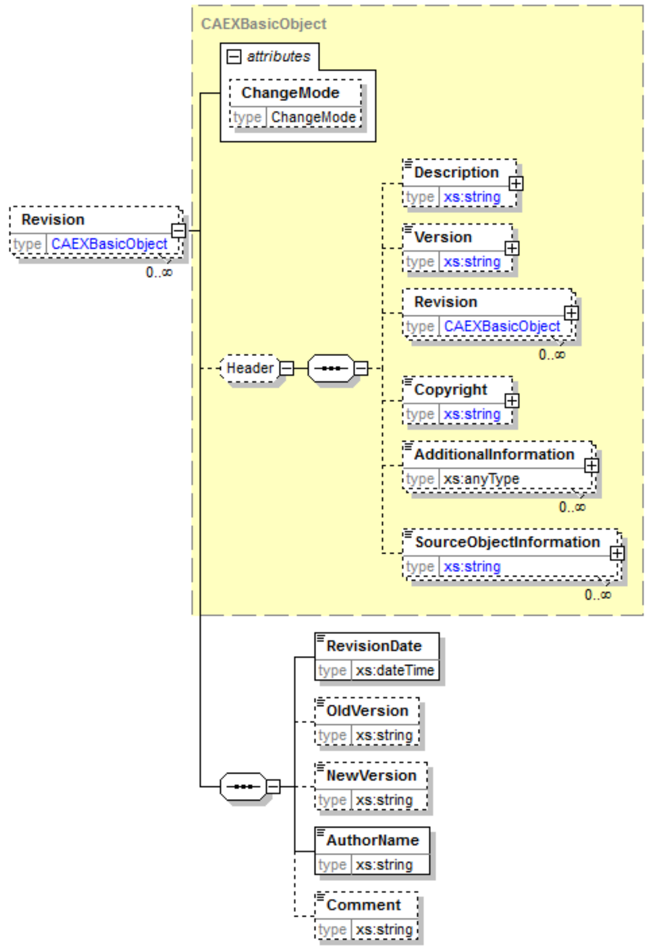
diagram	 Description type xs:string attributes ChangeMode type ChangeMode
namespace	http://www.dke.de/CAEX
type	extension of **xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

diagram	 Description type xs:string attributes ChangeMode type ChangeMode
namespace	http://www.dke.de/CAEX
type	extension of **xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

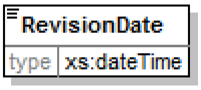
A.3.15.3 Element Header/Version

diagram	 Version type xs:string attributes ChangeMode type ChangeMode
namespace	http://www.dke.de/CAEX
type	extension of **xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

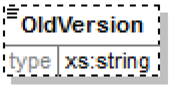
A.3.15.4 Element Header/Revision

diagram	
namespace	http://www.dke.de/CAEX
type	extension of **CAEXBasicObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **RevisionDate** **OldVersion** **NewVersion** **AuthorName** **Comment**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

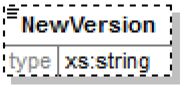
A.3.15.5 Element Header/Revision/RevisionDate

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:dateTime**
properties	isRef 0 content simple

A.3.15.6 Element Header/Revision/OldVersion

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

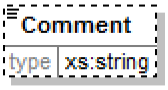
A.3.15.7 Element Header/Revision/NewVersion

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

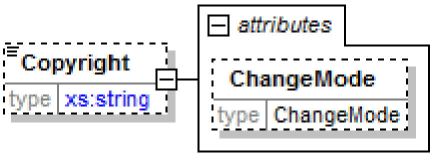
A.3.15.8 Element Header/Revision/AuthorName

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 content simple

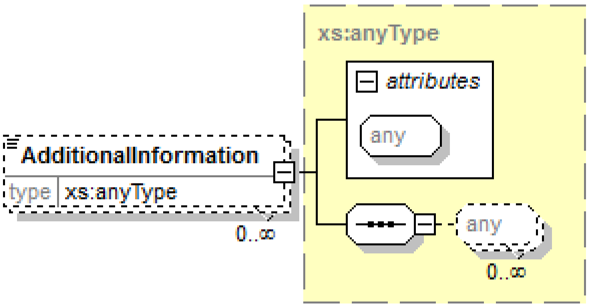
A.3.15.9 Element Header/Revision/Comment

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

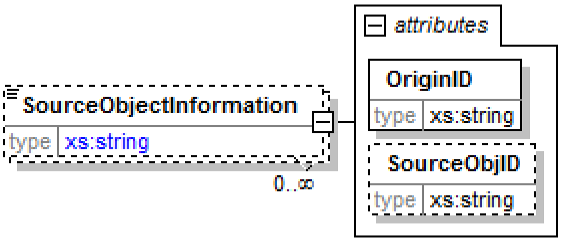
A.3.15.10 Element Header/Copyright

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of xs:string</td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>1</td></tr><tr><td>content</td><td>complex</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>ChangeMode</td><td>ChangeMode</td><td>optional</td><td>state</td><td></td><td></td></tr></table></td></tr>
</table>

A.3.15.11 Element Header/AdditionalInformation

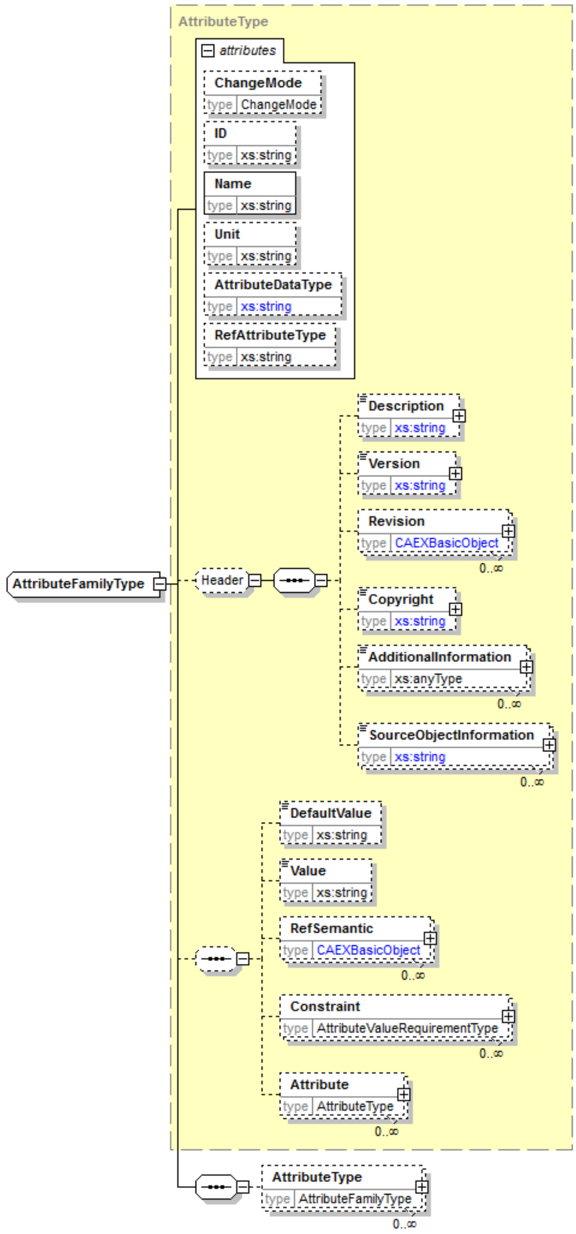
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>A.1.1.1.1 http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>A.1.1.1.2 xs:anyType</td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</td></tr><tr><td>mixed</td><td>true</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></table></td></tr>
</table>

A.3.15.12 Element Header/SourceObjectInformation

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>xs:string</b></td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</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>OriginID</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr><tr><td>SourceObjID</td><td><b>xs:string</b></td><td></td><td></td><td></td><td></td></tr></table></td></tr>
</table>

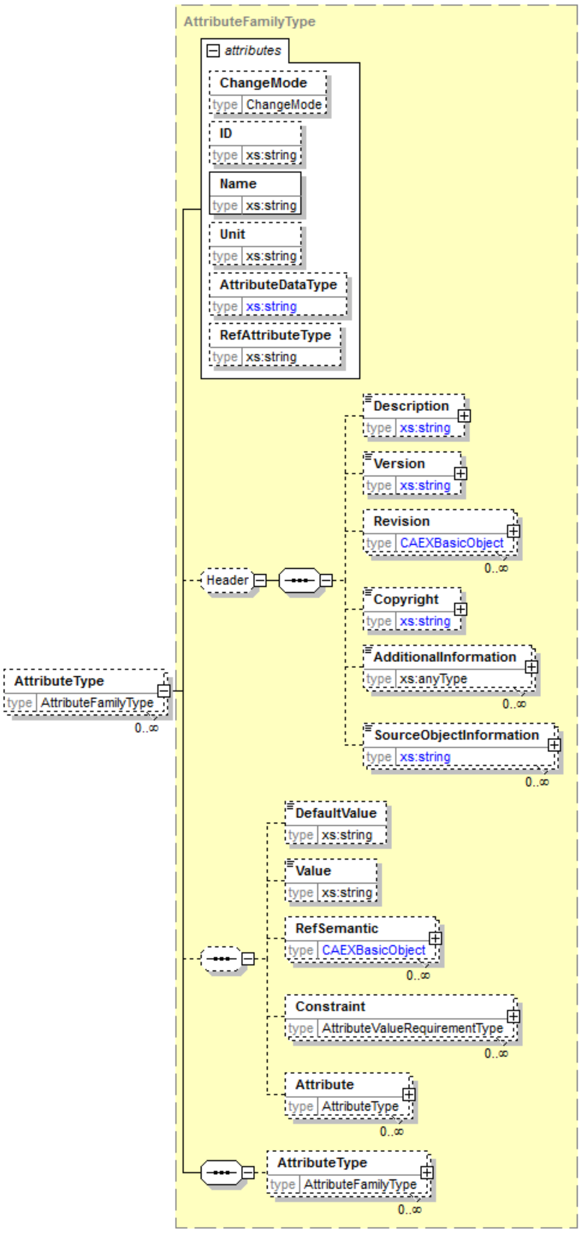
A.3.16 CAEX complex type AttributeFamilyType

This CAEX element defines base structures for modeling attribute type definitions. See A.2.4 and A.2.5.

<table>
<tr><td>diagram</td><td> </td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of AttributeType</td></tr>
<tr><td>properties</td><td>base AttributeType</td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute AttributeType</td></tr>
<tr><td>used by</td><td>elements CAEXFile/AttributeTypeLib/AttributeType
AttributeFamilyType/AttributeType</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>ChangeMode</td><td>ChangeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td>derived by: xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.17 CAEX complex type AttributeFamilyType/AttributeType

This CAEX element allows the class definition for attribute types in an attribute type library. Its application is described in A.2.4 and A.2.5.

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

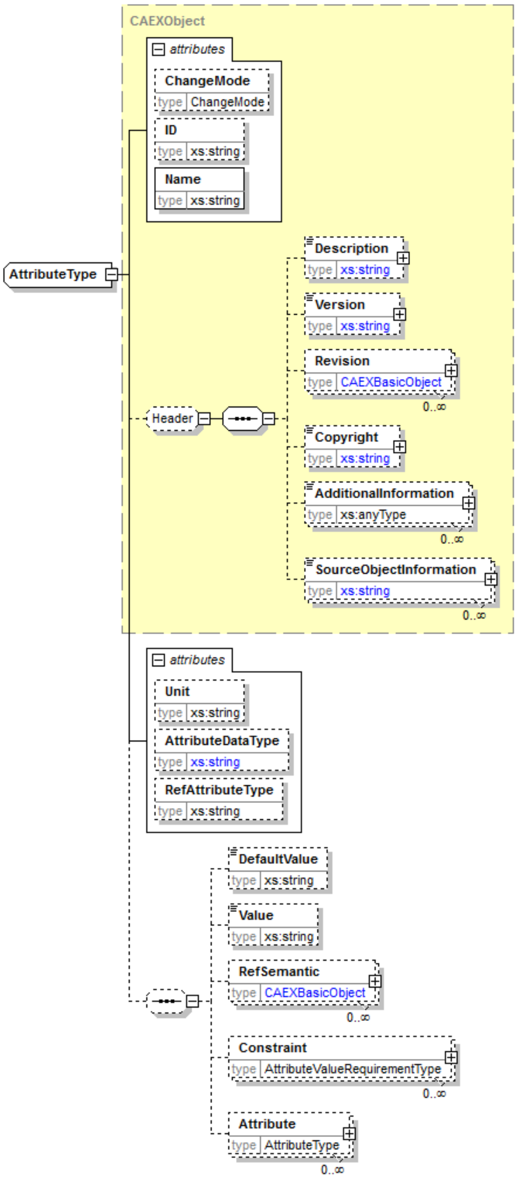
AttributeFamilyType
attributes
ChangeMode
type ChangeMode
ID
type xs:string
Name
type xs:string
Unit
type xs:string
AttributeDataType
type xs:string
RefAttributeType
type xs:string
Description
type xs:string
Version
type xs:string
Revision
type CAEXBasicObject
0..∞
Header
Copyright
type xs:string
AdditionalInformation
type xs:anyType
0..∞
SourceObjectInformation
type xs:string
0..∞
AttributeType
type AttributeFamilyType
0..∞
DefaultValue
type xs:string
Value
type xs:string
RefSemantic
type CAEXBasicObject
0..∞
Constraint
type AttributeValueRequirementType
0..∞
Attribute
type AttributeType
0..∞
AttributeType
type AttributeFamilyType
0..∞

</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><b>AttributeFamilyType</b></td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</b> <b>DefaultValue</b> <b>Value</b> <b>RefSemantic</b> <b>Constraint</b> <b>Attribute</b> <b>AttributeType</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td><b>derived by: xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

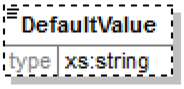
A.3.18 CAEX complex type AttributeType

A.3.18.1 General

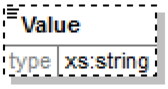
The CAEX type AttributeType is the base type for all CAEX attribute definitions. It is used for modeling attributes in the AttributeTypeLibrary, in class libraries and in object instances. See A.2.4 for details and examples.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of CAEXObject</td></tr>
<tr><td>properties</td><td>base CAEXObject</td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute</td></tr>
<tr><td>used by</td><td><table>
<tr><td>elements</td><td>InterfaceClassType/Attribute RoleClassType/Attribute SystemUnitClassType/Attribute InternalElementType/RoleRequirements/Attribute AttributeType/Attribute</td></tr>
<tr><td>complexType</td><td>AttributeFamilyType</td></tr>
</table></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>ChangeMode</td><td>ChangeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td>derived by: xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

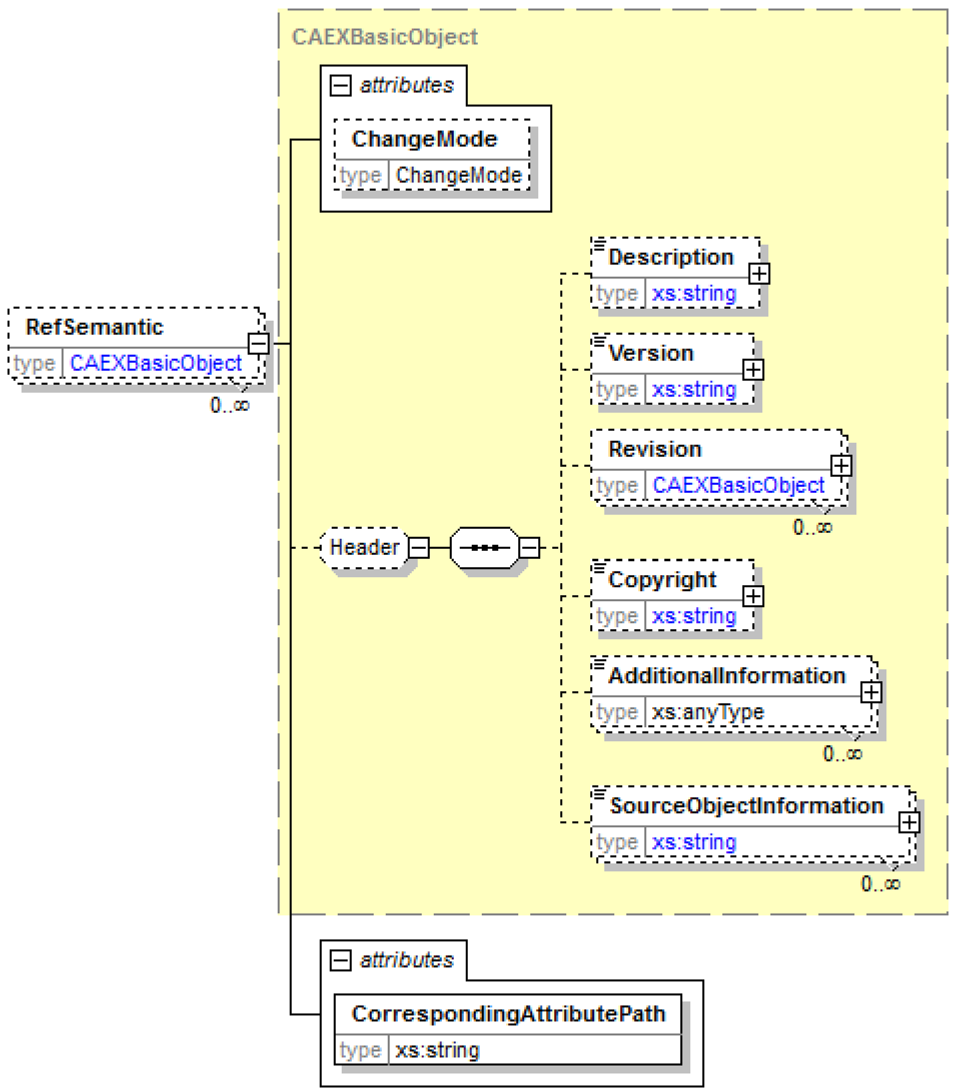
A.3.18.2 Element AttributeType/DefaultValue

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

A.3.18.3 Element AttributeType/Value

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

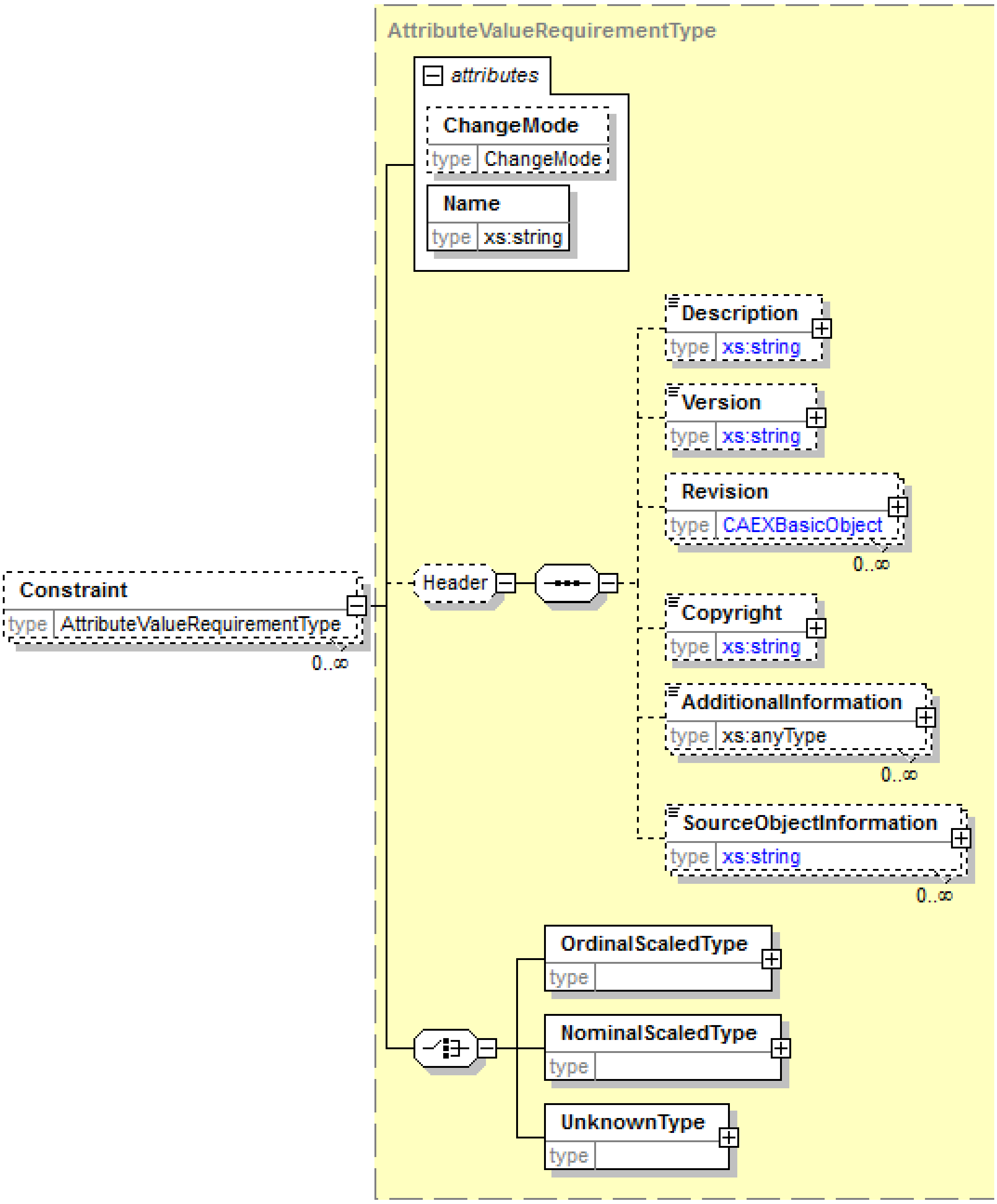
A.3.18.4 Element AttributeType/RefSemantic

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

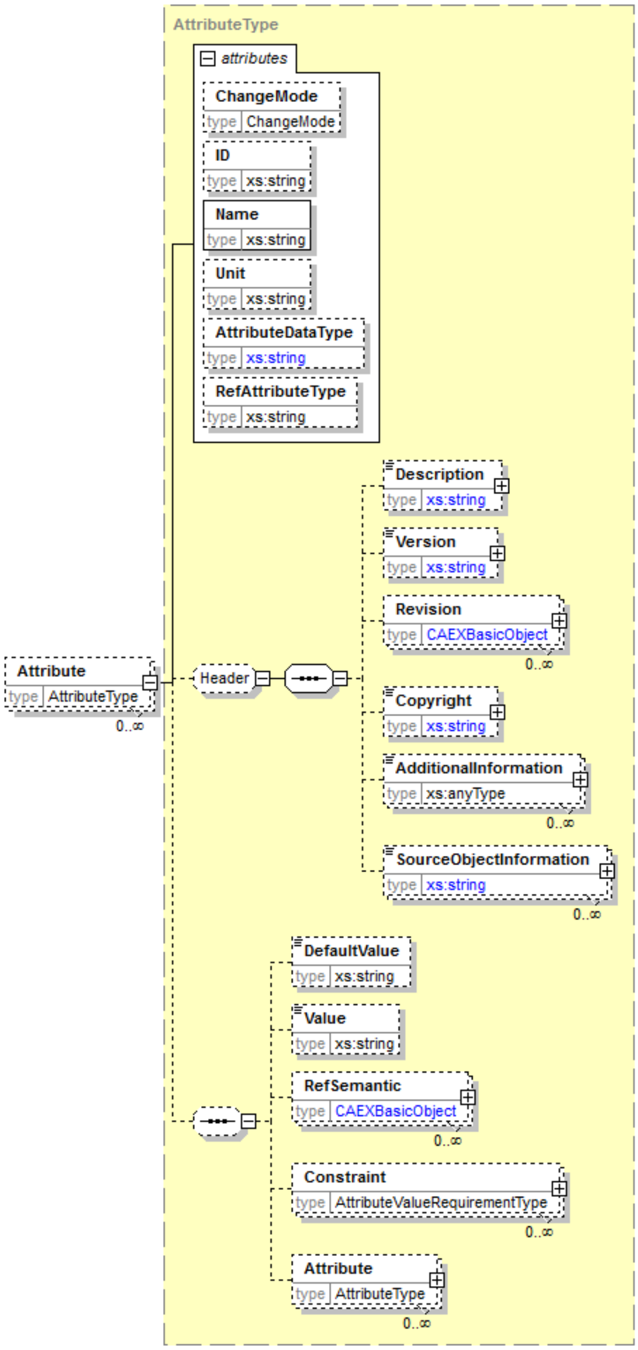
CAEXBasicObject
attributes
ChangeMode
type ChangeMode
RefSemantic
type CAEXBasicObject
0..∞
Description
type xs:string
Version
type xs:string
Revision
type CAEXBasicObject
0..∞
Header
Copyright
type xs:string
AdditionalInformation
type xs:anyType
0..∞
SourceObjectInformation
type xs:string
0..∞
attributes
CorrespondingAttributePath
type xs:string

</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXBasicObject</b></td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</td></tr></table></td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr><tr><td>CorrespondingAttributePath</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr></table></td></tr>
</table>

A.3.18.5 Element AttributeType/Constraint

diagram	
namespace	http://www.dke.de/CAEX
type	AttributeValueRequirementType
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation OrdinalScaledType NominalScaledType UnknownType
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state Name **xs:string** required

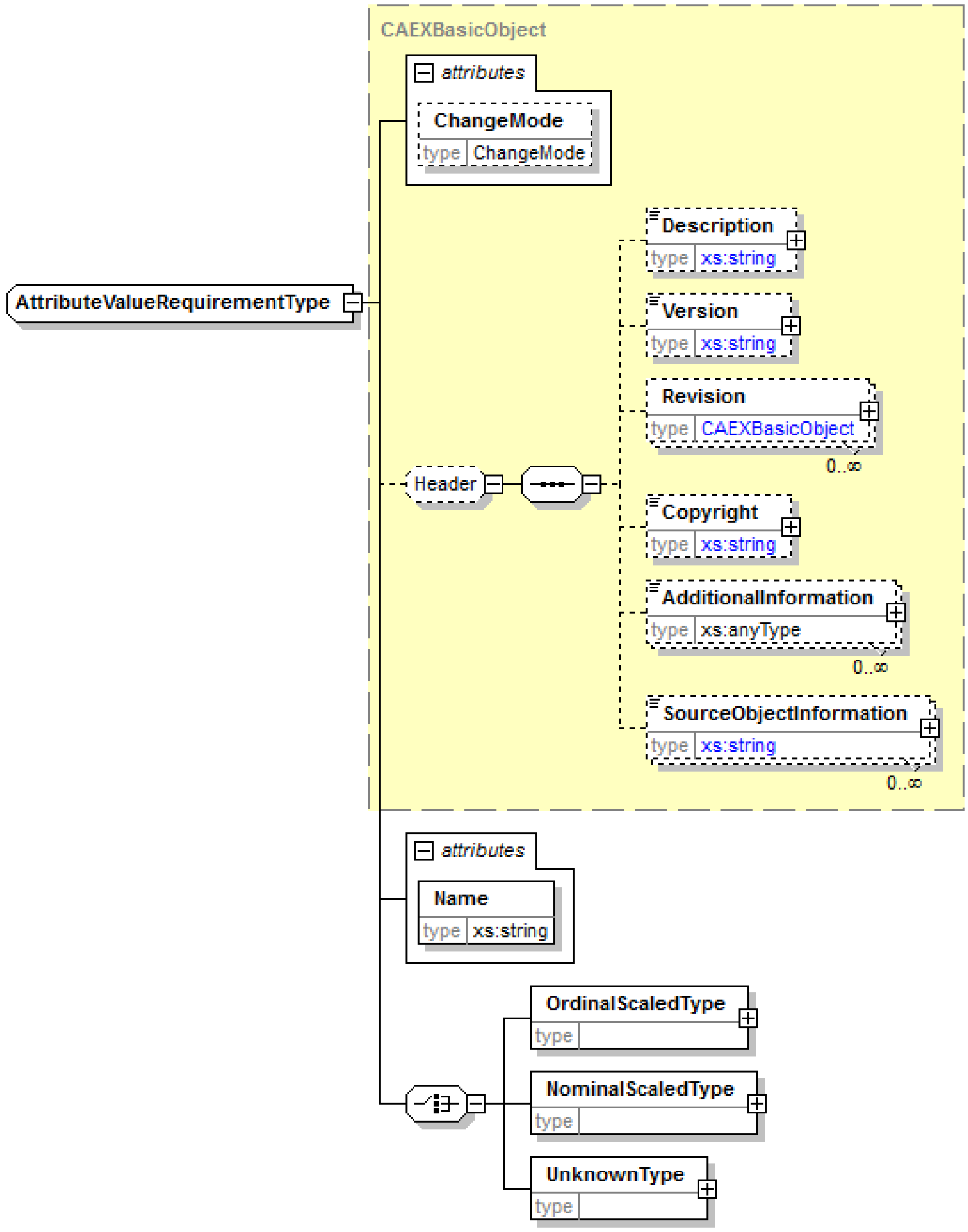
A.3.18.6 Element AttributeType/Attribute

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

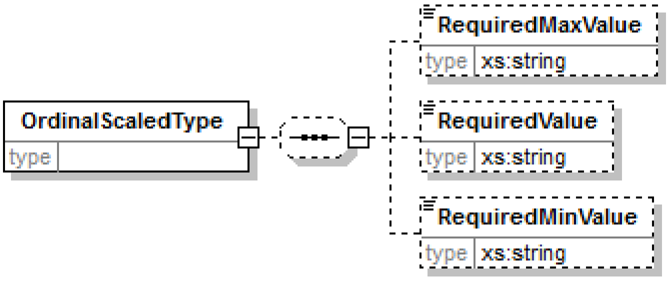
AttributeType
attributes
ChangeMode
type ChangeMode
ID
type xs:string
Name
type xs:string
Unit
type xs:string
AttributeDataType
type xs:string
RefAttributeType
type xs:string
Attribute
type AttributeType
0..∞
Header
Description
type xs:string
Version
type xs:string
Revision
type CAEXBasicObject
0..∞
Copyright
type xs:string
AdditionalInformation
type xs:anyType
0..∞
SourceObjectInformation
type xs:string
0..∞
DefaultValue
type xs:string
Value
type xs:string
RefSemantic
type CAEXBasicObject
0..∞
Constraint
type AttributeValueRequirementType
0..∞
Attribute
type AttributeType
0..∞

</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>AttributeType</td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td><b>derived by: xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

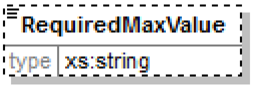
A.3.18.7 complexType AttributeValueRequirementType

diagram	 CAEXBasicObject attributes ChangeMode type ChangeMode Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ AttributeValueRequirementType Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ attributes Name type xs:string OrdinalScaledType type NominalScaledType type UnknownType type
namespace	http://www.dke.de/CAEX
type	extension of CAEXBasicObject
properties	base CAEXBasicObject
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation OrdinalScaledType NominalScaledType UnknownType
used by	element **AttributeType/Constraint**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state Name **xs:string** required

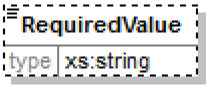
A.3.18.8 Element AttributeValueRequirementType/OrdinalScaledType

diagram	 RequiredMaxValue type xs:string OrdinalScaledType type RequiredValue type xs:string RequiredMinValue type xs:string
namespace	http://www.dke.de/CAEX
properties	isRef 0 content complex
children	RequiredMaxValue RequiredValue RequiredMinValue

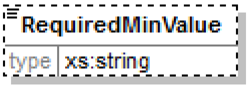
A.3.18.9 Element AttributeValueRequirementType/OrdinalScaledType/RequiredMaxValue

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

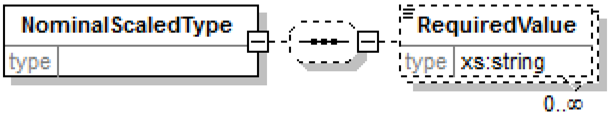
A.3.18.10 Element AttributeValueRequirementType/OrdinalScaledType/RequiredValue

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

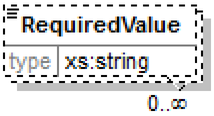
A.3.18.11 Element AttributeValueRequirementType/OrdinalScaledType/RequiredMinValue

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc 1 content simple

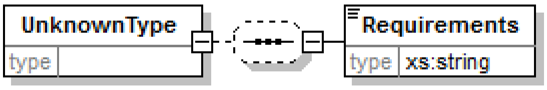
A.3.18.12 Element AttributeValueRequirementType/NominalScaledType

diagram	 NominalScaledType type RequiredValue type xs:string 0..∞
namespace	http://www.dke.de/CAEX
properties	isRef 0 content complex
children	RequiredValue

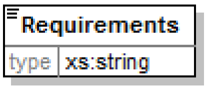
A.3.18.13 Element AttributeValueRequirementType/NominalScaledType/RequiredValue

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 minOcc 0 maxOcc unbounded content simple

A.3.18.14 Element AttributeValueRequirementType/UnknownType

diagram	
namespace	http://www.dke.de/CAEX
properties	isRef 0 content complex
children	Requirements

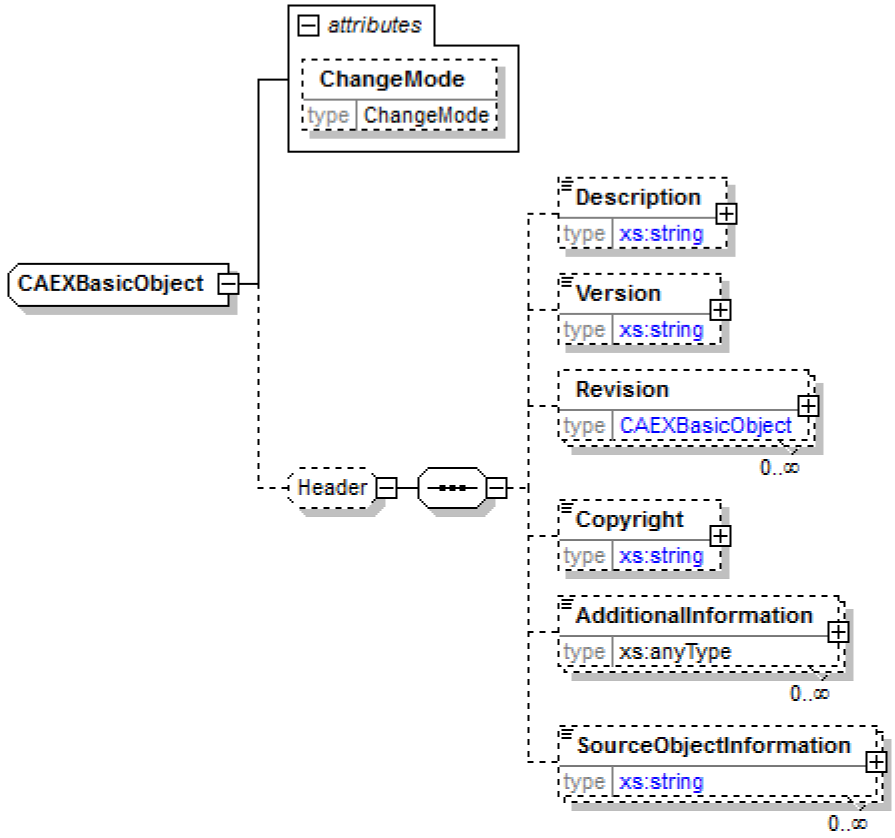
A.3.18.15 Element AttributeValueRequirementType/UnknownType/Requirements

diagram	
namespace	http://www.dke.de/CAEX
type	**xs:string**
properties	isRef 0 content simple

A.3.19 CAEX complex type CAEXBasicObject

The CAEX element CAEXBasicObject is the basis object for all CAEX elements. See A.2.2 and A.2.2.7 for details.

complexType **CAEXBasicObject**

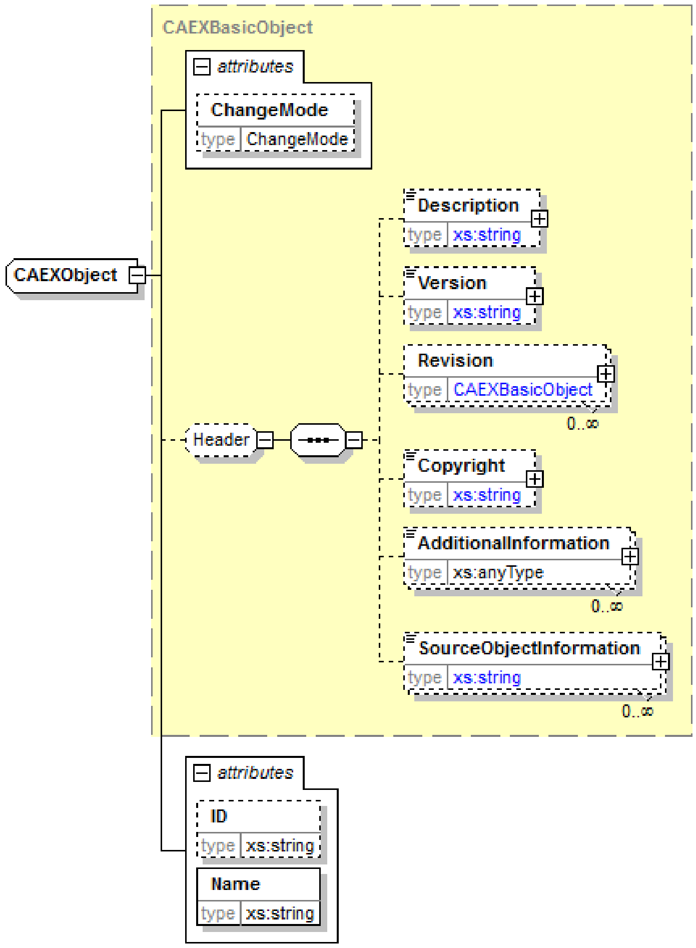
diagram	
namespace	http://www.dke.de/CAEX
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation**
used by	elements **MappingType/AttributeNameMapping** **CAEXFile** **CAEXFile/ExternalReference** **MappingType/InterfaceIDMapping** **AttributeType/RefSemantic** **Header/Revision** **InternalElementType/RoleRequirements** **SystemUnitClassType/SupportedRoleClass** complexTypes **AttributeValueRequirementType** **CAEXObject** **MappingType**

	Name	Type	Use	Default	Fixed	Annotation
attributes	ChangeMode	**ChangeMode**	optional	state		

A.3.20 CAEX complex type CAEXObject

The CAEX complex type CAEXObject is derived from the CAEXBasicObject and additionally defines the attributes "name" and "ID". This complex type is the base class for CAEX objects like classes, instances, attribute, interfaces, etc., which have a name.

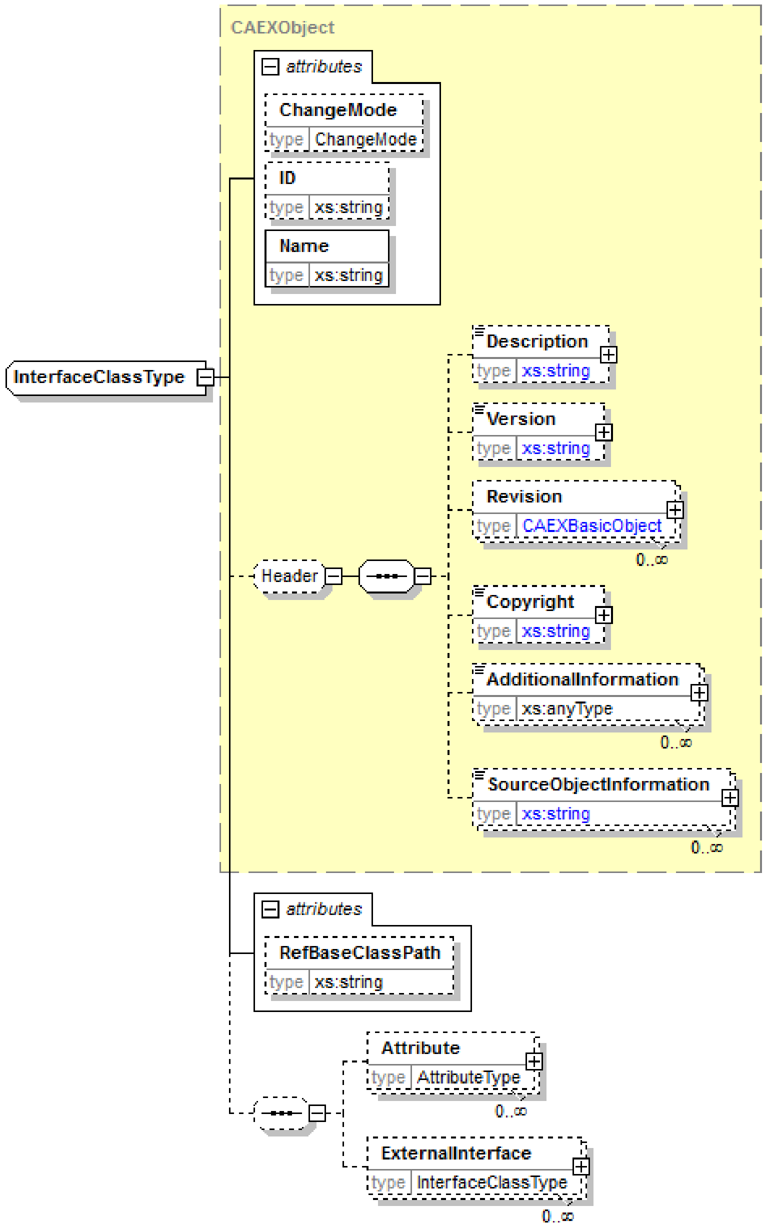
- The attribute "name" allows for the storage of a name string of individual objects, classes or types. Normative provisions regarding the usage of the name are defined in A.2.2.6.
- The attribute "ID" allows for the storage of unique identifiers of the individual objects. Normative provisions regarding the usage of this attribute are defined in A.2.2.6. The ID attribute helps corresponding exporter/importer tools to identify objects, e.g. if they have changed their name or their position within the system hierarchy.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXBasicObject</b></td></tr>
<tr><td>properties</td><td>base CAEXBasicObject</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation</b></td></tr>
<tr><td>used by</td><td><table>
<tr><td>elements</td><td><b>CAEXFile/AttributeTypeLib CAEXFile/InstanceHierarchy CAEXFile/InterfaceClassLib SystemUnitClassType/InternalLink CAEXFile/RoleClassLib CAEXFile/SystemUnitClassLib</b></td></tr>
<tr><td>complexTypes</td><td><b>AttributeType InterfaceClassType RoleClassType SystemUnitClassType</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.21 CAEX complex type InterfaceClassType

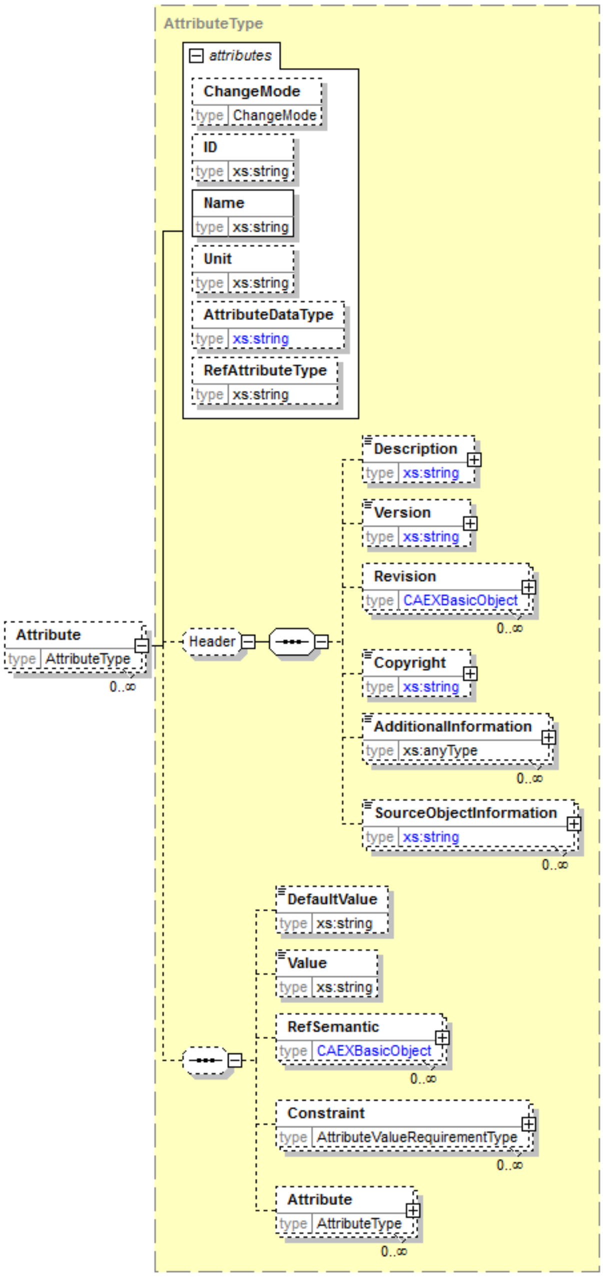
A.3.21.1 General

The CAEX element “InterfaceClassType” is the base type for InterfaceClass definitions. See A.2.6, A.3.8 and A.3.9 for details.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of CAEXObject</td></tr>
<tr><td>properties</td><td>base CAEXObject</td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface</td></tr>
<tr><td>used by</td><td><table>
<tr><td>elements</td><td><b>InterfaceClassType/ExternalInterface</b>
<b>RoleClassType/ExternalInterface</b>
<b>SystemUnitClassType/ExternalInterface</b>
<b>InternalElementType/RoleRequirements/ExternalInterface</b></td></tr>
<tr><td>complexType</td><td><b>InterfaceFamilyType</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseClass Path</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

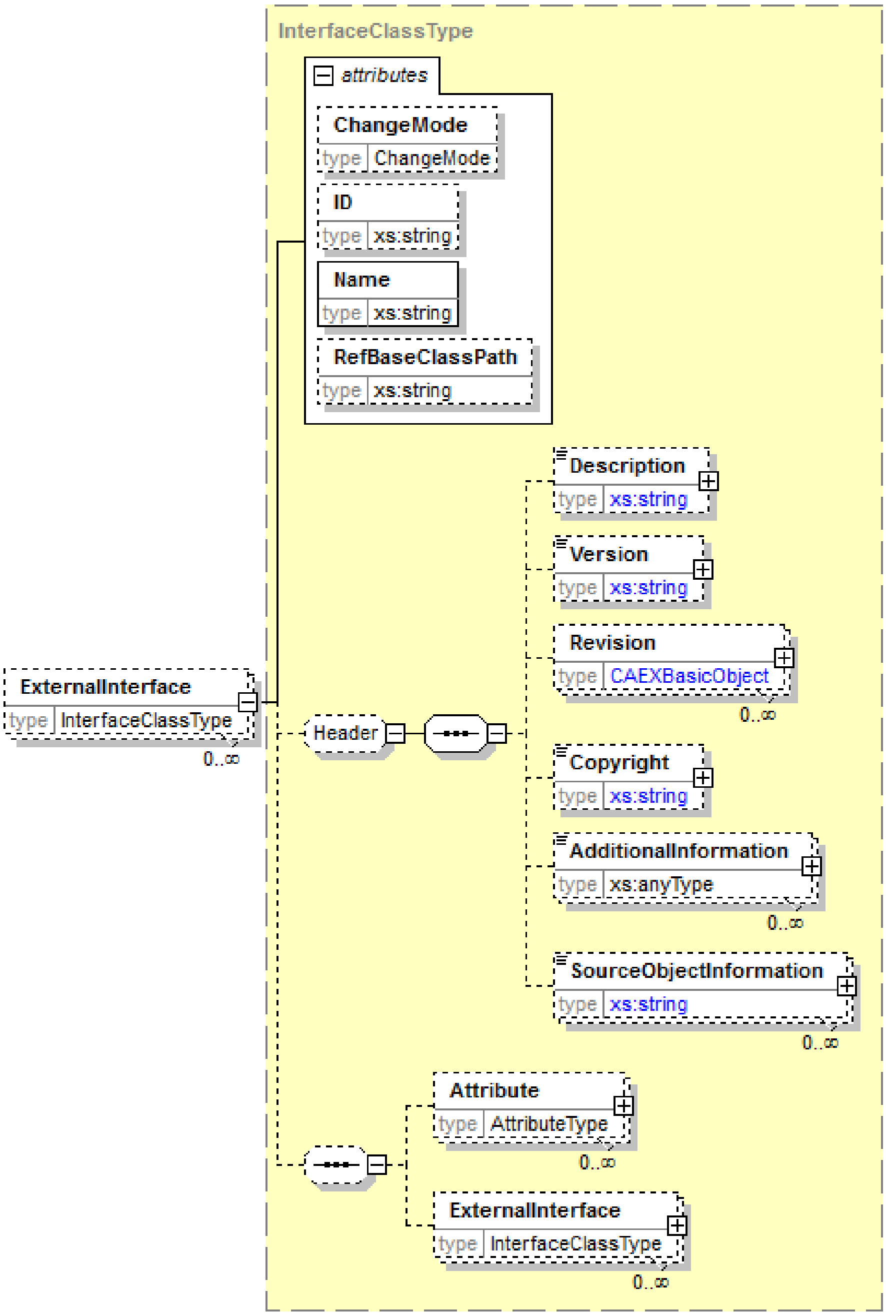
A.3.21.2 Element InterfaceClassType/Attribute

This element serves for the modeling of interface class attributes.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>AttributeType</td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td><b>derived by: xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.21.3 Element InterfaceClassType/ExternalInterface

This element serves for the modeling of nested interfaces.

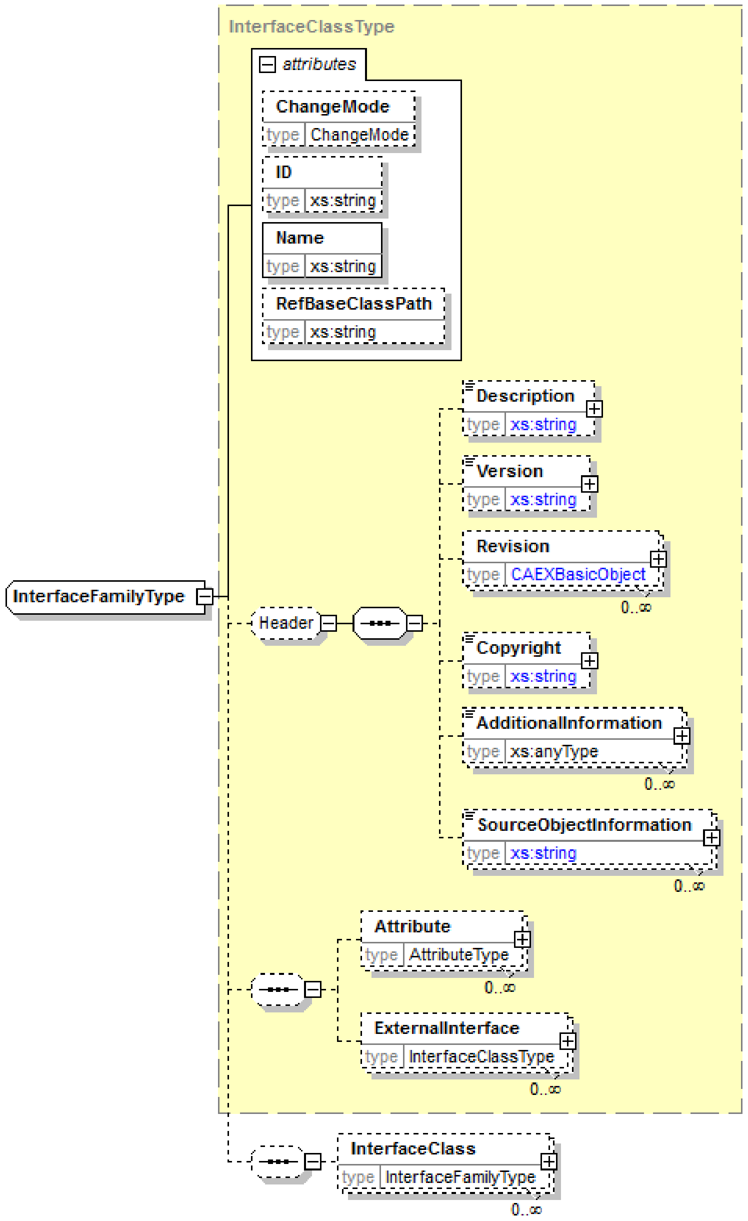
diagram	
namespace	http://www.dke.de/CAEX
type	**InterfaceClassType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface**

attributes	Name	Type	Use	Default	Fixed	Annotation
	ChangeMode	**ChangeMode**	optional	state		
	ID	**xs:string**	optional			
	Name	**xs:string**	required			
	RefBaseClass Path	**xs:string**	optional			

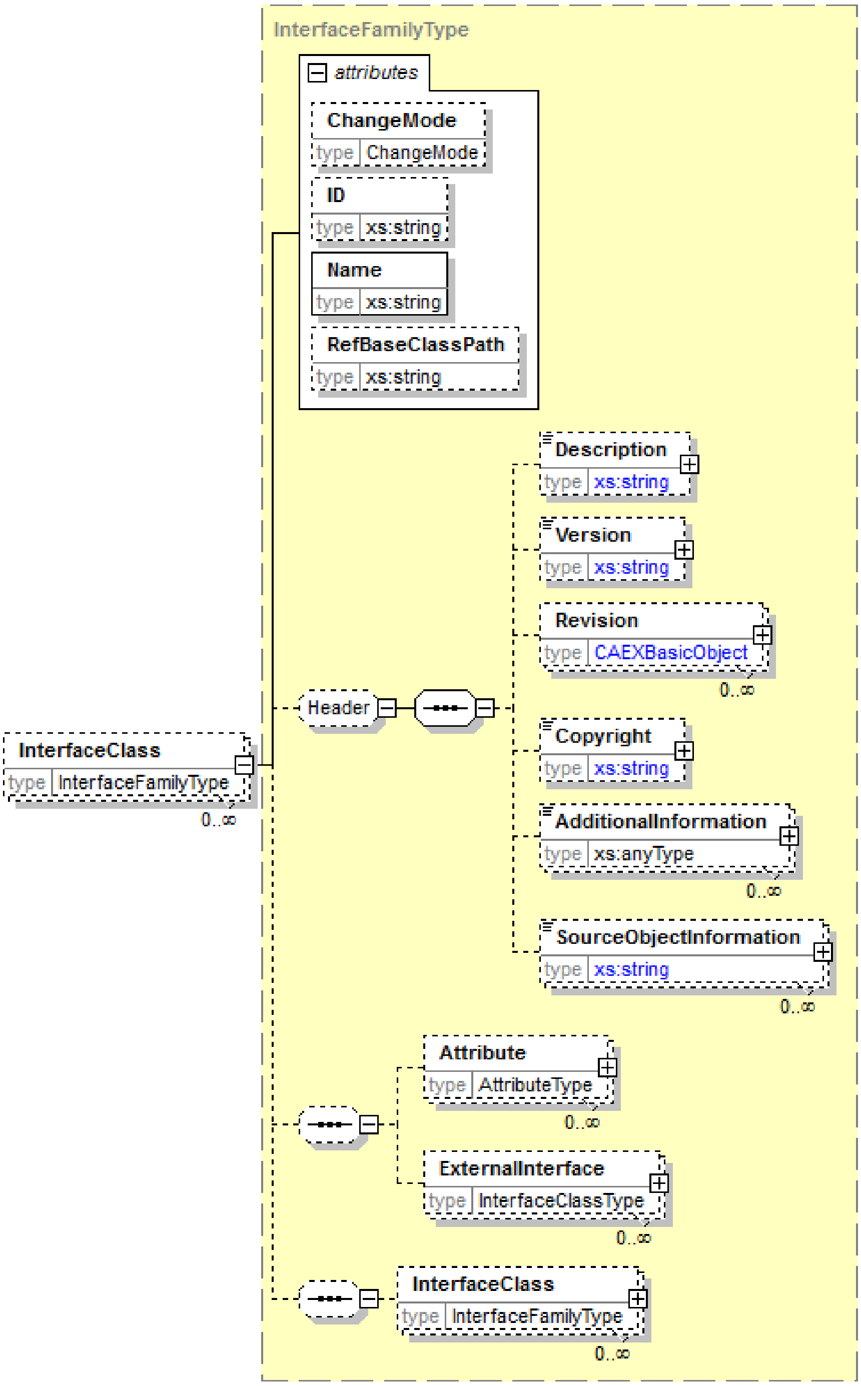
A.3.22 CAEX complex type InterfaceFamilyType

A.3.22.1 General

The CAEX element “InterfaceFamilyType” is an extension of the InterfaceClassType and additionally supports adding InterfaceClasses as children. This child is again of the type InterfaceFamilyType – this recursive definition allows for the storage of an arbitrary interface hierarchy tree. The parent-child relation between InterfaceClasses has no further semantics. See A.2.6 for details and examples.

diagram	 InterfaceClassType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string InterfaceFamilyType Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InterfaceClass type InterfaceFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **InterfaceClassType**
properties	base InterfaceClassType
children	**Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InterfaceClass**
used by	elements **CAEXFile/InterfaceClassLib/InterfaceClass** **InterfaceFamilyType/InterfaceClass**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state ID **xs:string** optional Name **xs:string** required RefBaseClassPath **xs:string** optional

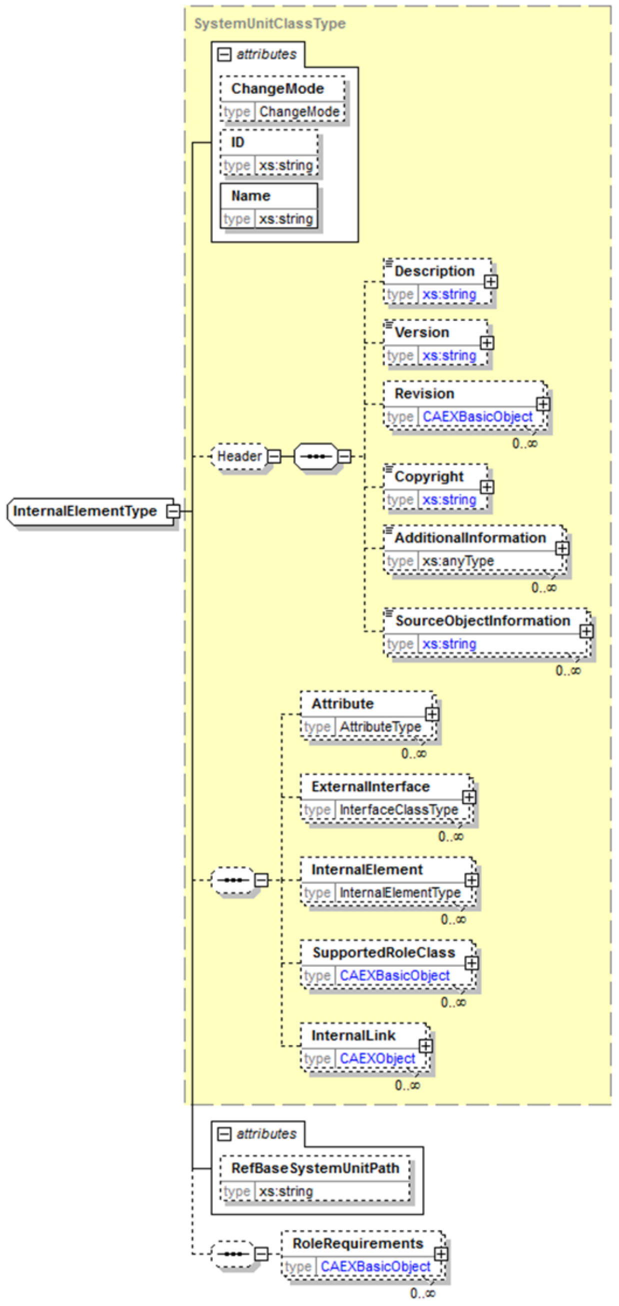
A.3.22.2 Element InterfaceFamilyType/InterfaceClass

diagram	 InterfaceFamilyType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ InterfaceClass type InterfaceFamilyType 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InterfaceClass type InterfaceFamilyType 0..∞
namespace	http://www.dke.de/CAEX
type	InterfaceFamilyType
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InterfaceClass
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / ChangeMode / optional / state ID / xs:string / optional Name / xs:string / required RefBaseClassPath / xs:string / optional

A.3.23 CAEX complex type InternalElementType

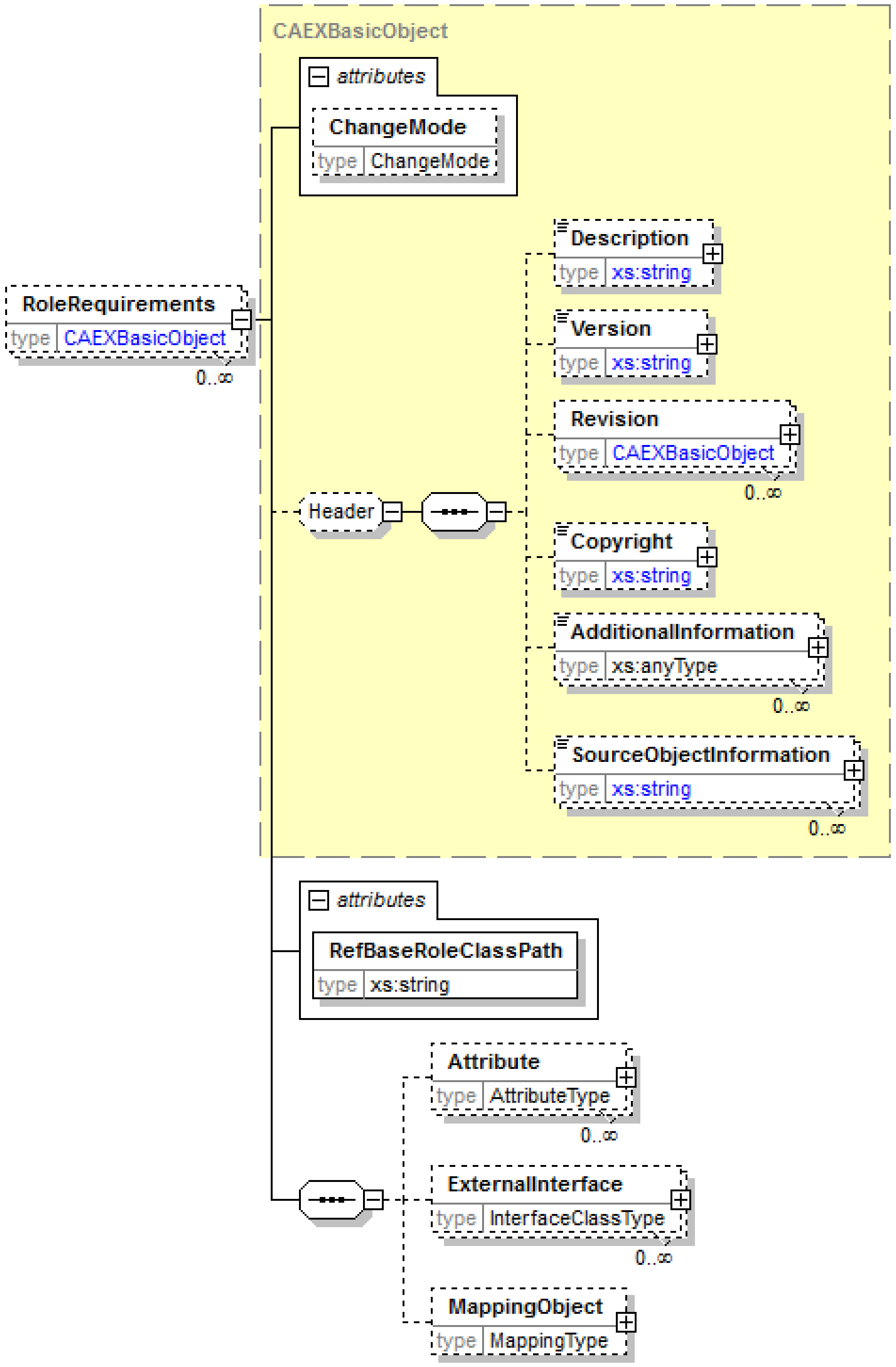
A.3.23.1 General

The CAEX element “InternalElementType” is the base type of the CAEX element “InternalElement”. See A.2.2.1, A.2.8.2 and A.3.7 for details.

diagram	 SystemUnitClassType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ InternalElementType Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InternalElement type InternalElementType 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ InternalLink type CAEXObject 0..∞ attributes RefBaseSystemUnitPath type xs:string RoleRequirements type CAEXBasicObject 0..∞					
namespace	http://www.dke.de/CAEX					
type	extension of **SystemUnitClassType**					
properties	base SystemUnitClassType					
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface** **InternalElement** **SupportedRoleClass** **InternalLink** **RoleRequirements**					
used by	elements **CAEXFile/InstanceHierarchy/InternalElement** **SystemUnitClassType/InternalElement**					
attributes	Name	Type	Use	Default	Fixed	Annotation
	ChangeMode	**ChangeMode**	optional	state		
	ID	**xs:string**	optional			
	Name	**xs:string**	required			
	RefBaseSystemUnitPath	**xs:string**	optional			

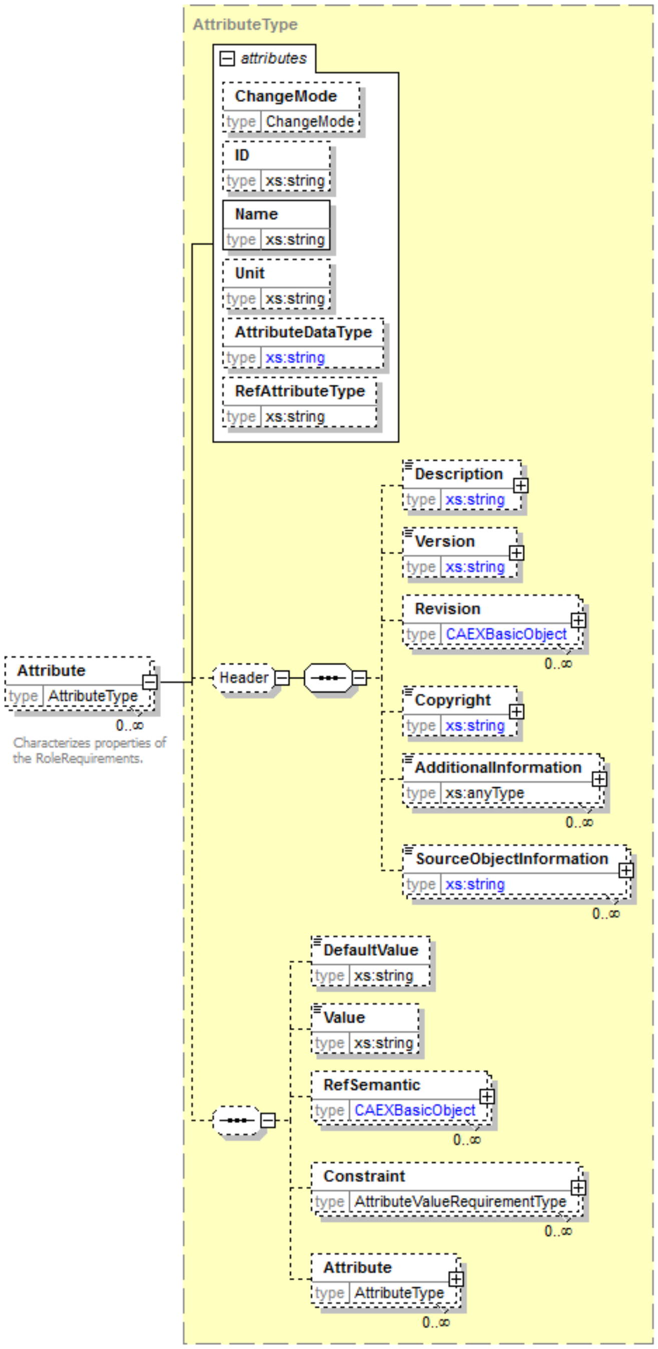
A.3.23.2 Element InternalElementType/RoleRequirements

The CAEX element “RoleRequirements” allows for the definition of a reference to a RoleClass as well as the definition of requirements of the corresponding object. See A.2.10 for details and examples.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of CAEXBasicObject</td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</td></tr></table></td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface MappingObject</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr><tr><td>RefBaseRoleClassPath</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr></table></td></tr>
</table>

A.3.23.3 Element InternalElementType/RoleRequirements/Attribute

diagram

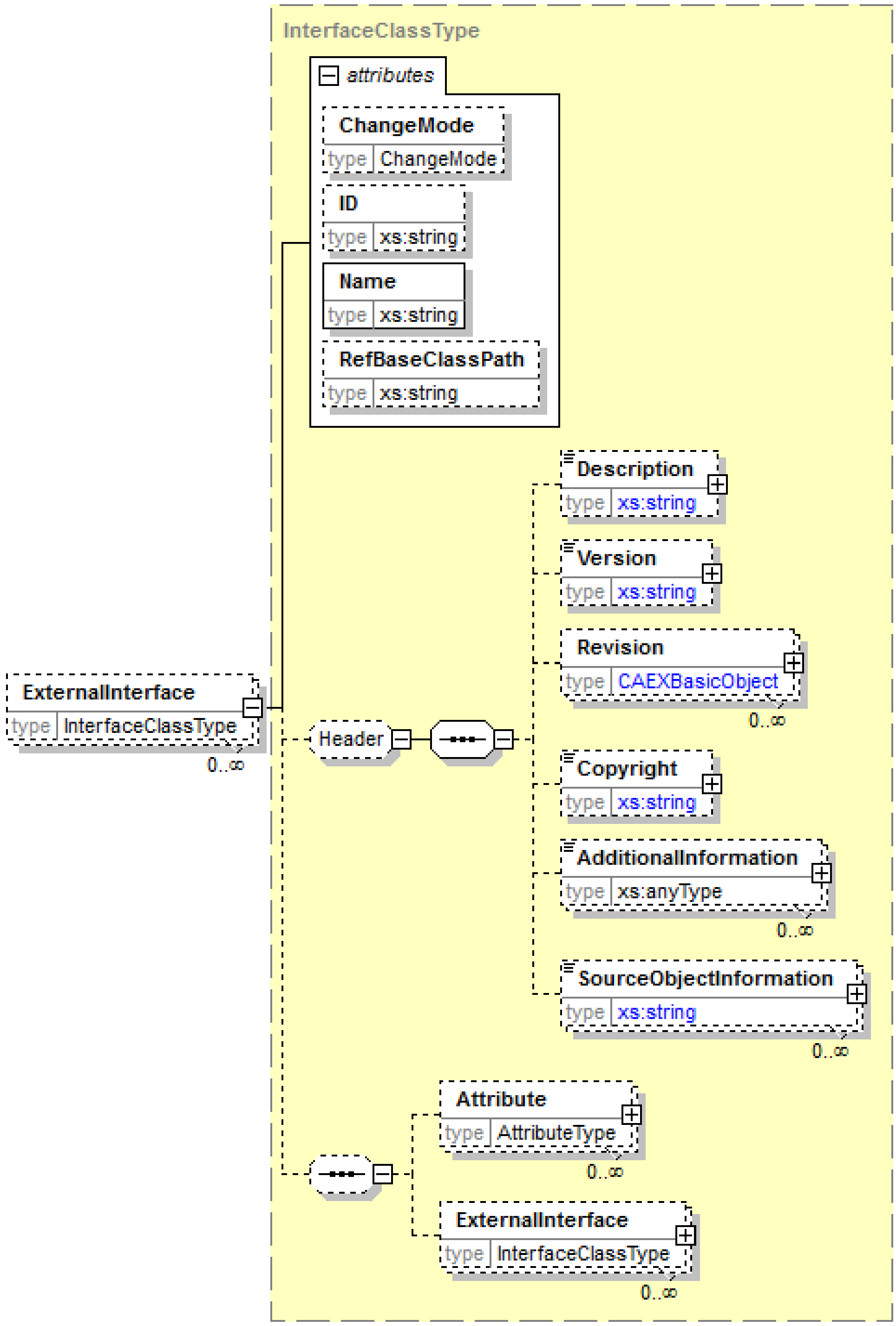


namespace	http://www.dke.de/CAEX
type	**AttributeType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **DefaultValue** **Value** **RefSemantic** **Constraint** **Attribute**

attributes

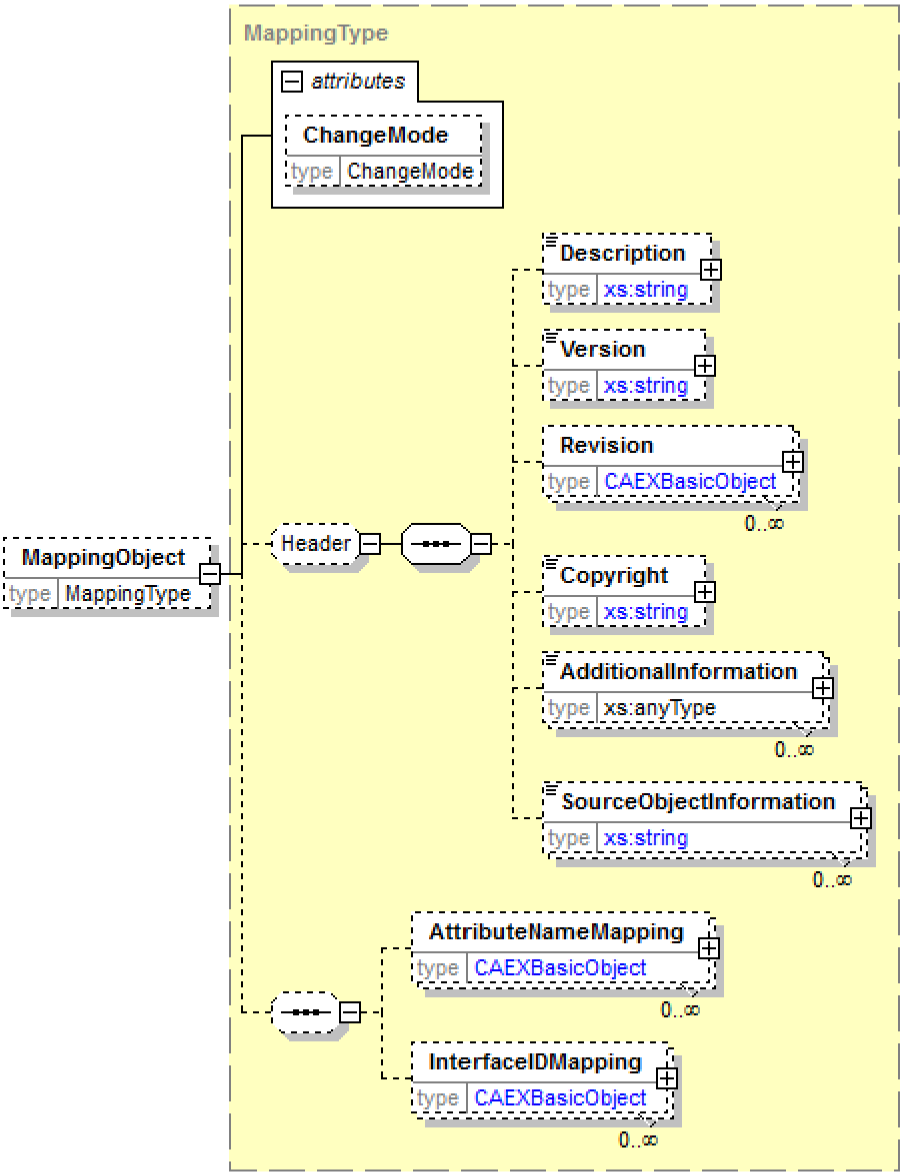
Name	Type	Use	Default	Fixed	Annotation
ChangeMode	**ChangeMode**	optional	state		
ID	**xs:string**	optional			
Name	**xs:string**	required			
Unit	**xs:string**	optional			
AttributeDataType	**derived by: xs:string**	optional			
RefAttributeType	**xs:string**	optional			

A.3.23.4 Element InternalElementType/RoleRequirements/ExternalInterface

diagram	 InterfaceClassType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string ExternalInterface type InterfaceClassType 0..∞ Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞
namespace	http://www.dke.de/CAEX
type	**InterfaceClassType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface**
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / **ChangeMode** / optional / state ID / **xs:string** / optional Name / **xs:string** / required RefBaseClassPath / **xs:string** / optional

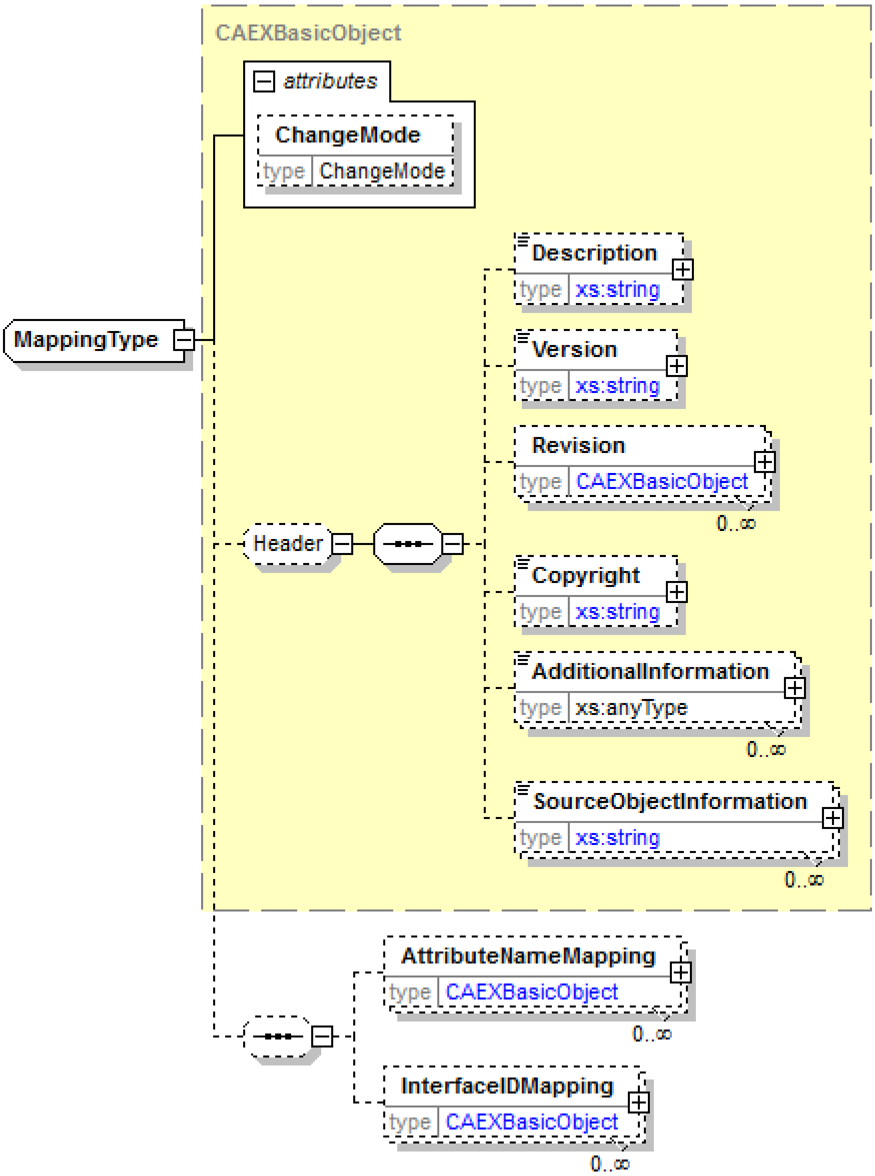
A.3.23.5 Element InternalElementType/MappingObject

See A.2.10 and A.2.11 for details and examples.

diagram	 MappingType attributes ChangeMode type ChangeMode MappingObject type MappingType Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ AttributeNameMapping type CAEXBasicObject 0..∞ InterfaceIDMapping type CAEXBasicObject 0..∞
namespace	http://www.dke.de/CAEX
type	**MappingType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **AttributeNameMapping** **InterfaceIDMapping**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

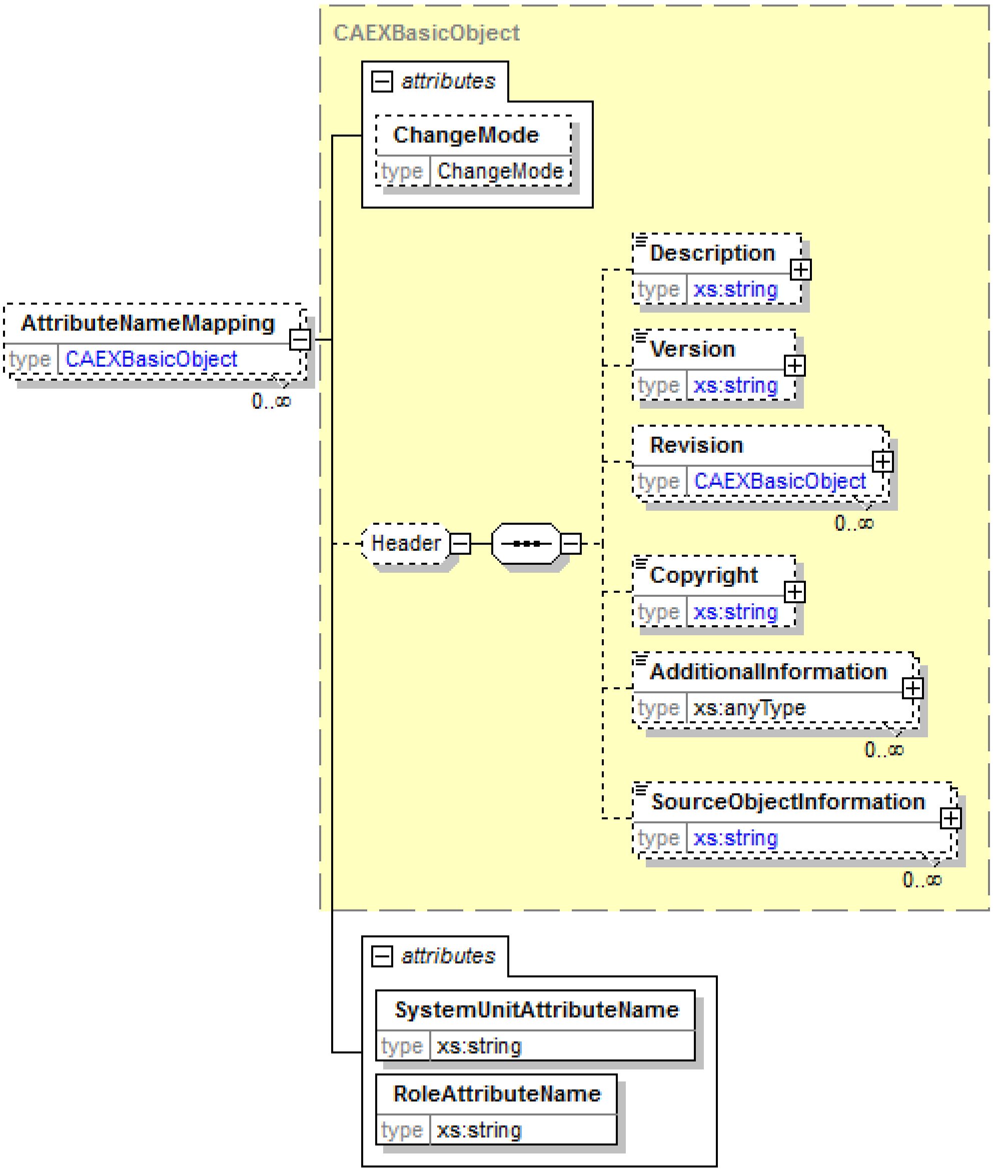
A.3.23.6 Complex type MappingType

This type is the base type for the CAEX MappingObject. See A.2.10 and A.2.11 for details and examples.

diagram	 CAEXBasicObject attributes ChangeMode type ChangeMode MappingType Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ AttributeNameMapping type CAEXBasicObject 0..∞ InterfaceIDMapping type CAEXBasicObject 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **CAEXBasicObject**
properties	base CAEXBasicObject
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **AttributeNameMapping** **InterfaceIDMapping**
used by	elements **SystemUnitClassType/SupportedRoleClass/MappingObject** **InternalElementType/MappingObject**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state

A.3.23.7 Element MappingType/AttributeNameMapping

diagram



namespace http://www.dke.de/CAEX

type extension of **CAEXBasicObject**

properties

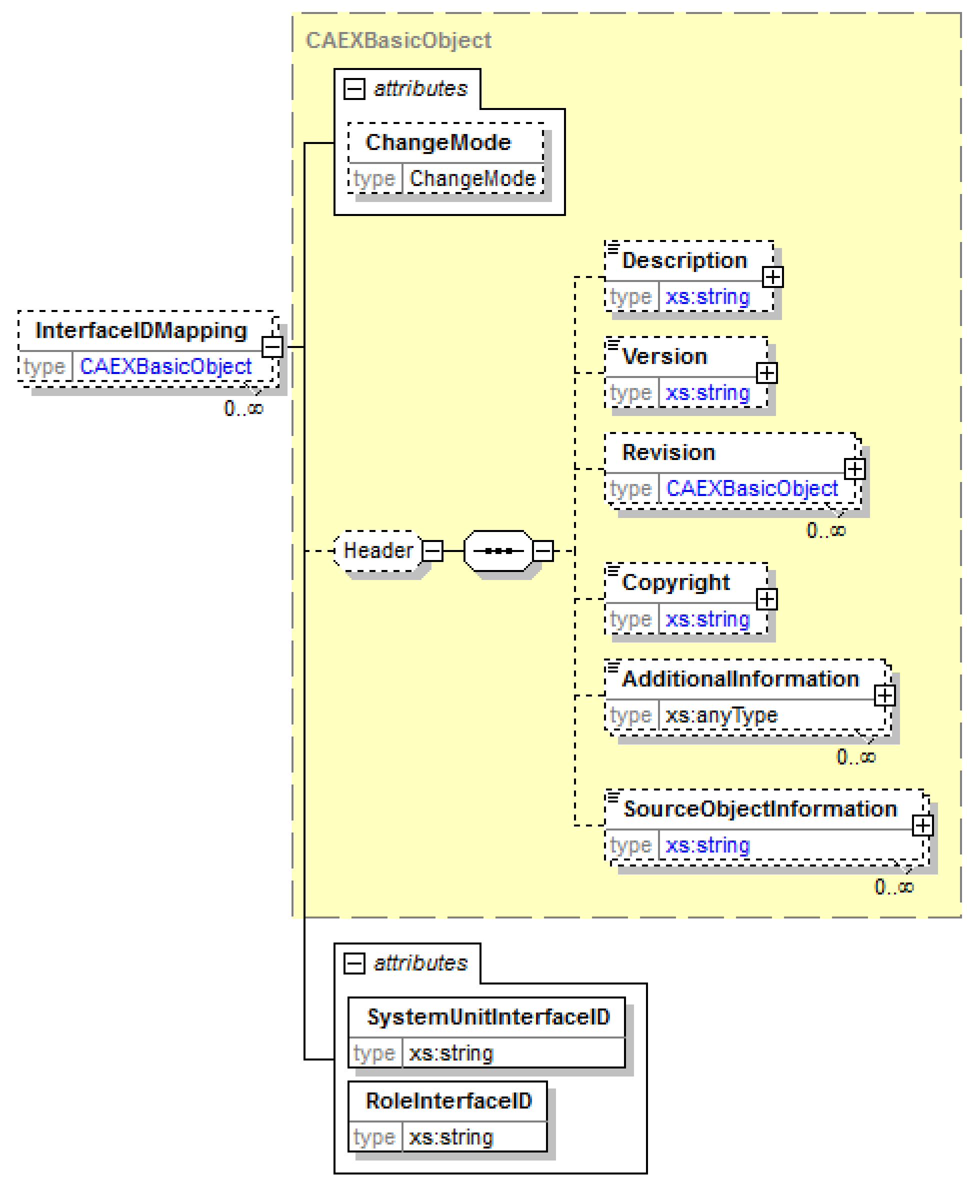
isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children **Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation**

attributes

Name	Type	Use	Default	Fixed	Annotation
ChangeMode	**ChangeMode**	optional	state		
SystemUnitAttributeName	**xs:string**	required			
RoleAttributeName	**xs:string**	required			

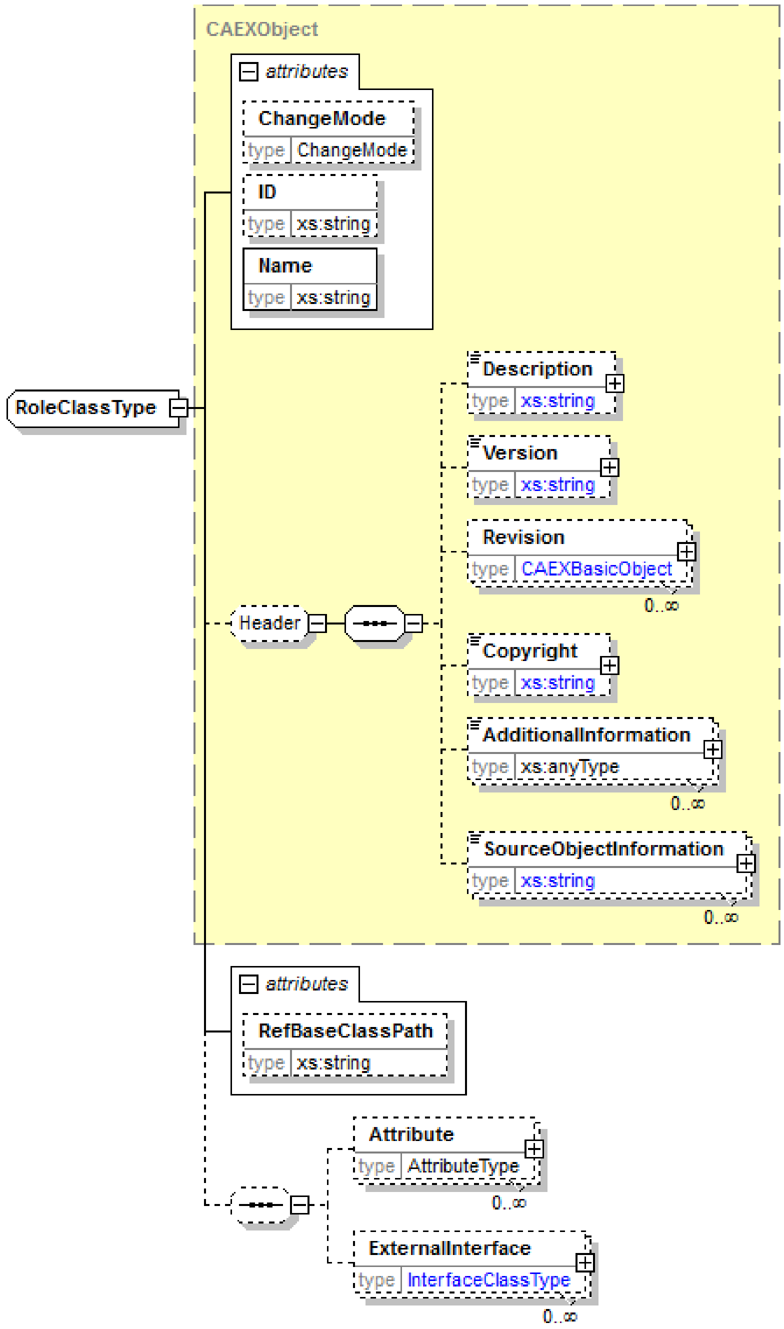
A.3.23.8 Element MappingType/InterfaceIDMapping

diagram	
namespace	http://www.dke.de/CAEX
type	extension of **CAEXBasicObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation**

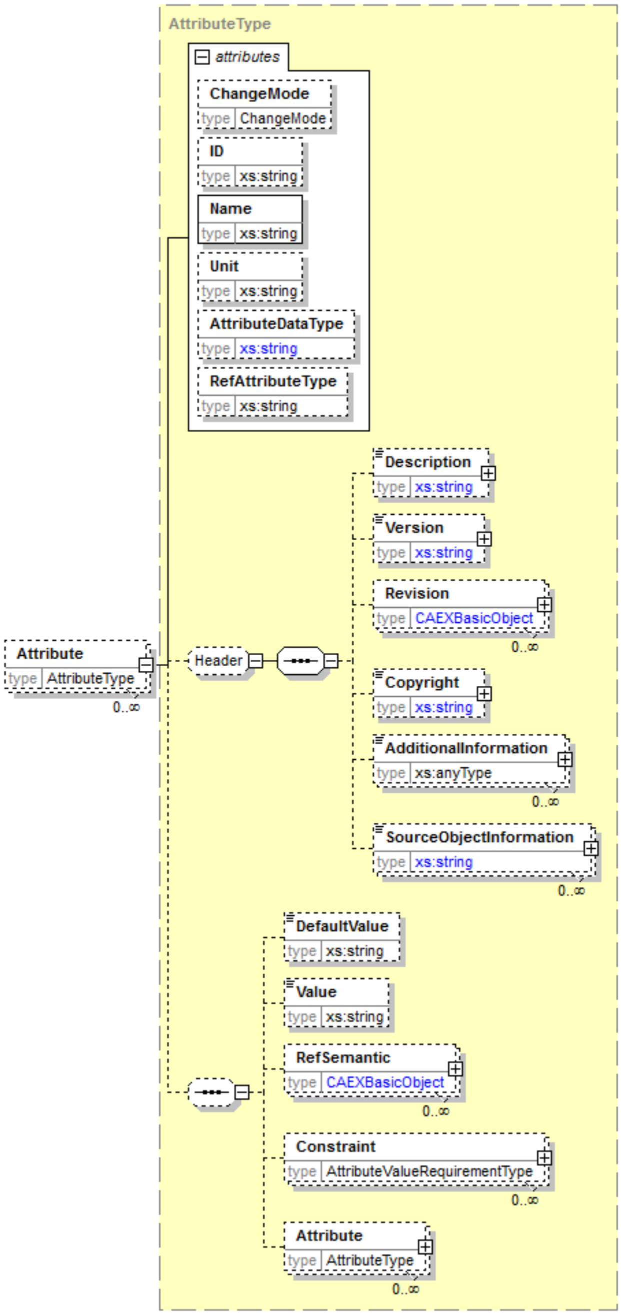
attributes	Name	Type	Use	Default	Fixed	Annotation
	ChangeMode	**ChangeMode**	optional	state		
	SystemUnitInterfaceID	**xs:string**	required			
	RoleInterfaceID	**xs:string**	required			

A.3.24 CAEX complex type RoleClassType

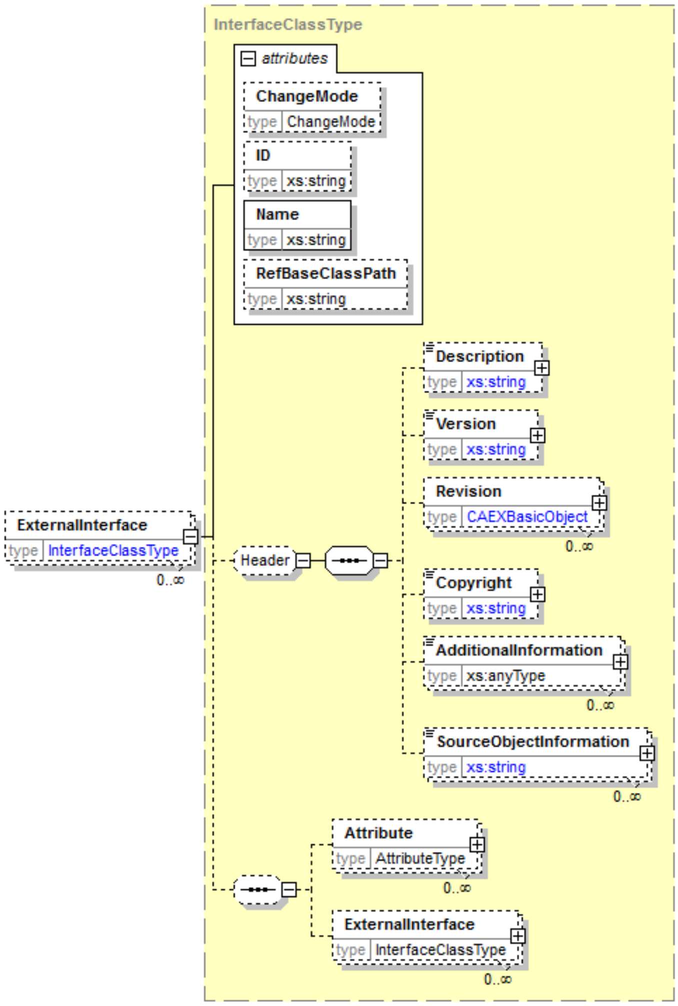
The CAEX element “RoleClassType” is the base type of the CAEX element RoleClass. See A.2.7 and A.3.11 for details.

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>CAEXObject</b></td></tr>
<tr><td>properties</td><td>base CAEXObject</td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</b> <b>Attribute</b> <b>ExternalInterface</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseClassPath</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.24.1 Element RoleClassType/Attribute

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>AttributeType</td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td>Description Version Revision Copyright AdditionalInformation SourceObjectInformation DefaultValue Value RefSemantic Constraint Attribute</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>ChangeMode</td><td>ChangeMode</td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>Unit</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>AttributeDataType</td><td>derived by: xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>RefAttributeType</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.24.2 Element RoleClassType/ExternalInterface

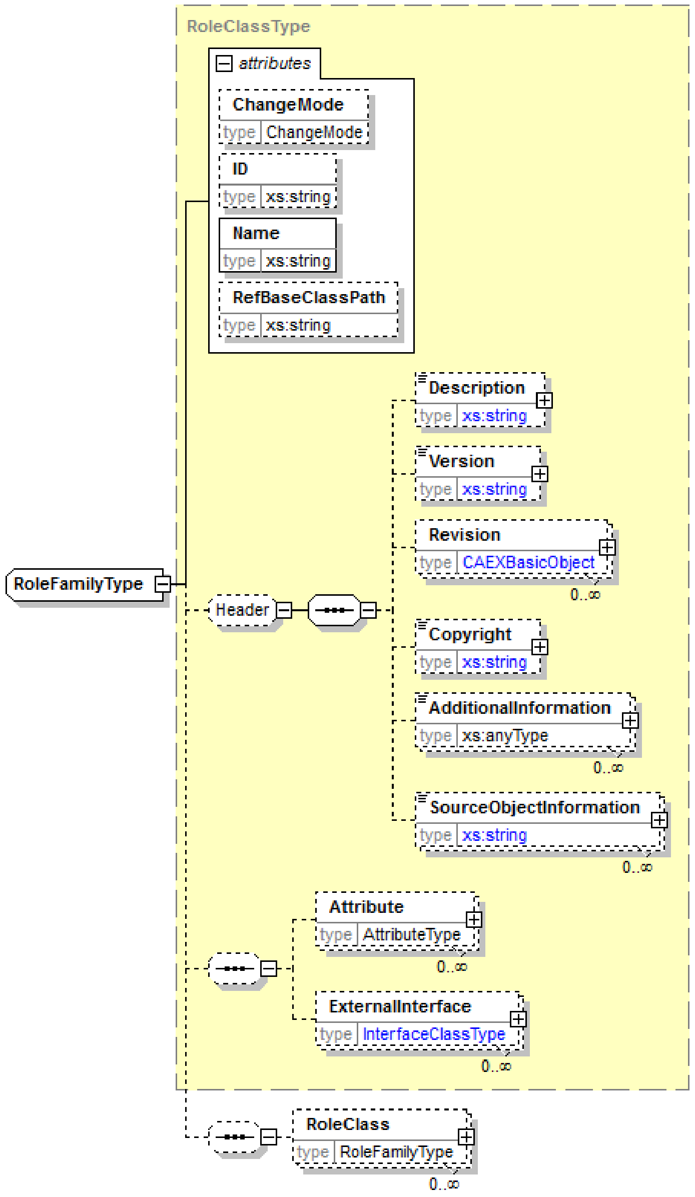
<table>
<tr><td>diagram</td><td></td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>InterfaceClassType</b></td></tr>
<tr><td>properties</td><td><table><tr><td>isRef</td><td>0</td></tr><tr><td>minOcc</td><td>0</td></tr><tr><td>maxOcc</td><td>unbounded</td></tr><tr><td>content</td><td>complex</td></tr></table></td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</b> <b>Attribute</b> <b>ExternalInterface</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr><tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr><tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr><tr><td>RefBaseClassPath</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr></table></td></tr>
</table>

A.3.25 CAEX complex type RoleFamilyType

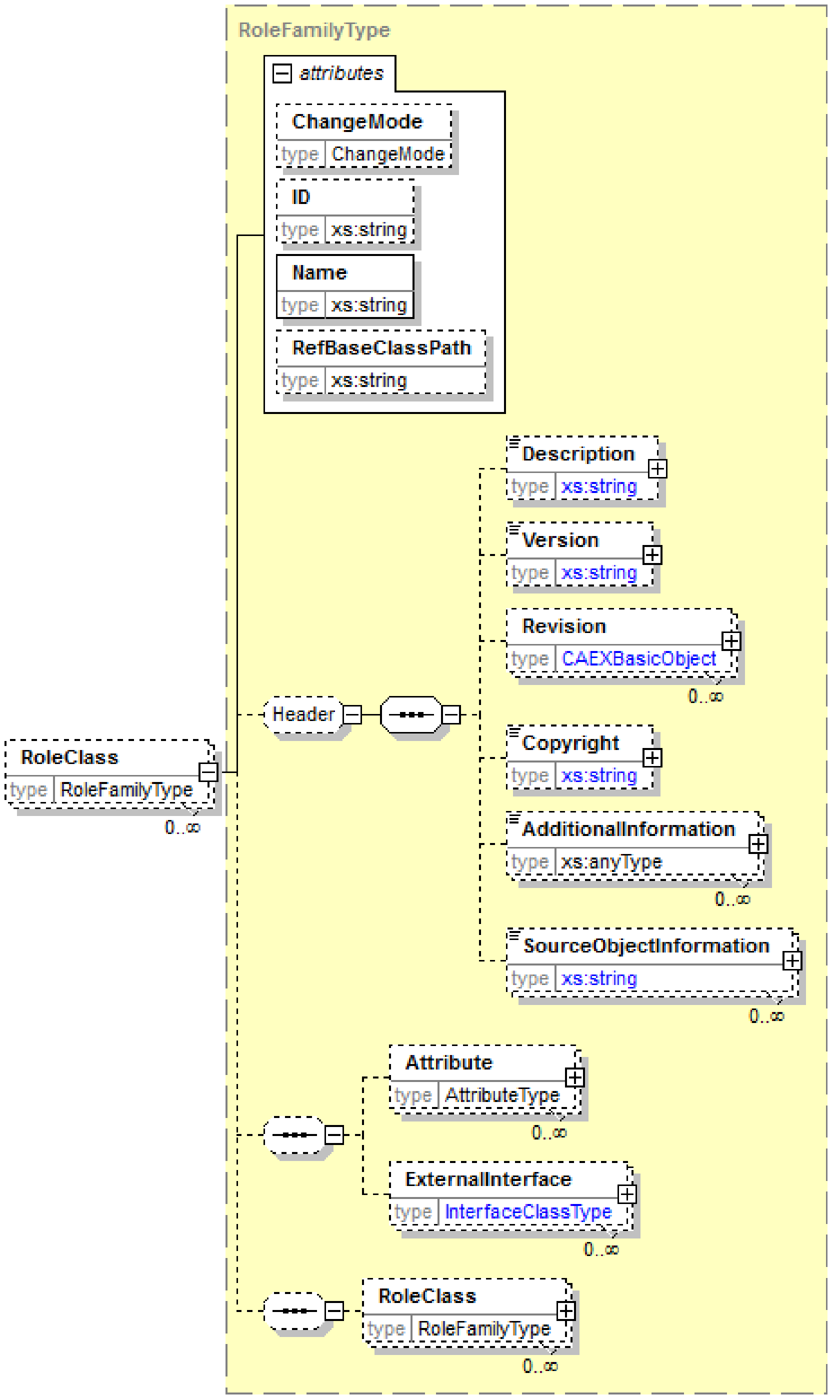
A.3.25.1 General

The CAEX element “RoleFamilyType” is an extension of the RoleClassType and additionally supports adding RoleClasses as children. This child is again of the type RoleFamilyType – this recursive definition allows for the storage of an arbitrary role hierarchy tree. See A.2.7 for details and examples.

A.3.25.2 Complex type RoleFamilyType

<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td>extension of <b>RoleClassType</b></td></tr>
<tr><td>properties</td><td>base RoleClassType</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface RoleClass</b></td></tr>
<tr><td>used by</td><td>elements <b>CAEXFile/RoleClassLib/RoleClass</b>
<b>RoleFamilyType/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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseClassPath</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

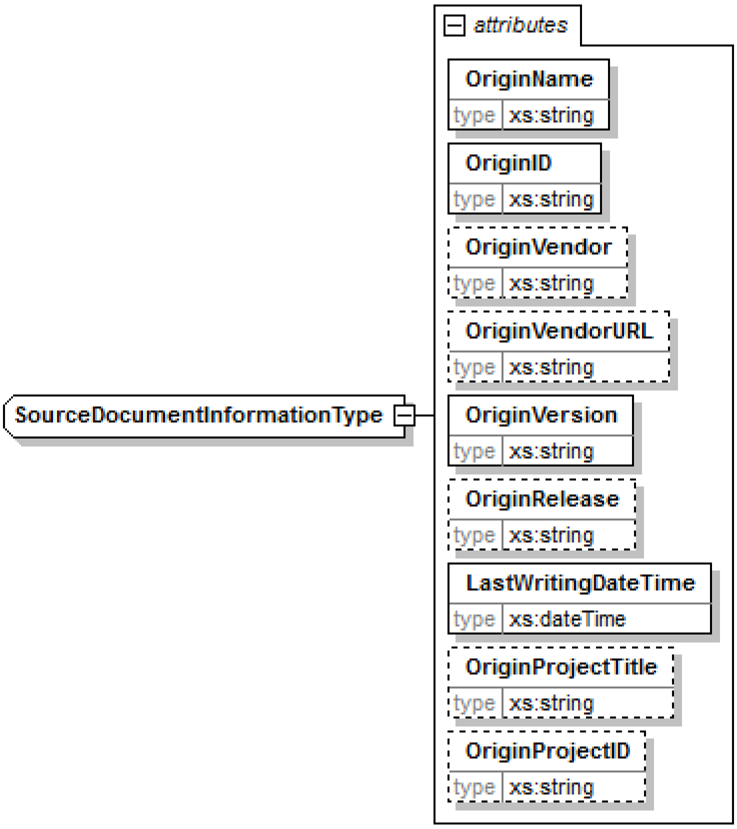
A.3.25.3 Element RoleFamilyType/RoleClass

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


</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><b>RoleFamilyType</b></td></tr>
<tr><td>properties</td><td>isRef 0
minOcc 0
maxOcc unbounded
content complex</td></tr>
<tr><td>children</td><td><b>Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface 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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseClassPath</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

A.3.26 CAEX complexType SourceDocumentInformationType

The CAEX element “SourceDocumentInformationType” defines a structure to model information about the data source of the present CAEX document. See A.2.2.5 and A.3.4 for details.

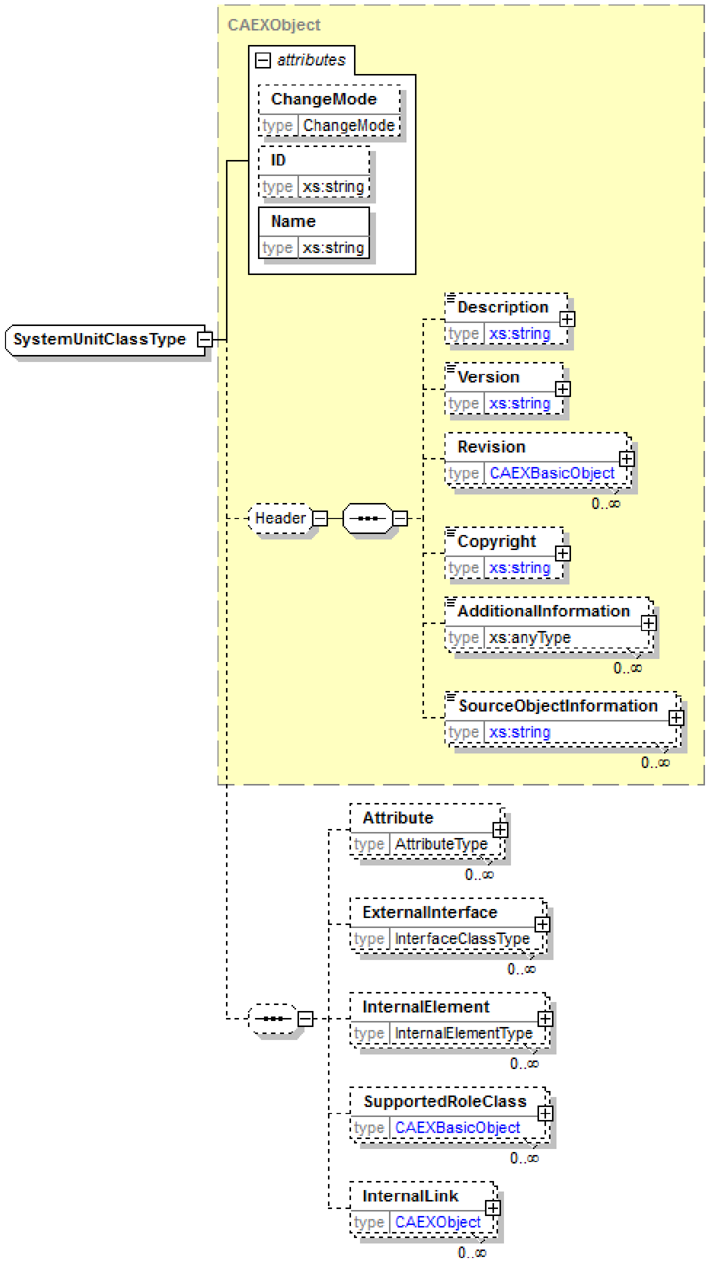
diagram	 attributes SourceDocumentInformationType OriginName type xs:string OriginID type xs:string OriginVendor type xs:string OriginVendorURL type xs:string OriginVersion type xs:string OriginRelease type xs:string LastWritingDateTime type xs:dateTime OriginProjectTitle type xs:string OriginProjectID type xs:string
namespace	http://www.dke.de/CAEX
used by	element **CAEXFile/SourceDocumentInformation**

	Name	Type	Use	Default	Fixed	Annotation
attributes	OriginName	**xs:string**	required			
	OriginID	**xs:string**	required			
	OriginVendor	**xs:string**	optional			
	OriginVendorURL	**xs:string**	optional			
	OriginVersion	**xs:string**	required			
	OriginRelease	**xs:string**	optional			
	LastWritingDateTime	**xs:dateTime**	required			
	OriginProjectTitle	**xs:string**	optional			
	OriginProjectID	**xs:string**	optional			

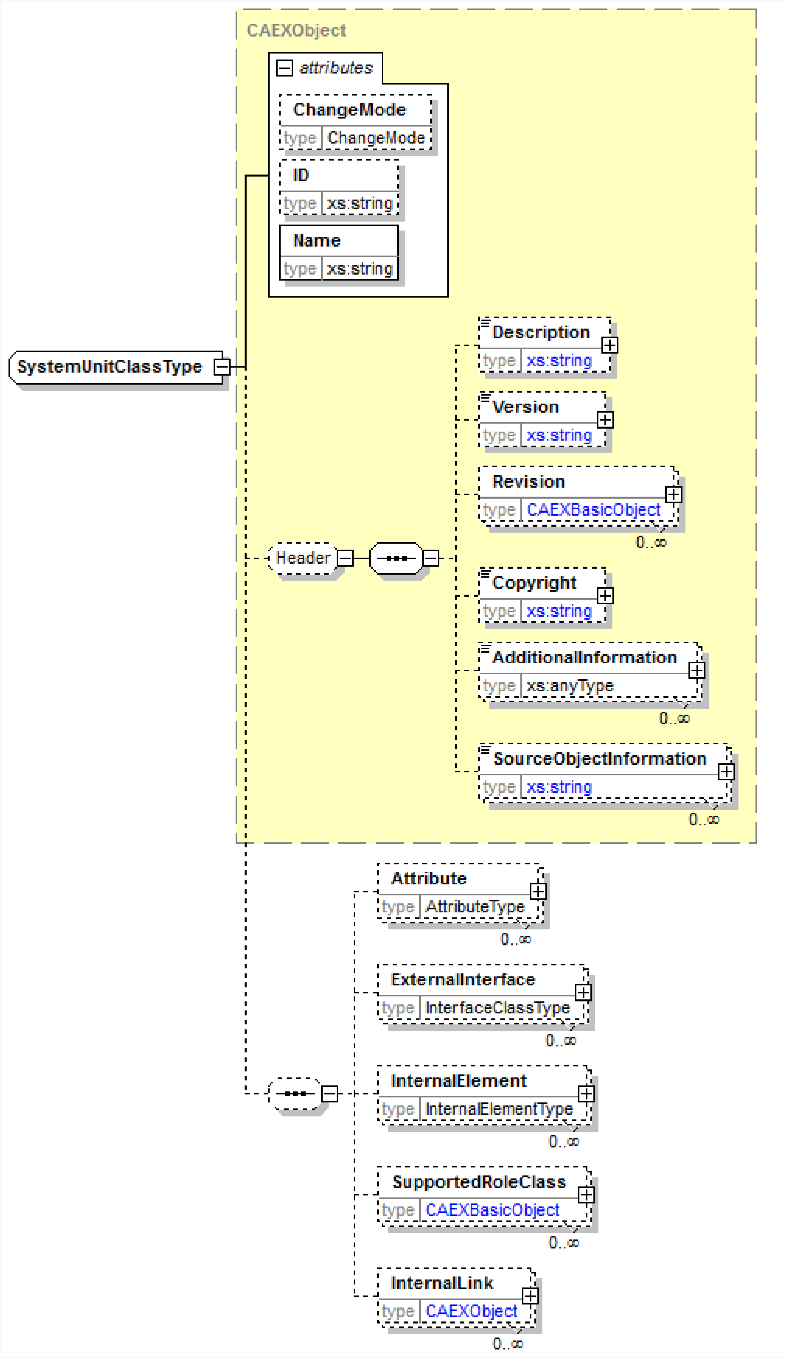
A.3.27 CAEX complex type SystemUnitClassType

A.3.27.1 General

The CAEX element “SystemUnitClassType” is the base type of the CAEX element SystemUnitClass. See A.2.3 and A.3.13 for details.

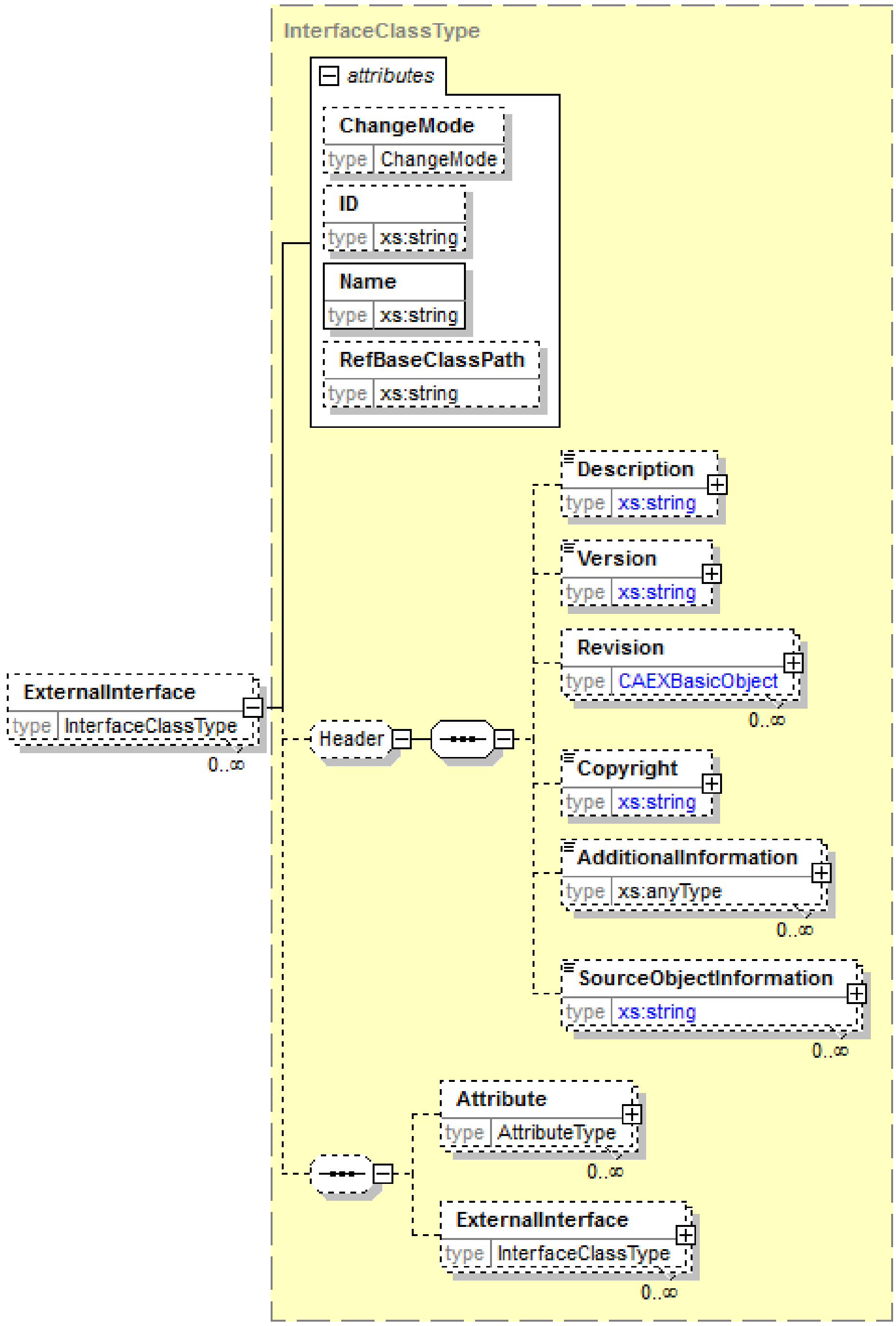
diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string SystemUnitClassType Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InternalElement type InternalElementType 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ InternalLink type CAEXObject 0..∞
namespace	http://www.dke.de/CAEX
type	extension of CAEXObject
properties	base CAEXObject
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InternalElement SupportedRoleClass InternalLink
used by	complexTypes **InternalElementType** **SystemUnitFamilyType**
attributes	Name | Type | Use | Default | Fixed ChangeMode | **ChangeMode** | optional | state | ID | **xs:string** | optional | | Name | **xs:string** | required | |

A.3.27.2 Element SystemUnitClassType/Attribute

diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string SystemUnitClassType Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞ InternalElement type InternalElementType 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ InternalLink type CAEXObject 0..∞
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	base CAEXObject
children	**Description Version Revision Copyright AdditionalInformation SourceObjectInformation Attribute ExternalInterface InternalElement SupportedRoleClass InternalLink**
used by	complexTypes **InternalElementType** **SystemUnitFamilyType**
attributes	see below

Name	Type	Use	Default	Fixed	Annotation
ChangeMode	**ChangeMode**	optional	state		
ID	**xs:string**	optional			
Name	**xs:string**	required			

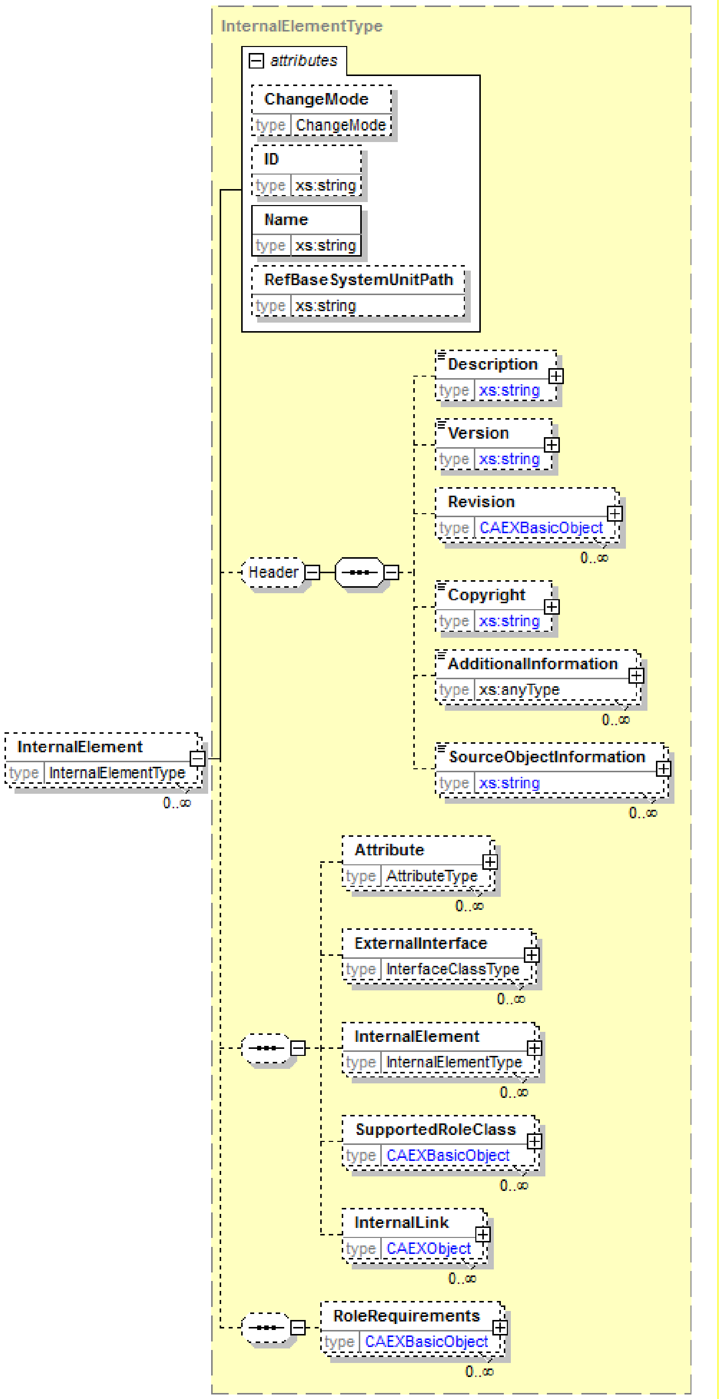
A.3.27.3 Element SystemUnitClassType/ExternalInterface

diagram	 InterfaceClassType attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string RefBaseClassPath type xs:string ExternalInterface type InterfaceClassType 0..∞ Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ Attribute type AttributeType 0..∞ ExternalInterface type InterfaceClassType 0..∞
namespace	http://www.dke.de/CAEX
type	**InterfaceClassType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface**

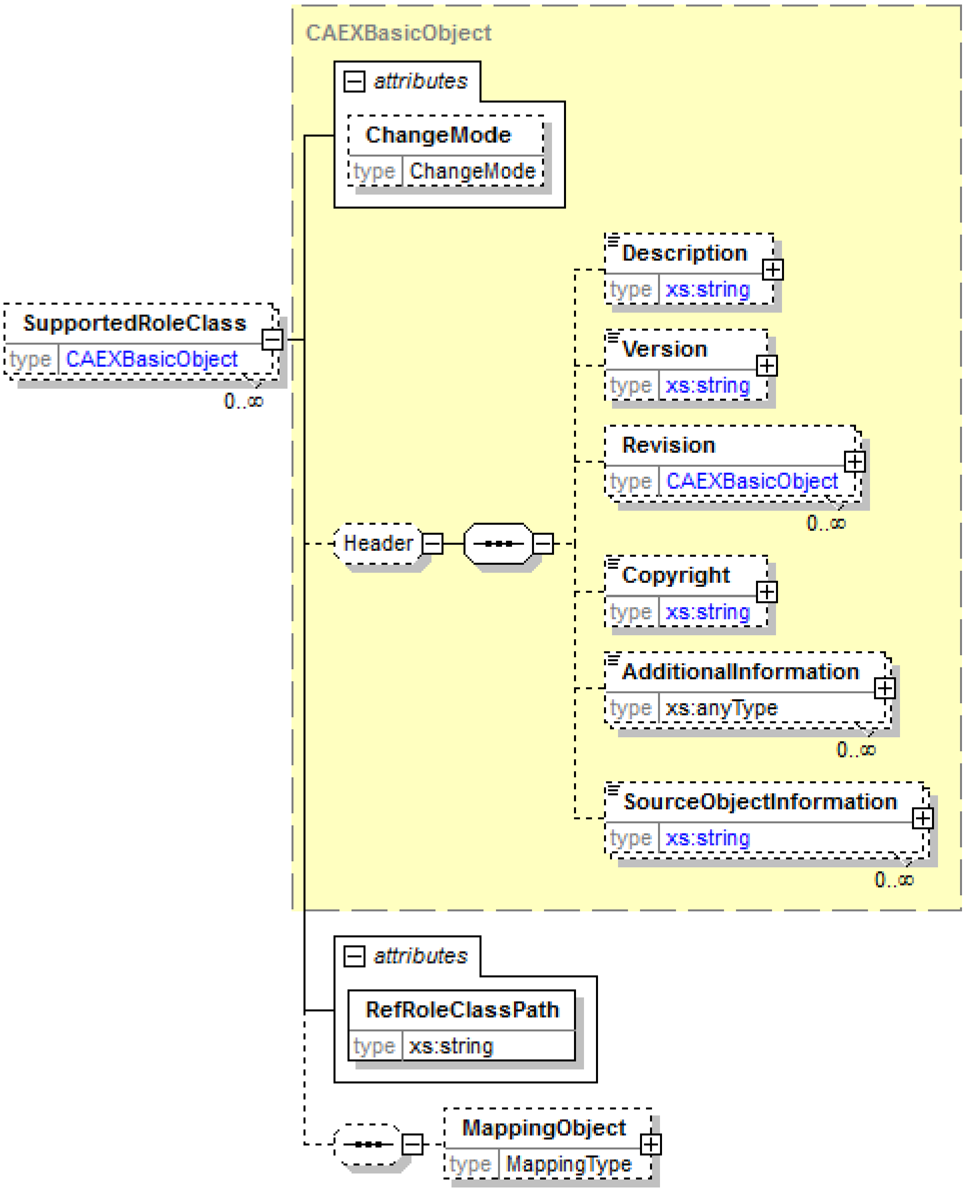
attributes

Name	Type	Use	Default	Fixed	Annotation
ChangeMode	**ChangeMode**	optional	state		
ID	**xs:string**	optional			
Name	**xs:string**	required			
RefBaseClassPath	**xs:string**	optional			

A.3.27.4 Element SystemUnitClassType/InternalElement

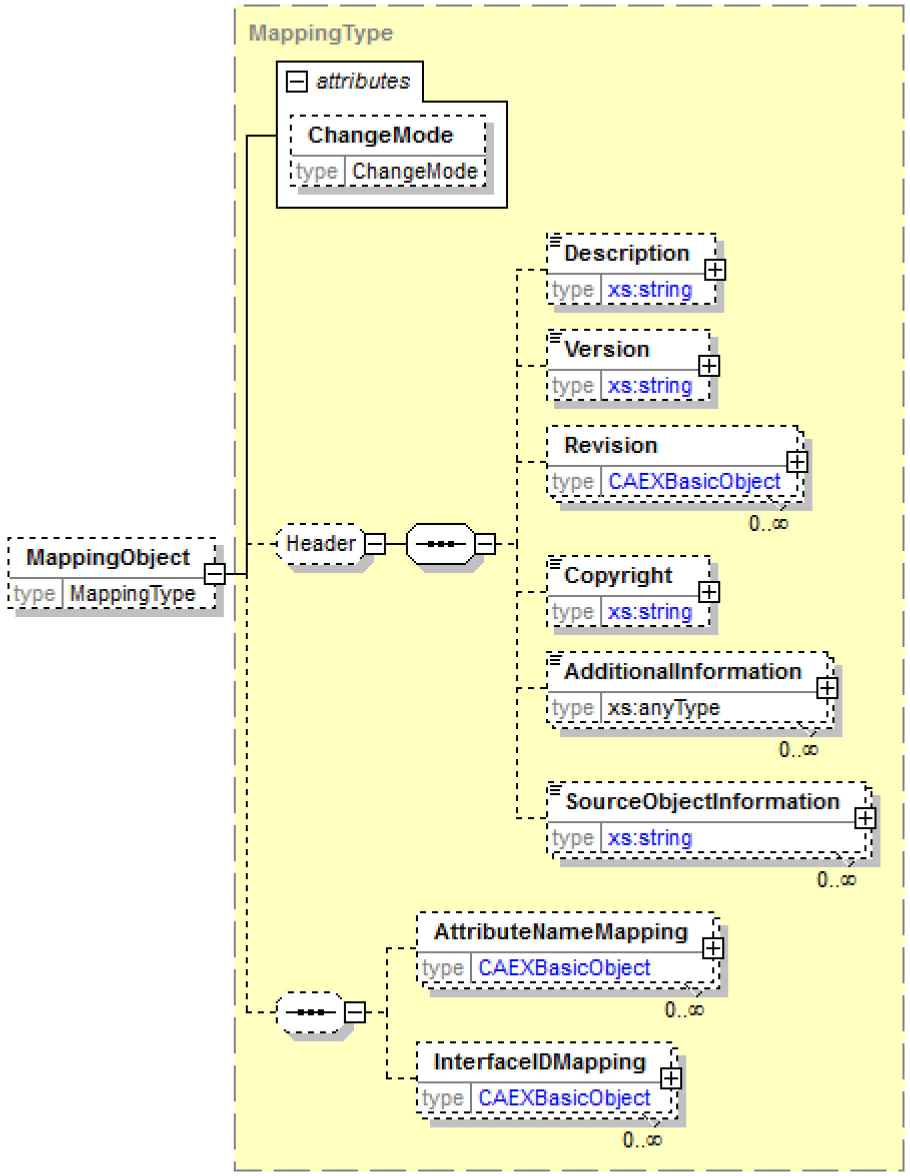
<table>
<tr><td>diagram</td><td>
</td></tr>
<tr><td>namespace</td><td>http://www.dke.de/CAEX</td></tr>
<tr><td>type</td><td><b>InternalElementType</b></td></tr>
<tr><td>properties</td><td><table>
<tr><td>isRef</td><td>0</td></tr>
<tr><td>minOcc</td><td>0</td></tr>
<tr><td>maxOcc</td><td>unbounded</td></tr>
<tr><td>content</td><td>complex</td></tr>
</table></td></tr>
<tr><td>children</td><td><b>Description</b> <b>Version</b> <b>Revision</b> <b>Copyright</b> <b>AdditionalInformation</b> <b>SourceObjectInformation</b> <b>Attribute</b> <b>ExternalInterface</b> <b>InternalElement</b> <b>SupportedRoleClass</b> <b>InternalLink</b> <b>RoleRequirements</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>ChangeMode</td><td><b>ChangeMode</b></td><td>optional</td><td>state</td><td></td><td></td></tr>
<tr><td>ID</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
<tr><td>Name</td><td><b>xs:string</b></td><td>required</td><td></td><td></td><td></td></tr>
<tr><td>RefBaseSystemUnitPath</td><td><b>xs:string</b></td><td>optional</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

A.3.27.5 Element SystemUnitClassType/SupportedRoleClass

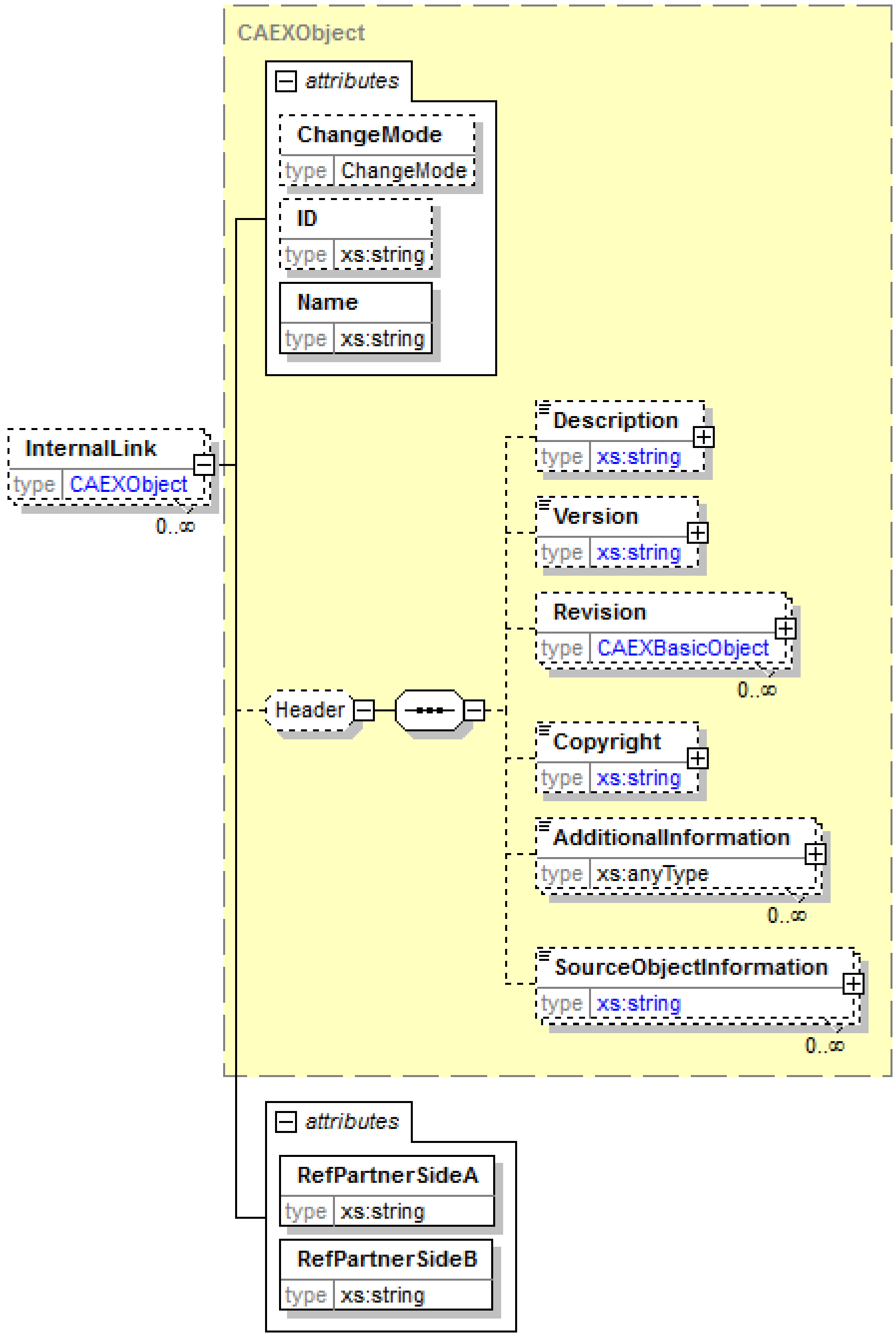
diagram	 CAEXBasicObject attributes ChangeMode type ChangeMode Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ SupportedRoleClass type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ attributes RefRoleClassPath type xs:string MappingObject type MappingType
namespace	http://www.dke.de/CAEX
type	extension of **CAEXBasicObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **MappingObject**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state RefRoleClassPath **xs:string** required

A.3.27.6 Element SystemUnitClassType/SupportedRoleClass/MappingObject

See A.2.11 for details and examples.

diagram	 MappingType attributes ChangeMode type ChangeMode MappingObject type MappingType Header Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ AttributeNameMapping type CAEXBasicObject 0..∞ InterfaceIDMapping type CAEXBasicObject 0..∞
namespace	http://www.dke.de/CAEX
type	MappingType
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	Description Version Revision Copyright AdditionalInformation SourceObjectInformation AttributeNameMapping InterfaceIDMapping
attributes	Name: ChangeMode Type: ChangeMode Use: optional Default: state Fixed: Annotation:

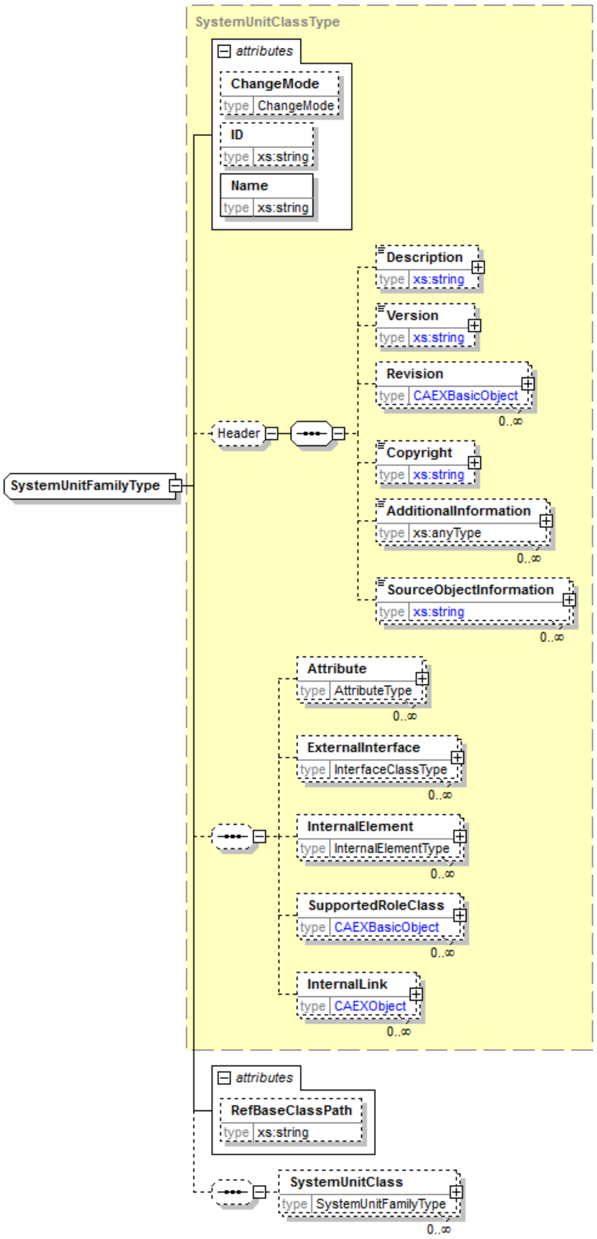
A.3.27.7 Element SystemUnitClassType/InternalLink

diagram	 CAEXObject attributes ChangeMode type ChangeMode ID type xs:string Name type xs:string InternalLink type CAEXObject 0..∞ Description type xs:string Version type xs:string Revision type CAEXBasicObject 0..∞ Header Copyright type xs:string AdditionalInformation type xs:anyType 0..∞ SourceObjectInformation type xs:string 0..∞ attributes RefPartnerSideA type xs:string RefPartnerSideB type xs:string
namespace	http://www.dke.de/CAEX
type	extension of **CAEXObject**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description Version Revision Copyright AdditionalInformation SourceObjectInformation**
attributes	Name / Type / Use / Default / Fixed / Annotation ChangeMode / **ChangeMode** / optional / state ID / **xs:string** / optional Name / **xs:string** / required RefPartnerSideA / **xs:string** / required RefPartnerSideB / **xs:string** / required

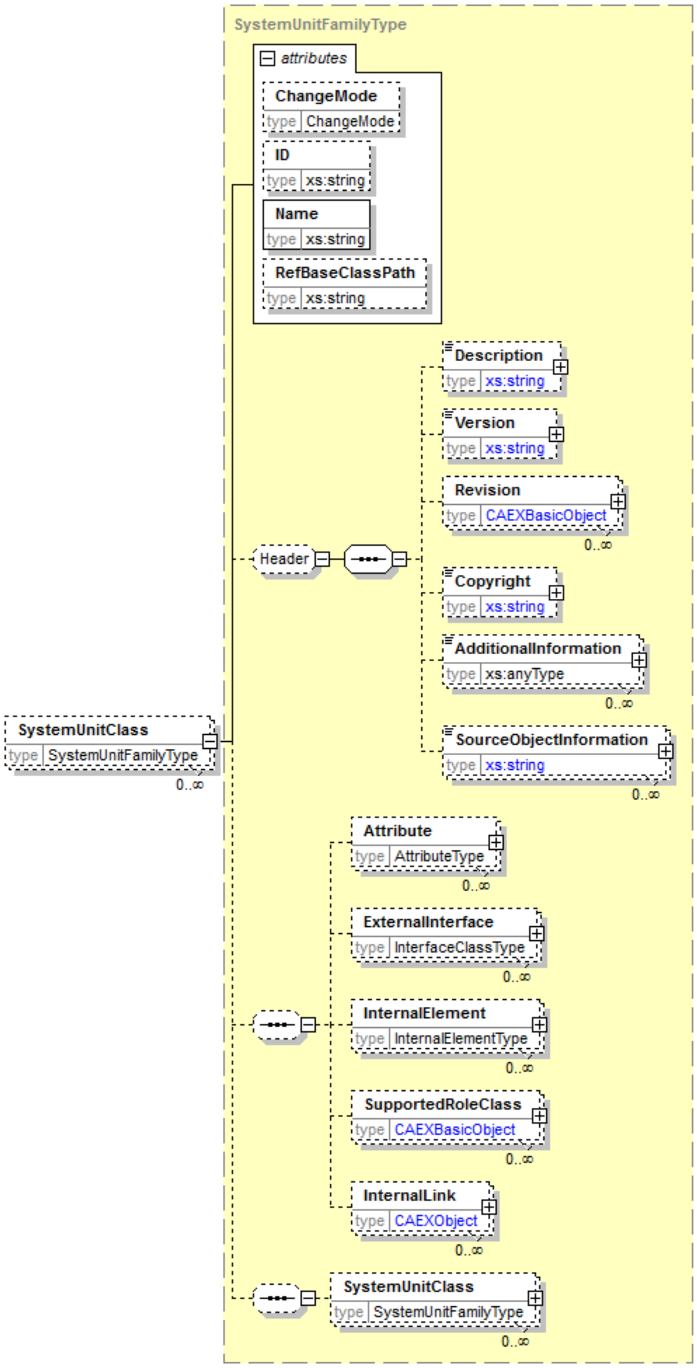
A.3.28 CAEX complex type SystemUnitFamilyType

A.3.28.1 General

The CAEX element “SystemUnitFamilyType” is an extension of the SystemUnitClassType and additionally supports adding SystemUnitClasses as children. This child is again of the type SystemUnitFamilyType – this recursive definition allows for the storage of an arbitrary SystemUnit hierarchy tree. See A.2.3, A.3.12 and A.3.13 for details and examples.

diagram	
namespace	http://www.dke.de/CAEX
type	extension of **SystemUnitClassType**
properties	base SystemUnitClassType
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface** **InternalElement** **SupportedRoleClass** **InternalLink** **SystemUnitClass**
used by	elements **CAEXFile/SystemUnitClassLib/SystemUnitClass** **SystemUnitFamilyType/SystemUnitClass**
attributes	Name Type Use Default Fixed Annotation ChangeMode **ChangeMode** optional state ID **xs:string** optional Name **xs:string** required RefBaseClassPath **xs:string** optional

A.3.28.2 Element SystemUnitFamilyType/SystemUnitClass

diagram	
namespace	http://www.dke.de/CAEX
type	**SystemUnitFamilyType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**Description** **Version** **Revision** **Copyright** **AdditionalInformation** **SourceObjectInformation** **Attribute** **ExternalInterface** **InternalElement** **SupportedRoleClass** **InternalLink** **SystemUnitClass**
attributes	Name ChangeMode ID Name RefBaseClassPath

Name	Type	Use	Default	Fixed	Annotation
ChangeMode	**ChangeMode**	optional	state		
ID	**xs:string**	optional			
Name	**xs:string**	required			
RefBaseClassPath	**xs:string**	optional			

A.3.29 CAEX simpleType ChangeMode

The CAEX type ChangeMode serves for the storage of version related information as defined in A.2.2.7.

namespace	http://www.dke.de/CAEX
type	restriction of **xs:string**
used by	attributes **CAEXBasicObject/@ChangeMode** **Header/Description/@ChangeMode** **Header/Version/@ChangeMode** **Header/Copyright/@ChangeMode**
facets	enumeration state enumeration create enumeration delete enumeration change

Annex B
(informative)

Examples of PCE requests

This Annex B (Figure B.1 to Figure B.36) provides examples of PCE requests.

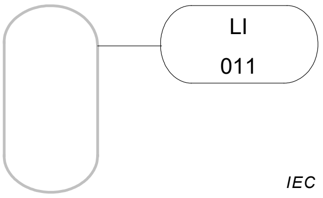


Figure B.1 – Local level indication, 1 process connection

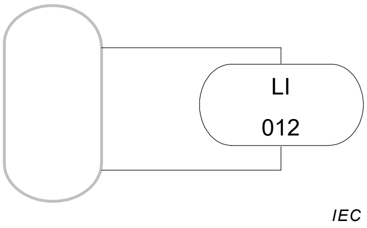


Figure B.2 – Local level indication, 2 process connections

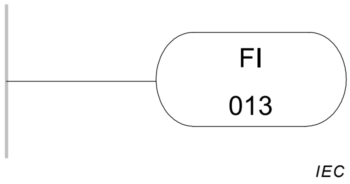


Figure B.3 – Local flow indication

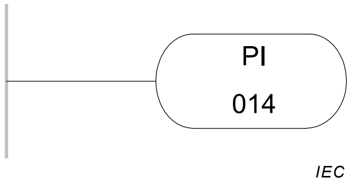


Figure B.4 – Local pressure indication

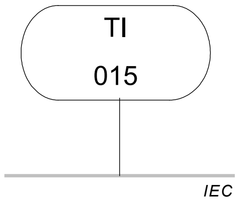


Figure B.5 – Local temperature indication

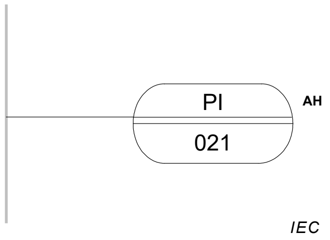
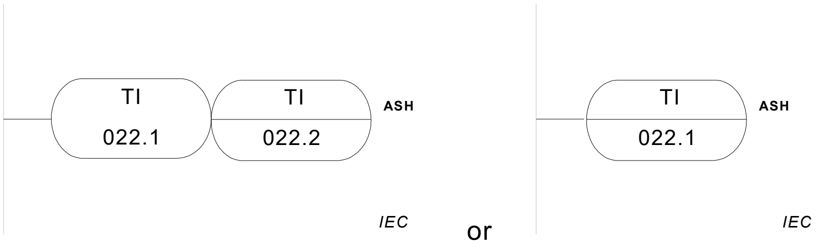


Figure B.6 – Local control panel, pressure indication, high alarm



If only one bubble is used a definition for a general explanation shall be given in the P&ID.

Figure B.7 – Local temperature indication, CCR temperature high alarm

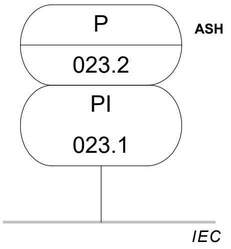


Figure B.8 – Local pressure indication, CCR pressure high alarm and switch

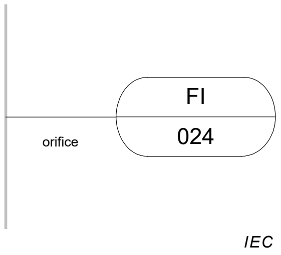


Figure B.9 – CCR flow indication, device information: Orifice Plate

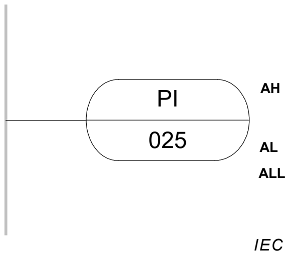


Figure B.10 – CCR pressure indication, low, low low and high alarm

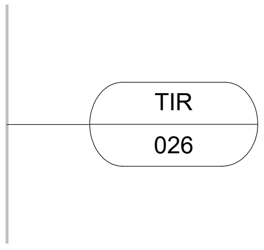


Figure B.11 – CCR temperature indication and registration

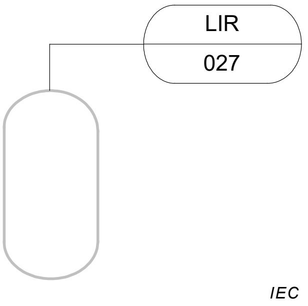


Figure B.12 – CCR level indication and registration, 1 process connection

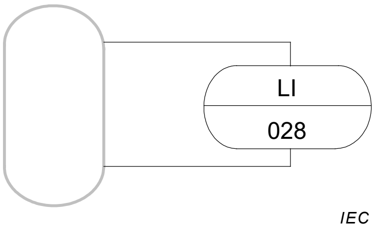


Figure B.13 – CCR level indication, 2 process connections

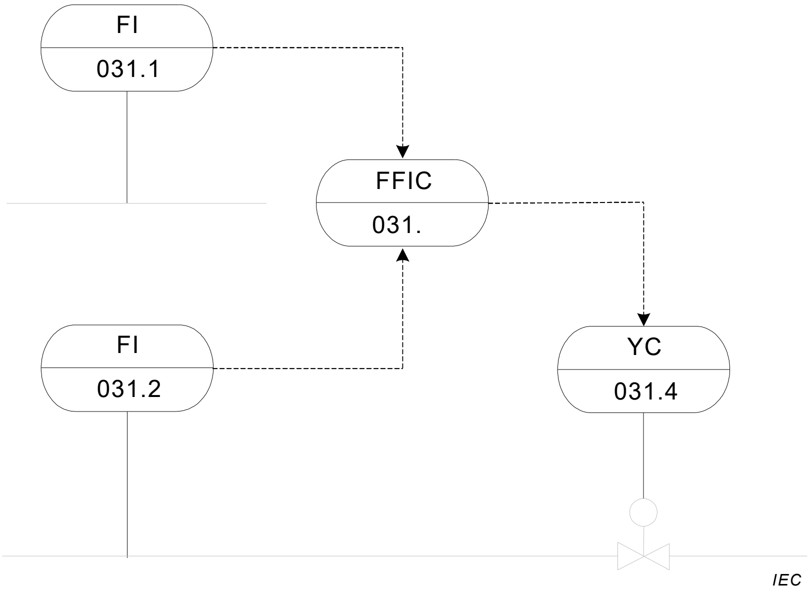


Figure B.14 – Two flow indications and flow ratio control in CCR

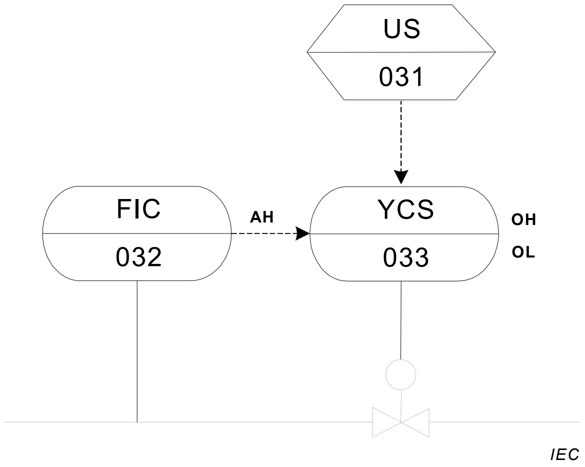


Figure B.15 – CCR flow indication and high alarm, flow control, control valve with extra interlock and open/close indication

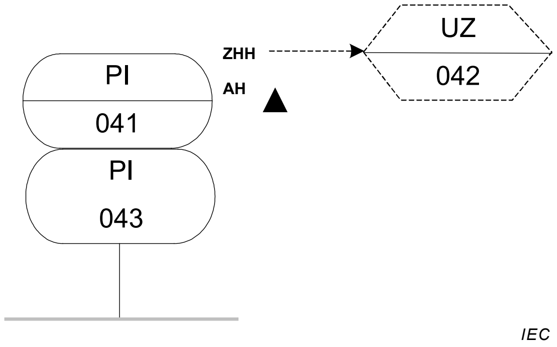


Figure B.16 – Local pressure indication, CCR pressure indication, high alarm and high high safety relevant switch; representation of transmitters with integrated local display (if not otherwise defined in a specification of the field device)

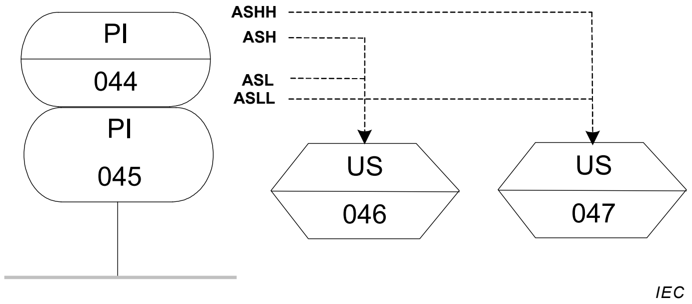


Figure B.17 – Local pressure indication, CCR pressure indication, alarms and switches

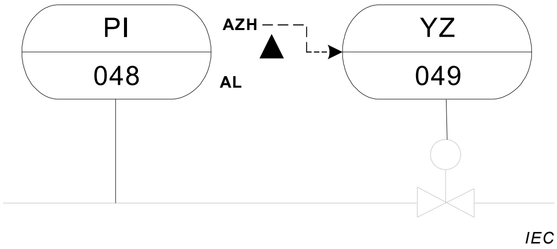


Figure B.18 – CCR pressure indication, high and low alarm, safety relevant switch action on on/off valve

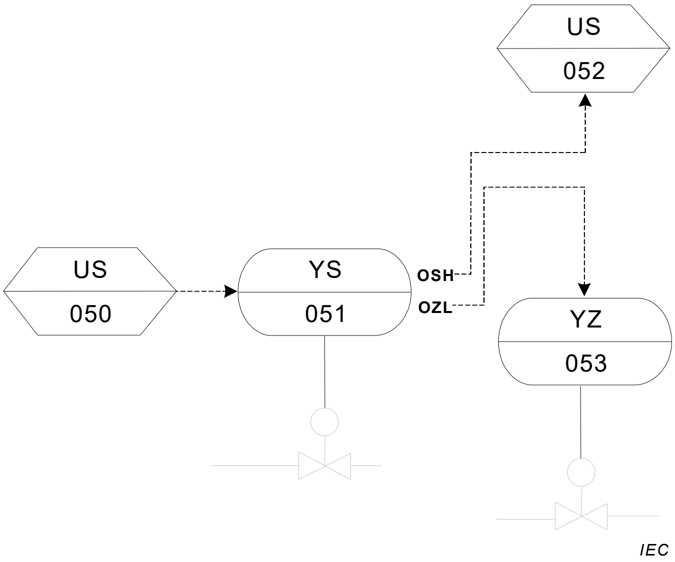


Figure B.19 – Switched valve with on/off indication and switching action, safety relevant switched valve

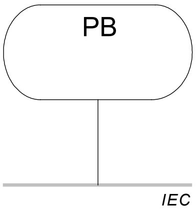


Figure B.20 – Pressure restriction

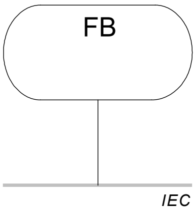


Figure B.21 – Flow restriction

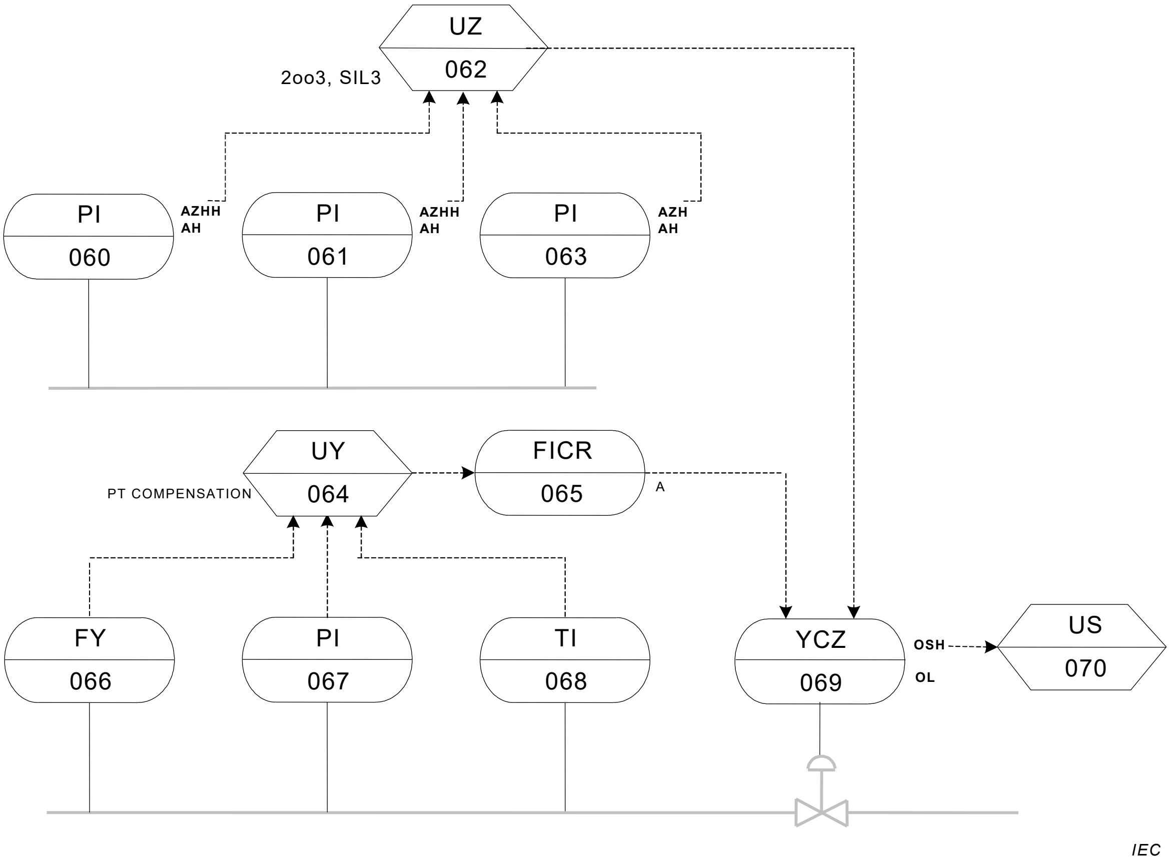


Figure B.22 – PT compensated flow control, safety-relevant pressure switch (two out of three (2oo3) shutdown), switched control valve with on/off indication and switching action at open position

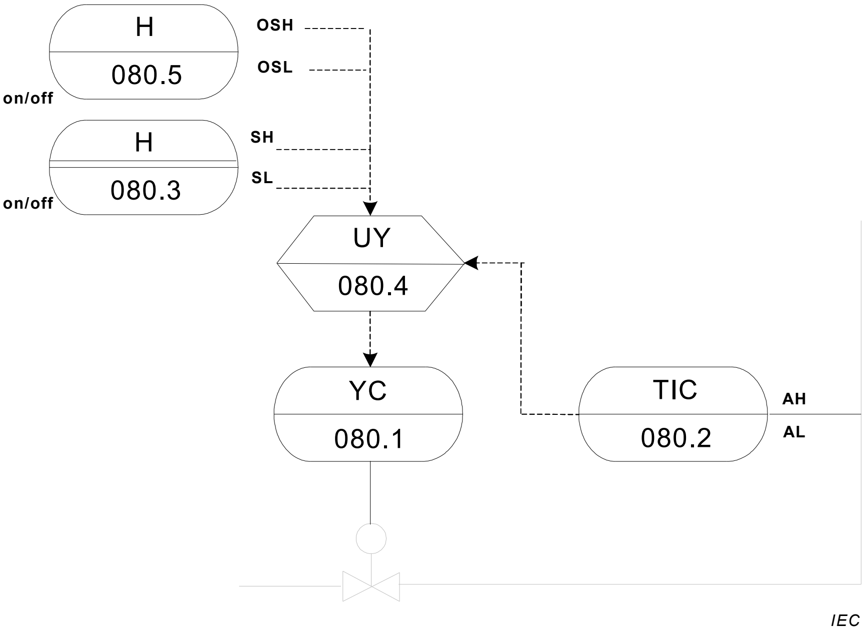


Figure B.23 – CCR temperature control, additional manual switch actions from CCR with indication and local control panel

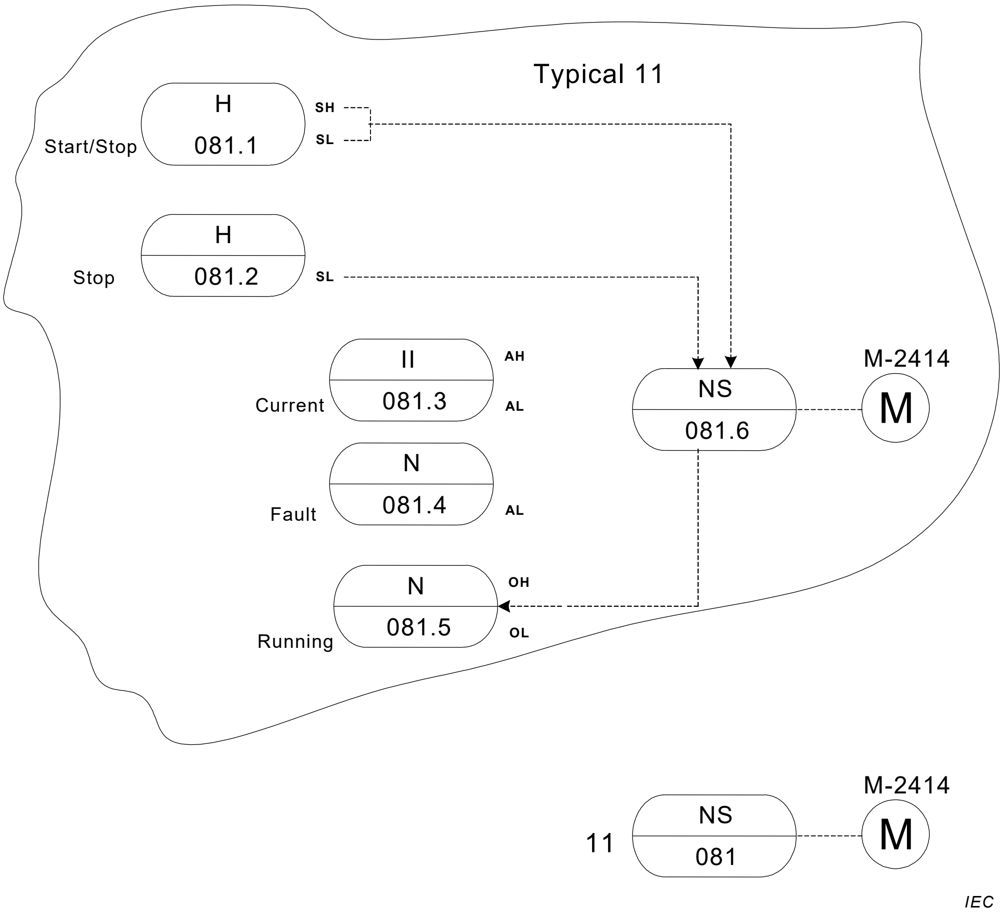


Figure B.24 – Motor typical, local on/off control, CCR off control, current, fault with alarm and running indication

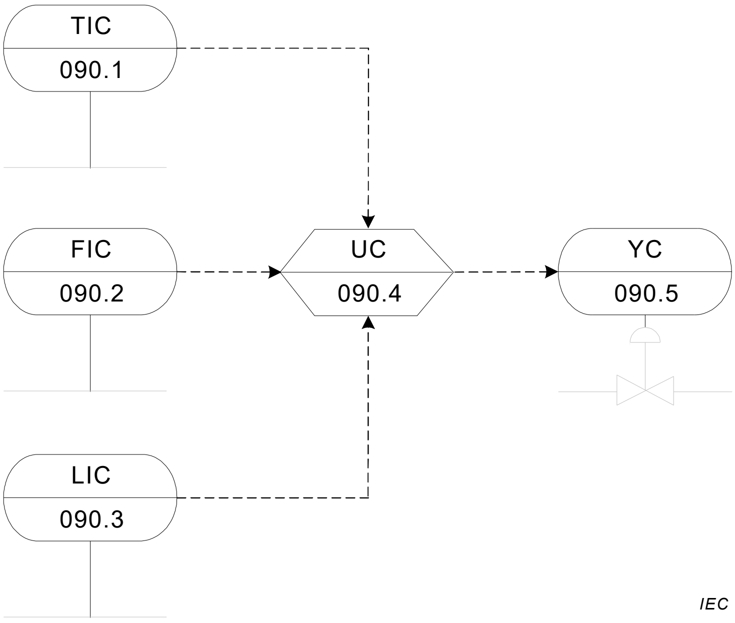


Figure B.25 – Multivariable controller

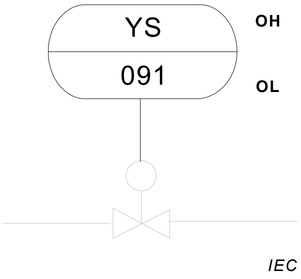


Figure B.26 – On/off valve with position indication

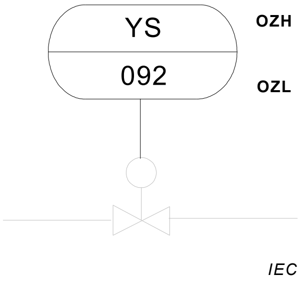


Figure B.27 – On/off valve with safety relevant switch and position indication

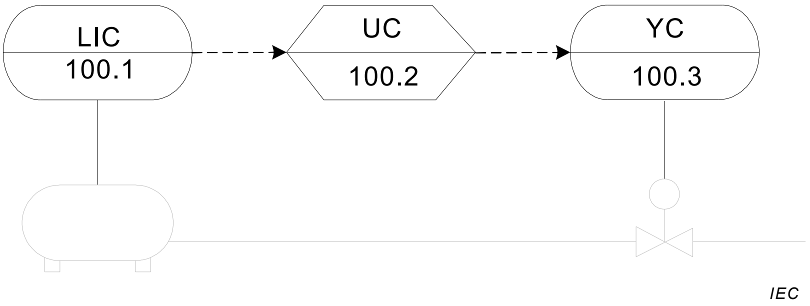


Figure B.28 – Level control with continuous controller

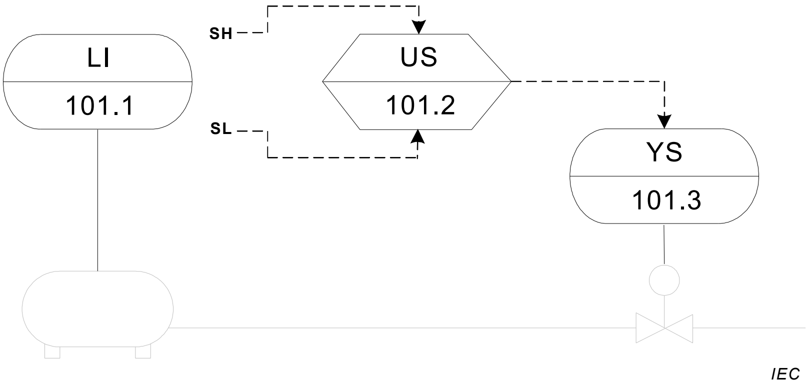


Figure B.29 – Level control with on/off switch

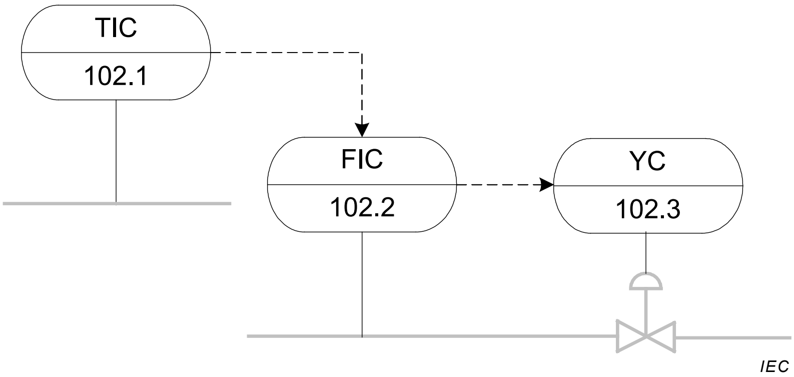


Figure B.30 – Cascade control for temperature as control input, flow control as follow-up controller

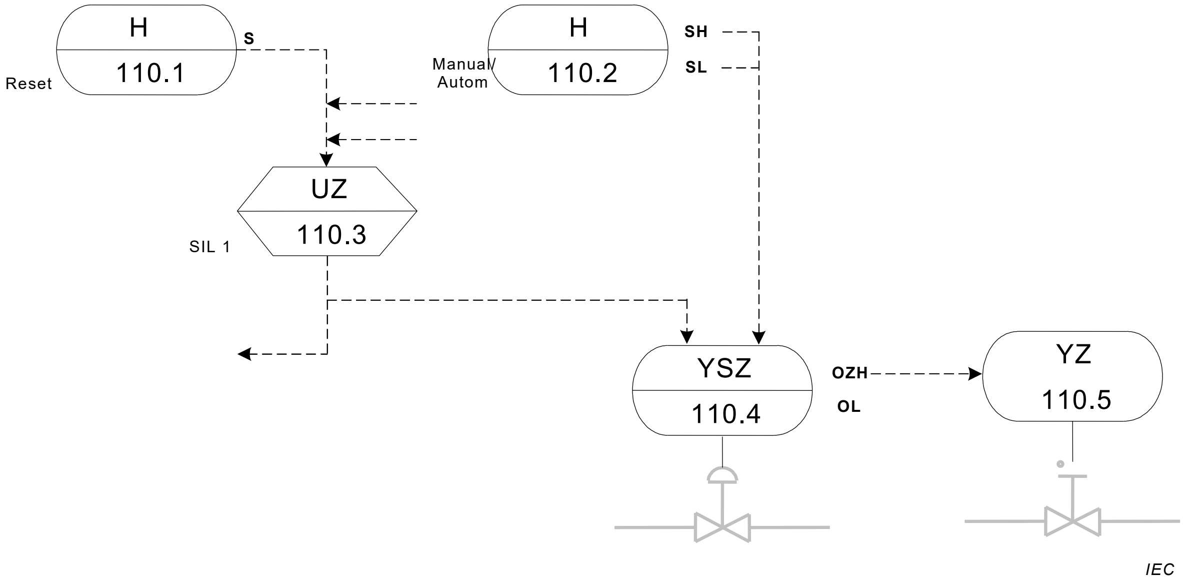


Figure B.31 – Safety directed high control to a subsequent valve, manual control for reset function and manual control for manual/automatic switch of the valve, valve with open/close indication and safety-relevant switch to subsequent valve

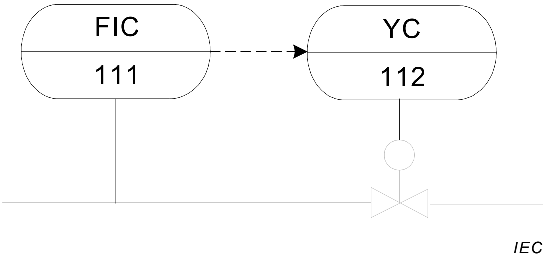


Figure B.32 – Flow control in CCR

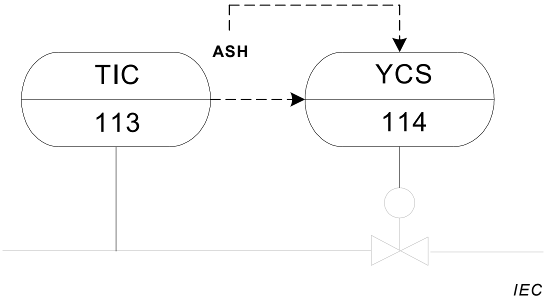


Figure B.33 – Temperature control with high alarm and high switch

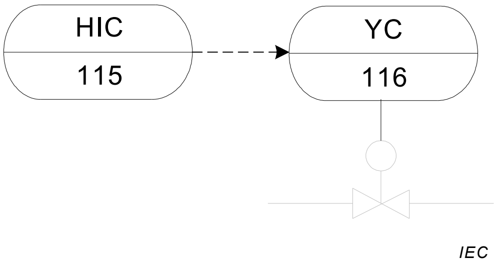


Figure B.34 – Manual control from CCR

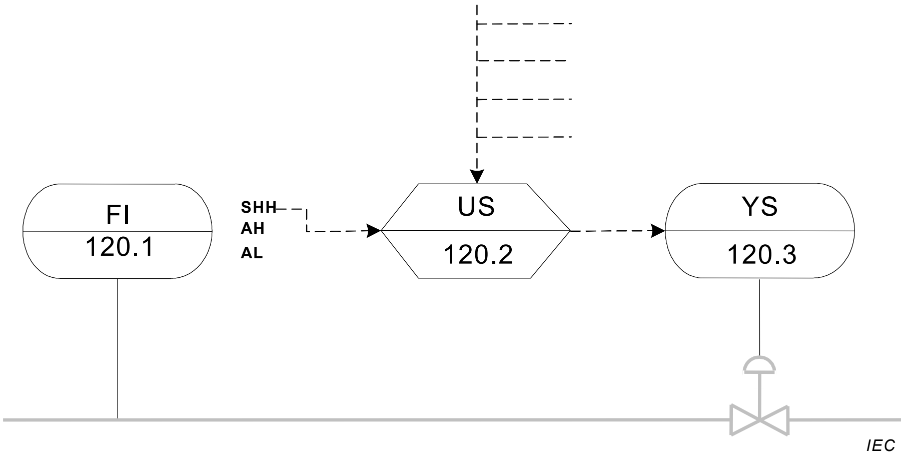


Figure B.35 – Flow measurement with display and alarms in CCR, high high switch on process control function and switch on/off valve

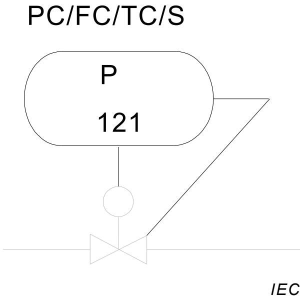


Figure B.36 – Local P-/F-/T-/S- control without auxiliary power (stand-alone)

Annex C
(normative)

Full XML schema of the CAEX model

Figure C.1 illustrates the full XML code of the CAEX schema. The file name of this XML Schema is according to A.2.2.2 “CAEX_ClassModel_V.3.0.xsd”.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- CAEX – Computer Aided Engineering Data-Exchange-Metamodel -->
<!-- Version 3.0, 31.05.2013 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.dke.de/CAEX"
targetNamespace="http://www.dke.de/CAEX" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:simpleType name="ChangeMode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="state"/>
      <xs:enumeration value="create"/>
      <xs:enumeration value="delete"/>
      <xs:enumeration value="change"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:group name="Header">
    <xs:sequence>
      <xs:element name="Description" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <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="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:complexContent>
            <xs:extension base="CAEXBasicObject">
              <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:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Copyright" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="ChangeMode" type="ChangeMode" use="optional" default="state"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="AdditionalInformation" type="xs:anyType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="SourceObjectInformation" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="OriginID" type="xs:string" use="required"/>
              <xs:attribute name="SourceObjID" type="xs:string"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:group>
<xs:complexType name="CAEXBasicObject">
  <xs:group ref="Header" minOccurs="0"/>
  <xs:attribute name="ChangeMode" type="ChangeMode" use="optional" default="state"/>
</xs:complexType>
<xs:complexType name="CAEXObject">
  <xs:complexContent>
    <xs:extension base="CAEXBasicObject">
      <xs:attribute name="ID" type="xs:string" use="optional"/>
      <xs:attribute name="Name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="InterfaceClassType">
  <xs:complexContent>
    <xs:extension base="CAEXObject">
      <xs:sequence minOccurs="0">
        <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ExternalInterface" type="InterfaceClassType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="RefBaseClassPath" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<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>
<xs:complexType name="RoleClassType">
  <xs:complexContent>
    <xs:extension base="CAEXObject">
      <xs:sequence minOccurs="0">
        <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ExternalInterface" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="InterfaceClassType"/>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="RefBaseClassPath" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<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>
<xs:complexType name="SystemUnitClassType">
  <xs:complexContent>
    <xs:extension base="CAEXObject">
      <xs:sequence minOccurs="0">
        <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ExternalInterface" type="InterfaceClassType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="InternalElement" type="InternalElementType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SupportedRoleClass" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXBasicObject">
                <xs:sequence minOccurs="0">
                  <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
                <xs:sequence>
                <xs:attribute name="RefRoleClassPath" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="InternalLink" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXObject">
                <xs:attribute name="RefPartnerSideA" type="xs:string" use="required"/>
                <xs:attribute name="RefPartnerSideB" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SystemUnitFamilyType">
  <xs:complexContent>
    <xs:extension base="SystemUnitClassType">
      <xs:sequence minOccurs="0">
        <xs:element name="SystemUnitClass" type="SystemUnitFamilyType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="RefBaseClassPath" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="InternalElementType">
  <xs:complexContent>
    <xs:extension base="SystemUnitClassType">
      <xs:sequence minOccurs="0">
        <xs:element name="RoleRequirements" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXBasicObject">
                <xs:sequence>
                  <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="ExternalInterface" type="InterfaceClassType" minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="MappingObject" type="MappingType" minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="RefBaseRoleClassPath" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="RefBaseSystemUnitPath" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttributeType">
  <xs:complexContent>
    <xs:extension base="CAEXObject">
      <xs:sequence minOccurs="0">
        <xs:element name="DefaultValue" type="xs:string" minOccurs="0"/>
        <xs:element name="Value" type="xs:string" minOccurs="0"/>
        <xs:element name="RefSemantic" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXBasicObject">
                <xs:attribute name="CorrespondingAttributePath" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="Constraint" type="AttributeValueRequirementType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Attribute" type="AttributeType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="Unit" type="xs:string" use="optional"/>
      <xs:attribute name="AttributeDataType" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="RefAttributeType" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttributeFamilyType">
  <xs:complexContent>
    <xs:extension base="AttributeType">
      <xs:sequence>
        <xs:element name="AttributeType" type="AttributeFamilyType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttributeValueRequirementType">
  <xs:complexContent>
    <xs:extension base="CAEXBasicObject">
      <xs:choice>
        <xs:element name="OrdinalScaledType">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="RequiredMaxValue" type="xs:string" minOccurs="0"/>
              <xs:element name="RequiredValue" type="xs:string" minOccurs="0"/>
              <xs:element name="RequiredMinValue" type="xs:string" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="NominalScaledType">
          <xs:complexType>
            <xs:sequence minOccurs="0">
              <xs:element name="RequiredValue" type="xs:string" 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="Name" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="MappingType">
  <xs:complexContent>
    <xs:extension base="CAEXBasicObject">
      <xs:sequence minOccurs="0">
        <xs:element name="AttributeNameMapping" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXBasicObject">
                <xs:attribute name="SystemUnitAttributeName" type="xs:string" use="required"/>
                <xs:attribute name="RoleAttributeName" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="InterfaceIDMapping" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="CAEXBasicObject">
                <xs:attribute name="SystemUnitInterfaceID" type="xs:string" use="required"/>
                <xs:attribute name="RoleInterfaceID" type="xs:string" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SourceDocumentInformationType">
  <xs:attribute name="OriginName" type="xs:string" use="required"/>
  <xs:attribute name="OriginID" type="xs:string" use="required"/>
  <xs:attribute name="OriginVendor" type="xs:string" use="optional"/>
```

```
    <xs:attribute name="OriginVendorURL" type="xs:string" use="optional"/>
    <xs:attribute name="OriginVersion" type="xs:string" use="required"/>
    <xs:attribute name="OriginRelease" type="xs:string" use="optional"/>
    <xs:attribute name="LastWritingDateTime" type="xs:dateTime" use="required"/>
    <xs:attribute name="OriginProjectTitle" type="xs:string" use="optional"/>
    <xs:attribute name="OriginProjectID" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:element name="CAEXFile">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="CAEXBasicObject">
          <xs:sequence>
            <xs:element name="SuperiorStandardVersion" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="SourceDocumentInformation" type="SourceDocumentInformationType" maxOccurs="unbounded"/>
            <xs:element name="ExternalReference" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXBasicObject">
                    <xs:attribute name="Path" type="xs:string" use="required"/>
                    <xs:attribute name="Alias" type="xs:string" use="required"/>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="InstanceHierarchy" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXObject">
                    <xs:sequence>
                      <xs:element name="InternalElement" type="InternalElementType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="InterfaceClassLib" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXObject">
                    <xs:sequence>
                      <xs:element name="InterfaceClass" type="InterfaceFamilyType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="RoleClassLib" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXObject">
                    <xs:sequence>
                      <xs:element name="RoleClass" type="RoleFamilyType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="SystemUnitClassLib" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXObject">
                    <xs:sequence>
                      <xs:element name="SystemUnitClass" type="SystemUnitFamilyType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="AttributeTypeLib" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="CAEXObject">
                    <xs:sequence>
                      <xs:element name="AttributeType" type="AttributeFamilyType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:extension>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="SchemaVersion" type="xs:string" use="required" fixed="3.0"/>
          <xs:attribute name="FileName" type="xs:string" use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

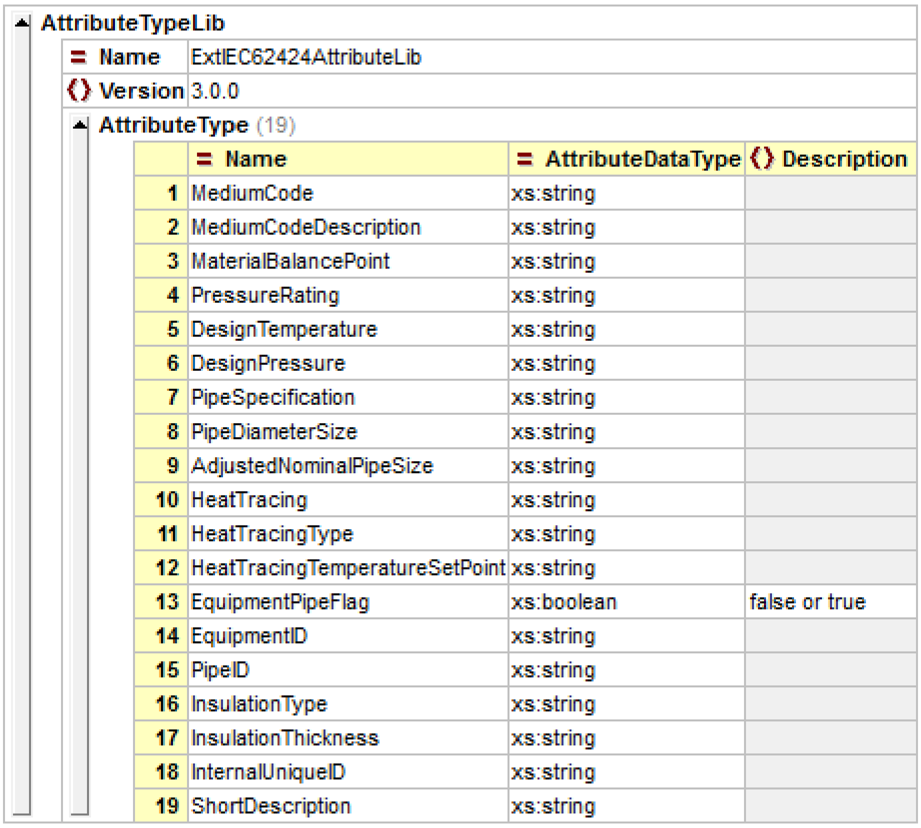
Figure C.1 – Full XML text of the CAEX Schema file “CAEX_ClassModel_V.3.0.xsd”

Annex D
(informative)

CAEX modelling examples

D.1 CAEX Attribute Type Library definition for additional attributes

Clause D.1 defines an attribute type library which models additional PCE request attributes according to Clause 8. This is illustrated in Figure D.1.



AttributeTypeLib

= Name	ExtIEC62424AttributeLib
{} Version	3.0.0

AttributeType (19)

	= Name	= AttributeDataType	{} Description
1	MediumCode	xs:string	
2	MediumCodeDescription	xs:string	
3	MaterialBalancePoint	xs:string	
4	PressureRating	xs:string	
5	DesignTemperature	xs:string	
6	DesignPressure	xs:string	
7	PipeSpecification	xs:string	
8	PipeDiameterSize	xs:string	
9	AdjustedNominalPipeSize	xs:string	
10	HeatTracing	xs:string	
11	HeatTracingType	xs:string	
12	HeatTracingTemperatureSetPoint	xs:string	
13	EquipmentPipeFlag	xs:boolean	false or true
14	EquipmentID	xs:string	
15	PipeID	xs:string	
16	InsulationType	xs:string	
17	InsulationThickness	xs:string	
18	InternalUniqueID	xs:string	
19	ShortDescription	xs:string	

IEC

Figure D.1 – Attribute type library with additional PCE request related attributes

The XML code of the attribute type library is shown in Figure D.2.

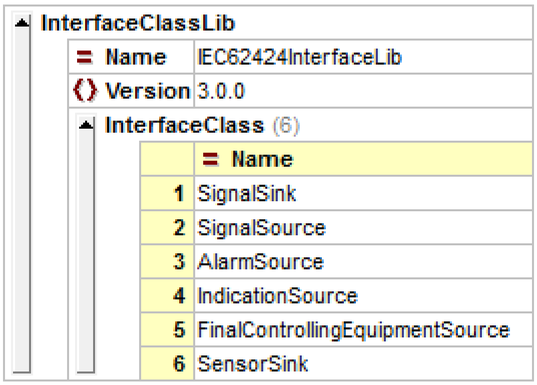
```
<AttributeTypeLib Name="ExtIEC62424AttributeLib">
    <Version>3.0.0</Version>
    <AttributeType Name="MediumCode" AttributeDataType="xs:string"/>
    <AttributeType Name="MediumCodeDescription" AttributeDataType="xs:string"/>
    <AttributeType Name="MaterialBalancePoint" AttributeDataType="xs:string"/>
    <AttributeType Name="PressureRating" AttributeDataType="xs:string"/>
    <AttributeType Name="DesignTemperature" AttributeDataType="xs:string"/>
    <AttributeType Name="DesignPressure" AttributeDataType="xs:string"/>
    <AttributeType Name="PipeSpecification" AttributeDataType="xs:string"/>
    <AttributeType Name="PipeDiameterSize" AttributeDataType="xs:string"/>
    <AttributeType Name="AdjustedNominalPipeSize" AttributeDataType="xs:string"/>
    <AttributeType Name="HeatTracing" AttributeDataType="xs:string"/>
    <AttributeType Name="HeatTracingType" AttributeDataType="xs:string"/>
    <AttributeType Name="HeatTracingTemperatureSetPoint" AttributeDataType="xs:string"/>
    <AttributeType Name="EquipmentPipeFlag" AttributeDataType="xs:boolean">
        <Description>false or true</Description>
    </AttributeType>
    <AttributeType Name="EquipmentID" AttributeDataType="xs:string"/>
    <AttributeType Name="PipeID" AttributeDataType="xs:string"/>
    <AttributeType Name="InsulationType" AttributeDataType="xs:string"/>
    <AttributeType Name="InsulationThickness" AttributeDataType="xs:string"/>
    <AttributeType Name="InternalUniqueID" AttributeDataType="xs:string"/>
    <AttributeType Name="ShortDescription" AttributeDataType="xs:string"/>
</AttributeTypeLib>
```

IEC

Figure D.2 – XML code of the Attribute type library

D.2 Example of CAEX InterfaceLib definition

Figure D.3 presents a CAEX interface library which defines all interface types according to 7.4.2.



InterfaceClassLib

= Name	IEC62424InterfaceLib
{} Version	3.0.0

InterfaceClass (6)

	= Name
1	SignalSink
2	SignalSource
3	AlarmSource
4	IndicationSource
5	FinalControllingEquipmentSource
6	SensorSink

IEC

Figure D.3 – Example of CAEX interface library

The full XML text for this example is shown in Figure D.4.

```
<InterfaceClassLib Name="IEC62424InterfaceLib">
      <Version>3.0.0</Version>
      <InterfaceClass Name="SignalSink"/>
      <InterfaceClass Name="SignalSource"/>
      <InterfaceClass Name="AlarmSource"/>
      <InterfaceClass Name="IndicationSource"/>
      <InterfaceClass Name="FinalControllingEquipmentSource"/>
      <InterfaceClass Name="SensorSink"/>
</InterfaceClassLib>
```

IEC

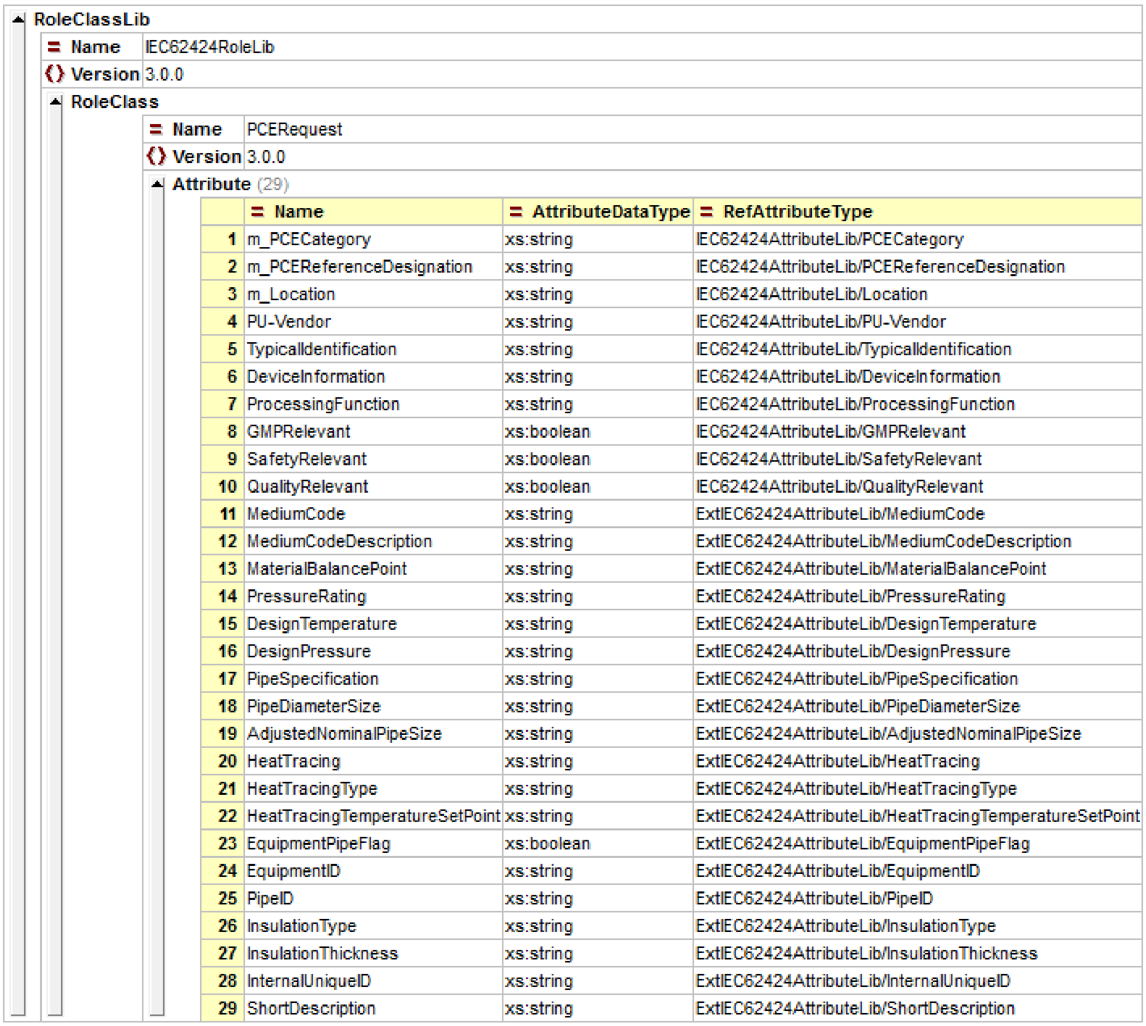
Figure D.4 – XML code of the example CAEX interface library

D.3 Example of a CAEX RoleLib definition

Clause D.3 specifies a CAEX role library with a predefined PCE request template in the form of a CAEX RoleClass which conforms to IEC 62424. This comprises the role library “IEC62424RoleLib” and a role class “PCERequest” (see Figure D.5). This role class does implement all mandatory attributes of a PCE request according to 7.5.3 and all additional attributes according to Clause D.1.

In the present role class, all mandatory attributes have modified names with a prefix “m_” which eases the application of those attributes. Since all attributes reference IEC 62424 conformant attribute types, they inherit the semantics of this standard.

If this role class is applied in an instance (see the example in Clause D.4), not all attributes have to be present. According to A.2.8.5, attributes which are not required may be removed on instance level.



RoleClassLib

= Name	IEC62424RoleLib
{} Version	3.0.0

RoleClass

= Name	PCERequest
{} Version	3.0.0

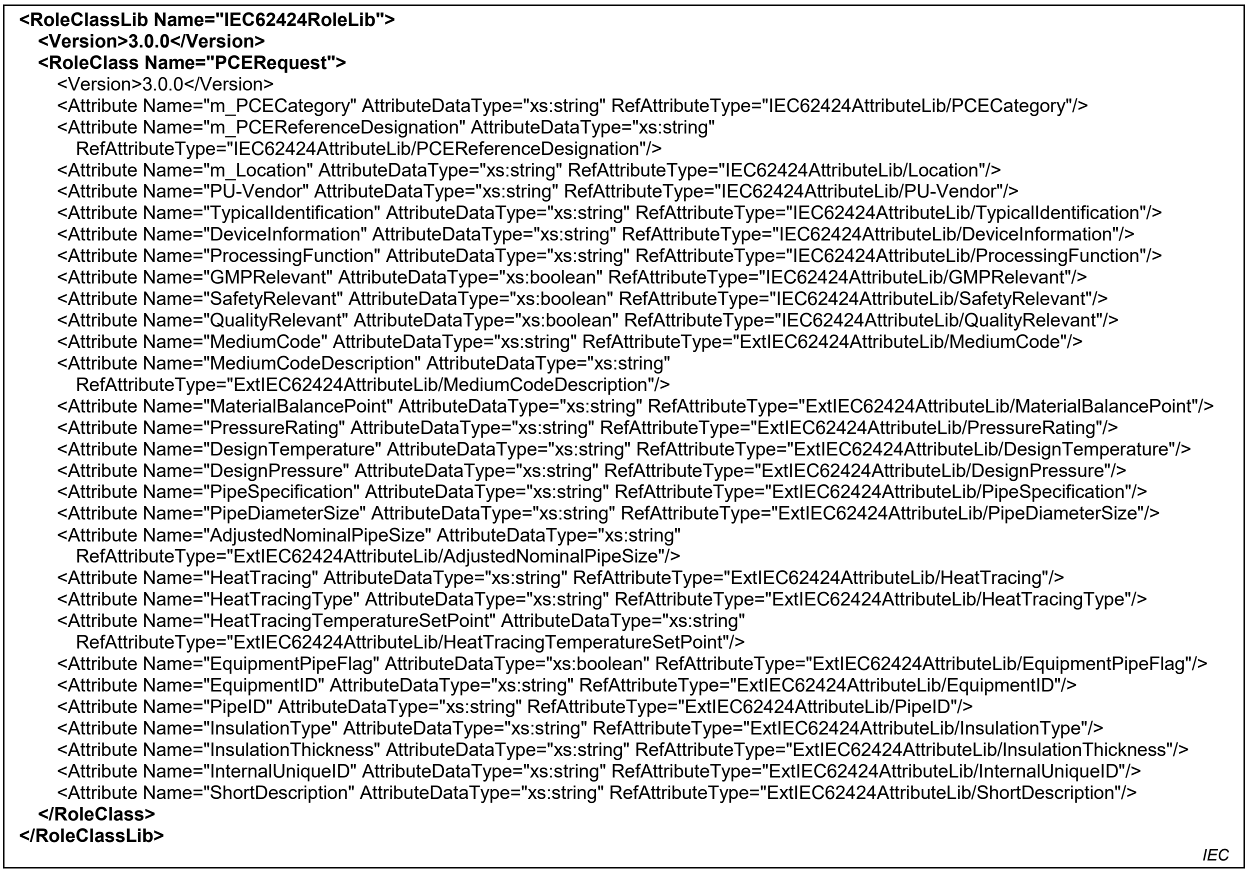
Attribute (29)

	= Name	= AttributeDataType	= RefAttributeType
1	m_PCECategory	xs:string	IEC62424AttributeLib/PCECategory
2	m_PCEReferenceDesignation	xs:string	IEC62424AttributeLib/PCEReferenceDesignation
3	m_Location	xs:string	IEC62424AttributeLib/Location
4	PU-Vendor	xs:string	IEC62424AttributeLib/PU-Vendor
5	TypicalIdentification	xs:string	IEC62424AttributeLib/TypicalIdentification
6	DeviceInformation	xs:string	IEC62424AttributeLib/DeviceInformation
7	ProcessingFunction	xs:string	IEC62424AttributeLib/ProcessingFunction
8	GMPRelevant	xs:boolean	IEC62424AttributeLib/GMPRelevant
9	SafetyRelevant	xs:boolean	IEC62424AttributeLib/SafetyRelevant
10	QualityRelevant	xs:boolean	IEC62424AttributeLib/QualityRelevant
11	MediumCode	xs:string	ExtIEC62424AttributeLib/MediumCode
12	MediumCodeDescription	xs:string	ExtIEC62424AttributeLib/MediumCodeDescription
13	MaterialBalancePoint	xs:string	ExtIEC62424AttributeLib/MaterialBalancePoint
14	PressureRating	xs:string	ExtIEC62424AttributeLib/PressureRating
15	DesignTemperature	xs:string	ExtIEC62424AttributeLib/DesignTemperature
16	DesignPressure	xs:string	ExtIEC62424AttributeLib/DesignPressure
17	PipeSpecification	xs:string	ExtIEC62424AttributeLib/PipeSpecification
18	PipeDiameterSize	xs:string	ExtIEC62424AttributeLib/PipeDiameterSize
19	AdjustedNominalPipeSize	xs:string	ExtIEC62424AttributeLib/AdjustedNominalPipeSize
20	HeatTracing	xs:string	ExtIEC62424AttributeLib/HeatTracing
21	HeatTracingType	xs:string	ExtIEC62424AttributeLib/HeatTracingType
22	HeatTracingTemperatureSetPoint	xs:string	ExtIEC62424AttributeLib/HeatTracingTemperatureSetPoint
23	EquipmentPipeFlag	xs:boolean	ExtIEC62424AttributeLib/EquipmentPipeFlag
24	EquipmentID	xs:string	ExtIEC62424AttributeLib/EquipmentID
25	PipeID	xs:string	ExtIEC62424AttributeLib/PipeID
26	InsulationType	xs:string	ExtIEC62424AttributeLib/InsulationType
27	InsulationThickness	xs:string	ExtIEC62424AttributeLib/InsulationThickness
28	InternalUniqueID	xs:string	ExtIEC62424AttributeLib/InternalUniqueID
29	ShortDescription	xs:string	ExtIEC62424AttributeLib/ShortDescription

IEC

Figure D.5 – Example CAEX role library illustrating the modeling of a PCE request role referencing PCE request related attributes

The full XML text for this example is shown in Figure D.6.



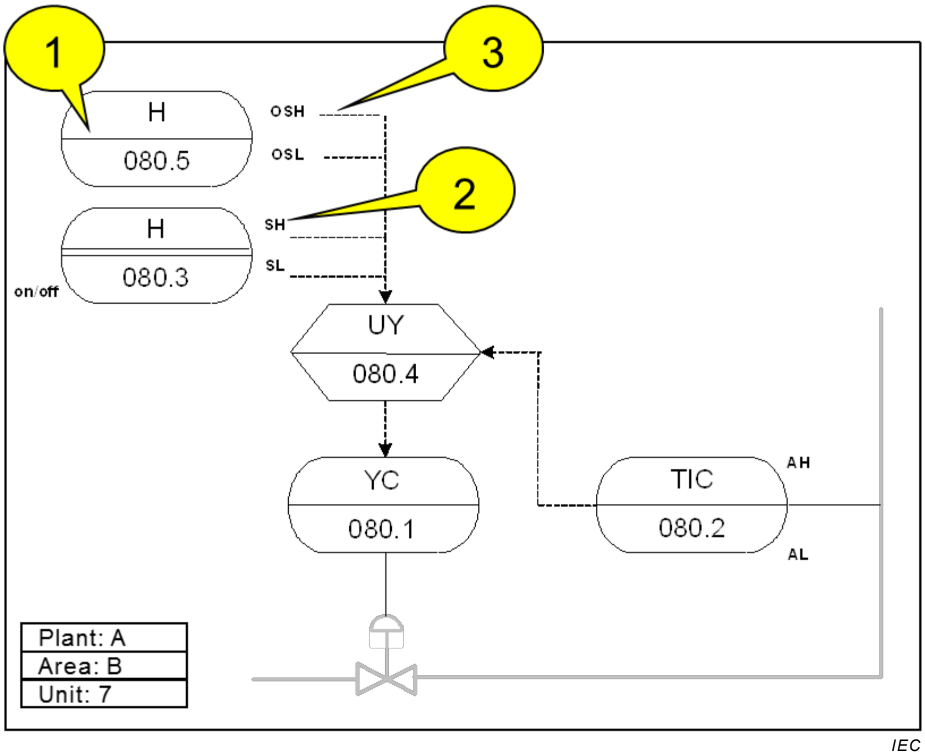
```
<RoleClassLib Name="IEC62424RoleLib">
  <Version>3.0.0</Version>
  <RoleClass Name="PCERequest">
    <Version>3.0.0</Version>
    <Attribute Name="m_PCECategory" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/PCECategory"/>
    <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string"
      RefAttributeType="IEC62424AttributeLib/PCEReferenceDesignation"/>
    <Attribute Name="m_Location" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/Location"/>
    <Attribute Name="PU-Vendor" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/PU-Vendor"/>
    <Attribute Name="TypicalIdentification" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/TypicalIdentification"/>
    <Attribute Name="DeviceInformation" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/DeviceInformation"/>
    <Attribute Name="ProcessingFunction" AttributeDataType="xs:string" RefAttributeType="IEC62424AttributeLib/ProcessingFunction"/>
    <Attribute Name="GMPRelevant" AttributeDataType="xs:boolean" RefAttributeType="IEC62424AttributeLib/GMPRelevant"/>
    <Attribute Name="SafetyRelevant" AttributeDataType="xs:boolean" RefAttributeType="IEC62424AttributeLib/SafetyRelevant"/>
    <Attribute Name="QualityRelevant" AttributeDataType="xs:boolean" RefAttributeType="IEC62424AttributeLib/QualityRelevant"/>
    <Attribute Name="MediumCode" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/MediumCode"/>
    <Attribute Name="MediumCodeDescription" AttributeDataType="xs:string"
      RefAttributeType="ExtIEC62424AttributeLib/MediumCodeDescription"/>
    <Attribute Name="MaterialBalancePoint" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/MaterialBalancePoint"/>
    <Attribute Name="PressureRating" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/PressureRating"/>
    <Attribute Name="DesignTemperature" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/DesignTemperature"/>
    <Attribute Name="DesignPressure" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/DesignPressure"/>
    <Attribute Name="PipeSpecification" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/PipeSpecification"/>
    <Attribute Name="PipeDiameterSize" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/PipeDiameterSize"/>
    <Attribute Name="AdjustedNominalPipeSize" AttributeDataType="xs:string"
      RefAttributeType="ExtIEC62424AttributeLib/AdjustedNominalPipeSize"/>
    <Attribute Name="HeatTracing" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/HeatTracing"/>
    <Attribute Name="HeatTracingType" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/HeatTracingType"/>
    <Attribute Name="HeatTracingTemperatureSetPoint" AttributeDataType="xs:string"
      RefAttributeType="ExtIEC62424AttributeLib/HeatTracingTemperatureSetPoint"/>
    <Attribute Name="EquipmentPipeFlag" AttributeDataType="xs:boolean" RefAttributeType="ExtIEC62424AttributeLib/EquipmentPipeFlag"/>
    <Attribute Name="EquipmentID" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/EquipmentID"/>
    <Attribute Name="PipeID" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/PipeID"/>
    <Attribute Name="InsulationType" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/InsulationType"/>
    <Attribute Name="InsulationThickness" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/InsulationThickness"/>
    <Attribute Name="InternalUniqueID" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/InternalUniqueID"/>
    <Attribute Name="ShortDescription" AttributeDataType="xs:string" RefAttributeType="ExtIEC62424AttributeLib/ShortDescription"/>
  </RoleClass>
</RoleClassLib>
```

IEC

Figure D.6 – XML code for the example CAEX role library

D.4 Example CAEX definition of PCE relevant P&ID information

The following example illustrates how to store PCE relevant information within a CAEX InstanceHierarchy. Figure D.7 depicts a P&ID example with focus on elements 1) to 3).



Key

1 PCE request “080.5”

2 signal “SH”

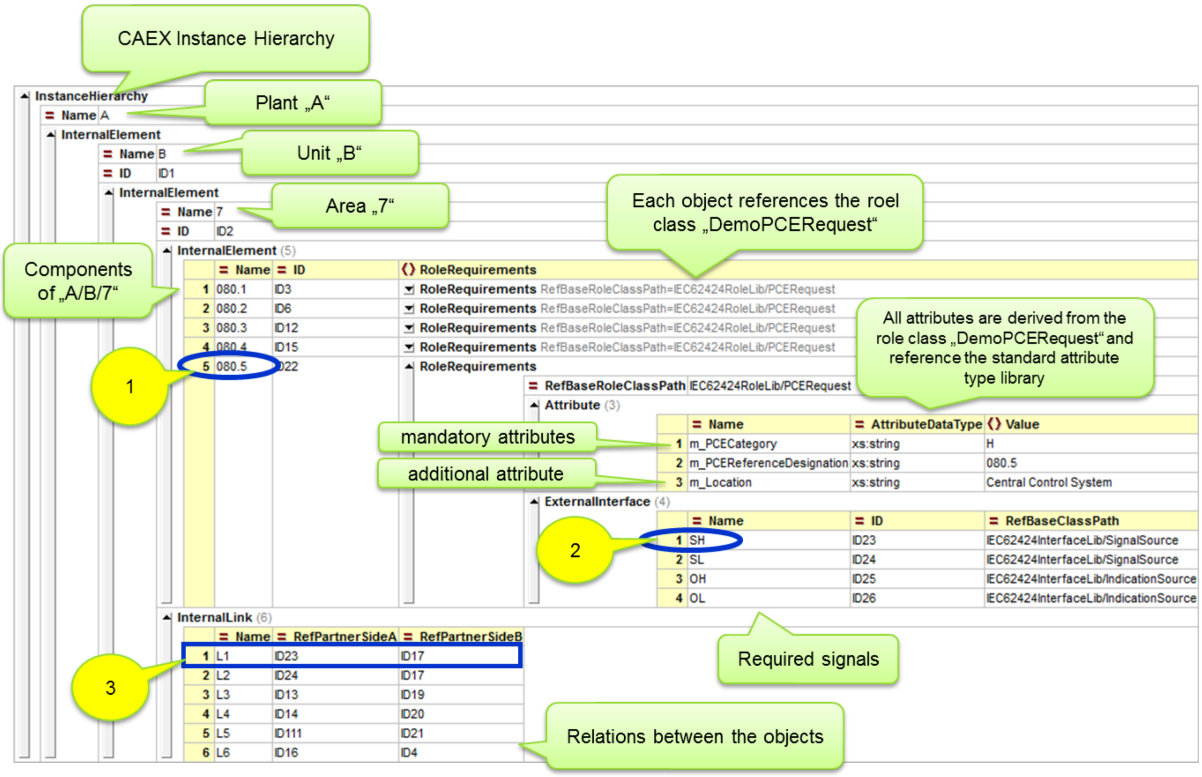
3 link between “080.4” and “OSH”

Figure D.7 – Example P&ID data to be mapped with CAEX

The whole system is described by means of the CAEX InternalElement “A/B/7”. Each PCE request, e.g. “080.5”, is described as a CAEX InternalElement which is associated to the example RoleClass “PCERequest”. Furthermore, this example specifies concrete values required for this PCE request. The PCE request might also be extended with optional attributes.

Figure D.8 depicts the corresponding CAEX XML structure. The InternalElements “B” and “7” are stored within the InstanceHierarchy “A”. The different PCE requests of this example are represented by means of nested InternalElements with each a RoleRequirements definition. The element “080.5” refers to the RoleClass “EC62424RoleLib/PCERequest”. Additionally, required additional signals are being defined. Finally, the relations between the objects are defined.

As described in Clause D.3, a RoleRequirement does not need to provide all attributes defined in the role class PCE request. According to A.2.8.5, unnecessary attributes are removed from the instances.



InstanceHierarchy

- Name A
- InternalElement
  - Name B
  - ID ID1
  - InternalElement
    - Name 7
    - ID ID2
    - InternalElement (5)

	Name	ID	RoleRequirements
1	080.1	ID3	RoleRequirements RefBaseRoleClassPath=IEC62424RoleLib/PCERequest
2	080.2	ID6	RoleRequirements RefBaseRoleClassPath=IEC62424RoleLib/PCERequest
3	080.3	ID12	RoleRequirements RefBaseRoleClassPath=IEC62424RoleLib/PCERequest
4	080.4	ID15	RoleRequirements RefBaseRoleClassPath=IEC62424RoleLib/PCERequest
5	080.5	ID22	RoleRequirements

RoleRequirements

- RefBaseRoleClassPath IEC62424RoleLib/PCERequest
- Attribute (3)

	Name	AttributeDataType	Value
1	m_PCECategory	xs:string	H
2	m_PCEReferenceDesignation	xs:string	080.5
3	m_Location	xs:string	Central Control System

- ExternalInterface (4)

	Name	ID	RefBaseClassPath
1	SH	ID23	IEC62424InterfaceLib/SignalSource
2	SL	ID24	IEC62424InterfaceLib/SignalSource
3	OH	ID25	IEC62424InterfaceLib/IndicationSource
4	OL	ID26	IEC62424InterfaceLib/IndicationSource

InternalLink (6)

	Name	RefPartnerSideA	RefPartnerSideB
1	L1	ID23	ID17
2	L2	ID24	ID17
3	L3	ID13	ID19
4	L4	ID14	ID20
5	L5	ID111	ID21
6	L6	ID16	ID4

CAEX Instance Hierarchy
Plant „A“
Unit „B“
Area „7“
Each object references the roel class „DemoPCERequest“
Components of „A/B/7“
1
All attributes are derived from the role class „DemoPCERequest“ and reference the standard attribute type library
mandatory attributes
additional attribute
2
Required signals
3
Relations between the objects
IEC

Key

1 PCE request “080.5”

2 signal “SH”

3 link between “080.4” and “OSH”

Figure D.8 – CAEX model of the example described in Figure D.7

The full XML text for the InstanceHierarchy example is shown in Figure D.9.

```
<InstanceHierarchy Name="A">
  <InternalElement Name="B" ID="ID1">
    <InternalElement Name="7" ID="ID2">
      <InternalElement Name="080.1" ID="ID3">
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest">
          <Attribute Name="m_PCECategory" AttributeDataType="xs:string">
            <Value>Y</Value>
          </Attribute>
          <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string">
            <Value>080.1</Value>
          </Attribute>
          <Attribute Name="M_Location" AttributeDataType="xs:string">
            <Value>Central Control System</Value>
          </Attribute>
          <Attribute Name="ProcessingFunction" AttributeDataType="xs:string">
            <Value>C</Value>
          </Attribute>
          <ExternalInterface Name="In000" ID="ID4" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
          <ExternalInterface Name="Y" ID="ID5" RefBaseClassPath="IEC62424InterfaceLib/FinalControllingEquipmentSource"/>
        </RoleRequirements>
      </InternalElement>
      <InternalElement Name="080.2" ID="ID6">
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest">
          <Attribute Name="m_PCECategory" AttributeDataType="xs:string">
            <Value>T</Value>
          </Attribute>
          <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string">
            <Value>080.2</Value>
          </Attribute>
          <Attribute Name="M_Location" AttributeDataType="xs:string">
            <Value>Central Control System</Value>
          </Attribute>
          <Attribute Name="ProcessingFunction" AttributeDataType="xs:string">
            <Value>IC</Value>
          </Attribute>
          <ExternalInterface Name="TIC" ID="ID7" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
          <ExternalInterface Name="AH" ID="ID8" RefBaseClassPath="IEC62424InterfaceLib/AlarmSource"/>
          <ExternalInterface Name="AL" ID="ID9" RefBaseClassPath="IEC62424InterfaceLib/AlarmSource"/>
          <ExternalInterface Name="In000" ID="ID10" RefBaseClassPath="IEC62424InterfaceLib/SensorSink"/>
          <ExternalInterface Name="I" ID="ID11" RefBaseClassPath="IEC62424InterfaceLib/IndicationSource"/>
          <ExternalInterface Name="SL" ID="ID111" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
        </RoleRequirements>
      </InternalElement>
      <InternalElement Name="080.3" ID="ID12">
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest">
          <Attribute Name="m_PCECategory" AttributeDataType="xs:string">
            <Value>H</Value>
          </Attribute>
          <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string">
            <Value>080.3</Value>
          </Attribute>
          <Attribute Name="M_Location" AttributeDataType="xs:string">
            <Value>Local Control Panel</Value>
          </Attribute>
          <ExternalInterface Name="SH" ID="ID13" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
          <ExternalInterface Name="SL" ID="ID14" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
        </RoleRequirements>
      </InternalElement>
      <InternalElement Name="080.4" ID="ID15">
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest">
          <Attribute Name="m_PCECategory" AttributeDataType="xs:string">
            <Value>U</Value>
          </Attribute>
          <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string">
            <Value>080.4</Value>
          </Attribute>
          <Attribute Name="M_Location" AttributeDataType="xs:string">
            <Value>Central Control System</Value>
          </Attribute>
          <Attribute Name="ProcessingFunction" AttributeDataType="xs:string">
            <Value>Y</Value>
          </Attribute>
          <ExternalInterface Name="Y" ID="ID16" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
          <ExternalInterface Name="In000" ID="ID17" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
          <ExternalInterface Name="In001" ID="ID18" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
          <ExternalInterface Name="In002" ID="ID19" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
          <ExternalInterface Name="In003" ID="ID20" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
          <ExternalInterface Name="In004" ID="ID21" RefBaseClassPath="IEC62424InterfaceLib/SignalSink"/>
        </RoleRequirements>
      </InternalElement>
      <InternalElement Name="080.5" ID="ID22">
        <RoleRequirements RefBaseRoleClassPath="IEC62424RoleLib/PCERequest">
          <Attribute Name="m_PCECategory" AttributeDataType="xs:string">
            <Value>H</Value>
          </Attribute>
          <Attribute Name="m_PCEReferenceDesignation" AttributeDataType="xs:string">
            <Value>080.5</Value>
          </Attribute>
          <Attribute Name="M_Location" AttributeDataType="xs:string">
            <Value>Central Control System</Value>
          </Attribute>
          <ExternalInterface Name="SH" ID="ID23" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
          <ExternalInterface Name="SL" ID="ID24" RefBaseClassPath="IEC62424InterfaceLib/SignalSource"/>
          <ExternalInterface Name="OH" ID="ID25" RefBaseClassPath="IEC62424InterfaceLib/IndicationSource"/>
          <ExternalInterface Name="OL" ID="ID26" RefBaseClassPath="IEC62424InterfaceLib/IndicationSource"/>
        </RoleRequirements>
      </InternalElement>
      <InternalLink Name="L1" RefPartnerSideA="ID23" RefPartnerSideB="ID17"/>
      <InternalLink Name="L2" RefPartnerSideA="ID24" RefPartnerSideB="ID17"/>
      <InternalLink Name="L3" RefPartnerSideA="ID13" RefPartnerSideB="ID19"/>
      <InternalLink Name="L4" RefPartnerSideA="ID14" RefPartnerSideB="ID20"/>
      <InternalLink Name="L5" RefPartnerSideA="ID111" RefPartnerSideB="ID21"/>
      <InternalLink Name="L6" RefPartnerSideA="ID16" RefPartnerSideB="ID4"/>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

Figure D.9 – XML code of the example described in Figure D.7

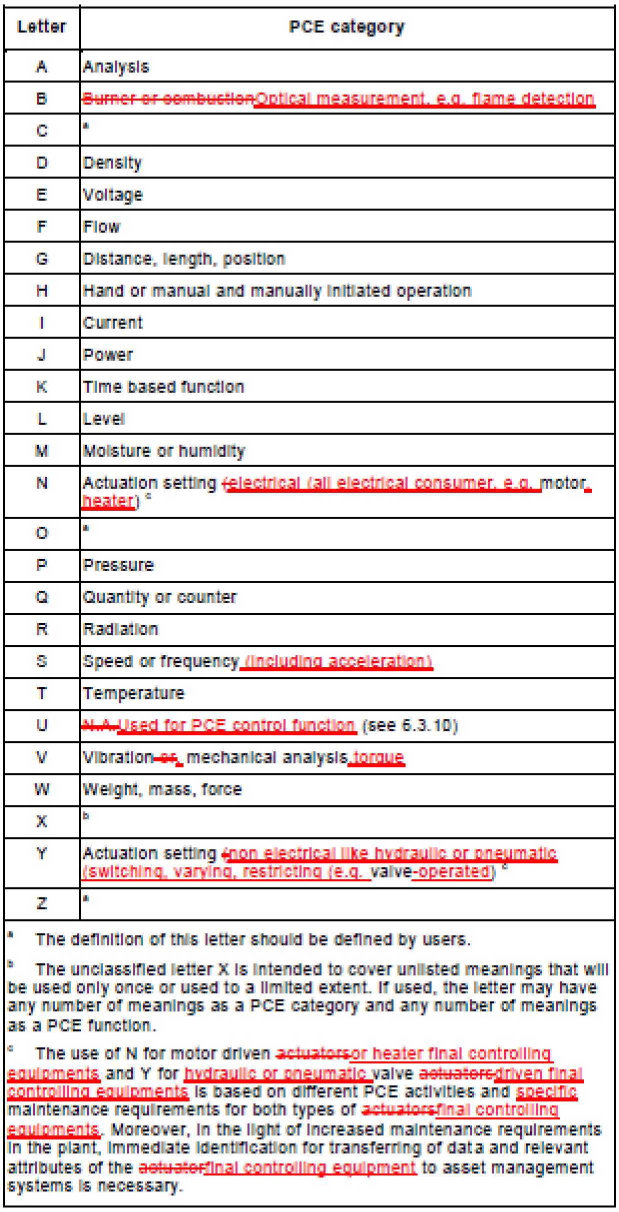
Annex E
(informative)

List of major changes and extensions of the second edition

This standard is fully compatible with IEC 62424:2008. The following list shows the major changes and extensions:

- updated and new definitions;
- identification replaced with reference designation;
- PCE categories and process functions updated.

Changes to Table 2 of IEC 62424:2008



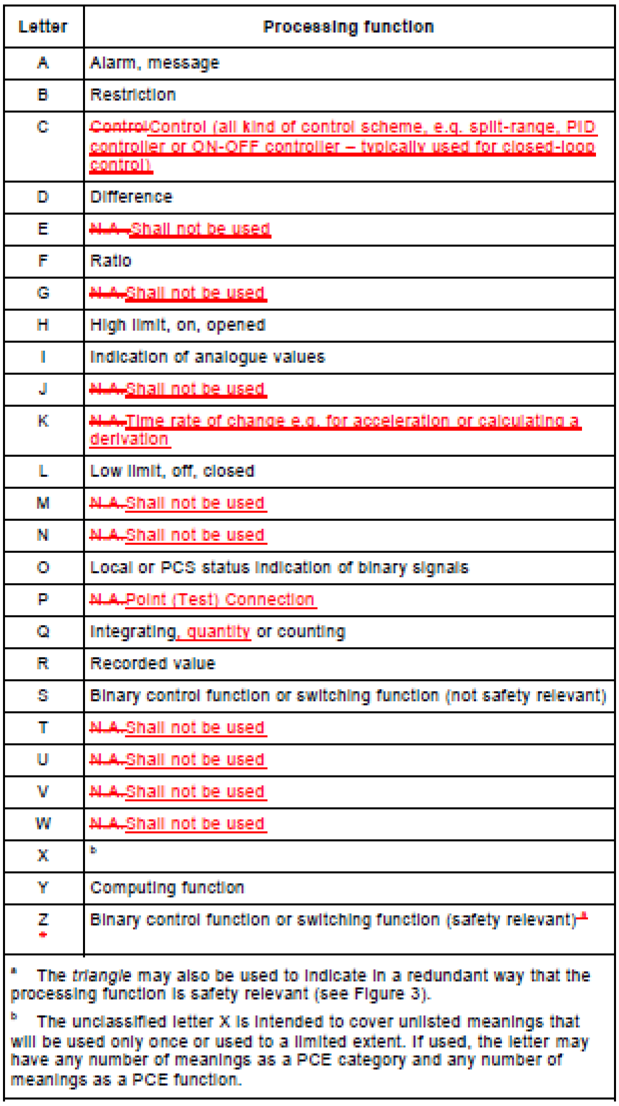
Letter	PCE category
A	Analysis
B	~~Burner or combustion~~Optical measurement, e.g. flame detection
C	[a]
D	Density
E	Voltage
F	Flow
G	Distance, length, position
H	Hand or manual and manually initiated operation
I	Current
J	Power
K	Time based function
L	Level
M	Moisture or humidity
N	Actuation setting ~~(~~electrical (all electrical consumer, e.g. motor, heater) [c]
O	[a]
P	Pressure
Q	Quantity or counter
R	Radiation
S	Speed or frequency (including acceleration)
T	Temperature
U	~~N.A.~~Used for PCE control function (see 6.3.10)
V	Vibration~~ or~~, mechanical analysis, torque
W	Weight, mass, force
X	[b]
Y	Actuation setting ~~(~~non electrical like hydraulic or pneumatic (switching, varying, restricting (e.g. valve-operated) [c]
Z	[a]

[a] The definition of this letter should be defined by users.

[b] 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.

[c] The use of N for motor driven ~~actuators~~or heater final controlling equipments and Y for hydraulic or pneumatic valve ~~actuators~~driven final controlling equipments is based on different PCE activities and specific maintenance requirements for both types of ~~actuators~~final controlling equipments. Moreover, in the light of increased maintenance requirements in the plant, immediate identification for transferring of data and relevant attributes of the ~~actuator~~final controlling equipment to asset management systems is necessary.

Changes to Table 3 of IEC 62424:2008



Letter	Processing function
A	Alarm, message
B	Restriction
C	~~Control~~Control (all kind of control scheme, e.g. split-range, PID controller or ON-OFF controller – typically used for closed-loop control)
D	Difference
E	~~N.A.~~ Shall not be used
F	Ratio
G	~~N.A.~~Shall not be used
H	High limit, on, opened
I	Indication of analogue values
J	~~N.A.~~Shall not be used
K	~~N.A.~~Time rate of change e.g. for acceleration or calculating a derivation
L	Low limit, off, closed
M	~~N.A.~~Shall not be used
N	~~N.A.~~Shall not be used
O	Local or PCS status indication of binary signals
P	~~N.A.~~Point (Test) Connection
Q	Integrating, quantity or counting
R	Recorded value
S	Binary control function or switching function (not safety relevant)
T	~~N.A.~~Shall not be used
U	~~N.A.~~Shall not be used
V	~~N.A.~~Shall not be used
W	~~N.A.~~Shall not be used
X	[b]
Y	Computing function
Z [a]	Binary control function or switching function (safety relevant)~~[a]~~

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

[b] 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.

Changes to Table 5 of IEC 62424:2008

Letter	Processing function
YS	non electrical actuating drive with open-loop-control function e.g. On/off valve
YC	non electrical actuating drive with closed-loop-control function e.g. Control valve
YCS	non electrical actuating drive with closed-loop-control function and open-loop-control (open/close) function e.g. Control valve with on/off function
YZ	non electrical actuating drive with open-loop-control function (safety related e.g. On/off valve (safety relevant)
YIC	non electrical actuating drive with closed-loop-control function and position indication e.g. Control valve ~~continuous~~with position indication
NS	electrical actuating drive with open-loop-control function e.g. On/off motor
NC	electrical actuating drive with closed-loop-control function e.g. Control motor

CAEX vers. 3.0

Introduction of:

- native multiple role support;
- nested interfaces;
- life cycle meta information, and
- a separate Attribute library;
- updated examples.

Updated electronic data model of the PCE request:

- new normative attribute library for basic PCE request attributes;
- new informative extended attribute library for further PCE request attributes;
- new informative electronic data model for the PCE request.

Bibliography

IEC 60050 (all parts), *International Electrotechnical Vocabulary* (available at http://www.electropedia.org)

IEC 60050-351, *International Electrotechnical Vocabulary – Part 351: Control Technology*

IEC 60848, *GRAFCET Specification language for sequential function charts*

IEC 61512-1, *Batch control – Part 1: Models and terminology*

IEC 61987-1, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 1: Measuring equipment with analogue and digital output*

ISO/IEC 9834-8, *Information technology – Procedures for the operation of object identifier registration authorities – Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

ISO 10628-1, *Diagrams for the chemical and petrochemical industry – Part 1: Specifications of diagrams*

ISO 10628-2, *Diagrams for the chemical and petrochemical industry – Part 2: Graphical symbols*

ISO 13628-6, *Petroleum and natural gas industries – Design and operation of subsea production systems – Part 6: Subsea production control systems*

ISO 13703*, Petroleum and natural gas industries – Design and installation of piping systems on offshore production platforms*

ISO TS 81346-3, *Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 3: Application rules for a reference designation system*

ISA 5.1:2009, *Instrumentation Symbols and Identification*, available at http://www.isa.org

____

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Copyright in BSI publications

All the content in BSI publications, including British Standards, is the property of and copyrighted by BSI or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use.

Save for the provisions below, you may not transfer, share or disseminate any portion of the standard to any other person. You may not adapt, distribute, commercially exploit, or publicly display the standard or any portion thereof in any manner whatsoever without BSI's prior written consent.

Storing and using standards

Standards purchased in soft copy format:

- A British Standard purchased in soft copy format is licensed to a sole named user for personal or internal company use only.
- The standard may be stored on more than 1 device provided that it is accessible by the sole named user only and that only 1 copy is accessed at any one time.
- A single paper copy may be printed for personal or internal company use only.

Standards purchased in hard copy format:

- A British Standard purchased in hard copy format is for personal or internal company use only.
- It may not be further reproduced – in any format – to create an additional copy. This includes scanning of the document.

If you need more than 1 copy of the document, or if you wish to share the document on an internal network, you can save money by choosing a subscription product (see 'Subscriptions').

Reproducing extracts

For permission to reproduce content from BSI publications contact the BSI Copyright & Licensing team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email subscriptions@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Useful Contacts

Customer Services
Tel: +44 345 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 345 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK