

61968-9 Message profiles for DLMS/COSEM

14th June 2010

Final

Study report by:

Don Taylor and Gyozo Kmethy

Gyozo Kmethy Secretary, IEC TC 13 and President, DLMS User Association

Don Taylor National expert, IEC TC13 WG15 and IEC TC57 WG14

CONTENTS

FOREWORD.....	3
INTRODUCTION.....	4
1 Scope	5
2 Conclusion	5
3 Recommendations	6
4 Practical use of the results.....	6
5 References.....	8
6 Terms, definitions and abbreviations	8
6.1 Abbreviations	8
7 DLMS/COSEM as extension of the CIM.....	9
8 Message profiles	13
8.1 GetRequest and GetResponse	13
8.2 SetRequest and SetResponse	18
8.3 ActionRequest and ActionResponse.....	21
8.4 EventNotification	24
8.5 XML Schema of GetRequest.....	26
Figure 1 – UML representation of DLMS_COSEM Object as extension to CIM.....	9
Figure 2 – XML Schema for GetRequest.....	13
Figure 3 – XML Schema for GetResponse	15
Figure 4 – Sequence of message exchange for GetRequest and GetResponse	16
Figure 5 – XML Schema for SetRequest.....	18
Figure 6 – XML Schema for SetResponse	19
Figure 7 – Sequence of message exchange for SetRequest and SetResponse	20
Figure 8 – XML Schema for ActionRequest.....	21
Figure 9 – XML Schema for ActionResponse	22
Figure 10 – Sequence of message exchange for ActionRequest and ActionResponse	23
Figure 11 – XML Schema for EventNotification	24
Figure 12 – Sequence of message exchange for EventNotification.....	25
Table 1 – Message and attribute truth table mapping	11

FOREWORD

- 1) IEC TC57 deals with the generation, transmission and distribution standards related to interfaces for message exchange between systems.
- 2) These interfaces are derived from UML information models created in the CIM that represents a common set of information found in the utilities business application domain.
- 3) TC57 WG14 deals with those interfaces for systems related to distribution networks and publish these in the IEC 61968 series of standards.
- 4) The scope of these interfaces is limited to facilitate communication/integration between ERP systems and the Metering System (MS) head-end and does not include the downstream protocols from the MS to the meter end device.
- 5) IEC TC13 deals with metering, tariff and load control and in particular data models and communications protocols used for data exchange between the MS head-end down to the meter end device.
- 6) The DLMS/COSEM specification, part of the IEC 62056 suite specifies the COSEM data model specifying the COSEM objects. The COSEM application layer specifies the DLMS messaging services to access the COSEM model. The resulting DLMS application messages are transported through any suitable set of lower layers. IEC TC 13 WG 14 works with the DLMS UA to promote, support, develop and maintain DLMS/COSEM. IEC 62056 has also been adopted by CEN TC 294 for data exchange with meters other than electricity (gas, water, heat).
- 7) TC57 WG14 approached TC13 with a request to investigate the feasibility of modelling end devices using the DLMS/COSEM object modelling the CIM with the aim to create 61968 compliant message profiles that will enable an ERP system to send messages via the MS head-end to COSEM Objects residing in end devices installed in the metering system.

NOTE: It is important to note that TC57 message profiles are not concerned with the protocols used between the metering system head-end and the downstream meters or devices. However, the contextual relationships between information elements in the metering domain are important to be known by the message profiles and therefore these elements need to be accurately modelled in the CIM.

INTRODUCTION

IEC TC57 standards relevant to this investigation:

IEC 62361-100 Naming and Design Rules covers the rules for XML structure of message profiles derived from the UML in the CIM.

IEC 61968-11 covers the CIM for Distribution in UML notation.

IEC 61968-9 covers interface message profiles for meter reading and load control.

TC13 standards relevant to this investigation:

IEC 62056 covers meter data exchange including the DLMS/COSEM suite.

See clause 5 for the list of references

DLMS User Association specifications relevant to this investigation:

The Blue Book covers the COSEM interface class specifications and the OBIS data identification system.

The Green Book specifies the protocol layers and the communication profiles.

The DLMS UA regularly offers the latest editions of the Blue Book and the Green Book for international standardization to IEC and CEN.

Other tools relevant to this investigation:

Enterprise Architect (EA) is a software tool that allows the operator to create UML notation models and sequence diagrams.

CIMTool is a software tool that imports the UML in XMI format and then enables the operator to create XML Schema (XSD) format documents for each message profile required.

XMLspy is a software tool that allows the viewing of XSD files in graphical notation.

1 Scope

The COSEM data model models physical devices as a set of one or more COSEM logical devices. One logical device, the Management Logical device is mandatory. Each COSEM logical device contains two or more COSEM objects. The two mandatory objects are the Current Association object controlling the access to the resources and the Logical Name object, uniquely identifying the Logical Device.

NOTE In the Management Logical Device, the Logical Device Name object may be replaced by the SAP assignment object, holding the names and the SAPs of each logical device present.

A UML model of a COSEM physical device with logical device was created as an extension to the CIM, based on `iec61970cim14v12_iec61968cim10v28_combined_DLMS(1.02).eap`.

NOTE that this model is a first-approach only and still needs some further work for optimisation in the final version.

The UML was exported to an XMI format file and then imported to CIMTool as a project.

The following message profiles were then created in CIMTool:

- ActionRequest.xsd
- ActionResponse.xsd
- EventNotification.xsd
- GetRequest.xsd
- GetResponse.xsd
- SetRequest.xsd
- SetResponse.xsd

NOTE also the following:

- 1) CIMTool creates message profiles in conformance with the Naming and Design Rules.
- 2) The XSD message profile, so produced, maintains the attribute definitions and the class relationships as modelled in the CIM.

XMLspy was then used to view these XSD files and displayed in graphical diagram notation for ease of interpretation by the reader. These are detailed in the below text for further reading and interest.

EA was used to create use case sequence diagrams to show the interaction and message interchange between the 3 actors: ERP_System, MS_System and Meter_EndDevice.

2 Conclusion

The COSEM model is a relatively simple structure as captured in UML notation in the CIM.

It is also relatively simple to create DLMS/COSEM message profiles in XSD that covers almost all the functionality required for an ERP system to talk to DLMS/COSEM compliant end devices via the MS head-end.

- a) ERP can selectively “write” values down to individual attribute level of COSEM Objects in Logical Devices located in physical devices.
- b) ERP can selectively “read” values down to individual attribute level of COSEM Objects in Logical Logical Devices located in physical devices.

- c) ERP can “invoke” actions down to individual method level of COSEM Objects in Logical Devices located in physical devices.
- d) ERP can receive unsolicited event notifications from individual COSEM Objects in Logical Devices located in physical devices.

The shortfall of messages not yet covered can be completed on a next round of refinements.

3 Recommendations

This report is to be circulated to the following parties:

- Team of experts that participated on the teleconference call in November 2009, from whence came the initial enquiry for this investigation;
- IEC TC57 WG14;
- IEC TC13 WG14;
- DLMS User Association management committee and core team;
- Mr. Eric Lambert at ERDF, who expressed an independent interest in this work

A decision needs to be made whether it is desirable to progress this work further into an IEC standard under the IEC 61968 suite.

If the answer is affirmative, then the content of this report could be submitted to TC57 WG14 as a work item for inclusion into the current work program for the revision of IEC 61968-9 Ed2. It will thus be an extension of IEC 61968-9 that will enable ERP systems to communicate to the metering head-end when the meters are DLMS/COSEM compliant.

Such a submission can possibly be done via a D-Liaison between DLMS User Association and TC57 WG14 specifically set up for this purpose.

Don Taylor is willing to take the lead of a sub-team in the current Part-9 team in TC57 WG14 to deal with the DLMS-COSEM work and Gyoza Kmety is willing to play the lead role from the DLMS User Association side with a team of DLMS experts supporting him.

4 Practical use of the results

The standards produced by TC57 WG14 are message profiles to be used for integration of systems at the ERP level related to the utility business domain in the distribution network, extending to the meters and other end devices.

The message exchange is not directly between the ERP system and the end device, but always indirectly via the metering system head-end, by means of a request/response message exchange protocol and is always asynchronous (no system state).

The following example may illustrate this more clearly:

Let's say the customer information system (CIS) wants to obtain a certain set of readings from a particular known group of meters.

CIS has a copy of the DLMS/COSEM specification and thus it knows which logical device, which object and which attributes it wants the values from, but it cannot directly speak to the meters. So, it plugs into the GetRequest message (the one we created) the meter serial numbers of interest, the logical device name for each meter, the objects of interest for each logical device and the attributes of interest for each object. It now sends this message to the metering system head-end (HE) with the request to please obtain the information specified in the message for it from the meters specified in the message. HE then goes away and in its

own time performs a DLMS-protocol session with the meters of interest, reads the values and when it is ready, puts these into the GetResponse message, which it sends back to CIS in reply to its original request. No session state is held between CIS and HE or CIS and meter and the exchange is totally asynchronous. Thus the CIS never speaks directly to the meter, but always via the HE.

It may rightfully be asked: what is the specific advantage of the particular message profiles that we have created for this purpose? It may also be argued that CIS could have compiled any message in any format agreed between CIS and HE to achieve this. It is true, but the difference is a) we have defined a standard message structure and b) the message structure maintains the contextual relationship between physical device, logical device, object, attribute and method exactly as is defined in the DLMS/COSEM standards.

This is an example of ERP to HE integration, but the same is true for peer-peer integration. For example MDM or MDC may also send the same message request to HE and obtain the same information in the same standard format.

This is ultimately the purpose of the 61968 interface standards - to provide a standard set of message profiles for system integration at the back-end. I would also refer the reader to 61968-1-1 Enterprise Service Bus, which is one such integration bus defined exactly for this purpose and using the 61968 message profiles.

Now, 61968-9 specifically deals with message profiles relating to meter reading and control, which has a direct interest to TC13. However, there are other parts of 61968, each dealing with a different part of the business application domain in the utility distribution network (see table below).

Below is a current list of work items in TC57 WG14 in the IEC 61968 suite. Parts 3 to 10 and 12 to 14 are message profiles for those particular business domains. Part 11 contains the UML model of the CIM from which the message profiles are derived (in the same way as we have done for this DLMS/COSEM investigation) and that is also where the UML part in our report will end up, while the message profiles will end up in Part 9.

Part	Title	Stage
61968-1	Interface Architecture	IS
61968-1-1	Enterprise Service Bus	WD
61968-1-2	Webservices Implementation Profile	WD
61968-2	Glossary	TR
61968-3	Network Operations	IS
61968-4	Records & Asset Management	FDIS
61968-5	Operational Planning	WD
61968-6	Maintenance & Construction	WD
61968-7	Network Extension	WD
61968-8	Customer Support	WD
61968-9	Meter Reading & Control	ED1
61968-10	Distributed Energy Resources	WD
61968-11	CIM for Distribution	CDV
61968-12	Compliance & Interoperability	WD
61968-13	Common Distribution Power System Model	FDIS
61968-14-1	MultiSpeak mapping to 61968	WD
61968-14-2	MultiSpeak profile	WD

5 References

IEC 61968-11, SYSTEM INTERFACES FOR DISTRIBUTION MANAGEMENT – Part 11: Common information model (CIM) extensions for distribution

IEC 62056-53 Ed 2.0:2006, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application layer*

IEC 62056-61 Ed 2.0:2006, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: OBIS Object identification system*

IEC 62056-62 Ed 2.0:2006, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface classes*

DLMS UA 1000-1 Ed. 9.0:2009, *COSEM Interface Classes and the OBIS Identification System (the "Blue Book")*

DLMS UA 1000-1 Ed.7.0:2009, *DLMS/COSEM Architecture and Protocols (the "Green Book")*

NOTE Lower layer standards from the IEC 62056 DLMS/COSEM suite are not listed here.

6 Terms, definitions and abbreviations

6.1 Abbreviations

DLMS	Device Language Message Specification
COSEM	Companion Specification for Energy Metering
CIM	Common Information Model
UML	Unified Modelling Language
XML	Extensible Markup Language
XSD	XML Schema Document
IEC	International Electrotechnical Commission
TC57	Technical Committee 57
TC13	Technical Committee 13
TC57 WG14	TC 57 Working Group 14
TC13 WG15	TC13 Working Group 15
TC13 WG14	TC13 Working Group 14
ERP	Enterprise Resource Planning
MS	Metering System
EA	Enterprise Architect
XMI	XML Metadata Interchange
LDN	Logical device Name

7 DLMS/COSEM as extension of the CIM

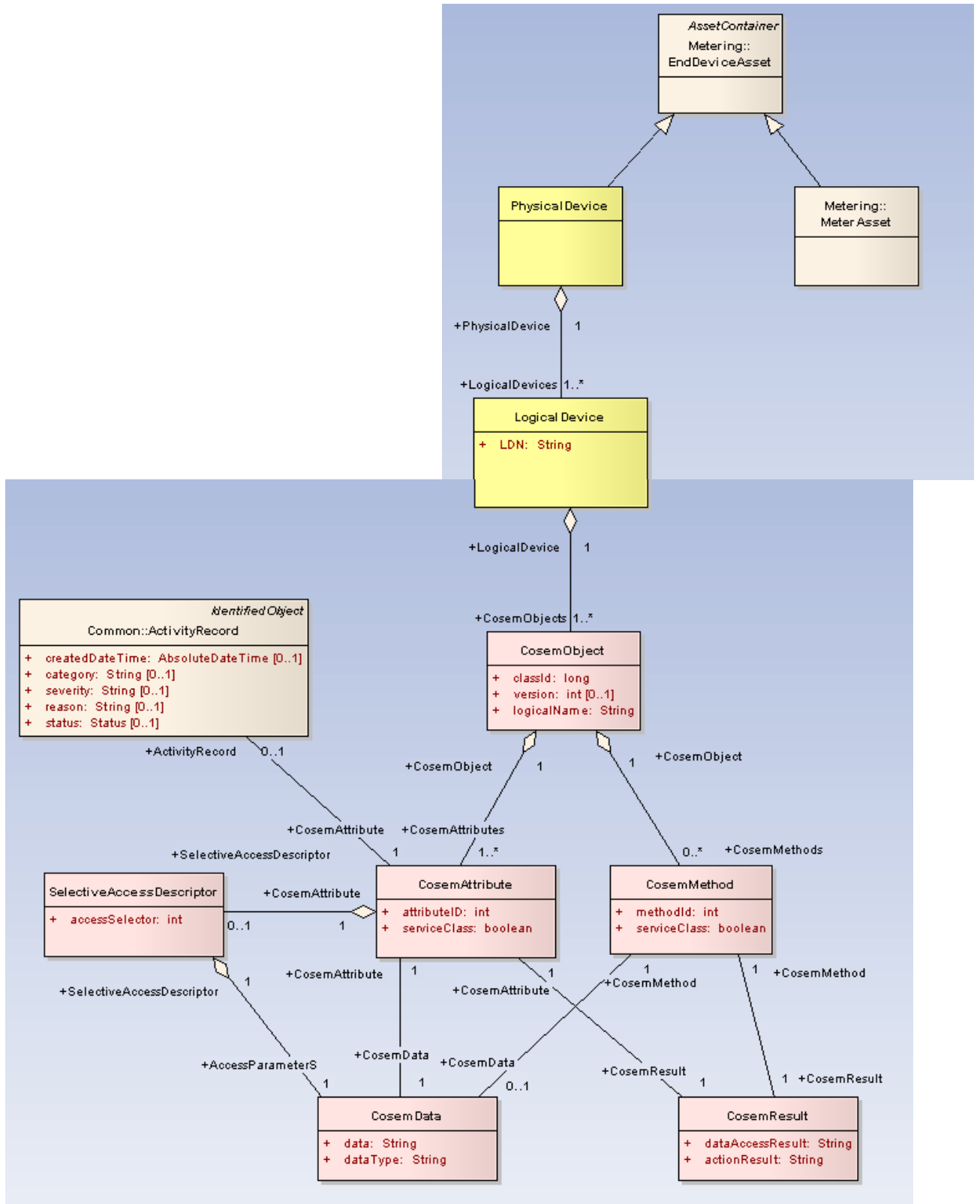


Figure 1 – UML representation of DLMS_COSEM Object as extension to CIM

How to read Figure 1:

EndDeviceAsset is a child class in the inheritance hierarchy of asset classes modelled in the CIM. End devices are conceptually those device physically located at or near to (but associated with) the ServiceDeliveryPoint at the end of the distribution network. The ServiceDeliveryPoint is the point on the network where ownership and risk transfers from the service supplier to the customer and is practically the point where a dwelling is connected to the network.

In this context a billing meter is such an EndDeviceAsset and is typically connected to the ServiceDelivery point. Thus MeterAsset is a child class of (or a “kind of”) EndDeviceAsset

In the DLMS/COSEM context we model the PhysicalDevice as a “kind of” EndDeviceAsset (rather than a MeterAsset) because COSEM objects may locate in devices other than meters.

A PhysicalDevice is an aggregation of one or more logical devices. Each LogicalDevice is identified by a unique logical device name (LDN).

NOTE The required functionality at the customer’s premises may be implemented in one or more physical devices.

A LogicalDevice is an aggregation of one or more COSEM objects. Each CosemObject has a classId – defining the number and kind of attributes and methods - a version (of the class) and a logicalName. The version of the interface class need to be known by the HE. It can be retrieved from the meter, by reading the appropriate attributes of the Association objects. Otherwise, the version information is not part of the data exchanges between the HE and the meter.

Each COSEM object is an aggregation of one or more attributes. Each Attribute is identified by its attributeId. Some COSEM object attributes may be accessed selectively, as specified by the SelectiveAccessDescriptor. The serviceClass refers to the DLMS service used to access the attributes, it may be confirmed (response is expected) or unconfirmed (response is not expected).

Each COSEM objects may have zero or more methods. The service class refers to the DLMS service used to access the attributes, it may be confirmed or unconfirmed.

Access to COSEM object attributes – for reading or writing – results in data or data access result sent by the meter.

Access to COSEM object methods may include sending data from the HE to the meter as method invocation parameters. The response may include data, actionResult or both.

Note that this model is not optimised for final inclusion into the CIM, but primarily serves to demonstrate the required functionality.

For example: CosemData and CosemResult could be modelled as compound data types that would improve the model.

Table 1 shows the attributes and their use by each message in accordance with which rules the message profiles were created.

Note that the presence of the various elements is shown for the data exchange between the ERP and the HE. The presence of these elements in the data exchange between the HE and the end devices is specified in the DLMS/COSEM standards.

M = mandatory

O = optional

Blank = not used

Table 1 – Message and attribute truth table mapping

Message	GetRequest	GetResponse	SetRequest	SetResponse	ActionRequest	ActionResponse	EventNotification
LDN	M	M	M	M	M	M	M
classID	M	M	M	M	M	M	M
logicalName	M	M	M	M	M	M	M
version	O	O	O	O	O	O	O
attributeID	M	M	M	M			M
serviceClass	M		M				
accessSelector	O		O				
methodID					M	M	
serviceClass					M		

CosemData.data		O	M		O	O	M
CosemData.dataType			M				M
AccessParameter.data	M		M				
AccessParamete.dataType	M		M				
dataAccessResult		O		M		O	
actionResult						M	
ActivityRecord.mRID							O
ActivityRecord.category							O
ActivityRecord.createdDateTime							O
ActivityRecord.description							O
ActivityRecord.localName							O
ActivityRecord.name							O
ActivityRecord.reason							O
ActivityRecord.severity							O
PhysicalDevice.mRID	O	O	O	O	O	O	O
PhysicalDevice.description	O	O	O	O	O	O	O
PhysicalDevice.localName	O	O	O	O	O	O	O
PhysicalDevice.name	O	O	O	O	O	O	O
PhysicalDevice.serialNumber	O	O	O	O	O	O	O

8 Message profiles

8.1 GetRequest and GetResponse

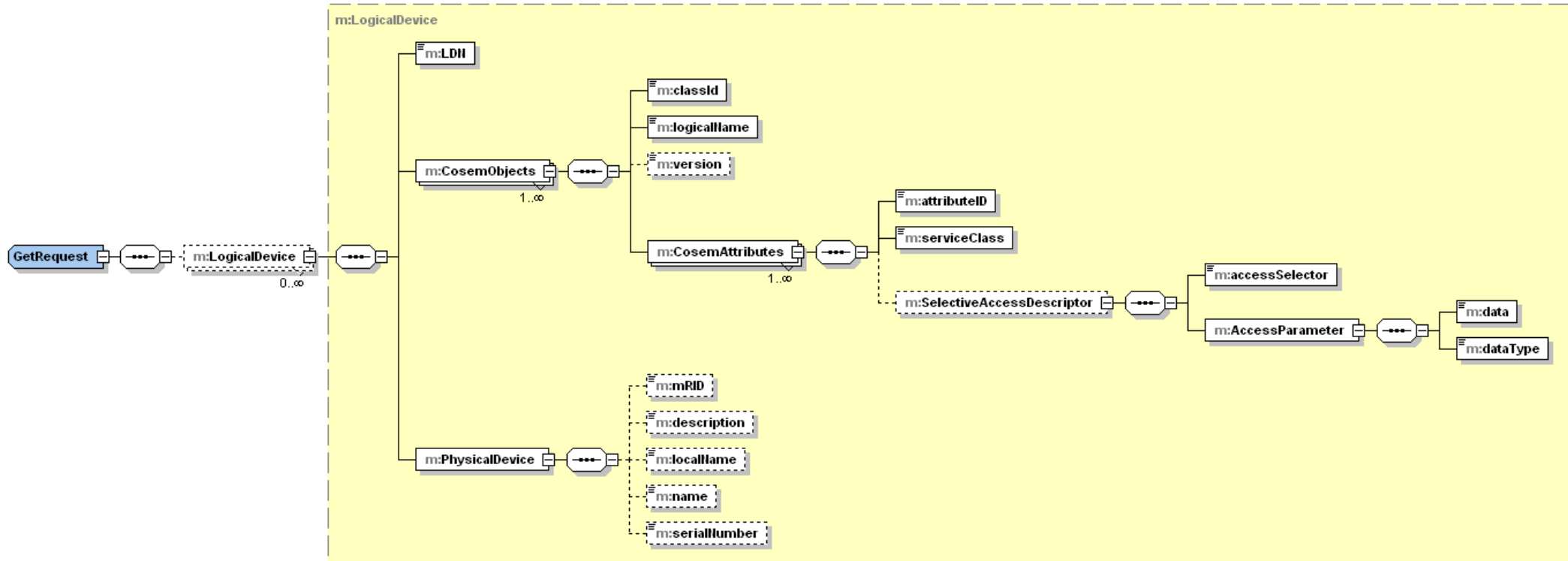


Figure 2 – XML Schema for GetRequest

How to read Figure 2

The solid boxes are mandatory data elements contained in the message payload.

The dotted line boxes are optional data elements contained in the message payload.

- GetRequest is a document referencing one or more logical devices in a physical device.
- Each physical device contains one or more Logical Devices, identified by their unique Logical Device Name. This LDN is held by a Logical Device Name object. Note, that in the Management Logical Device, the Logical Device Name object may be replaced by the SAP assignment object, holding the names and the SAPs of each logical device present.
- Each LogicalDevice contains one or more CosemObjects.
- Each CosemObject is identified by its classID, logicalName, and version and has one or more CosemAttributes. Attribute 1 holds the LogicalName.
- Each CosemAttribute reference contains the class_id, and the LogicalName of the object, the attributeID and optionally a SelectiveAccessDescriptor (optional). The serviceClass specifies if a response is expected or not.
- The SelectiveAccessDescriptor (if present) contains one accessSelector and one set of AccessParameters
- The set of AccessParameters contains one data element and one dataType element
- The PhysicalDevice contains one mRID, one description, one localName, one name and one serialNumber (these are all optional)

Note that the absence of a cardinality (multiplicity) number below an element implies a value of 1.

A similar interpretation is used for the other figures in this document.

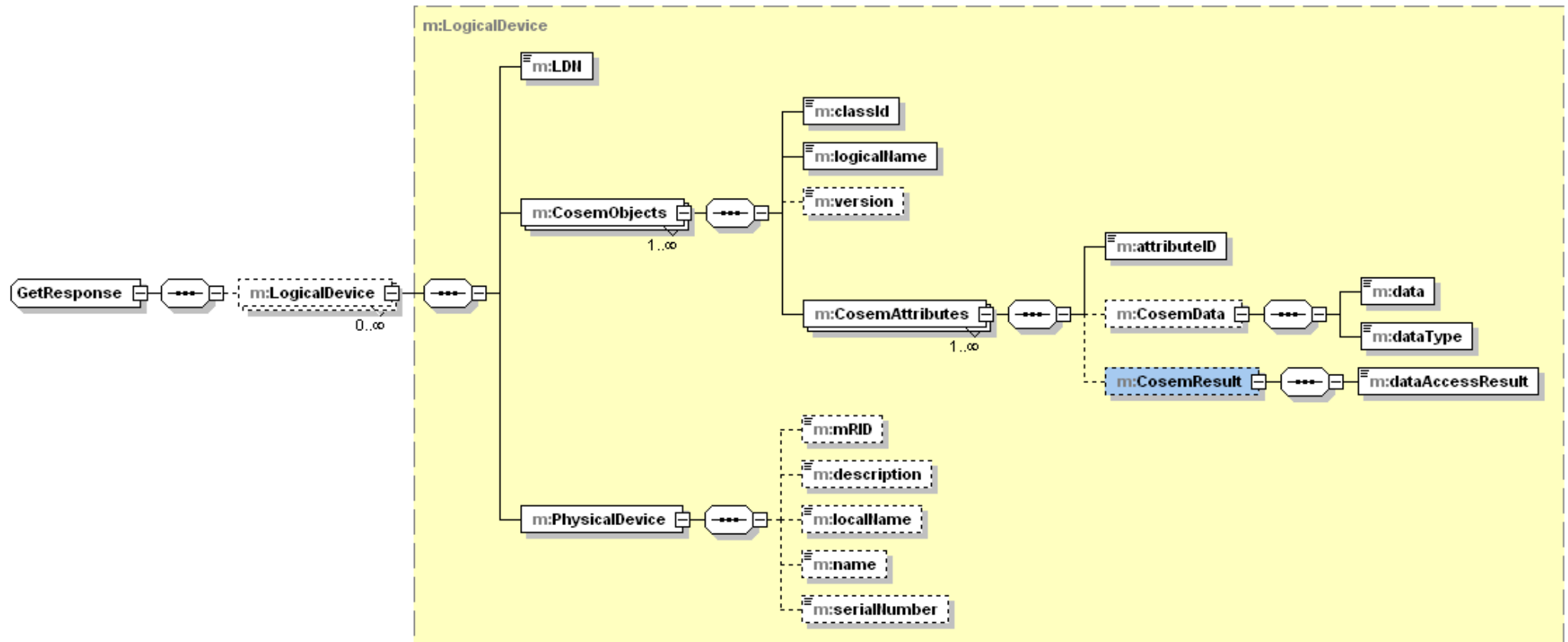


Figure 3 – XML Schema for GetResponse

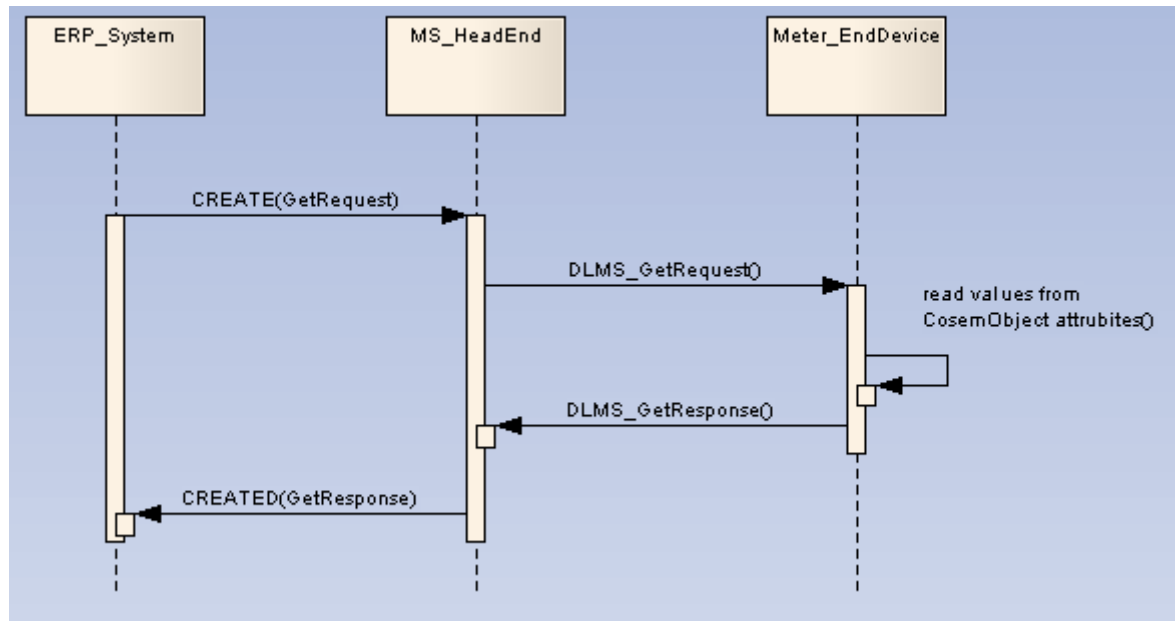


Figure 4 – Sequence of message exchange for GetRequest and GetResponse

How to read Figure 4:

ERP_System sends a GetRequest message to MS_HeadEnd in the prescribed format given by IEC 61968 series.

It uses the verb CREATE (as defined in 61968 series) to tell MS_HeadEnd system to create a new set of data and to obtain the values from the PhysicalDevice, LogicalDevice and Attributes specified in the message profile GetRequest.

MS_HeadEnd translates the received message into DLMS notation and sends the translated message on to the specified Meter_EndDevice.

MS_EndDevice receives the message and performs the required operation to read the specified values in the meter.

Meter_EndDevice sends the values back to MS-HeadEnd in DLMS notation.

MS-HeadEnd translates the received message back into the prescribed format given by IEC 61968 series and sends it on to the ERP_System as a GetResponse message.

It uses the verb CREATED (as defined in 61968 series) to tell ERP_System that this is the reply to the original request.

The message exchange process is now complete.

8.2 SetRequest and SetResponse

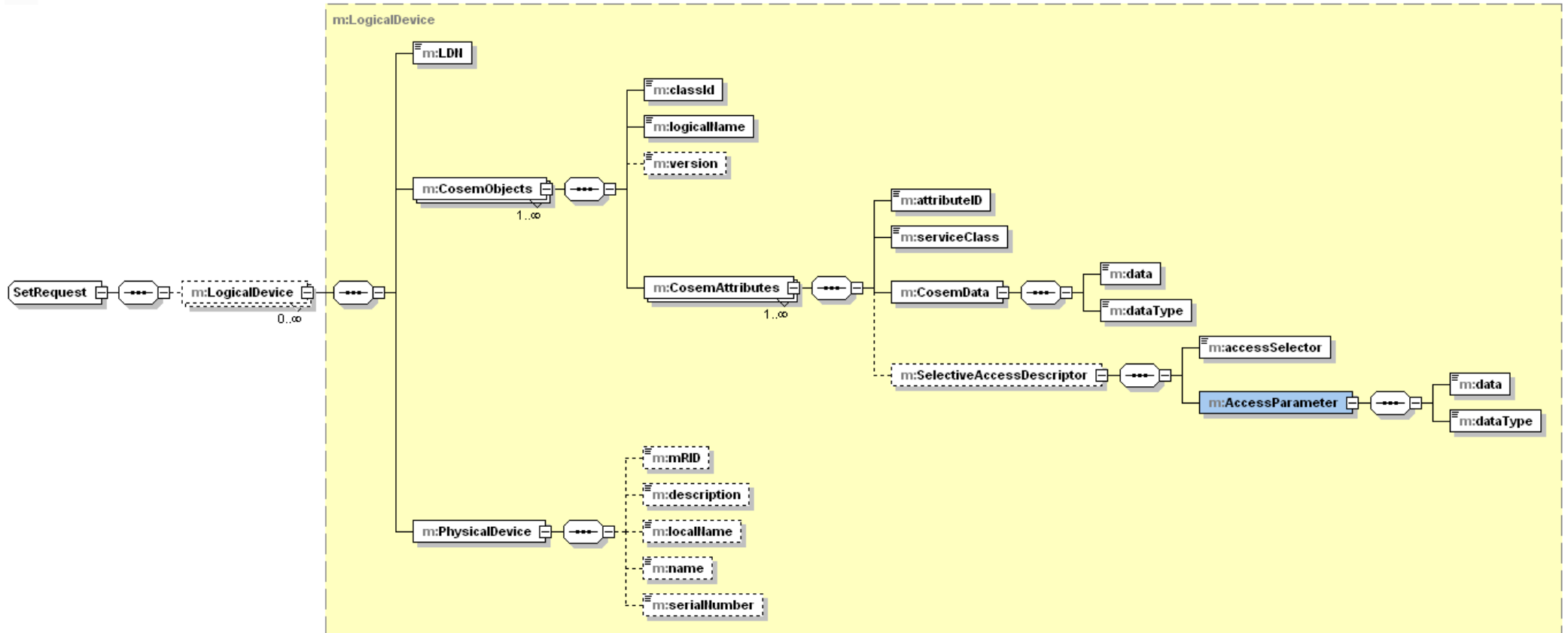


Figure 5 – XML Schema for SetRequest

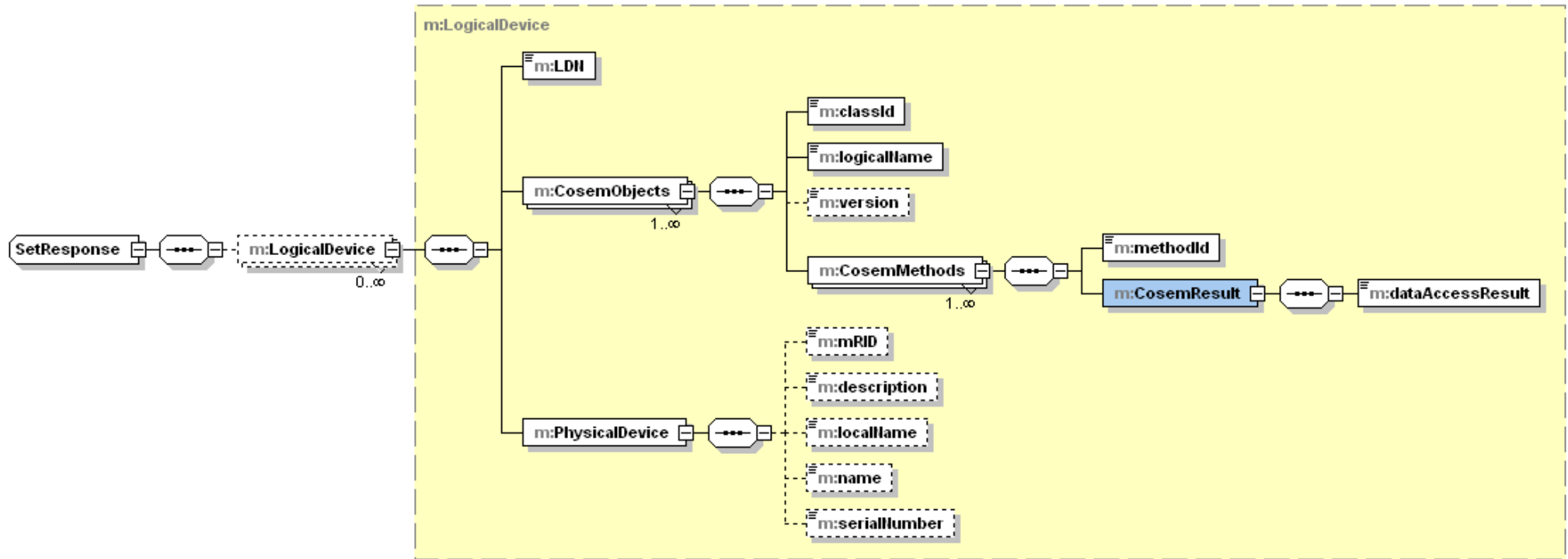


Figure 6 – XML Schema for SetResponse

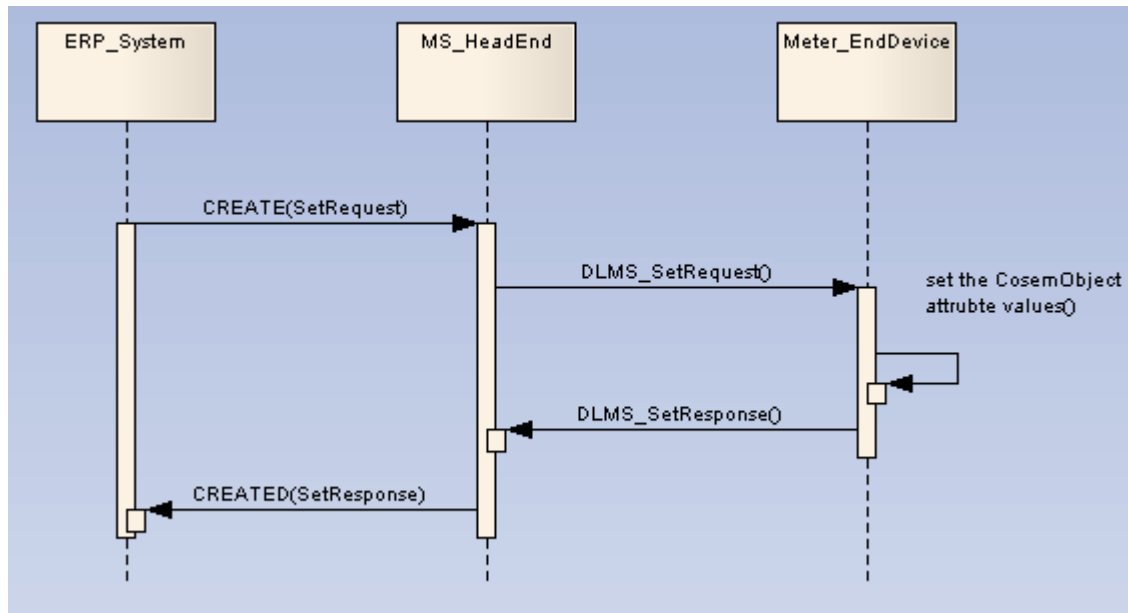


Figure 7 – Sequence of message exchange for SetRequest and SetResponse

8.3 ActionRequest and ActionResponse

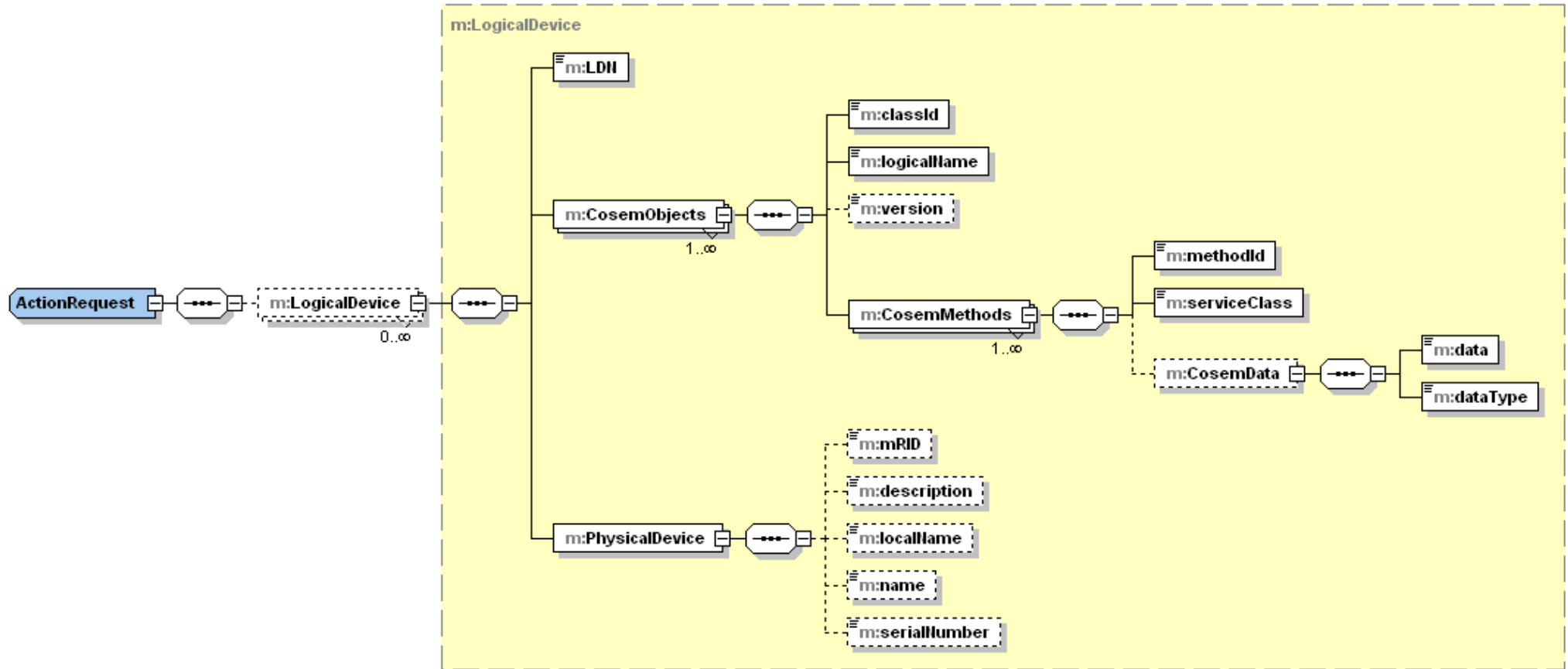


Figure 8 – XML Schema for ActionRequest

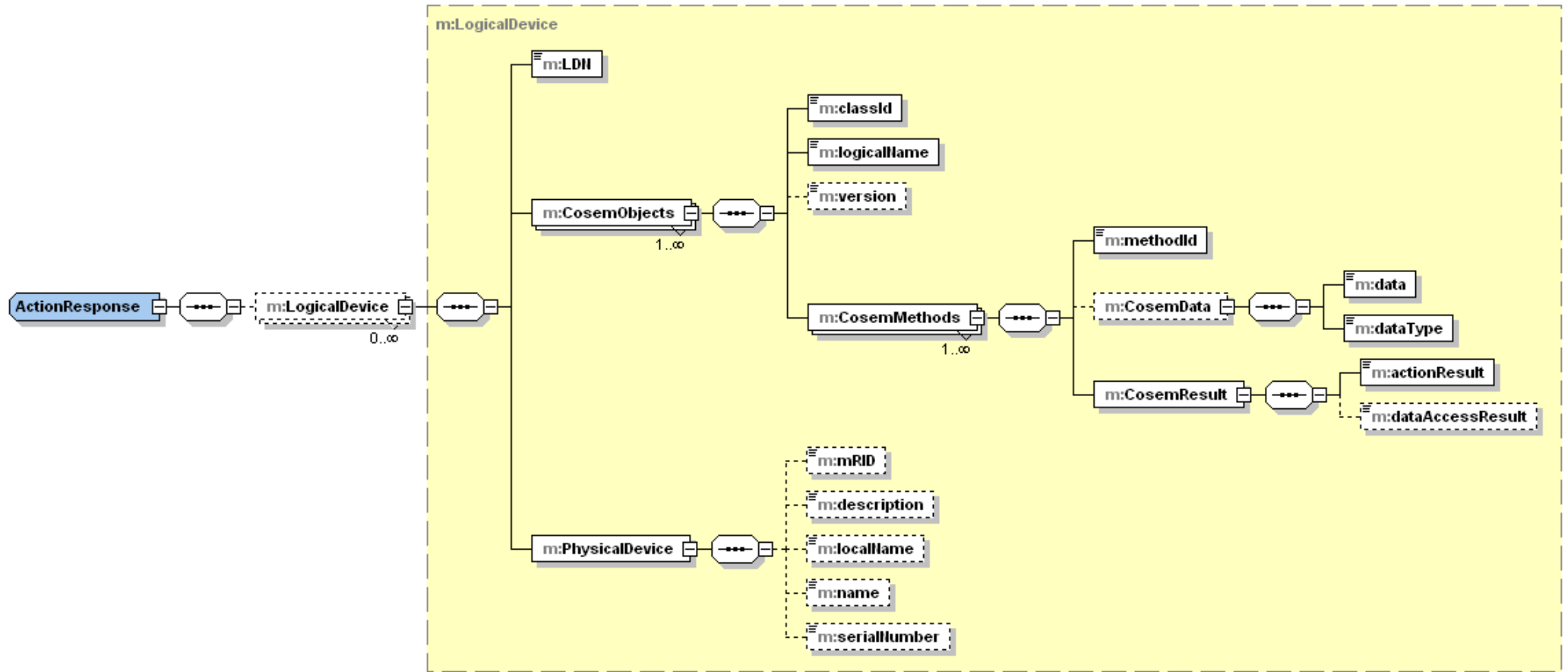


Figure 9 – XML Schema for ActionResponse

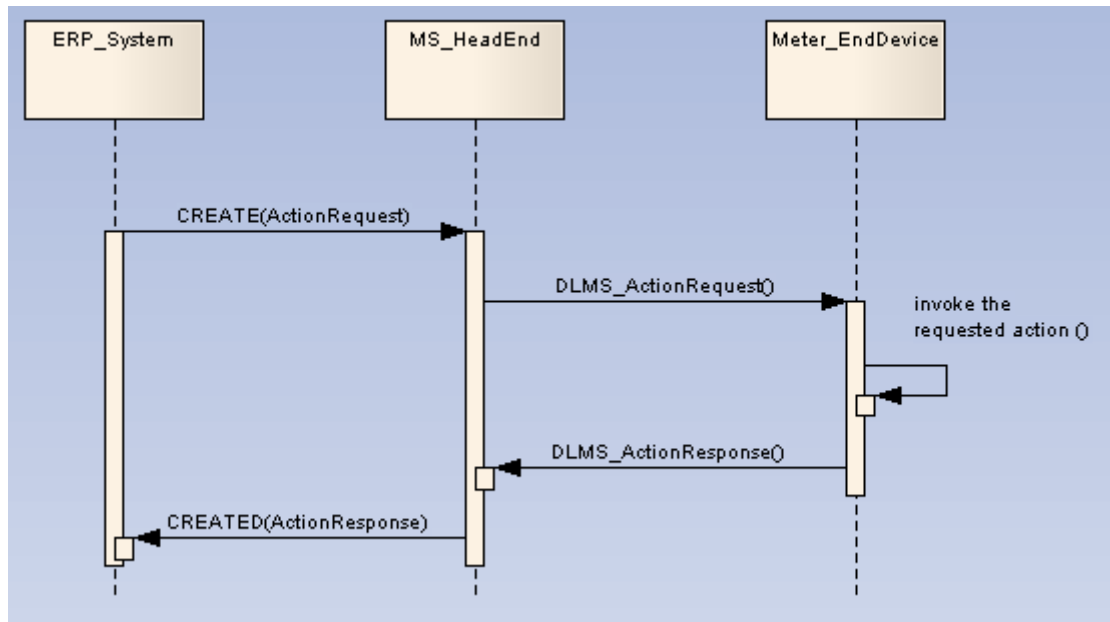


Figure 10 – Sequence of message exchange for ActionRequest and ActionResponse

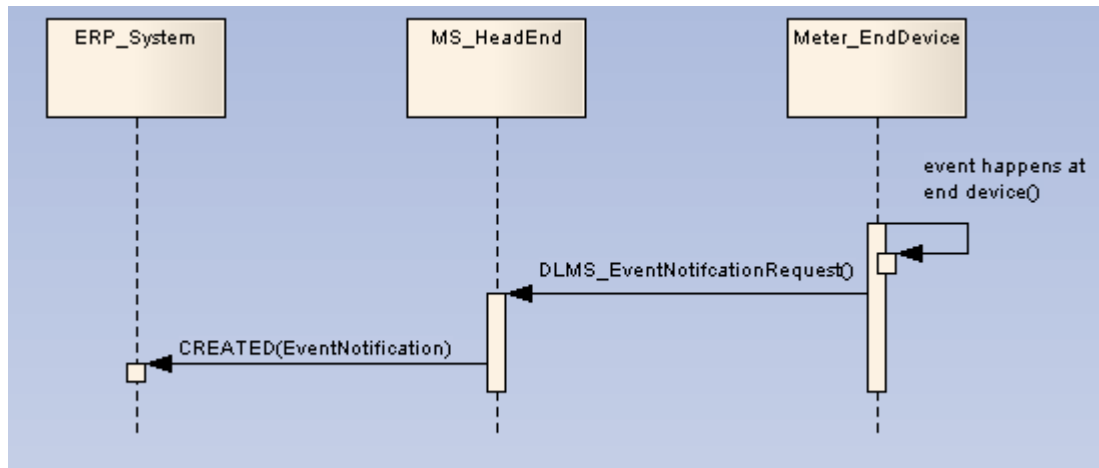


Figure 12 – Sequence of message exchange for EventNotification

8.5 XML Schema of GetRequest

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:a="http://langdale.com.au/2005/Message#"
  xmlns:sawSDL="http://www.w3.org/ns/sawSDL" xmlns="http://langdale.com.au/2005/Message#"
  xmlns:m="http://iec.ch/TC57/2007/GetRequest#" targetNamespace="http://iec.ch/TC57/2007/GetRequest#"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
- <xs:annotation>
<xs:documentation>Generated by CIMTool Beta , see http://cimtool.org</xs:documentation>
  </xs:annotation>
  <xs:element name="GetRequest" type="m:GetRequest" />
- <xs:complexType name="GetRequest">
- <xs:sequence>
  <xs:element name="LogicalDevice" type="m:LogicalDevice" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <xs:complexType name="LogicalDevice" sawSDL:modelReference="http://iec.ch/TC57/CIM-generic#LogicalDevice">
- <xs:annotation>
<xs:documentation>at least one Association LN or Association SN object must be present</xs:documentation>
  </xs:annotation>
- <xs:sequence>
- <xs:element name="LDN" type="xs:string" sawSDL:modelReference="http://iec.ch/TC57/CIM-generic#LogicalDevice.LDN">
<xs:annotation />
  </xs:element>
- <xs:element name="CosemObjects" maxOccurs="unbounded" sawSDL:modelReference="http://iec.ch/TC57/CIM-
  generic#LogicalDevice.CosemObjects">
<xs:annotation />
- <xs:complexType sawSDL:modelReference="http://iec.ch/TC57/CIM-generic#CosemObject">
- <xs:sequence>
- <xs:element name="classId" type="xs:long" sawSDL:modelReference="http://iec.ch/TC57/CIM-
  generic#CosemObject.classId">
<xs:annotation />

```

```

</xs:element>
-   <xs:element      name="logicalName"      type="xs:string"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemObject.logicalName">
<xs:annotation />
</xs:element>
-   <xs:element      name="version"      type="xs:int"      minOccurs="0"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemObject.version">
- <xs:annotation>
<xs:documentation>optional if ERP system wants to specify which version this message applies to</xs:documentation>
</xs:annotation>
</xs:element>
-   <xs:element      name="CosemAttributes"      maxOccurs="unbounded"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemObject.CosemAttributes">
<xs:annotation />
- <xs:complexType sawsdl:modelReference="http://iec.ch/TC57/CIM-generic#CosemAttribute">
- <xs:sequence>
-   <xs:element      name="attributeID"      type="xs:int"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemAttribute.attributeID">
<xs:annotation />
</xs:element>
-   <xs:element      name="serviceClass"      type="xs:boolean"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemAttribute.serviceClass">
- <xs:annotation>
<xs:documentation>values: 1= confirmed, 0=unconfirmed</xs:documentation>
</xs:annotation>
</xs:element>
-   <xs:element      name="SelectiveAccessDescriptor"      minOccurs="0"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemAttribute.SelectiveAccessDescriptor">
<xs:annotation />
- <xs:complexType sawsdl:modelReference="http://iec.ch/TC57/CIM-generic#SelectiveAccessDescriptor">
- <xs:sequence>
-   <xs:element      name="accessSelector"      type="xs:int"      sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#SelectiveAccessDescriptor.accessSelector">
<xs:annotation />

```

```

</xs:element>
- <xs:element name="AccessParameter" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#SelectiveAccessDescriptor.AccessParameter">
<xs:annotation />
- <xs:complexType sawsdl:modelReference="http://iec.ch/TC57/CIM-generic#CosemData">
- <xs:sequence>
- <xs:element name="data" type="xs:string" sawsdl:modelReference="http://iec.ch/TC57/CIM-generic#CosemData.data">
<xs:annotation />
</xs:element>
- <xs:element name="dataType" type="xs:string" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#CosemData.dataType">
<xs:annotation />
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="PhysicalDevice" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#LogicalDevice.PhysicalDevice">
<xs:annotation />
- <xs:complexType sawsdl:modelReference="http://iec.ch/TC57/CIM-generic#PhysicalDevice">
- <xs:sequence>
- <xs:element name="mRID" type="xs:string" minOccurs="0" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#IdentifiedObject.mRID">
- <xs:annotation>

```

```

<xs:documentation>A Model Authority issues mRIDs. Given that each Model Authority has a unique id and this id is part
of the mRID, then the mRID is globally unique.</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="description" type="xs:string" minOccurs="0" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#IdentifiedObject.description">
- <xs:annotation>
<xs:documentation>The description is a free human readable text describing or naming the object. It may be non unique
and may not correlate to a naming hierarchy.</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="localName" type="xs:string" minOccurs="0" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#IdentifiedObject.localName">
- <xs:annotation>
<xs:documentation>The localName is a human readable name of the object. It is only used with objects organized in a
naming hierarchy. The simplest naming hierarchy has just one parent (the root) giving a flat naming hierarchy.
However, the naming hierarchy usually has several levels, e.g. Substation, VoltageLevel, Equipment etc. Children of
the same parent have names that are unique among them. If the uniqueness requirement cannot be met
IdentifiedObject.localName shall not be used, use IdentifiedObject.name instead.</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="name" type="xs:string" minOccurs="0" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#IdentifiedObject.name">
- <xs:annotation>
<xs:documentation>The name is a free text human readable name of the object. It may be non unique and may not
correlate to a naming hierarchy.</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="serialNumber" type="xs:string" minOccurs="0" sawsdl:modelReference="http://iec.ch/TC57/CIM-
generic#Asset.serialNumber">
- <xs:annotation>
<xs:documentation>Serial number of this asset.</xs:documentation>
</xs:annotation>
</xs:element>

```

```
</xs:sequence>  
</xs:complexType>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
</xs:schema>
```