1

Basic Ground Rules

2

For

3

Model Documentation

4

5

6
7
8
9

Version 6.1

10
11
12
13
14
15
16
17
18
19
20
21

Document Version: 3

22
23

24   # Table of Contents
25

63

1 Model Detail

COPYRIGHT & LIABILITY

The Edig@s Workgroup (EASEE-Gas Message and Workflow Design Working Group) disclaims and excludes, and any user of the Edig@s Workgroup Implementation Guidelines acknowledges and agrees to the Edig@s Workgroup disclaimer of, any and all warranties, conditions or representations, express or implied, oral or written, with respect to the guidelines or any part thereof, including any and all implied warranties or conditions of title, non-infringement, merchantability, or fitness or suitability for any particular purpose (whether or not the Edig@s Workgroup knows, has reason to know, has been advised, or is otherwise in fact aware of any such purpose), whether alleged to arise by law, by reason of custom or usage in the trade, or by course of dealing. Each user of the guidelines also agrees that under no circumstances will the Edig@s Workgroup be liable for any special, incidental, exemplary, punitive or consequential damages arising out of any use of, or errors or omissions in, the guidelines.

2 Basic Ground Rules

2.1 Introduction

The basic ground rules outlined in this document are applicable for all the messages developed for version 6 of the *Edig@s* Message Implementation Guidelines.

2.2 Time identification

It is mandatory to use UTC as the standard time metrology in the Edig@s messages as recommended in the CBP 2003-002/01.

When parties involved are located in different time zones this will largely simplify the correct definition and understanding of the time indications. Additionally the annual switch to and from daylight saving time does not affect UTC.

Dates and times shall use the standardised format from ISO 8601(i.e. a date and time shall be expressed as:

YYYY-MM-DD**T**hh:mm:ss**Z**

whereas a date and time interval shall be expressed as

YYYY-MM-DD**T**hh:mm**Z**/ yyyy-mm-dd**T**hh:mm**Z**

In the above formulae the following terms mean:

> **YYYY = Year;**
> **MM = month;**
> **DD = day;**
> **hh = hour;**
> **mm = minutes;**
> **ss = seconds.**

ATTENTION:

It is mandatory that all times included in a message are provided with the same time definition.

2.2.1 Values for hour definition

The identification of an hourly period in the *Edig@s* messages is defined respecting the following rules:

- Values for hours range from **00h** to **23h**

i.e.: the last hour of a day is from **23h00** to **00h00**

The first hour of the next day is from **00h00** to **01h00**

It is thus clear that **the first hour** in a combination **is always inclusive** and **the last hour** of a combination **is always exclusive**.

- Values for minutes range from 00 to 59

E.g.: 03:00, 03:01, 03:02, ..., 03:58, 03:59, 04:00

2.2.2 Daylight saving time

The use of UTC, mandatory in Edig@s, makes the messages independent of any impact due to Daylight Saving Time. However as a convention used in the gas and electricity industries on the change to daylight saving time (summertime) the new day in question has only 23 hours. On contrary on the change from daylight saving time (wintertime) the day in question has 25 hours.

The first gas day therefore in the case of CEST summertime day (last weekend of March between Saturday to Sunday) is as follows:

The time interval on the day of the changeover is AAAA-MM-DDT05:00Z/ AAAA-MM-DD+1T04:00Z (i.e. 23 hours)

For the rest of the summer the time interval corresponds to:

AAAA-MM-DDT04:00Z/AAAA-MM-DD+1T04:00Z Note: the first hour is inclusive; the last hour is exclusive.

The first gas day in the case of CET Summertime day (last weekend of October between Saturday to Sunday) is as follows:

The time interval on the day of the changeover is AAAA-MM-DDT04:00Z/ AAAA-MM-DD+1T05:00Z (i.e. 25 hours)

125 For the rest of the winter the time interval corresponds to:
126 AAAA-MM-DDT05:00Z/AAAA-MM-DD+1T05:00Z Note: the first hour is inclusive; the last hour is exclusive.

2.3 Edig@s document uniqueness

128 Each Edig@s electronic document that is not a version of a previous transmission shall have an identification which
129 shall be unique over time.
130 An Edig@s document shall be identified by the concatenation of the identification of the issuer of the message, the
131 document identification, document type and the document version.

2.3.1 Changing previously issued messages

133 A previously issued message has a unique document identification as well as a version. If a change takes place, and
134 the revised message has to be issued, then the document identification remains unchanged and the version is increased.
135 The new version of the document completely replaces the previous version of the document.
136 The first version number for a given document identification shall normally be 1.
137 The document version number must be incremented for each retransmission of a document that contains changes to
138 the previous version.
139 The receiving system shall only accept a document with a version number which is greater than the previous version
140 number of the same document.
141 Unless stated otherwise in separate message implementation guideline documents, a new message with either a new
142 identification or the same identification and same or lower version number should be rejected.

2.3.2 Document uniqueness during the nomination process

144 The nomination process is broken down into 2 phases:
145 1. The transmission of a day ahead nomination for the beginning of a gas day.
146 2. The transmission of within day nominations for all the rest of the day respecting the market area submission rules.

2.3.2.1 Nomination

148 The nomination covers the full gas day. The nomination shall be identified by the issuer of the nomination, a document
149 identification that is unique and a version number. If prior to the first period for the gas day it is necessary to correct
150 the document or it is necessary to renominate within the renomination deadlines, it shall have the same identifiers
151 (issuer and document) as the initial submission with the exception of the version which shall be superior to the previous
152 version.

2.4 Document header validity period scope

154 The concatenation of the time intervals in an electronic document shall cover the complete validity period. Any
155 exceptions to this rule shall be explicitly defined in the documentation of the relevant implementation guides.

2.5 Units used in messages

157 In line with the EASEE-Gas Recommendation "CBP Harmonisation of Units" *Edig@s* recommends the use of the
158 following unit standards in the *Edig@s* message:
159   Pressure: bar
160   Energy: kWh (with a combustion reference temperature of 25°C)
161   Volume: m3 (at 0°C and 1.01325 bar) (normal m3)
162   Gross Calorific Value: kWh/m3 (normal m3), with a combustion reference temperature of 25°C
163
164 In case other unit references or scales are used within *Edig@s* messages, a bilateral operating agreement between the
165 sending and receiving parties shall state the references used. No conversion parameters shall be included in the *Edig@s*
166 message in case other references are used for pressure, energy, volume or Gross Calorific Value.

2.6 Code values

168 When coded entries are required, the valid code values are found in the electronic document implementation guide
169 decision table.
170
171 The definition of the various code values is only provided in the relevant list in the *Edig@s* Code list.
172 **Attention:**

173   Missing code values should be reported to the *Edig@s* Workgroup who will arrange for adequate action.
174
175   See the *Edig@s* Maintenance Procedure.

176   ## 2.7 Message examples

177   Examples, where provided, illustrate how message templates can be implemented. Those examples are fictitious and
178   they DO NOT necessarily represent an actual operational situation. Under actual operational conditions those
179   messages may be more complex while involving more information. Since such examples are only illustrative it MUST
180   NEVER be used as a basis for programming or implementing this message.

181   ## 2.8 Debit and credit definition

182   A debit refers to a quantity that decreases a balance account.
183   A credit refers to a quantity that increases a balance account.

184   ## 2.9 The definition of input and output time series

185   An **Input quantity is the quantity entering a system operated by a System Operator** and an **Output quantity is**
186   **the quantity exiting a system operated by a System Operator**.
187   This information is provided in the Attribute "Direction" which is tightly coupled to the quantity itself and is normally
188   found in the Period class.
189   Consequently as long as there is no difference in the characterising information on which the Period class is dependent
190   the direction, Z02 for input and Z03 for output may be mixed in the same Period set. For example the first hourly
191   period could be Z02, the second hour Z03, the third hour Z03, the fourth hour Z02, etc..
192
193   This rule may vary if it is necessary to convey the direction as submitted to a System Operator. In such a case this
194   condition shall be defined in the relevant implementation guide.

195   ### 2.9.1 Example of direction use

196
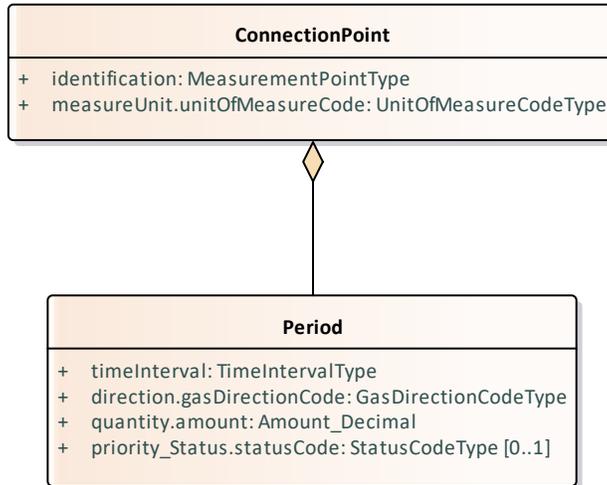
197
198   Figure: 1 **Example of direction use**
199   In the above example the Period class is dependent on the Connection Point class. Consequently as long as there is
200   the same connection point, and associated attributes, the Period class may be repeated. Otherwise a new Connection
201   Point class with different attributes will be required.
202   This implies that there is only one Connection Point class where its attributes remain the same.

203   ## 2.10 Flow direction convention

204

205   ### 2.10.1   Use case

206 In case the quantities do not directly relate to a system operated by a System Operator, for instance under Gas Sales
207 Agreements or Service Agreements, Input quantities are quantities put into the contract and Output quantities are
208 quantities delivered out of the contract. Quantities provided by a Seller to a Buyer are Output quantities (delivered out
209 of the sales agreement). Quantities provided by a Shipping user to a Shipping provider are Input Quantities (into the
210 shipping agreement).
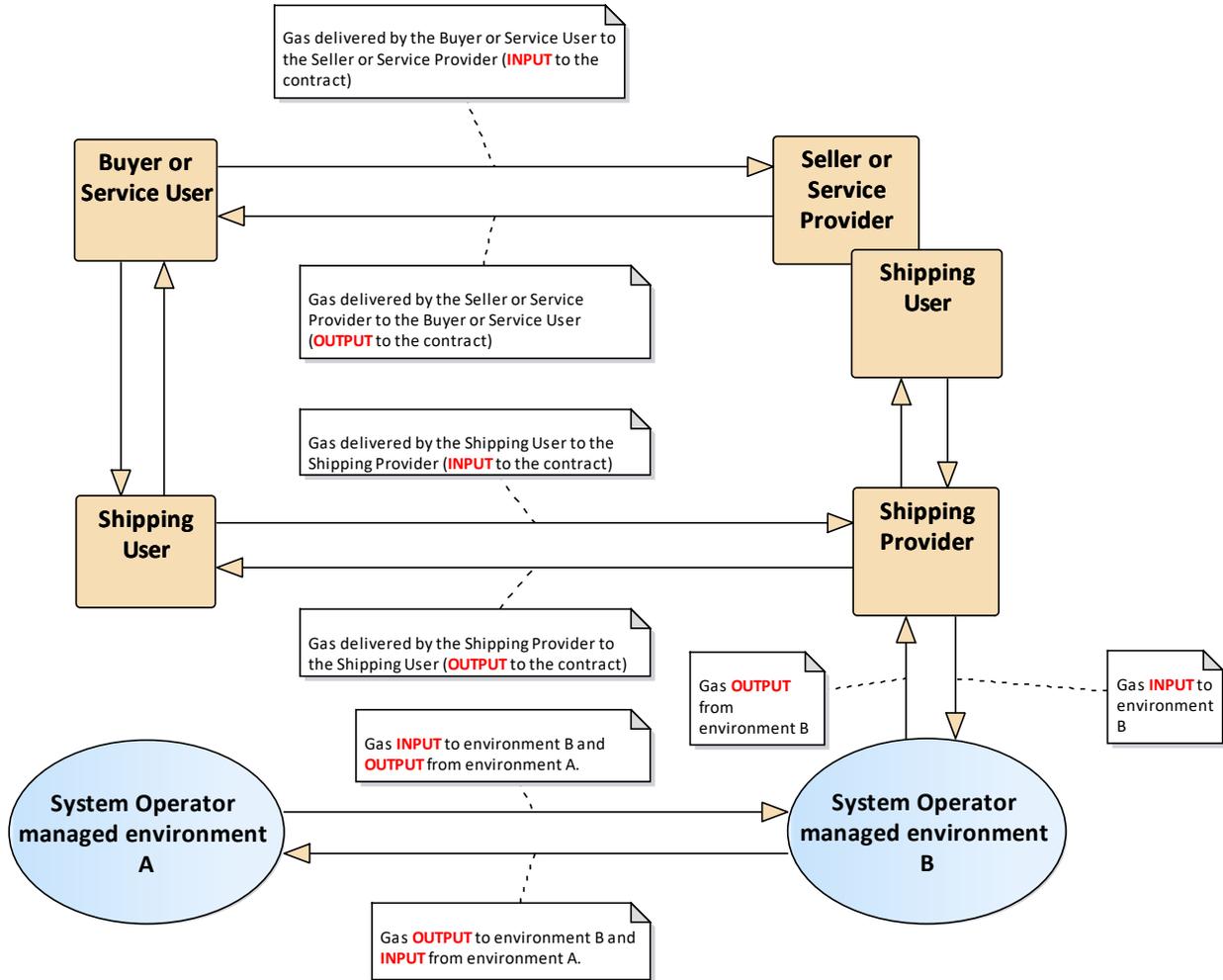
Figure: 2 **Use case**

2.10.2   Direction switch

215 A direction switch between 2 Periods (between two hours) is implemented directly where the first Period has an input
216 direction (Z02) and a direction switch to output occurs in the second period. It is sufficient to change the direction for
217 the second period to output (Z03). This indicates that the direction has changed from an input direction to an output
218 direction.

2.11 Quantity assignment

220 All quantities assigned to the Quantity attribute shall only be expressed as a numeric value using the characters in the
221 range 0 to 9 in addition to a single decimal sign. Only a negative decimal sign is permitted.
222 In addition all quantities with a unit of measure such as Kilowatts shall not contain a decimal sign since the sign is not
223 an integral part of the quantity value and is generally used merely to express an additional concept.
224 Negative values are permissible for quantities with a unit of measure such as temperature since the sign is an integral
225 part of the quantity value and expresses no additional concept.

2.11.1   Decimal mark

227   The decimal mark is the point (".").

2.11.2   Measure assignment

229   All values assigned to the MeasureUnit attribute shall respect the same rules defined for Quantity assignment.

2.11.3   Zero quantity values

231   If a quantity has a value of zero then the following rules shall be followed to define the direction of the zero value:
232   1.   If the complete time series is for an input direction then the zero quantity shall be input
233   2.   If the complete time series is for an output direction then the zero quantity shall be output
234   3.   If the time series is a mix of input and output directions then the zero quantity may be indifferently an input or an
235   output direction.

238   In the case of debit and credit zero values the same basic rules shall apply, namely:
239   1.   If the complete time series is for debit values then the zero quantity shall be debit
240   2.   If the complete time series is for credit values then the zero quantity shall be credit
241   3.   If the time series is a mix of debit and credit values then the zero quantity may be indifferently a debit or a credit
242   value.

2.12 Internal and external account definitions

245   An Internal Account corresponds to an account defined by a System Operator responsible for the area covered for a
246   Balance Responsible Party.
247   An External Account corresponds to an account defined by a System Operator that is not responsible for the area and
248   is known to both System Operators.

2.13 Rules concerning the use of a particular coding scheme for party identifications

251   For international trade all party identifications shall respect the CBP 2020-001/01 "Energy identifier coding". The
252   recommended coding scheme shall in this case always be the Energy Identification Coding scheme (EIC).
253   For internal trade it is recommended to use the same coding scheme but local rules may dictate the use of other
254   schemes.

2.14 Edig@s packages

256   An Edig@s package provides a homogeneous set of processes defined within the Edig@s scope. Each process and
257   electronic document are identified by the targetNamespace URN which is broken down as follows:
258   "urn:easee-gas.eu:edigas:[Process]:[ElectronicDocument]:6:1" with a mandatory attribute "schemaVersion" which
259   should indicate which version of the XSD schema which is in use. Each electronic document is documented in a
260   message implementation guide (MIG document), where a document version and a corresponding schema version is
261   present. This enables an electronic document to evolve and cater for corrections or minor evolutions.

263   The package in the targetNamespace is identified with the indication "6.1".

265   Parties opting to use the Edig@s standard should always implement the most recently published package. All parties
266   should make provisions to be able to handle at least two versions of the standard, one being the latest published
267   package. Which package will be used between two parties should be agreed bilaterally in the Operational Agreement
268   or in the Interconnection Agreement.

270   An Edig@s revision of a package will only occur at a minimum of two years.

2.15 Multiple versions of Edig@s messages

273   In order to be usable, a standard need to evolve and to adapt to changing operating environments as well as to evolving
274   and new market requirements. This is why there are multiple versions of an Edig@s message.

275 Edig@s shall only make changes or evolutions to the last version of a message that has been published on the Edig@s
276 website.
277
278 New requirements resulting in a change of an existing message model (name, multiplicity, datatype, constraint,
279 relation, role, etc.) will result in a new model being issued and a new schema version of the message structure in the
280 current package.
281
282 There are two levels of version and release information provided in an XML schema instance:
283 The version and release of the document set of implementation guides (i.e., Version 6 Release 1), and the version of
284 the schema. The information is provided in the default namespace of the XML schema instance:
285
286 (i.e.   urn:easee-gas.eu:edigas:[Process]:[ElectronicDocument]:6:1   schemaVersion="n").   The   last   attribute,
287 "schemaVersion", corresponds to the "schema version" that can be found on the cover page of the implementation
288 guide where the document for the electronic document is defined. This information may be found in every Edig@s
289 schema in the targetNameSpace and schemaVersion attributes.
290
291 For example:
292
293 <?xml version="1.0" encoding="UTF-8"?>
294 <AuctionBid_Document   xmlns="urn:easee-gas.eu:edigas:CapacityAllocationAuction:AuctionBidDocument:6:1"
295 schemaVersion="1">
296 <identification>001</identification>
297
298 In the case of a corrigendum the following release structure shall be used:
299 · [document version] [C] [corrigendum version]
300 · Document version = existing document version of the process
301 · C = corrigendum
302 · Corrigendum version = latest corrigendum version.
303
304 For example, if there is a first corrigendum for document version "2" the release value shall be "2C1".

2.16 The use of "_", "." and upper and lower camelcase in XML schema tag names

307 XML Schema tag names are constructed from the tag names provided in the document assembly model. The document
308 assembly model is automatically generated from the document contextual model. The assembly process integrates
309 several contextual classes into a more concise class whenever possible. The naming convention provided by the
310 assembly process enables in a precise manner to be able to identify the referenced object in the document contextual
311 model.
312 The basic assembly rules are:
313 A class name of a contextual model class (Aggregate Business Information Entity – ABIE) begins with an upper case
314 letter. In a multi word name, the beginning letter of every succeeding name is uppercase.
315 • An attribute name of an ABIE contextual class always begins with a lower case letter. In a multi word name, the
316   beginning letter of every succeeding name is uppercase.
317 • The role name of the "association end" to an ABIE contextual class always begins with an upper case letter. In a
318   multi word name, the beginning letter of every succeeding name is uppercase.
319 • An underscore "_" separates a qualified name from a basic name that it is contextualising.
320 A class name of an assembly model class (Message Business Information Entity – MBIE) begins with an upper case
321 letter. In a multi word name, the beginning letter of every succeeding name is uppercase.
322 • An attribute name of an MBIE assembly class always begins with a lower case letter. In a multi word name, the
323   beginning letter of every succeeding name is uppercase.
324 • The role name of the "association end" to an MBIE assembly class always begins with an upper case letter. In a
325   multi word name, the beginning letter of every succeeding name is uppercase.
326 • An underscore "_" separates a qualified name from a basic name that it is contextualising.
327 When at assembly level, two aggregated ABIE contextual classes have been grouped into one MBIE assembly class:

328   • The name of the assembly class is the name of the aggregating contextual class.
329   • All of the attributes of the aggregating contextual class become attributes of the assembly class.
330   The attributes of the aggregated class that have become attributes of the assembly class have the following naming
331   rule:
332   • The name of the assembly class attribute is the name of the aggregated contextual class attribute prefixed with
333   the contextual aggregation end role name followed by a period "." and the first letter of the role name is reverted
334   to lowercase.
335   A typical example of the application of these rules can be found in every assembly document header with the tag name
336   "issuer_MarketParticipant.marketRole.code".
337   For example where the ABIE contextual model indicates the following:

2.16.1   Contextual model

339
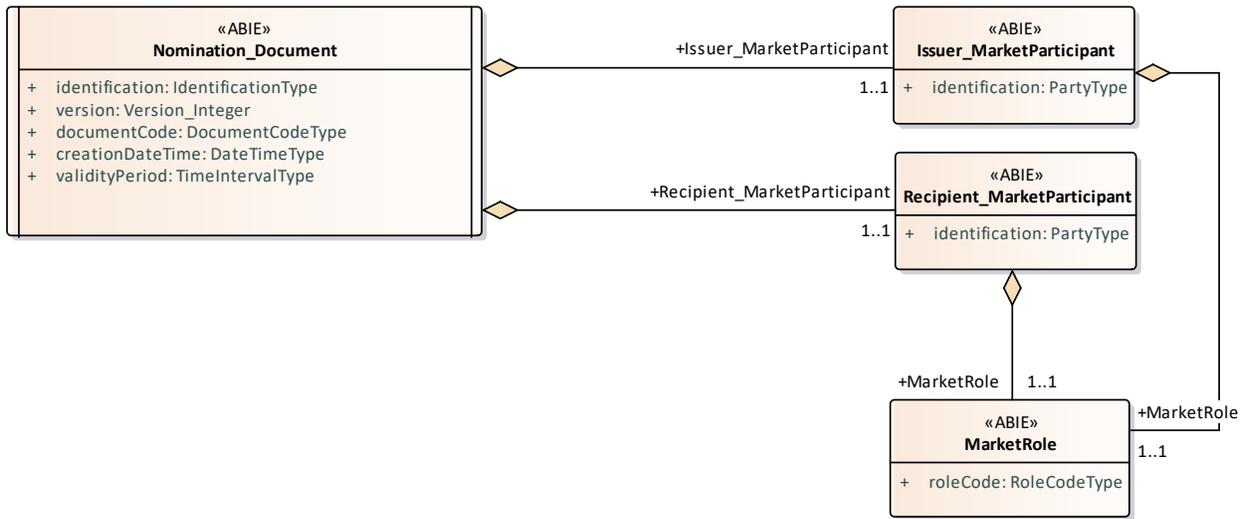
340
341   Figure: 3 **Contextual model**
342   This is transformed automatically into the following assembly model:

2.16.2   Assembly model

344
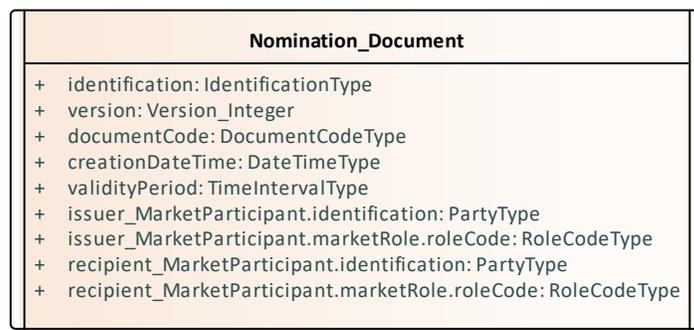
345
346   Figure: 4 **Assembly model**
347   Where the four attributes:
348   • issuer_MarketParticipant.identification;
349   • issuer_MarketParticipant.marketRole.roleCode;
350   • recipient_MarketParticipant.identification;
351   • recipient_MarketParticipant.marketRole.roleCode;
352   have all been aggregated from the two classes MarketParticipant and MarketRole as well as the three roles Issuer,
353   Recipient and MarketRole as follows:

354 • The generic class "MarketParticipant" has been assembled into the class "MarketDocument" and becomes an
355   attribute of MarketDocument".
356 • The role "Issuer" qualifies the generic class "MarketParticipant". However, because it is now an attribute of the
357   class "MarketDocument" the first letter is transformed to lowercase.
358 • The generic class "MarketRole" has been assembled into the class "MarketParticipant" and becomes an attribute
359   of "MarketParticipant". As an attribute of "MarketParticipant" the first letter of the class name is transformed to
360   lowercase since it is not qualified and it is separated from "MarketParticipant" with a period.
361 • "roleCode" is an attribute of the class "MarketRole" and is separated from it with a period.".

2.17 Use of message restriction XSDs

363 All Edig@s message XSDs include in the base schema a schema that restricts the Edig@s codelist set to allow only
364 the codes permitted in the message in question. The Figure below shows the case of an Edig@s message that is
365 compliant with the Edig@s standard.

2.17.1   Schema setup that is Edig@s compliant

367
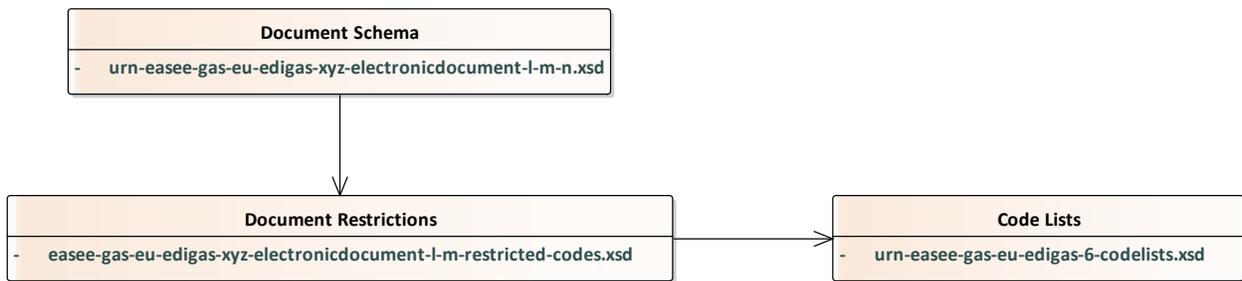
Figure: 5 **Schema setup that is Edig@s compliant**

370 The standard Edig@s document schema includes within it a document restriction schema that contains all the codes
371 from the Standard Edig@s codelist that are permitted for use in the message. The Edig@s schema is named "urn-
372 easee-gas-eu-edigas-xyz-electronicdocument-n-m.xsd" Where this corresponds to the target namespace of the
373 document in question and where the ":"s have been replaced by a "-".
374 The electronic document restriction schema for a given message is named "urn-easee-gas-eu-edigas-xyz-
375 electronicdocument-l-m-restricted-codes.xsd" where the part "xyz-electronicdocument-l-m" is the same as that of the
376 message schema in which it is included.
377 The Edig@s schema design also permits a Transmission System Operator to add for local market use additional codes
378 over and above the permitted set of codes. In order to enable this possibility the local code schemas as outlined in
379 figure 3 have been added.

2.17.2   Local code extension schema

381
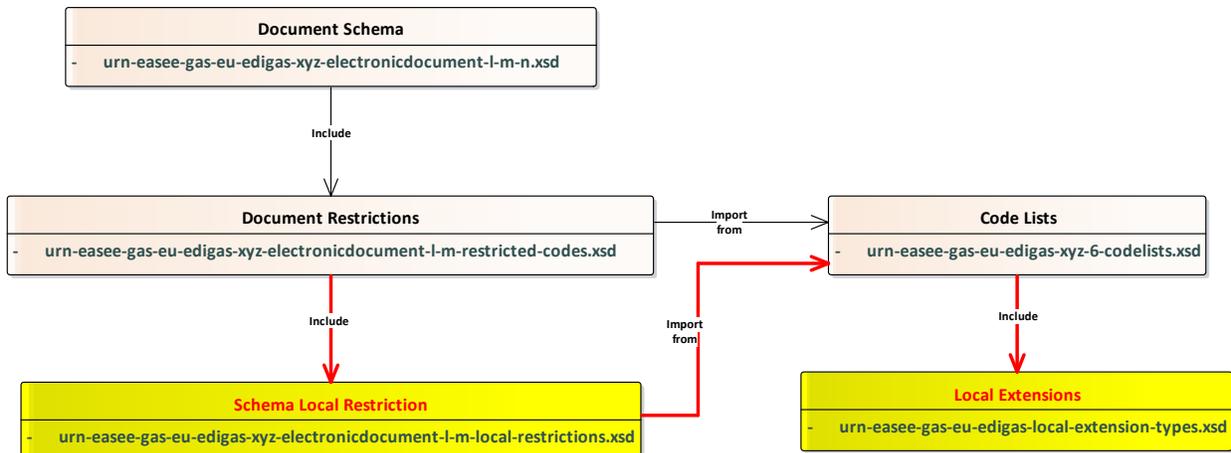
Figure: 6 **Local code extension schema**

384 In order to enable a Transmission System Operator to add local codes exclusively for use within the local market two
385 codelist schema have been integrated into the standard schema set.
386 The first extension schema is associated with the standard set of codelists and contains all the codes that a Transmission
387 System Operator wishes to use within the local market. This codelist "extends" in fact the standard codelist and any
388 local codes that are to be permitted for use within the Edig@s document set must be added to this list.
389 The second extension schema is associated with the standard document restriction schema and enables the list of
390 standard permitted restricted codes to be extended with the local codes. All codes used within this list must appear in
391 the general local extension codelist. This schema is related to a specific message and has the name in the form "urn-
392 easee-gas-eu-edigas-xyz-electronicdocument-l-m-local-restrictions.xsd"   where   "xyz-electronicdocument-l-m"
393 corresponds to the message schema in question.
394 The release of a given set of electronic document schema will always contain these two codelist schema which by
395 default only contain codes within the permitted codelist set. It is the responsibility of the Transmission System
396 Operator to maintain these codelists up to date with any local variants that have been added for the local market. The
397 addition of codes to these codelists makes the use of the Edig@s messages non-compliant with the standard.
398 Some local market conditions may restrict the standard restricted codelist even further. This is carried out through the
399 modification of the "Document restrictions" XSD file. It should however be noted this makes the standard codelist
400 restrictions non-compliant.
401 The use of the functions outlined in yellow and red in the diagram above make the implementation non standard since
402 these introduce local requirements which deviate from a harmonised implementation approach.

403 ## 2.18 Use of net values

404 In Edig@s 6 it has been decided to only permit net-values in nomination messages. This means that for a given
405 timeseries (for the same connection point) in a Nomination & Matching message (NOMINT, NOMRES, DELORD,
406 DELRES) only one value and one direction code should be present.
407
408 There is an exception for messages that are exchanged between systems that matches with different granularity. When
409 a nomination message is exchanged from a daily system to a system that is matching on an hourly basis (and both
410 input and output nominations is possible), the daily message needs to have both directions present in the message.
411
412 The Edig@s version 6 does not provide any provisions for validation of the usage of net values in the XSD, so even
413 though it is possible to provide both directions in the messages through the Edig@s schemas, the guidelines presented
414 here are mandatory to follow.
415

416 # 3 Document Change Log

417 ## 3.1 Version

Version	Description
Version 1 2020-06-29	Initial release
Version 2 2021-07-02	Release 6.1 Corrected section for document uniqueness, added document type as part of the concatenation. Updated section for Edig@s packages, versions and namespace.
Version 3 2023-07-15	Updated section 2.3.1 for changes on previously issued messages. Deleted section 2.3.2.2 Within day nomination

418