TECHNICAL REPORT

ISO/IEC TR 19583-23

First edition
2020-06

Information technology — Concepts and usage of metadata —

Part 23: Data element exchange (DEX) for a subset of ISO/IEC 11179-3

Technologies de l'information — Concepts et utilisation des métadonnées —

Partie 23: Échange d'éléments de données (DEX) pour un sous-ensemble de l'ISO/IEC 11179-3

Reference number
ISO/IEC TR 19583-23:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

A list of all parts in the ISO/IEC 19583 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO/IEC 11179 series addresses the semantics of data, the representation of data, and the registration of the descriptions of that data, i.e. metadata. While ISO/IEC 11179-3 provides the basic conceptual model for a metadata registry (MDR) in which information about metadata can be recorded and maintained, implementers and users of the registries described in the ISO/IEC 11179 series require further guidance to exchange data element definitions with each other via a standard-based protocol. It is necessary to have a common protocol and message semantics to be able to communicate with an MDR to locate the data elements given the search criteria and exchange metadata of data elements by addressing the technical and semantic interoperability challenges.

This document was developed to describe a message exchange framework specification for communicating a subset of data element metadata with an ISO/IEC 11179-3 MDR.

Information technology — Concepts and usage of metadata —

Part 23: Data element exchange (DEX) for a subset of ISO/IEC 11179-3

1 Scope

This document specifies the message exchange framework for communicating data element definitions with an ISO/IEC 11179-3 metadata registry (MDR). It defines message semantics, protocols and bindings for a set of transactions allowing the exchange of a commonly used subset of data element metadata with an ISO/IEC 11179-3 MDR.

This document establishes the following data element message exchange interoperability specifications:

— retrieving data element list from an ISO/IEC 11179-3 MDR matching the selection criteria;

— retrieving metadata of a selected data element from an ISO/IEC 11179-3 MDR.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11179-3 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1.1
target data model
information view of an application schema

Note 1 to entry: Implementation-dependent realizations of logical information models such as XML serializations, RDF serializations, JSON serializations and relational database schemas.

[SOURCE: ISO/TS 19129:2009, 4.1.2, modified — Note 1 to entry has been added.]

3.1.2
mapping specification
procedural functions to locate the data element in a given *target data model* (3.1.1)

Note 1 to entry: A mapping specification includes a *mapping script* (3.1.3), the type of the script (e.g. XPATH) and the target data model on which the mapping script can be executed.

3.1.3
mapping script
executable script on a given *target data model* (3.1.1)

Note 1 to entry: A mapping script can be an XPATH expression to be executed on XML documents[1].

3.1.4
metadata consumer
actor that retrieves the metadata created by the *metadata source* (3.1.5)

Note 1 to entry: A metadata consumer can optionally query the *metadata source* (3.1.5) for a list of data elements matching the *selection criteria* (3.1.6).

3.1.5
metadata source
actor responsible for creation of the data element list matching the *selection criteria* (3.1.6) and the creation of metadata for a selected data element per request from the *metadata consumer* (3.1.4)

Note 1 to entry: A metadata source is associated with an ISO/IEC 11179-3 metadata registry.

3.1.6
selection criteria
logical expression, the criteria being satisfied only if the expression evaluates to the value TRUE

[SOURCE: ISO 10303-14:2005, 3.3.9]

3.1.7
value set
finite set in a specific correspondence with the index-set of the list

Note 1 to entry: A value set is identified with a unique ID and version.

[SOURCE: ISO 8485:1989, 5.3.1.4, modified — Note 1 to entry has been added.]

3.2 Abbreviated terms

CDASH	Clinical Data Acquisition Standards Harmonization
CDISC	Clinical Data Interchange Standards Consortium
DEX	data element exchange
FHIR	fast healthcare interoperability resources
HITSP	Healthcare Information Technology Standards Panel
JSON	JavaScript object notation
MDR	metadata registry
NAV	navigation error (due to unknown data element)
NCI	National Cancer Institute

POSIX	portable operating system interface
RDF	resource description framework
regex	regular expressions
REST	representational state transfer
SDTM	study data tabulation model
SPARQL	simple protocol and RDF query language
SQL	structured query language
XML	eXtensible Markup Language
XPATH	XML path language
XSD	XML schema definition
VERUNK	version unknown

4 Overview of data element exchange (DEX) specification

4.1 General

The objective of DEX specification is to describe a message exchange framework for communicating a subset of administered data element definitions with an ISO/IEC 11179-3 MDR. It defines message semantics, protocols and bindings for a set of transactions allowing the exchange of a subset of administered data element metadata with an ISO/IEC 11179-3 MDR.

Two actors are defined as a part of this specification:

— Metadata source: The metadata source is responsible for the creation of the data element list matching the selection criteria and the creation of metadata for a selected data element per request from the metadata consumer. The metadata source is associated with an ISO/IEC MDR.

— Metadata consumer: The metadata consumer is responsible for the importation of metadata created by the metadata source. The metadata consumer can optionally query the metadata source for a list of data elements matching the selection criteria.

The following data element message exchange patterns are supported between the metadata source and metadata consumer actors:

— Retrieving a data element list from an ISO/IEC 11179-3 MDR matching the selection criteria.

— Retrieving metadata of a selected data element from an ISO/IEC 11179-3 MDR.

Corresponding to these message exchange patterns, two transaction specifications are provided:

— retrieve data element list;

— retrieve metadata.

The retrieve data element list transaction is an optional preparatory act to retrieve the identification information for the list of data elements matching the given the selection criteria, which can be used in the second retrieve metadata transaction to collect the metadata of the selected data element. The core content to be exchanged as a result of the retrieve metadata message exchange pattern is “data element metadata”. In 4.2, details of this content are examined first since the search criteria as a part of retrieve data element list also depends on this content specification. In 4.3 and 4.4, the transaction specifications of retrieve data element list and retrieve metadata are presented.

4.2 Data element metadata

The data element metadata is a flattened subset of the metadata attributes defined in the ISO/IEC 11179 series as depicted in Table 1. A mapping of DEX attributes to ISO/IEC 11179-3 attributes is available in Annex C. On top of the ISO/IEC 11179 series-based metadata of a data element, the attribute Mapping_Specification has been added to specify the mapping of an abstract data element definition to different target data models.

The attribute names in Tables 1 through 9 are based on ISO/IEC 11179-3. An ISO/IEC 11179 series class attribute is specified by using the class name and attribute name separated by a period e.g. 11179_class_name.attribute_name. The attribute name is written in the form of 11179_Class_Name if the attribute is composing inner attributes defined within a separate table. The optionality field can have the following values with their associated meanings:

R required

R2 required if the information is available

O optional

The string data type corresponds to xsd:string and the date data type corresponds to xsd:date. The format for xsd:date is YYYY-MM-DD where Y is the year, M is the month, D is the day of month.

Table 1 — Data_Element metadata details

Attribute name	Optionality	Is repeatable	Type	Description
identifier	R	No	string	The universally unique identifier of the data element. This identifier should be the same as the identifier in the received retrieve metadata request message.
registration_authority_identifier	R	No	string	The authority who has defined and registered the data element to the metadata source. EXAMPLES CDISC, HITSP, NCI. See ISO/IEC 11179-3:2013 6.3.8.2. This attribute is comprised of international_code_designator, organization_identifier, organization_part_identifer. All these attributes can be used to create the string for the registration_authority_identifier. This is a mandatory field since the data elements must be administered for this specification.
version	R	No	string	The version of the data element.
designation.sign	R	No	string	The designation (name) of the data element that can be used for display purposes.
definition.text	R	No	string	The definition that gives an unambiguous description of the data element and its use.

Table 1 *(continued)*

Attribute name	Optionality	Is repeatable	Type	Description
registry_specifiation.context	R2	No	string	The specific domain in which this data element is defined. EXAMPLES CDASH, SDTM. If such a context is defined by the registration authority for this data element in the MDR, then this attribute is mandatory.
creation_date	R	No	date	The date when this data element is created.
effective_date	R	No	date	The date when this data element becomes effective to be used.
until_date	R2	No	date	The date when the data element is no longer expected to be used.
last_change_date	R2	No	date	The date when the data element was last revised.
change_description	R2	No	string	A note that indicates the revision reason and description of the updates.
Data_Element_Concept	R	No	See Table 2 for the details of Data_Element_Concept	The concept which is the meaning part of the data element definition. A data element is created with an association of a data element concept and a value domain.
Value_Domain	R	No	See Table 3 for the details of Value_Domain	The domain from which the data element takes its values. Each data element is composed of a data element concept and a value domain.
Mapping_Specification	R	Yes	See Table 5 for the details of Mapping_Specification	The exact specification to locate the data element in a target data model. This attribute is an extension on top of ISO/IEC 11179-3.

Table 2 — Data_Element_Concept details

Attribute name	Optionality	Is repeatable	Type	Description
identifier	R	No	string	The universally unique identifier of the data element concept.
version	R	No	string	The version of the data element concept.
designation.sign	R	No	string	The textual representation of the data element concept.
object_class.designation.sign	R2	No	string	An object class represents a set of ideas, abstractions, or things in the real world that are identified with explicit boundaries and meaning and whose properties and behaviour follow the same rules. This attribute is the name of the object class of the data element concept.
property.designation.sign	R2	No	string	A property is a characteristic common to all members of an object class. This attribute is the name of the property of the data element concept.

Table 3 — Value_Domain details

Attribute name	Optionality	Is repeatable	Type	Description
identifier	R	No	string	The universally unique identifier of the value domain.
type	R	No	string	The type of the value domain. Valid types are defined, enumerated and described.
datatype.name	R	No	string	The data type which represents the characteristics of the permissible values for the property of the data element. EXAMPLE xsd:string.
unit_of_measure	R2	No	string	Actual units in which the associated values of the property of the data element are measured.
source_uri	R2	No	string	A reference to the external value set, if the value domain's type is "defined" (ISO/IEC 11179-3:2013/Amd 1:2020).
Permissible_Value	R2	Yes	See Table 4 for the details of Permissible_Value.	The permissible value set from which the values of this data element can be selected.

Table 4 — Value_Domain Permissible_Value details

Attribute name	Optionality	Is repeatable	Type	Description
permitted_value	R	No	string	The permitted value.
value_meaning.designation.sign	R	No	string	The textual representation of the meaning of the permitted value.
begin_date	R	No	date	The date that this value becomes effective.
end_date	R2	No	date	The date that this value becomes invalid.

Table 5 — Mapping_Specification summary

Attribute name	Optionality	Is repeatable	Type	Description
Target_Data_Model	R	No	See Table 7 for the details of Target_Data_Model	The data model that the data element is interrelated with. It can be a database schema, an OWL schema, an XML schema, etc.
type	R	No	string	The type of the mapping specification. The type should be selected from the mapping specification type value set (See Table 6).
mapping_script	R	No	string	The executable script to locate the data element in a target data model. EXAMPLE XPATH scripts, SPARQL or SQL queries.

Table 6 — Mapping specification type value set

Mapping specification type	Description
XPATH	XPath is a language that describes a way to locate and process items in XML documents by using an addressing syntax based on a path through the document's logical structure or hierarchy.
SQL	SQL is an industry-standard language for creating, updating and querying relational database management systems.
SPARQL	SPARQL is a standard query language and data access protocol for use with the RDF data model.
FHIR Query	A query with a set of parameters (http://www.hl7.org/implement/standards/fhir/query.html).
Other	Any mapping specification that is not one of the above types.

Table 7 — Target_Data_Model details

Attribute name	Optionality	Is repeatable	Type	Description
name	R	No	string	The name of the target data model. EXAMPLE ASTM/HL7 CCD.
description	R	No	string	Textual description of the target data model. If available, the unique identifier for the target data model can be presented here. EXAMPLE 2.16.840.1.113883.10.20.1 for ASTM/HL7 CCD.
url	R2	No	string	If available a reference to the schema of the target data model.

4.3 Retrieve data element list

4.3.1 General

This transaction is used by the metadata consumer to retrieve a list of data elements from the metadata source matching the given selection criteria. The interaction diagram is depicted in Figure 1. In 4.3.2 through 4.3.4 the triggering events for the request and response messages involved in this transaction, the message semantics of the request and response messages and the expected action that needs to be carried out by the receiving actors are presented.

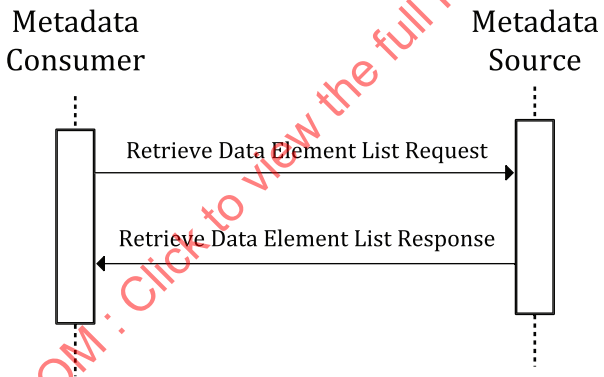

Figure 1 — Retrieve metadata element list transaction

4.3.2 Retrieve data element list request

4.3.2.1 Trigger events

The metadata consumer wants to retrieve the metadata of a list of data elements matching the selection criteria by defining several filters. The metadata consumer sends a retrieve data element list request to the metadata source.

4.3.2.2 Message semantics

The metadata consumer sends a retrieve data element list request message to specify the selection criteria to filter and return the metadata of a list of data elements. The selection criteria are represented as a set of filters as depicted in Table 8. The metadata consumer should send zero or more filters in the retrieve data element list request. Each filter applies a filtering criterion on the specified attribute of the data element. Providing no filters inside the retrieve data element list request means that the metadata consumer queries for all available data elements. More than one filter means that the metadata source

should semantically create an intersection of the matching data elements for each filter. That is, logical AND operator is implicitly assumed on multiple filters. In 4.3.4 the protocol requirements and the format of the message are described in full detail.

Table 8 — Filter details

Attribute name	Optionality	Is repeatable	Type	Description
name	R	No	string	The name of the data element attribute to apply the filter on.
operator	R	No	string	The operation of the filter. The operator should be selected from the filter operator value set (see Table 9).
value	R	No	string	The value of the filtered attribute to apply while executing the operator.

Table 9 — Filter operator value set

Filter Operator	Description
equals	Checks for mathematical equality of the given filter value with the value of the data element attribute.
match	This operator should compare the contents of the referenced attribute with the regex using the POSIX[4] matching rules. If the regex matches the attribute, the data element matches. All regex matches should be executed in case insensitive mode.
before	Date comparisons expects the argument in the format of xsd:date (YYYY-MM-DD) and compare it with the attribute using a date comparison. This operator should filter out the dates before or equals to the given filter date.
after	Date comparisons expects the argument in the format of xsd:date (YYYY-MM-DD) and compare it with the attribute using a date comparison. This operator should filter out the dates after or equals to the given filter date.

4.3.2.3 Expected action

The metadata source should perform matching in accordance with the filter operator rules described in Table 9. Any data element which has metadata that matches all of the provided filters at the same time should be included in the response. That is, multiple filters should be linked together with logical AND operator.

4.3.3 Retrieve data element list response

4.3.3.1 Trigger events

This message will be triggered by a retrieve data element list request message.

4.3.3.2 Message semantics

The response should be a retrieve data element list response message which should have one Data_Element_Summary attribute (presented in Table 10) for each matching data element. If no matching data elements are found, data element list response message should be empty. In 4.3.4 the protocol requirements and the format of the message are described in full detail.

Table 10 — Data_Element_Summary in the retrieve data element list response message

Attribute name	Optionality	Is repeatable	Type	Description
Data_Element_Summary	R2	Yes	See Table 11 for the details of Data_Element_Summary	The summary information about the data element.

The atttributes of the Data_Element_Summary are presented in Table 11. The string data type corresponds to xsd:string and the date data type corresponds to xsd:date. The format for xsd:date is YYYY-MM-DD where Y is the year, M is the month, D is the day of month.

Table 11 — The attributes of the Data_Element_Summary in the retrieve data element list response message

Attribute name	Optionality	Is repeatable	Type	Description
identifier	R	No	string	The universally unique identifier of the data element.
registration_authority_identifier	R	No	string	The authority who has defined and registered the data element to the metadata source (EXAMPLES CDISC, HITSP, NCI).
version	R	No	string	The version of the data element.
designation.sign	R	No	string	The name of the data element that can be used for display purposes
definition.text	R	No	string	Definition that gives an unambiguous description of the data element and its use.
registry_specification.context	R2	No	string	The specific context in which this data element is defined (EXAMPLES CDASH, SDTM, HITSP C154). If such a contextualDomain is defined by the registrationAuthority for this data element in the MDR, then it is mandatory.
Value_Domain.type	R	No	string	The type of the value domain. valid types are defined, enumerated and described
Value_Domain.datatype.name	R	No	string	The data type of the value domain.
Value_Domain.source_uri	R2	No	string	A reference to the external value set, if the value domain's type is ''defined' (ISO/IEC 11179-3:2013:Amd1:2020) (EXAMPLES http://snomed.info/ct, http://loinc.org)
Permissible_Value	R2	Yes	Table 4	Provide 3 permissible_value elements. Provide all if there are fewer permitted values.

4.3.3.3 Expected action

A metadata consumer processes the Data_Element_Summary elements according to its business process logic.

4.3.4 Protocol requirements

4.3.4.1 General

There are two different protocol requirements for the retrieve data element list transaction. The protocol of this transaction should be based on SOAP 1.2[3] and it should support HTTP REST.

SOAP

The relevant XML namespace definitions can be seen in Table 12.

Table 12 — WSDL namespace definitions

soap12	http://schemas.xmlsoap.org/wsdl/soap12/
wsdl	http://schemas.xmlsoap.org/wsdl/
xsd	http://www.w3.org/2001/XMLSchema
dex	urn:iso:it:metadata:dex:2016

These are the requirements for the retrieve data element list transaction presented in the order in which they should appear in the WSDL definition[2] (see Annex A for an informative WSDL):

— The following types should be included (xsd:include) in the /definitions/types: namespace="urn: iso:it:metadata:dex:2016", schema="DEX.xsd"

  — The /definitions/message/part/@element attribute of the Retrieve Data Element List Request message should be defined as “dex:RetrieveDataEementListRequest”

  — The /definitions/message/part/@element attribute of the Retrieve Data Element List Response message should be defined as “dex:RetrieveDataElementListResponse”

  — The /definitions/portType/operation/input/@message attribute for the Retrieve Data Element List Operation should be defined as “dex:RetrieveDataElementListRequestMessage”

  — The /definitions/portType/operation/output/@message attribute for the Retrieve Data Element List Operation should be defined as “dex:RetrieveDataElementListResponseMessage”

  — The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:iso:it:metadata:dex:2016:RetrieveDataElementList”

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are presented in 4.3.4.2 and 4.3.4.3.

A full XML schema document for the DEX types is available in Annex A.

REST

For the HTTP REST binding, this transaction is required to be served under “/DataElements” endpoint after the base endpoint which can be set according to the requirements of the serving host. One example can be http://example.com/service/mdr/api/2016/DataElements. Retrieve data element list transaction should bind to HTTP GET method of the identified endpoint.

The protocol requirements for the request and response messages are given in 4.3.4.2 and 4.3.4.3.

4.3.4.2 Retrieve data element list request message

SOAP

Within the request message payload the <dex:RetrieveDataElementListRequest/> element is defined as:

— Zero or more /dex:RetrieveDataElementListRequest/dex:filter element containing

  — a required /dex:RetrieveDataElementListRequest/dex:filter/dex:name element with type “xsd:string”

  — a required /dex:RetrieveDataElementListRequest/dex:filter/dex:operator element with type “xsd:string”

  — a required /dex:RetrieveDataElementListRequest/dex:filter/dex:value element with type “xsd:string”

A sample retrieve data element list SOAP request is given as follows:

```
soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsa=http://www.
w3.org/2005/08/addressing
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Header>
        <wsa:MessageID>urn:uuid:f43f7bda-a5f9-42b1-b8dc-e78be1a2a180</wsa:MessageID>
         <wsa:Action>urn:iso:it:metadata:dex:2016:RetrieveDataElementList</wsa:Action>
   </soap:Header>
   <soap:Body>
        <dex:RetrieveDataElementListRequest xmlns:dex="urn:iso:it:metadata:dex:2016">
               <dex:filter>
                   <dex:name>designation.sign</dex:name>
                   <dex:operator>match</dex:operator>
                   <dex:value>.*sex.*</dex:value> <!—Regular expression-->
               </dex:filter>
        </dex:RetrieveDataElementListRequest>
   </soap:Body>
</soap:Envelope>
```

REST

Since the transaction is bound to HTTP GET, the search parameters are required to be query parameters which will be visible in the query link. The query parameters should be given in a filter query within the format "filter=name:operator:value". The name of the parameter should be followed by a colon and then the operator followed by another colon, and finally the value for that parameter. It should be noted that the value itself may contain colons. Multiple filter parameters mean the matching data elements for the filters should be intersected such as applying a logical AND operator. The names of the HTTP query parameters are the same as the names of the attributes identified as Data Element Metadata in Table 1.

Values of the operator should be selected from the set defined in Table 8.

A sample request message which is the REST equivalent of the given SOAP request example is as follows:

http://example.com/service/mdr/api/2016/DataElements?filter=designation.sign:match:.*sex.*

4.3.4.3 Retrieve data element list response message

SOAP

Metadata source should include within the response message payload for the SOAP binding option the <dex:RetrieveDataElementListResponse/> element which contains:

— Zero or more /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary element, containing

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:identifier element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex: registration_authority_identifier element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:version element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex: designation.sign element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:definition.text element with type “xsd:string”

— an optional /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:registry_specification.context element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Value_Domain.type element with type “xsd:string”

— a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Value_Domain.datatype.name element with type “xsd: string”

— an optional /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Value_Domain.source_uri element with type “xsd:string”

— zero or more /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Permissible_Value element containing

  — a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Permissible_Value/dex:permitted_value element with type “xsd:string”

  — a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Permissible_Value/dex:value_meaning.designation.sign element with type “xsd:string”

  — a required /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Permissible_Value/dex:begin_date element with type “xsd:date”

  — an optional /dex:RetrieveDataElementListResponse/dex:Data_Element_Summary/dex:Permissible_Value/dex:end_date element with type “xsd:date”

A sample Retrieve Data Element List SOAP Response is given as follows:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
   <soap:Header>
         <wsa:Action>urn:iso:it:metadata:dex:2016:RetrieveDataElementListResponse</
wsa:Action>
         <wsa:RelatesTo>urn:uuid:7ec4961a-712f-11e7-8cf7-a6006ad3dba0</wsa:RelatesTo>
   </soap:Header>
   <soap:Body>
         <dex:RetrieveDataElementListResponse xsi:schemaLocation="urn:iso:it:metadata:d
ex:2016 dex.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dex="urn:iso:i
t:metadata:dex:2016">
               <dex:Data_Element_Summary>
                     <dex:identifier>8426f5a8-712f-11e7-8cf7-a6006ad3dba0</dex:identifier>

   <dex:registration_authority_identifier>CDISC:ClinicalResearch:DataElements</
dex:registration_authority_identifier>
                     <dex:version>0.1</dex:version>
                     <dex:designation.sign>DMSEX</dex:designation.sign>
                     <dex:definition.text>The assemblage of physical properties or qualities
by which male is distinguished from female; the physical difference between male and female;
the distinguishing peculiarity of male or female (NCI - CDISC Definition). Record the
appropriate sex (e.g., F (female), M (male).</dex:definition.text>
                     <dex:registry_specification.context>CDASH</dex:registry_specification.
context>
                     <dex:Value_Domain.type>Enumerated</dex:Value_Domain.type>
                     <dex:Value_Domain.datatype.name>xsd:string</dex:Value_Domain.datatype.
name>
                     <dex:Permissible_Value>
                           <dex:permitted_value>F</dex:permitted_value>

   <dex:value_meaning.designation.sign>Female</dex:value_meaning.designation.sign>
                           <dex:begin_date>2016-01-01</dex:begin_date>
                     </dex:Permissible_Value>
                     <dex:Permissible_Value>
                           <dex:permitted_value>M</dex:permitted_value>

   <dex:value_meaning.designation.sign>Male</dex:value_meaning.designation.sign>
                           <dex:begin_date>2016-01-01</dex:begin_date>
                     </dex:Permissible_Value>
               </dex:Data_Element_Summary>
         </dex:RetrieveDataElementListResponse>
   </soap:Body>
</soap:Envelope>
```

REST

The REST binding should return a JSON serialization of the retrieve data element list response message. The JSON response is a list of JSON objects where each JSON object represents a dex:Data_Element _Summary conceptually. Metadata source should include within the response message payload for the HTTP REST binding option a JSON list which can be empty if there are no matching data elements.

A sample retrieve data element list JSON response is given as follows:

```
 [{
   "identifier": "8426f5a8-712f-11e7-8cf7-a6006ad3dba0",
   "registration_authority_identifier": "CDISC:ClinicalResearch:DataElements",
   "version": "0.1",
   "designation.sign": "DMSEX",
   "definition.text": "The assemblage of physical properties or qualities by which
male is distinguished from female; the physical difference between male and female;
the distinguishing peculiarity of male or female (NCI - CDISC Definition). Record the
appropriate sex (e.g., F (female), M (male).",
   "registry_specification.context": "CDASH",
   "Value_Domain.type": "Enumerated",
   "Value_Domain.datatype.name": "xsd:string",
   "Permissible_Values": [
           {
                   "permitted_value": "F",
                   "value_meaning.designation.sign": "Female",
                   "begin_date": "2016-01-01"
           },
           {
                   "permitted_value": "M",
                   "value_meaning.designation.sign": "Male",
                   "begin_date": "2016-01-01"
           }
   ]
}]
```

4.4 Retrieve metadata

4.4.1 General

The metadata consumer uses the retrieve metadata transaction to retrieve the metadata of a selected data element from the metadata source. The interaction diagram is depicted in Figure 2. In 4.4.2 through 4.4.4 the triggering events for the request and response messages involved in this transaction, the message semantics of the request and response messages and the expected action that needs to be carried out by the receiving actors are presented.

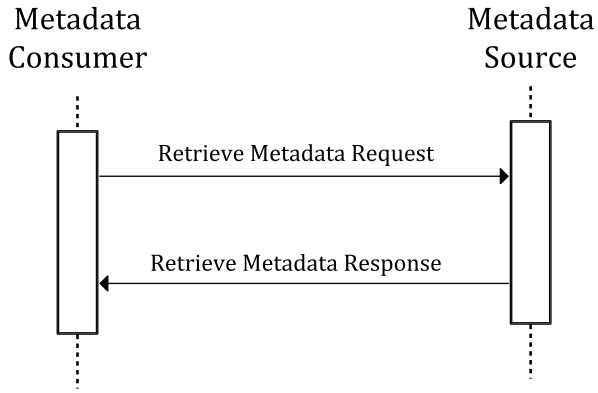

Figure 2 — Retrieve metadata transaction

4.4.2 Retrieve metadata request

4.4.2.1 Trigger events

The metadata consumer wants to retrieve a specific data element. The metadata consumer knows the identifier (ID) of the data element, either by performing a retrieve data element list transaction or by other means not defined by this document.

4.4.2.2 Message semantics

The retrieve metadata request should carry the following information presented in Table 13:

— A required identifier that identifies the data element.

— A required registration authority identifier that indicates the authority who has defined and registered the data element to the metadata source.

— An optional version that identifies a specific version of the data element. If no version is specified, the metadata consumer is requesting the most recent version of the data element.

The string data type corresponds to xsd:string.

Table 13 — Summary of the attributes in the retrieve metadata request message

Attribute name	Optionality	Type	Description
identifier	R	string	The universally unique identifier of the data element.
registration_authority_identifier	R	string	The authority who has defined and registered the data element to the metadata source (EXAMPLES CDISC, HITSP, NCI).
version	O	string	The version of the data element. If no version is specified, the metadata consumer is requesting the most recent version of the data element.

In 4.4.4, the protocol requirements and the format of the message are described in full detail.

4.4.2.3 Expected action

When receiving a retrieve metadata request, a metadata source should generate a retrieve metadata response containing the metadata of the data element that corresponds to the request parameters or

an error code if the data element could not be retrieved. If no version is specified in the request, then the most recent version should be returned.

The following error responses may be returned:

a) A SOAP fault, whose code value is NAV, with the reason being: “unknown data element”.

b) A SOAP fault, whose code value is VERUNK, with the reason being: “version unknown”.

4.4.3 Retrieve metadata response

4.4.3.1 Trigger events

This message will be triggered by a retrieve metadata request message.

4.4.3.2 Message Semantics

The retrieve metadata response message should carry the metadata of the data element presented in Table 1. In 4.4.4, the protocol requirements and the format of the message are described in full detail.

4.4.3.3 Expected action

A metadata consumer processes the data element according to its business process logic.

4.4.4 Protocol requirements

4.4.4.1 General

There are two different protocol requirements for the retrieve metadata transaction. The protocol of this transaction should be based on SOAP 1.2 and it should support HTTP REST.

SOAP

The relevant XML namespace definitions can be seen in Table 12.

These are the requirements for the retrieve metadata transaction presented in the order in which they should appear in the WSDL definition (see Annex A for an informative WSDL):

— The following types should be included (xsd:include) in the /definitions/types:

  — The /definitions/message/part/@element attribute of the retrieve metadata request message should be defined as “dex:RetrieveMetadataRequest”

  — The /definitions/message/part/@element attribute of the retrieve metadata response message should be defined as “dex:RetrieveMetadataResponse”

  — The /definitions/portType/operation/input/@message attribute for the retrieve metadata operation should be defined as “dex:RetrieveMetadataRequestMessage”

  — The /definitions/portType/operation/output/@message attribute for the retrieve metadata operation should be defined as “dex:RetrieveMetadataResponseMessage”

  — The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:iso:it:metadata:dex:2016:RetrieveMetadata”

These are the requirements that affect the wire format of the SOAP message. The other WSDL properties are only used within the WSDL definition and do not affect interoperability. Full sample request and response messages are in 4.4.4.2 and 4.4.4.3.

A full XML schema document for the DEX types is available in Annex A.

REST

For the HTTP REST binding, this transaction is required to be served under "/Metadata" endpoint after the base endpoint which can be set according to the requirements of the serving host. One example can be http://example.com/service/mdr/api/2016/Metadata. Retrieve metadata transaction should bind to HTTP GET method of the identified endpoint.

The protocol requirements for the request and response messages are given in 4.4.4.2 and 4.4.4.3.

4.4.4.2 Retrieve metadata request message

SOAP

Within the request message payload the <dex:RetrieveMetadataRequest/> element is defined as:

— A required /dex:RetrieveMetadataRequest/dex:identifier element that contains the ID of the requested data element within the metadata source

— A required /dex:RetrieveMetadataRequest/dex:registration_authority_identifier element with type "xsd:string"

— An optional /dex:RetrieveMetadataRequest/dex:version element with type "xsd:string"

A sample retrieve metadata SOAP Request is given as follows:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsa=http://www.
w3.org/2005/08/addressing
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Header>
        <wsa:MessageID>urn:uuid:f43f7bda-a5f9-42b1-b8dc-e78be1a2a183</wsa:MessageID>
        <wsa:Action>urn:iso:it:metadata:dex:2016:RetrieveMetadata</wsa:Action>
   </soap:Header>
   <soap:Body>
        <dex:RetrieveMetadataRequest xmlns:dex="urn:iso:it:metadata:dex:2016">
               <dex:identifier>8426f5a8-712f-11e7-8cf7-a6006ad3dba0</dex:identifier>
               <dex:registration_authority_identifier>
                    CDISC:ClinicalResearch:DataElements
               </dex:registration_authority_identifier>
               <dex:version>0.1</dex:version>
        </dex:RetrieveMetadataRequest>
   </soap:Body>
</soap:Envelope>
```

REST

Since the transaction is bound to HTTP GET, the parameters are required to be query parameters which will be visible in the query link. The query parameters should be given in a filter query within the format "filter=name:operator:value". The name of the parameter should be followed by a colon and then the operator followed by another colon, and finally the value for that parameter. It should be noted that the value itself may contain colons. The names of the HTTP query parameters are the same as the ones identified for the SOAP binding of the retrieve metadata request message, that is the attribute names under <dex: RetrieveMetadataRequest />.

Only valid operator is "equals" for the query parameters for this request.

A sample request message which is the REST correspondent of the give SOAP Request example is given as follows:

http://example.com/service/mdr/api/2016/Metadata/208426f5a8-712f-11e7-8cf7-a6006ad3dba0?filter=registration_authority_identifier:equals:CDISC:ClinicalResearch:DataElements&filter=version:equals:0.1

4.4.4.3 Retrieve metadata response message

SOAP

Metadata source should include within the response message payload for the SOAP binding option the <dex:RetrieveMetadataResponse/> element which contains:

— A required /dex:RetrieveMetadataResponse/dex:Data_Element element, containing

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:identifier element with type “xsd:string”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:registration_authority_identifier element with type “xsd:string”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:version element with type “xsd:string”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:designation.sign element with type “xsd:string”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:definition.text element with type “xsd:string”

  — an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:registry_specification.context element with type “xsd:string”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:creation_date element with type “xsd:date”

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:effective_date element with type “xsd:date”

  — an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:until_date element with type “xsd:date” (Required if available)

  — an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:last_change_date element with type “xsd:date” (Required if available)

  — an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:change_description element with type “xsd:string” a required (Required if available)

  — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Data_Element_Concept containing

    — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Data_Element_Concept/dex:identifier with type “xsd:string”

    — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:dataElementConcept/dex:version element with type “xsd:string”

    — a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:dataElementConcept/dex:designation.sign element with type “xsd:string”

- an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Data_Element_Concept/dex:object_class.designation.sign element with type “xsd:string”
- an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Data_Element_Concept/dex:property.designation.sign element with type “xsd:string”

— a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain element containing

- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:identifier element with type “xsd:string”
- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:type element with type “xsd:string”. The value should be “Defined”, “Enumerated” or “Described”.
- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:datatype.name element with type “xsd:string”.
- an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:unit_of_measure element with type “xsd:string” (Required if available)
- an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:source_uri element with type “xsd:string”. (Required if the type of this Value Domain is “Defined”).
- zero or more /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:Permissible_Value element containing (Required if available)
  - a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:Permissible_Value/dex:permitted_value element with type “xsd:string”
  - a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:Permissible_Value/dex:value_meaning.designation.sign element with type “xsd:string”
  - a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:Permissible_Value/dex:begin_date element with type “xsd:date”
  - an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Value_Domain/dex:Permissible_Value/dex:end_date element with type “xsd:date”

— one or more /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification element containing

- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:Target_Data_Model element containing
  - a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:Target_Data_Model/dex:name element with type “xsd:string”
  - a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:Target_Data_Model/dex:description element with type “xsd:string”
  - an optional /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:Target_Data_Model/dex:url element with type “xsd:string”
- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:type element with type “xsd:string”
- a required /dex:RetrieveMetadataResponse/dex:Data_Element/dex:Mapping_Specification/dex:mapping_script element with type “xsd:string”

A sample retrieve metadata SOAP response is given as follows:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
   <soap:Header>
         <wsa:Action>urn:iso:it:metadata:dex:2016:RetrieveMetadataResponseResponse</
wsa:Action>
         <wsa:RelatesTo>urn:uuid:f43f7bda-a5f9-42b1-b8dc-e78be1a2a183</wsa:RelatesTo>
   </soap:Header>
   <soap:Body>
         <dex:RetrieveMetadataResponse xsi:schemaLocation="urn:iso:it:metadata:dex:2016
iso-dex.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dex="urn:iso:it:
metadata:dex:2016">
               <dex:Data_Element>
                     <dex:identifier>8426f5a8-712f-11e7-8cf7-a6006ad3dba0</
dex:identifier>

   <dex:registration_authority_identifier>CDISC:ClinicalResearch:DataElements</
dex:registration_authority_identifier>
                     <dex:version>0.1</dex:version>
                     <dex:designation.sign>DMSEX</dex:designation.sign>
                     <dex:definition.text>The assemblage of physical properties or
qualities by which male is distinguished from female; the physical difference between male
and female; the distinguishing peculiarity of male or female (NCI - CDISC Definition).
Record the appropriate sex (e.g., F (female), M (male).</dex:definition.text>
                     <dex:registry_specification.context>CDASH</dex:registry_
specification.context>
                     <dex:creation_date>2010-01-01</dex:creation_date>
                     <dex:effective_date>2010-01-01</dex:effective_date>
                     <dex:until_date>2020-01-01</dex:until_date>
                     <dex:Data_Element_Concept>
                           <dex:identifier>42ca71ce-7149-11e7-8cf7-a6006ad3dba0</
dex:identifier>
                           <dex:version>0.1</dex:version>
                           <dex:designation.sign>SEX</dex:designation.sign>

   <dex:object_class.designation.sign>DM</dex:object_class.designation.sign>
                           <dex:property.designation.sign>SEX</dex:property.
designation.sign>

                     </dex:Data_Element_Concept>
                     <dex:Value_Domain>
                           <dex:identifier>926c6908-7149-11e7-8cf7- a6006ad3dba0</
dex:identifier>
                           <dex:type>Enumerated</dex:type>
                           <dex:datatype.name>xsd:string</dex:datatype.name>
                           <dex:Permissible_Value>
```

```
                                    dex:permitted_value>F</dex:permitted_value>

   <dex:value_meaning.designation.sign>Female</dex:value_meaning.designation.sign>
                                    <dex:begin_date>2016-01-01</dex:begin_date>
                              </dex:Permissible_Value>
                              <dex:Permissible_Value>
                                    <dex:permitted_value>M</dex:permitted_value>

   <dex:value_meaning.designation.sign>Male</dex:value_meaning.designation.sign>
                                   <dex:begin_date>2016-01-01</dex:begin_date>
01</dex:begin_date>
                              </dex:Permissible_Value>
                        </dex:Value_Domain>
                        <dex:Mapping_Specification>
                              <dex:Target_Data_Model>
                                    <dex:name>HL7 CCD</dex:name>
                                    <dex:description>HL7 Continuity of Care Document</
dex:description>
                              </dex:Target_Data_Model>
                              <dex:type>XPATH</dex:type>

   <dex:mapping_script>./ClinicalDocument/recordTarget/patientRole/patient/
administrativeGenderCode</dex:mapping_script>

                        </dex:Mapping_Specification>
                  </dex:Data_Element>
         </dex:RetrieveMetadataResponse>
   </soap:Body>
</soap:Envelope>
```

REST

The REST binding should return a JSON serialization of the retrieve metadata response message. The JSON response is a JSON object representing dex:Data_Element conceptually. Metadata source should include within the response message payload for the HTTP REST binding option a JSON object representing the Data_Element.

A sample retrieve matadata JSON response is given as follows:


```
{
   "identifier": "8426f5a8-712f-11e7-8cf7-a6006ad3dba0",
   "registration_authority_identifier": "CDISC:ClinicalResearch:DataElements",
   "version": "0.1",
   "designation.sign": "DMSEX",
   "definition.text": "The assemblage of physical properties or qualities by which male is distinguished from
female; the physical difference between male and female; the distinguishing peculiarity of male or female (NCI
- CDISC Definition). Record the appropriate sex (e.g., F (female), M (male).",
   "registry_specification.context": "CDASH",
   "creation_date": "2010-01-01",
   "effective_date": "2010-01-01",
   "until_date": "2020-01-01",
   "Data_Element_Concept": {
        "identifier": "42ca71ce-7149-11e7-8cf7-a6006ad3dba0",
        "version": "0.1",
        "designation.sign": "SEX",
        "object_class.designation.sign": "DM",
        "property.designation.sign": "SEX"
   },
   "Value_Domain": {
        "identifier": "926c6908-7149-11e7-8cf7-a6006ad3dba0",
        "type": "Enumerated",
        "datatype.name": "xsd:string",
        "Permissible_Values": [
              {
                     "permitted_value": "F",
                     "value_meaning.designation.sign": "Female",
                     "begin_date": "2016-01-01"
              },
              {
                     "permitted_value": "M",
                     "value_meaning.designation.sign": "Male",
                     "begin_date": "2016-01-01"
              }
        ]
   },
   "Mapping_Specifications": [
        {
              "Target_Data_Model": {
                     "name": "HL7 CCD",
                     "description": "HL7 Continuity of Care Document"
        },
        "type": "XPATH",
        "mapping_script":
"./ClinicalDocument/recordTarget/patientRole/patient/administrativeGenderCode"
        }
   ]
}
```

Annex A
(informative)

A.1 Schema and WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
XML Schema for IHE Data Element Exchange Profile (DEX) with XDS/MPQ/XCA Document Type
Binding Option
for use in WSDL definitions.
-->
<?xml version="1.0" encoding="UTF-8"?>

<!--

XML Schema for ISO/IEC Data Element Exchange (DEX) Standard with XDS/MPQ/XCA Document Type
Binding Option for use in WSDL definitions.

-->

<xsd:schema xmlns="urn:iso:it:metadata:dex:2016" xmlns:dex="urn:iso:it:metadata:dex:2016"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://www.w3.org/2005/08/
addressing" targetNamespace="urn:iso:it:metadata:dex:2016" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xsd:element name="RetrieveMetadataRequest" type="dex:RetrieveMetadataRequestType" />
  <xsd:element name="RetrieveMetadataResponse" type="dex:RetrieveMetadataResponseType" />
  <xsd:element name="RetrieveDataElementListRequest" type="dex:RetrieveDataElementListRequ
estType" />
  <xsd:element name="RetrieveDataElementListResponse" type="dex:RetrieveDataElementListRes
ponseType" />

  <xsd:complexType name="RetrieveMetadataRequestType">
    <xsd:sequence>
      <xsd:element name="identifier" type="xsd:string" />
      <xsd:element name="registration_authority_identifier" type="xsd:string" />
      <xsd:element name="version" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="RetrieveMetadataResponseType">
    <xsd:sequence>
      <xsd:element name="Data_Element" type="dex:DataElementType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="RetrieveDataElementListRequestType">
    <xsd:sequence>
      <xsd:element name="filter" type="dex:FilterType" minOccurs="0" maxOccurs="unbounded"
/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="FilterType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="operator" type="dex:OperatorType" />
      <xsd:element name="value" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="OperatorType" final="restriction" >
```

```
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="equals" />
        <xsd:enumeration value="match" />
        <xsd:enumeration value="before" />
        <xsd:enumeration value="after" />
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="RetrieveDataElementListResponseType">
    <xsd:sequence>
      <xsd:element name="Data_Element_Summary" type="dex:DataElementSummaryType"
minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DataElementSummaryType">
    <xsd:sequence>
      <xsd:element name="identifier" type="xsd:string" />
      <xsd:element name="registration_authority_identifier" type="xsd:string" />
      <xsd:element name="version" type="xsd:string" />
      <xsd:element name="designation.sign" type="xsd:string" />
      <xsd:element name="definition.text" type="xsd:string" />
      <xsd:element name="registry_specification.context" type="xsd:string"  />
      <xsd:element name="Value_Domain.type" type="dex:ValueDomainTypeType" />
      <xsd:element name="Value_Domain.datatype.name" type="xsd:string" />
      <xsd:element name="Value_Domain.source_uri" type="xsd:string" minOccurs="0" />
      <xsd:element name="Permissible_Value" type="dex:PermissibleValueType" minOccurs="0"
maxOccurs="3" />
     </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DataElementType">
    <xsd:sequence>
      <xsd:element name="identifier" type="xsd:string" />
      <xsd:element name="registration_authority_identifier" type="xsd:string" />
      <xsd:element name="version" type="xsd:string" />
      <xsd:element name="designation.sign" type="xsd:string" />
      <xsd:element name="definition.text" type="xsd:string" />
      <xsd:element name="registry_specification.context " type="xsd:string" minOccurs="0"
/>
      <xsd:element name="creation_date" type="xsd:date" />
      <xsd:element name="effective_date" type="xsd:date" />
      <xsd:element name="until_date" type="xsd:date" minOccurs="0" />
      <xsd:element name="last_change_date" type="xsd:date" minOccurs="0" />
      <xsd:element name="change_description" type="xsd:string" minOccurs="0" />
      <xsd:element name="Data_Element_Concept" type="dex:DataElementConceptType" />
      <xsd:element name="Value_Domain" type="dex:ValueDomainType" />
      <xsd:element name="Mapping_Specification" type="dex:MappingSpecificationType"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="DataElementConceptType">
    <xsd:sequence>
      <xsd:element name="identifier" type="xsd:string" />
      <xsd:element name="version" type="xsd:string" />
      <xsd:element name="designation.sign" type="xsd:string" />
      <xsd:element name="object_class.designation.sign" type="xsd:string" minOccurs="0" />
      <xsd:element name="property.designation.sign" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="ValueDomainType">
    <xsd:sequence>
      <xsd:element name="identifier" type="xsd:string" />
      <xsd:element name="type" type="dex:ValueDomainTypeType" />
      <xsd:element name="datatype.name" type="xsd:string" />
      <xsd:element name="unit_of_measure" type="xsd:string" minOccurs="0" />
      <xsd:element name="source_uri" type="xsd:string" minOccurs="0" />
      <xsd:element name="Permissible_Value" type="dex:PermissibleValueType" minOccurs="0"
maxOccurs="unbounded" />
```

```
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="ValueDomainTypeType" final="restriction">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Defined" />
        <xsd:enumeration value="Enumerated" />
        <xsd:enumeration value="Described" />
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="PermissibleValueType">
    <xsd:sequence>
      <xsd:element name="permitted_value" type="xsd:string" />
      <xsd:element name="value_meaning.designation.sign" type="xsd:string" />
      <xsd:element name="begin_date" type="xsd:date" />
      <xsd:element name="end_date" type="xsd:date" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="MappingSpecificationType">
    <xsd:sequence>
      <xsd:element name="Target_Data_Model" type="dex:TargetDataModelType" />
      <xsd:element name="type" type="xsd:string" />
      <xsd:element name="mapping_script" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="TargetDataModelType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="description" type="xsd:string" />
      <xsd:element name="url" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!--

      IHE Data Element Exchange Profile (DEX) WSDL definition.

-->

<wsdl:definitions
      xmlns="urn:iso:it:metadata:dex:2016"
      targetNamespace="urn:iso:it:metadata:dex:2016"
      xmlns:dex="urn:iso:it:metadata:dex:2016"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
      xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:wsa="http://www.w3.org/2005/08/addressing">

      <wsdl:types>
            <xsd:schema targetNamespace="urn:iso:it:metadata:dex:2016">
                  <xsd:include schemaLocation="DEX.xsd"/>
            </xsd:schema>
      </wsdl:types>

      <wsdl:message name="RetrieveMetadataRequestMessage">
            <wsdl:part name="body" element="dex:RetrieveMetadataRequest"/>
      </wsdl:message>
      wsdl:message name="RetrieveMetadataResponseMessage">
            <wsdl:part name="body" element="dex:RetrieveMetadataResponse"/>
      </wsdl:message>
```

```
	<wsdl:message name="RetrieveDataElementListRequestMessage">
		<wsdl:part name="body" element="dex:RetrieveDataElementListRequest"/>
	</wsdl:message>
	<wsdl:message name="RetrieveDataElementListResponseMessage">
		<wsdl:part name="body" element="dex:RetrieveDataElementListResponse"/>
	</wsdl:message>


	<wsdl:portType name="DataElementExchangePortType">
		<wsdl:operation name="RetrieveMetadata">
			<wsdl:input message="dex:RetrieveMetadataRequestMessage"/>
			<wsdl:output message="dex:RetrieveMetadataResponseMessage"/>
		</wsdl:operation>
		<wsdl:operation name="RetrieveDataElementList">
			<wsdl:input message="dex:RetrieveDataElementListRequestMessage"/>
			<wsdl:output message="dex:RetrieveDataElementListResponseMessage"/>
		</wsdl:operation>


	</wsdl:portType>

	<wsdl:binding name="DataElementExchangeBinding" type="dex:DataElementExchangePortT
ype">
		<soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/
http"/>
		<wsdl:operation name="RetrieveMetadata">
			<soap12:operation soapAction="urn:iso:it:metadata:dex:2016:RetrieveMet
adata"/>
			<wsdl:input>
				<soap12:body use="literal"/>
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			</wsdl:input>
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			<wsdl:output>
				<soap12:body use="literal"/>
			</wsdl:output>
		</wsdl:operation>
		<wsdl:operation name="RetrieveDataElementList">
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			<soap12:operation soapAction="urn:iso:it:metadata:dex:2016:RetrieveDat
aElementList"/>
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			<wsdl:input>
				<soap12:body use="literal"/>
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			</wsdl:input>
soapAction="urn:iso:it:metadata:dex:2016:RetrieveMetadata"/>
			<wsdl:output>
				<soap12:body use="literal"/>
			</wsdl:output>
		</wsdl:operation>


	</wsdl:binding>

	<wsdl:service name="DataElementExchangeService">
		<wsdl:documentation>SOAP Web Service for IHE Data Element Exchange Profile</
wsdl:documentation>
		<wsdl:port name="DataElementExchangePort" binding="dex:DataElementExchangeBin
ding">
			<soap12:address/>
		</wsdl:port>
	</wsdl:service>

</wsdl:definitions>
```

Annex B
(informative)

B.1 JSON schema of Data_Element for the REST binding

```
{
  "$schema": "http://json-schema.org/schema#",
  "title": "Data_Element",
 "description": "Data_Element object for the REST Binding of the ISO/IEC Data Element Exchange (DEX)",
  "type": "object",
  "properties": {
    "identifier": {
     "type": "string"
    },
      "registration_authority_identifier": {
      "type": "string"
    },
      "version": {
      "type": "string"
    },
      "designation.sign": {
      "type": "string"
    },
      "definition.text": {
      "type": "string"
    },
      "registry_specification.context": {
      "type": "string"
    },
      "creation_date": {
      "type": "string"
```

```
},
   "effective_date": {
   "type": "string"
},
   "until_date": {
   "type": "string"
},
   "last_change_date": {
   "type": "string"
},
   "change_description": {
   "type": "string"
},
   "Data_Element_Concept": {
   "type": "object",
   "properties": {
      "identifier": {
         "type": "string"
      },
      "version": {
         "type": "string"
      },
      "designation.sign": {
         "type": "string"
      },
      "object_class.designation.sign": {
         "type": "string"
      },
      "property.designation.sign": {
         "type": "string"
      }
   },
   "required": ["identifier", "version", "designation.sign"]
```

```
},
  "Value_Domain": {
  "type": "object",
  "properties": {
    "identifier": {
      "type": "string"
    },
    "type": {
      "type": "string",
      "enum": ["Defined", "Enumerated", “Described"]
    },
    "datatype.name": {
      "type": "string"
    },
    "unit_of_measure": {
      "type": "string"
    },
    "source_uri": {
      "type": "string"
    },
    "Permissible_Values": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "permitted_value": {
            "type": "string"
          },
          "value_meaning.designation.sign": {
            "type": "string"
          },
          "begin_date": {
            "type": "string"
```