

**Information technology – Metamodel framework for interoperability (MFI)
– Part 13: Metamodel for Form Registration**

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard

Document subtype:

Document stage: (20) Preparatory

Document language: E

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

ISO copyright office

Case postale 56 • CH-1211 Geneva 20

Tel. + 41 22 749 01 11

Fax + 41 22 749 09 47 E-mail copyright@iso.ch

Web www.iso.ch

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

Foreword	vii
Introduction.....	viii
1 Scope	1
2 Conformance	1
2.1 General	1
2.2 Degree of conformance	1
2.2.1 General	1
2.2.2 Strictly conforming implementation.....	1
2.2.3 Conforming implementation.....	2
2.3 Implementation Conformance Statement (ICS).....	2
3 Normative references	2
4 Terms, definitions and abbreviated terms	2
4.1 Terms and definitions	2
4.2 Terms for concepts used in this part of 19763.....	2
4.2.1	2
4.2.2	2
4.2.3	3
4.2.4	3
4.2.5	3
4.2.6	3
4.2.7	3
4.2.8	3
4.2.9	3
4.2.10	4
4.2.11	4
4.3 Abbreviated terms	4

5	Structure of MFI form registration	4
5.1	Overview of MFI form registration	4
5.2	Detail provided in each metaclass definition	8
5.3	Metaclasses in MFI for Form registration	2
5.3.1	Form_Design.....	2
5.3.2	Form_Language.....	2
5.3.3	Form_Template.....	2
5.3.4	Compliance_Rule	3
5.3.5	Form_Design_Element.....	3
5.3.6	Section_Element.....	3
5.3.7	Presentation_Element.....	4
5.3.8	Text_Element	4
5.3.9	Media_Element	4
5.3.10	Question_Element.....	5
5.3.11	Prompt.....	5
5.3.12	Input_Field.....	6
5.3.13	Text_Field.....	6
5.3.14	List_Field.....	6
5.3.15	Lookup_Field	7
5.3.16	List_Item	7
5.3.17	Question_Number	7
5.3.18	Question	8
5.3.19	Additional_Text.....	8
5.3.20	Local_Definition.....	8
5.3.21	Expression	9
5.3.22	Constant	9
5.3.23	Variable.....	9
5.3.24	Binary_Operation	9
	Annex A (informative) Description of the metamodel	11

5.4	Structure	11
5.4.1	Ordering	11
5.4.2	Containment and repetition	11
5.4.3	Questions and answers	11
5.5	Logic	11
5.6	Representation	12
5.7	Templates	12
Annex B (informative) Relationship of metaclasses to the MDR Metamodel		13
Annexe C (informative) Example design patterns		14
Annex D (informative) Bibliography		15

Figures

Figure 1: Form Structure Metamodel	5
Figure 2: Inheritance model for ISO19763-13	6
Figure 3: Question structure detail	6
Figure 4: Local definition	7

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 19763-13 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 32, *Data management and Interchange*.

ISO/IEC 19763 consists of the following parts, under the general title *Information technology — Metamodel framework for interoperability (MFI)*:

Part 1: Reference model

Part 3: Metamodel for ontology registration

Part 5: Metamodel for process model registration

Part 6: Registry Summary

Part 7: Metamodel for service registration

Part 8: Metamodel for role and goal registration

Part 9: On Demand Model Selection (ODMS) [Technical Report]

Part 10: Core model and mapping

Part 11: Advanced Mapping [Technical Report]

Part 12: Metamodel for information model registration

Part 13: Metamodel for form registration

Introduction

There is an increasing demand for systems to interoperate by exchanging data. For these data exchanges to be meaningful it is essential that the business information requirements that are met by the data stored in these systems are understood so that suitable data exchange mechanisms can be developed.

Not only does this require a clear understanding of the meaning of the data, it also frequently requires the coordination of data capture. Where data input is manual, an important source of data semantics is the design and behavior of the form used for data entry - indeed if we do not understand the encoding of knowledge in the database schema or we suspect some anomaly in the data captured, we inspect the original form and the context of its use. Furthermore, if we wish to gather interoperable data, it is frequently necessary to harmonise aspects of form design and behavior before data is captured.

This need is recognised and addressed locally in a number of commercial and open source products, particularly in the medical research domain. RedCAP, OpenClinica, Oracle Clinical, Medidata Rave, and the NCI caDSR all provide facilities to design, deploy and share form designs. OpenClinica additionally will output parts of its form design in the Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM) which provides some standard facilities for the basic description of forms that have been used to collect medical research data. However, there is no abstract, universal metamodel for form registration that allows the registration of forms, the faithful exchange of form designs between systems and supports the definition of standard instruments for the coordination of data capture. This is the intent of 19763-13.

Given a standard metamodel for the registration of forms, ISO/IEC 19763 Metamodel Framework for Interoperability (MFI) and ISO/IEC 11179 Metamodel for metadata registries provide important facilities for the creation and annotation of form designs. ISO/IEC19763 supports the registration of forms and form sections as models and model elements, provides facilities to record associations between form sections - particularly derivation, specialisation, extension and reuse - and allows the association of forms with the data models that are used to store data captured by their instances. ISO/IEC11179 supports the creation and exchange of standard question banks and defines a rich source of question-level metadata attributes with which to explain the meaning of individual data items. Together, both standards can support the rapid design and reuse of forms, wrap and hide the complexity of semantic annotation from subject matter experts, and provide a ready reference of associations and transformations for users seeking to collect and use interoperable data.

Information technology – Metamodel framework for interoperability (MFI) – Part 13: Metamodel for Forms

1 Scope

The primary purpose of the multipart standard ISO/IEC 19763 is to specify a metamodel framework for interoperability. This part of ISO/IEC 19763 specifies a metamodel for registering forms.

The Oxford English Dictionary defines a form as 'a formulary document with blanks for the insertion of particulars'.

This standard provides a metamodel to describe the structure and semantics of a form devoid of any specific, domain semantics, for purposes that include: the registration and documentation of form designs, both paper and electronic; the association of form designs with appropriate entity-relationship diagrams and data models (ISO/IEC19763-12) so that data may be faithfully exchanged between system components and between systems; the association between a set of forms whose data maybe compared, joined or composed for analysis.

The standard does not supplant or replace computer languages such as XForms, Windows forms, InfoPath, Adobe Forms or relevant parts of HTML which describe how a form model might be implemented within an application framework. The standard is devoid of any domain semantics and thus abstracts some common functions between standards in the healthcare domain such as Health Level 7 (HL7) Clinical Document Architecture (CDA), the OpenEHR Archetype Definition Language (ADL) and the Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM).

Editors note: too healthcare specific – look for other examples. Applies to whole of document

2 Conformance

2.1 General

An implementation claiming conformance with this part of ISO/IEC 19763 shall support the metamodel specified in 5.3, depending on a degree of conformance as described below.

2.2 Degree of conformance

2.2.1 General

The distinction between “strictly conforming” and “conforming” implementations is necessary to address the simultaneous needs for interoperability and extensions. This part of ISO/IEC 19763 describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions and industries, but are not specified by this part of ISO/IEC 19763.

A strictly conforming implementation may be limited in usefulness but is maximally interoperable with respect to this part of ISO/IEC 19763. A conforming implementation may be more useful, but may be less interoperable with respect to this part of ISO/IEC 19763.

2.2.2 Strictly conforming implementation

A strictly conforming implementation

- a) shall support the metamodel specified in 5;
- b) shall not support any extensions to the metamodel specified in 5.

2.2.3 Conforming implementation

A conforming implementation

- a) shall support the metamodel specified in 5;
- b) may support extensions to the metamodel specified in 5 that are consistent with the metamodel specified in 5.

2.3 Implementation Conformance Statement (ICS)

An implementation claiming conformance with this part of ISO/IEC 19763 shall include an Implementation Conformance Statement stating

- a) whether it is a strictly conforming implementation or a conforming implementation (2.2);
- b) what extensions are supported if it is a conforming implementation.

Conformance statements for systems that implement this standard shall additionally describe the languages used to convey slotRepresentation, scopeRepresentation and Constraints, and the relationship types available for the Mapping_Relation class.

3 Normative references

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

ISO/IEC 19763-1, Information technology – Metamodel framework for interoperability (MFI) – Part 1: Reference model

ISO/IEC 19763-10, Information technology – Metamodel framework for interoperability (MFI) – Part 10: Core model and basic mapping

ISO/IEC 11179-3 (full title to be inserted)

4 Terms, definitions and abbreviated terms

4.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19763-1, ISO/IEC 19763-10, ISO/IEC 11179-3:[Ed 3 - date to follow] and the following apply.

4.2 Terms for concepts used in this part of 19763

Editors note: Format of definitions does not comply with the directive. Should be as 4.2.1/4.1.2 Should be able to take out word, put in definition and it still make sense – so phrase the

4.2.1

Annotation

note

Note: made on an item of interest by way of explanation or comment; a reference from an item to arbitrary metadata whose type is not directly modeled

4.2.2

Binary large object

BLOB

collection of binary data stored as a single entity in a database management system

Note: typically images, audio or other multimedia objects

4.2.3**Constraint**

limitation or constriction

4.2.4**Form**

formulary document with blanks for the insertion of particulars

Note: a particular form includes

- The amazon.com new customer registration form
- The UK HMRC form SA1 for obtaining a unique taxpayer reference for self-assessment of income tax

4.2.5**Predicate**

property or relation

4.2.6**String**

datatype composed by a finite sequence of characters

4.2.7**Form template**

partial form design that establishes a pattern for the creation of other form designs

Note: instances of form templates include

- The Cancer Genome Atlas (TCGA) standard participant follow up form
- The Royal College of Pathology proforma for bone tumour reports

4.2.8**Compliance rule**

specification for some aspect of a form design that must be satisfied for that design to be a correct implementation of a form template

4.2.9**Form design**

specification for the creation of semantically equivalent form implementations in different applications and media

Note: Instances of form designs include

- The US Cancer Therapy Evaluation Program Radiotherapy Oncology Group 0320 study Multiple Brain Metastases Followup Form design in the Cancer Data Standards repository
- The specification of the data entry webpage for the amazon.com new customer registration form

4.2.10

Form implementation

artifact created when a form design is deployed or implemented in an application framework or rendered in digital or analog media

4.2.11

Completed form

document created by completing the fields in a form implementation according to its instructions and prompts

Editor's note: careful thought must be given to the terminology as metamodel (this standard), model (a particular form design), instance (a particular form design implemented deployed to an application framework) and the document created by the implementation need to be simultaneously discussed. A complete and consistent management of this task will be essential for the first committee draft

4.3 Abbreviated terms

MFI Core and mapping

ISO/IEC 19763-10, Information technology – Metamodel Framework for Interoperability – Part-2: Core model and basic mapping

MDR Metamodel

ISO/IEC 11179-3:2013, Information technology – Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes

MFI Form registration

Information technology – Metamodel framework for interoperability (MFI) – Part 13: Metamodel for Form Registration

5 Structure of MFI form registration

5.1 Overview of MFI form registration

Figure 1 shows an overview of the metamodel for the registration of forms, figure 2 the inheritance model and relationship to MFI Core and mapping, figure 3 the detailed structure of a question element and figure 4 the model for local expressions.

The basic mapping package should be used as described in part 10, with specialised semantics for the Model_Element_Set_Mapping_Degree as included in figure 2 appropriate for the mapping between form model elements found in the enumeration Form_Element_Set_Mapping_Degree.

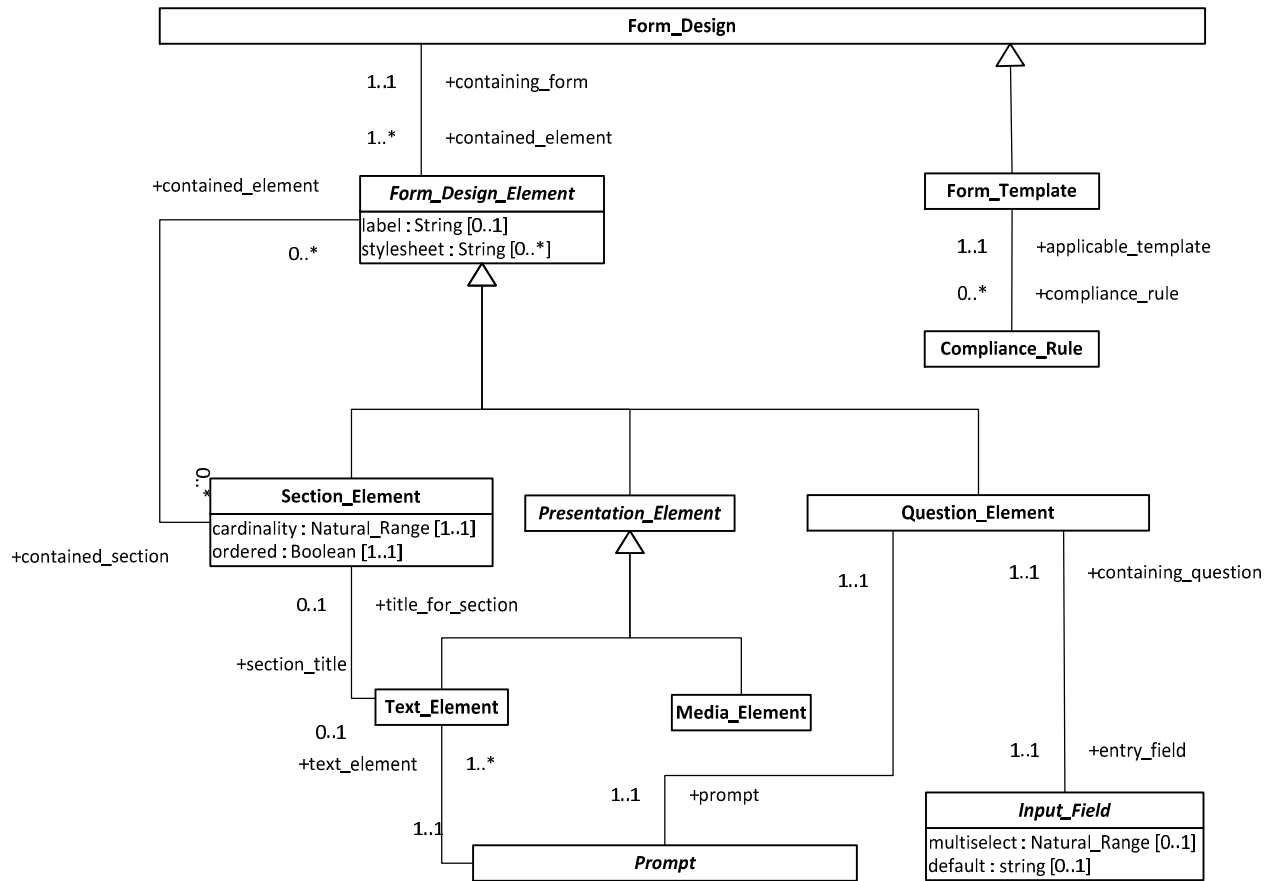


Figure 1: Form Structure Metamodel

Editor’s note – to be completed: all association names, remove public (+) annotations

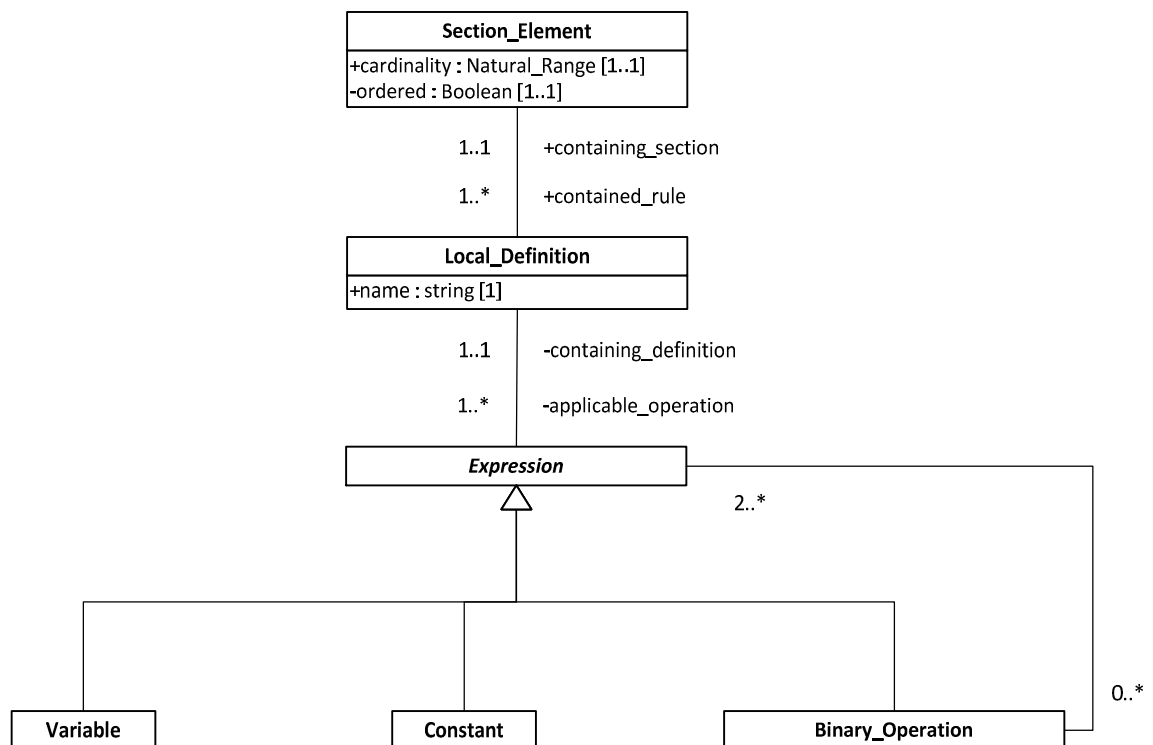


Figure 4: Local definition

The metamodel for information model registration comprises the following metaclasses:

- **Additional_Text**
- **Binary_Operation**
- **Compliance_Rule**
- **Constant**
- **Expression**
- **Form_Design**
- **Form_Template**
- **Form_Design_Element**
- **Form_Language**
- **Input_Field**
- **List_Field**
- **List_Item**
- **Local_Definition**
- **Lookup_Field**
- **Media_Element**
- **Presentation_Element**
- **Prompt**
- **Question**
- **Question_Element**
- **Question_Number**
- **Section_Element**
- **Text_Element**
- **Text_Field**
- **Variable**

The purpose and use of the metamodel is described in detail in Annex A (informative). Detailed specifications of the metaclasses are provided in Annex B (informative).

5.2 Detail provided in each metaclass definition

For each metaclass the following detail is shown:

- A definition that describes the role or significance of instances of the metaclass.
- The name of its immediate supertype.
- Any alternative names (synonyms or aliases) for the metaclass.
- A list of attributes.
- A list of references.

For each attribute the following detail is shown:

- The name of the attribute; where the attribute is one that is provided by the type defined in the MDR metamodel by which the instances of the metaclass are extended the name is italicised.
- The datatype for values of the attribute.
- The multiplicity of the attribute.
- A description that describes the role or significance of values of the attribute.

For each reference the following detail is shown:

- The name of the reference; this is the role name that describes the role played by the referenced metaclass with respect to the association identified by this reference.
- The name of the referenced metaclass.
- The multiplicity of the reference.
- A description that describes the role or significance of the instance, or instances, of the referenced metaclass with respect to an instance of this metaclass.
- The name of the reference in the referenced metaclass that provides the inverse definition for the association.
- An indication as to whether this metaclass is responsible for the maintenance of the association, i.e. the precedence of the metaclass with respect to the association.

5.3 Metaclasses in MFI for Form registration

Editor's note: definitions are draft. Table borders will be hidden in committee drafts

5.3.1 Form_Design

Form_Design is a metaclass, each instance of which represents the design of a specific form which is formulary document with blanks for the insertion of particulars.					
Superclass					
Model from MFI Core and mapping,					
Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_element	Form_Design_Element	1..*	The set of form elements that comprise this form	containing_form	yes

5.3.2 Form_Language

Form_Language is a metaclass, each instance of which represents the language used to express the design of the associated Form_Design.					
Editor's note: the 'described_by' relation inherited from 19763-10 specifies the language used in the registration of the form. Is this a single language or is this a combination of several different languages – for instance do we allow an independent declaration of the regular expression language used to specify field patterns, the version of HTML used to render a web form or the version of CSS used in its display?					
Superclass					
Model from MFI Core and mapping,					
Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_element	Form_Design_Element	1..*	The set of form elements that comprise this form	containing_form	yes

5.3.3 Form_Template

Form_Template is a metaclass, each instance of which represents a specific form template which is a partially complete form design intended to guide the creation of similar form designs					
Superclass					
Form					
Reference	Class	Multiplicity	Description	Inverse	Precedence
compliance_rule	Compliance_Rule	0..*	A rule that must be followed or satisfied to indicate compliance of some aspect of a Form_Model instance to its associated Form_Template	containing_template	yes

			instance		
--	--	--	----------	--	--

5.3.4 Compliance_Rule

Compliance_Rule is a metaclass, each instance of which is a specification of some aspect of design that must be satisfied if an instance of a Form is a valid implementation of an instance of a Form_Template.					
Superclass					
None					
Reference	Class	Multiplicity	Description	Inverse	Precedence
applicable_template	Form_Template	1..1	The instance of Form_Template to which this rule applies	compliance_rule	no

5.3.5 Form_Design_Element

Form_Design_Element is an abstract metaclass, each instance of which represents some component of an instance of the class Form_Design.					
Superclass					
Model_Element (defined in MFI Core and mapping)					
Attribute	Data Type	Multiplicity	Description		
label	String	0..1	A name, label or identifier		
stylesheet	String	0..*	A statement in some stylesheet language about component		
Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_section	Section_Element	0..*	The set of Section_Element instances contained by this instance of Form_Element	contained_element	no
containing_form	Form_Design	1..1	The instance of class Form_Design within which this Form_Design_Element instance is contained by	contained_element	no

5.3.6 Section_Element

Section_Element is a metaclass, each instance of which is a section of an instance of the class Form that describes a structural association between a set of Form_Design_Element instances.					
Superclass					
Form_Design_Element					
Attribute	Data Type	Multiplicity	Description		
cardinality	Natural_Range Editor's note This will be	1..1	The allowed range of times a Section_Element instance may be repeated in the completed by operators completing a form that conforms to the design described by the Form instance to		

	changed to two separate attributes in line with part 12		which the Section_Element instance belongs.		
ordered	Boolean	1..1	A flag that indicates if the order of child Form_Element instances is semantically significant.		
Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_element	Form_Element	0..*	The set of Form_Element instances contained by this Section_Element instance.	contained_section	yes
contained_rule	Local_Definition	0..*	A set of Local_Definitions that are used to calculate values, set defaults, and gate certain sections, questions and list items where irrelevant	containing_section	yes
sectionTitle	Text_Element	0..1	An optional association that declares a particular Text_Element within the section as its title	titleForSection	

5.3.7 Presentation_Element

Presentation_Element is an abstract metaclass, each instance of which is a presentation component of the form.
Superclass
Form_Element

5.3.8 Text_Element

Text_element is a metaclass, each instance of which is a textual presentation element of a form intended to instruct or explain to the user of the form what the data should mean, how it should be completed and any actions that must be taken with the completed form.					
Superclass					
Presentation_Element					
Reference	Class	Multiplicity	Description	Inverse	Precedence
titleForSection	Section_Element	0..1	The section element for which this Text_Element is optionally a title for	sectionTitle	

5.3.9 Media_Element

Media_Element is a metaclass, each instance of which represents some image, audio or video element within a Form
--

Superclass
Form_Element

5.3.10 Question_Element

Question_Element is a metaclass each instance of which represents a question on a form					
Superclass					
Form_Element					
Reference	Class	Multiplicity	Description	Inverse	Precedence
containedField	Input_Field	1	The input field associated with the question	containingQuestionElement	
Prompt	Prompt	1	The collection of text elements that are associated with this Question_Element and provide the user-facing instructions for its completion		

5.3.11 Prompt

Prompt is an abstract metaclass each instance of which represents the set of textual elements of a Question_Element					
Superclass					
Form_Element					
Reference	Class	Multiplicity	Description	Inverse	Precedence
additionalText	Additional_Text	0..*	Additional text intended to aid the user of the form: e.g. an indication of acceptable units of measure for the dimensions of a parcel on a parcel courier order form	questionPrompt	
containingQuestionElement	Question_Element	1..1	The question element to which this prompt belongs	prompt	
questionNumberText	Question_Number	0..1	The text of the question number providing the answer sequence of the collection of Question_Elements	questionPrompt	
questionText	Question	1..1	The text of the question providing the immediate semantics of the answer	questionPrompt	

5.3.12 Input_Field

Input_Field is an abstract metaclass each instance of which represents the part of a Question_Element which allows entry of a value					
Superclass					
Form_Element					
Attribute	DataType	Multiplicity	Description		
multiselect	Natural_Range	0..1	The number of answers to the question that may be provided. If the attribute is missing, it is assumed that the input field only allows a single answer		
default	string	0..1	The default value of the input field		
Reference	Class	Multiplicity	Description	Inverse	Precedence
containing_question_element	Question_Element	1..1	The Question_Element to which this Input_Field belongs	input_field	no

5.3.13 Text_Field

Text_Field is a metaclass each instance of which represents a field in which any value may be entered, subject to the pattern and length constraints					
Superclass					
Input_Field					
Attribute	DataType	Multiplicity	Description		
length	Int	0..1	The maximum number of characters that the Text_Field may accept. If this value is missing there is no limit on the number of characters the field may accept		
pattern	string	0..1	A pattern in a regular expression language that constrains data entry: e.g. "yyyy-mm-dd" constrains data entry in many form languages to		

5.3.14 List_Field

List_Field is a metaclass each instance of which represents a field in which only predefined answers are allowed					
Superclass					
Input_Field					

Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_listItem	List_Item	2..*	The set of pre-defined list items that are allowed answers to the question	inContainingListField	yes

5.3.15 Lookup_Field

Lookup_Field is a metaclass each instance of which represents a field which – like a List_Field – has a valid list of answers from a defined domain, but where the members of the domain vary with time and between implementations: e.g. a view providing a valid set of active customer IDs for a sales order system; a terminology approved for tagging an experimental result; a web service ; open issue lookup in bug tracking software.					
Superclass					
Input_Field					
Attribute	Data Type	Multiplicity	Description		
end_point	string	[1..*]	The location of the endpoint providing the value; a service or function call, a URI call that returns the value list		

5.3.16 List_Item

List_Item is a metaclass each instance of which represents an available answer for a list field					
Superclass					
Attribute	Data Type	Multiplicity	Description		
name	String	0..1	An optional name that allows the list item to participate in a Local_Definition, either as part of a predicate or as a consequence of a predicate		
guard	Boolean	0..1	An attribute that indicates that if the value this List_Item instance represents is selected during the completion of the form, then the collection of stopped elements will be unavailable for data input		
Reference	Class	Multiplicity	Description	Inverse	Precedence
inList	List_Field	1	The List_Field to which the List_Item belongs	hasListItem	
stopped_element	Form_Element	0..*	An association to a Form_Element instance	stopping_answer	yes

5.3.17 Question_Number

Question_Number is a metaclass each instance of which is a Text_Element instance indicating the position of the parent Question_Element instance in its containing Section_Element instance					
---	--	--	--	--	--

Superclass					
Prompt					
Reference	Class	Multiplicity	Description	Inverse	Precedence
number_for_question_element	Prompt	1	The Prompt instance to which this question number refers	question_number_text	No

5.3.18 Question

Question is a metaclass each instance of which is a Text_Element instance containing the main text of the Question_Element instance					
Superclass					
Prompt					
Reference	Class	Multiplicity	Description	Inverse	Precedence
text_for_question_element	Prompt	1	The Prompt instance to which this question refers	question_text	No

5.3.19 Additional_Text

Additional_Text is a metaclass each instance of which is a Text_Element instance holding any extra text relevant to the completion of the question					
Superclass					
Prompt					
Reference	Class	Multiplicity	Description	Inverse	Precedence
additional_text_for_question_element	Prompt	1	The Prompt instance to which this additional text refers	additional_text	No

5.3.20 Local_Definition

Local_Definition is a metaclass whose instance groups a set of mathematical and logical statements pertaining to completion order, relevancy, validation and other dependencies between Question_Element instances					
Superclass					

Reference	Class	Multiplicity	Description	Inverse	Precedence
containing_section	Section_Element	1..1	The Section_Element instance that provides the scope for the Local_Definition instance	contained_rule	No

5.3.21 Expression

Expression is an abstract metaclass which allows the recording of expressions declaring and relating Constants, Variables and Binary_Operations					
Superclass					
Reference	Class	Multiplicity	Description	Inverse	Precedence
partOfDefinition	Local_Definition	1	The Local_Definition to which the Expression belongs	hasExpression	

5.3.22 Constant

Constant is an abstract metaclass that supports the declaration of a named constant in a local expression within the scope of a Section_Element					
Superclass					
Expression					

5.3.23 Variable

Variable is an abstract metaclass that supports the declaration of a named variable in a local expression within the scope of a Section_Element					
Superclass					
Expression					

5.3.24 Binary_Operation

Text_Field is a metaclass each instance of which represents a field in which any value may be entered, subject to the pattern and length constraints					
Superclass					
Expression					

Reference	Class	Multiplicity	Description	Inverse	Precedence
hasExpression	Expression	2	The set of precisely 2 constants, variables and binary_operations associated by this binary operation		

Annex A (informative) Description of the metamodel

In the following description of the metamodel the symbol § alongside an attribute name indicates that the attribute is included in this description for completeness but is provided by the type defined in the MDR metamodel by which the metaclasses defined in this part of MFI are extended.

The metamodel for form registration is intended to provide facilities for recording the logical design of a form and associating it with - in particular - data standards (ISO11179-3), other forms and other relevant data (ISO19763-12) and logical (ISO19763-3, ISO11179-3) models. This provides structured documentation on the nature and design of an individual form, ensures that data captured according to the form might be better understood and that design decisions in the creation of the form from other forms, data models or ER diagrams may be explained in support of data sharing Master Data Management and the Semantic Web.

5.4 Structure

The structure of a form is modelled as an ordered tree of Form_Elements inside a Form container. Form_Elements may be Section_Elements, that logically group other form elements, Presentation_Elements such as textual regions, pictures and audio clips, and Question_Elements that allow the user to enter data onto a form.

Form_elements share optional name and style attributes: the name attribute allows an element to participate in a Local_Definition; the style attribute allows the registration of a stylesheet or other description of the layout of the components of a Form_Element.

It is not intended that a Question_Element can contain a Section_Element although some Presentation_Element instances could – e.g. a box

5.4.1 Ordering

Form_Elements may be ordered or un-ordered. Ordering is declared within a Section_Element and must always be selected when modelling or defining forms that have functional dependencies between questions, overt semantic links or where questions are used to skip, hide or indicate the relevance of following sections or questions. Common examples of constructs that require ordering include

- Amplification, explanation or extemporisation. Q1.prompt: what is your favourite colour?; Q2.prompt: why? Q1.prompt: Does your partner smoke?; Q2.prompt: does this bother you?
- Form workflow. Section 2 Q1.prompt: is the study subject alive? (yes/no) Following text element: 'if the answer is yes then please go to section 3'
- Functional dependency. Section_Element.Title: for the previous tax year; Q1.prompt: Total taxable income (pounds sterling); Q2.prompt: Total tax paid (pounds sterling); Section_Element.Local_Definition total_tax_paid less_than total_taxable_income/2

Order must be the normal navigation order of the form, frequently the normal reading order for the (human) language and script of the form. Ordering of question elements may be achieved through assignment of Question_Number prompts or through document order in strictly complying implementations.

5.4.2 Containment and repetition

The cardinality attribute of the form section class allows the modeling of tables: the table is a repeating Section_Element with Question_Elements for each column. Tabular presentation is effected by the stylesheet attribute.

5.4.3 Questions and answers

5.5 Logic

predicate *dead.value=true*

consequence *performance-status.relevance=false*

consequence *cause-of-death.relevance=true*

consequence *date-of-death.value > last-date-seen-alive-from-previous-followup.value*

5.6 Representation

5.7 Templates

The cardinality of the relationships is designed to allow the creation of empty `Section_Element` instances. One use of this is the definition of a `Form_Template`, where an empty section is declared for content extension. In a clinical trials followup form template, standard content would be declared to ensure that censor dates are uniformly collected, and a logical, empty, sections provided for content specific to the study implementing the template such as disease specific questions, study name and logo.

Annex B (informative) Relationship of metaclasses to the MDR Metamodel

As explained in ISO/IEC 19763 Part 10, instances of the metaclasses defined in this part of ISO/IEC 19763 may be extended by the types, or inherit from classes defined in the MDR Metamodel as follows:

- **Form** maybe of type **Identified_Item**, **Designatable_Item**, **Administered_Item** and **Classifiable_Item**.
- **Form_Element** maybe of type **Identified_Item**, **Designatable_Item** and **Classifiable_Item**. If it of type **Designatable_Item** then its *label* attribute may be omitted
- **Question_Element** may be a subclass of **Data_Element**
- **List_Item** may be a subclass of **Concept** or **Valid_Value**

Editor's note: this will contain example UML showing integrated structures

Annexe C (informative) Example design patterns

Editors note: this section will show two or three example implementation models that could be used to support interoperability between forms: one for simple form design and exchange; one that is connected to a ISO/IEC11179-3 metadata registry; one that maps form structures to a ISO19763-12 Information Model Registry

Annex D (informative) Bibliography