



ISO/IEC JTC1/SC7  
Software Engineering  
Secretariat: CANADA (SCC)

**ISO/IEC JTC1/SC7 N1541**  
**July 14, 1996**

**DOC TYPE:** CD Ballot

**TITLE:** CD 15474-2: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 2 : Framework for Modeling and Extensibility

**SOURCE:** WG11

**PROJECT:** 07.28.1.2

**STATUS:** First CD ballot

**REFERENCES:** Resolution 404, SC7 N1314

**ACTION ID:** ACT

**DUE DATE:** 1996-10-28. It would be most appreciated that a copy of the comments, if any, be send to the project 07.28 editor, Mr Jean Bérubé (jberube@ibm.net) before 1996-10-18.

**DISTRIBUTION:** P, O and L

**MEDIUM:** E-Mail - uuencoded compressed Word ; Diskette - compressed Word

**NO. OF PAGES:** 107 [347 kB]

**DISK NUMBER:** 8



<b>ISO/IEC JTC1/SC7 CD 15474-2</b>	
Date <b>1996-07-14</b>	Reference number <b>ISO/JTC 1/SC 7 N 1541</b>
Supersedes document SC 7 N 1314	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/JTC 1/SC 7 Committee Title <b>Software Engineering</b> Secretariat: Standards Council of Canada (SCC)	Circulated to P- and O-members, and to technical committees and organizations in liaison for: X discussion at Ottawa on 1996-10-21 X comment by 1996-10-21 X voting by (P-members only)  <b>1996-10-28</b>  Please return all votes and comments in electronic form directly to the SC 7 Secretariat by the due date indicated.
---	--

ISO/IEC JTC1/SC7

Title: Information Technology - Software Engineering Data  
Definition and Interchange - Overview and Framework: Part 2  
: Framework for Modeling and Extensibility

Project: 07.28.1.2

---

Introductory note: See Attachements

Medium: E-Mail - uuencoded compressed Word ; Diskette - compressed Word

No. of pages: 107

---



<b>Vote on Committee Draft ISO/IEC 15474-2</b>	
Date of circulation <b>1996-07-14</b>	Reference number ISO/JTC 1/SC 7 <b>N 1541</b>
Closing date <b>1996-10-28</b>	

ISO/JTC 1/SC 7 Committee Title <b>Software Engineering</b> Secretariat: Standards Council of Canada (SCC)	Circulated to P-members of the committee for voting  Please return all votes and comments in electronic form directly to the SC 7 Secretariat by the due date indicated.
---	--

ISO/IEC JTC1/SC7

Title: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 2 : Framework for Modeling and Extensibility

Project: 07.28.1.2

**Vote:**

- APPROVAL OF THE DRAFT AS PRESENTED
- APPROVAL OF THE DRAFT WITH COMMENTS AS GIVEN ON THE ATTACHED
  - general:
  - technical:
  - editorial:
- DISAPPROVAL OF THE DRAFT FOR REASONS ON THE ATTACHED
  - Acceptance of these reasons and appropriate changes in the text will change our vote to approval
- ABSTENTION (FOR REASONS BELOW):

P-member voting:  
National Body (Acronym)

Date:  
YYCC-MM-DD

Submitted by:

Name

ISO IEC CD 15474-2  
EIA/IS-107  
SC7 WG11 N046R1

**Information Technology - Software Engineering Data  
Definition and Interchange - Overview and Framework  
Part 2 : Framework for Modeling and Extensibility**

<b>Author:</b>	EIA CDIF	<b>Source:</b>	SC7 WG11
<b>Project:</b>	7.28.1.2	<b>Date:</b>	1996.05.31
<b>Action:</b>	Circulation to JTC1 SC7 For CD Ballot, CD Review and Comments	<b>Status:</b>	Committee Draft
<b>Reference:</b>	SC7 N1220	Resolution #343 Ottawa Plenary (94.06.17)	
	SC7 N1314	CDIF Framework Working Draft	
	SC7 N1415	Resolution #404 Brisbane Plenary (96.06.01)	

## Introduction

ISO/IEC JTC1 SC7 WG11, in its Project 7.28, has the objective of enabling the interchange of information between a variety of heterogeneous software engineering tools.

Many of the input documents for Project 7.28 are the result of developmental work performed by the CASE Data Interchange Format (CDIF) technical committee of the Electronics Industries Association (EIA). This is one of those documents, and is distributed as received from the CDIF editor.

At its Brisbane meeting, in June 95, SC7 approved the distribution of the committee drafts.

This document is intended to provide one part within the multipart standard resulting from Project 7.28. Documents will be reformatted to ISO Directives as part of the Committee Draft ballot and comment resolution.

## Comments

Comments are requested within the timetable provided by the SC7 cover sheet (generally 100 days) to facilitate preparation for the next WG11 meeting. Comments should be sent by normal formal channels.

To facilitate planning and preparatory resolution proposals, WG11 would appreciate comments to be submitted to the WWW ([www.cdif.org/comments.html](http://www.cdif.org/comments.html)) site identified for that purpose).

Comments are to be provided in electronic format, generally following the format used for the WWW page. If comments are also submitted on paper, they should use a similar format.

Please identify clearly the status of comments submitted, such as:

National Body Comments,  
National Body Approved Expert Comments.

Jean Bérubé  
Project Editor, 7.28

IDEgenic inc  
87 Burton,  
Greenfield Park, Québec,  
Canada J4V 2Y6

Tél: 1 514 465 0245  
Fax: 1 514 465 2398  
[jberube@ibm.net](mailto:jberube@ibm.net)

711PRA-11  
Explanatory Notes for SC7 Reviewers  
of the  
WG11-CDIF Documents  
(Project 07.28)

- The “Overview” document (WG11 N047R1, SC7 N1540) should be read first since it describes all the other project 7.28 CD documents to be reviewed at this time (as well as several documents planned for the future). It also describes the overall architecture that applies to these documents.
- The “Framework” document (WG11 N046R1, SC7 N1541) should be read second, as it describes the entity-relationship-attribute modeling notation that is used in many of the documents.
- The “Data Modeling” document (WG11 N036, SC7 N1548) and the “Data Flow Modeling” (WG11 N035, SC7 N1549) document provide the substantive coverage of software engineering techniques within the first set of documents. These documents permit the structured representation of software designs prepared by the use of any of a number of well established software engineering techniques that produce what are commonly called data flow diagrams and/or entity-relationship- attribute diagrams. These two documents will be of the greatest interest to software engineering practitioners.
- The “Foundation”, “Common”, and “Data Definition” documents (WG11 N048R1, N049, N050 and respectively SC7 N1545, N1546, 1547) are “modeling infrastructure” documents that support the contents of the software engineering techniques document, Data Modeling and Data Flow Modeling. Elements in these three infrastructure documents are incorporated by reference into the technique documents as necessary.
- The Transfer Format documents, “General Rules”, “Syntax”, and “Encoding” (WG11 N040R1, N041R1, N042R1 and respectively SC7 N1542, N1543, N1544) describe how character sequences are formed in order to communicate software design representations. These may be read separately from the technique and infrastructure documents described above.
- Of special importance for this review is the “Foundation” document (WG11 N048R1, SC7 N1545). This document illustrates the format intended to be applied to all the documents in this set for subsequent balloting. WG11 specifically solicits reviewer comments related to the format of this document so that the reformatting effort for the other documents in this set need be done only once.

WG11 will greatly appreciate the national body review comments on these documents, and thanks in advance the reviewers for their efforts. Any early submissions of comments will be very helpful.

## WG11 Foreword

In order to fully ensure that all comments received are processed efficiently and consistently, WG11 requests of all reviewers to please utilize the comments template that is provided with each document. Submitting comments in electronic form rather than on paper will be especially helpful.

If assistance with the usage of the comment forms is required, or if there are any other questions, please contact either Jean Berube, the Project 7.28 coordinating editor, or myself.

Sincerely,

Peter Eirich  
WG11 Convenor

**CD 15474-2****Information Technology - Software Engineering Data  
Definition and Interchange - Overview and Framework:****Part 2 : Framework for Modeling and Extensibility**

CDIF TITLE:

***CDIF - Framework for Modeling and Extensibility***

2 EIA/IS-107 January, 1994

3

4 Revised 4/24/96 to incorporate ISO/IEC JTC1 SC7/WG11 comments submitted as  
5 input to the 1995 Brisbane meeting

6 ELECTRONIC INDUSTRIES ASSOCIATION

7 ENGINEERING DEPARTMENT



1

## NOTICE

2 EIA Engineering standards and publications are designed to serve the public interest through  
3 eliminating misunderstandings between manufacturers and purchasers, facilitating  
4 interchangeability and improvement of products, and assisting the purchaser in selecting and  
5 obtaining with minimum delay the proper product for his particular need. Existence of such  
6 standards and publications shall not in any respect preclude any member or nonmember of EIA  
7 from manufacturing or selling products not conforming to such standards and publications, nor  
8 shall the existence of such standards and publications preclude their voluntary use by those other  
9 than EIA members, whether the standard is to be used either domestically or internationally.

10 Recommended standards and publications are adopted by EIA without regard to whether or not  
11 their adoption may involve patents on articles, materials or processes. By such action, EIA does  
12 not assume any liability to any patent owner, nor does it assume any obligation whatever to  
13 parties adopting the standard or publication.

14

## EIA INTERIM STANDARDS

15 EIA Interim Standards contain information deemed to be of technical value to the industry, and  
16 are published at the request of the originating Committee without necessarily following the  
17 rigorous public review and resolution of comments which is a procedural part of the development  
18 of an EIA Recommended Standard.

19 EIA Interim Standards should be reviewed on an annual basis by the formulating Committee and a  
20 decision made on whether to proceed to develop an EIA Recommended Standard on this subject.  
21 EIA Interim Standards must be canceled by the Committee and removed from the EIA Standards  
22 Catalog before the end of their third year of existence.

23 This EIA standard is considered to have international standardization implication, but the  
24 International Standardization Organization activity has not progressed to the point where a valid  
25 comparison between the EIA standard and the ISO work can be made.

26

## COPYRIGHT 1994

27 Published by  
28 ELECTRONIC INDUSTRIES ASSOCIATION  
29 Engineering Department  
30 2500 Wilson Blvd. Suite 203  
31 Arlington, VA 22201  
32

33 PRICE: Please refer to the current  
34 Catalog of EIA & JEDEC STANDARDS & ENGINEERING PUBLICATIONS  
35 or contact the EIA Electronic Information Group  
36 (Phone: (703) 527-7001, Fax: (703) 527-0685)  
37

38 All rights reserved  
39 Printed in U.S.A.  
40

41 ISBN 0-7908-0013-6

---



## PREFACE

This standard will assist the vendors and users of CASE tools in developing mechanisms for interchanging information between CASE tools. This standard specifies an element of a family of related standards. When used together, these standards specify a mechanism for transferring information between CASE tools.

***CDIF - CASE Data Interchange Format - Overview (ISO CD 15474-1: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 1 : Overview)* should be read first when initially exploring CDIF.** It explains the overall CDIF Architecture and how the family of standards fits together.

**This standard, *CDIF - Framework for Modeling and Extensibility (ISO CD 15474-2: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 2 : Framework for Modeling and Extensibility)*, should also be read before the other standards in the CDIF Family of Standards.** It defines the CDIF Meta-meta-model and the modeling concepts used throughout CDIF and the extensibility mechanism.

This standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

The material contained in this publication has been copyrighted by the CASE Data Interchange Format (CDIF) Division of the Electronic Industries Association (EIA). Permission to copy this material in paper and electronic form has been granted to the international software engineering standards development organization (ISO/IEC JTC1 SC7) for the sole purpose of international standards development work. Organizations receiving copies of this document may not duplicate and distribute additional copies internally for any reason other than to facilitate their organization's review and comment back to ISO/IEC JTC1/SC7.



1

This page is intentionally blank

# CONTENTS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45

<b>1. SCOPE.....</b>	<b>1</b>
<b>2. INTRODUCTION.....</b>	<b>2</b>
<b>2.1 Audience .....</b>	<b>2</b>
<b>2.2 Structure of this Document .....</b>	<b>2</b>
<b>2.3 Conventions .....</b>	<b>4</b>
<b>3. META-META-MODEL CONCEPTS .....</b>	<b>5</b>
<b>3.1 Introduction .....</b>	<b>5</b>
<b>3.2 Modeling Approach.....</b>	<b>5</b>
3.2.1 Introduction .....	5
3.2.2 Naming Conventions .....	6
3.2.3 Many-to-many Meta-relationships .....	6
3.2.4 Attributed Meta-relationships .....	6
3.2.5 Subtyping of Meta-entities.....	6
3.2.6 Multiple Inheritance in Meta-entities.....	7
3.2.7 Subtyping of Meta-relationships .....	9
3.2.8 Multiple Inheritance of Meta-relationships .....	10
3.2.9 Arity of Meta-relationships .....	10
3.2.10 Mutual Exclusivity of Meta-relationships .....	10
<b>3.3 CDIF Graphical Notation.....</b>	<b>10</b>
3.3.1 Introduction .....	10
3.3.2 Meta-entity Symbol .....	11
3.3.3 Meta-relationship Symbol .....	11
3.3.4 Meta-relationship Cardinality Symbol .....	12
3.3.5 Meta-entity Subtype Hierarchy Symbol.....	13
3.3.6 Meta-relationship Subtype Hierarchy .....	14
3.3.7 Mutual Exclusivity of Meta-relationships .....	14
<b>3.4 Subject Areas.....</b>	<b>15</b>
3.4.1 Definition .....	15
3.4.2 Working Meta-model .....	15
3.4.3 Explicit and Implicit Use of <i>CollectableMetaObjects</i> in Subject Areas .....	15
3.4.4 Rules for Defining Subject Area Scope .....	16
3.4.5 Subject Area Diagrams .....	17
<b>3.5 Data Types .....</b>	<b>18</b>
3.5.1 Introduction .....	18
3.5.2 Bitmap .....	19
3.5.3 Boolean.....	19
3.5.4 Date .....	19
3.5.5 Enumerated.....	19
3.5.6 Float.....	19
3.5.7 Identifier .....	19
3.5.8 Integer .....	19
3.5.9 IntegerList.....	19
3.5.10 Point .....	20
3.5.11 PointList.....	20

---

1	3.5.12 String .....	20
2	3.5.13 Text .....	20
3	3.5.14 Time.....	20
4	<b>4. META-MODEL EXTENSIBILITY .....</b>	<b>21</b>
5	<b>4.1 Introduction .....</b>	<b>21</b>
6	<b>4.2 Extensibility .....</b>	<b>21</b>
7	4.2.1 Add SubjectArea .....	21
8	4.2.2 Add CollectableMetaObject.IsUsedIn.SubjectArea .....	22
9	4.2.3 Add Meta-entities.....	22
10	4.2.4 Add Meta-relationships .....	22
11	4.2.5 Add AttributableMetaObject.HasSubtype.AttributableMetaObject.....	22
12	4.2.6 Add Meta-attribute to a Meta-entity or Meta-relationship.....	22
13	4.2.7 Add Legal Enumerated Value to a Meta-attribute.....	22
14	<b>5. EXPORTER RESPONSIBILITIES .....</b>	<b>23</b>
15	<b>5.1 Introduction .....</b>	<b>23</b>
16	<b>5.2 Extensibility .....</b>	<b>23</b>
17	<b>5.3 Maximum Output.....</b>	<b>23</b>
18	<b>5.4 Meta-attributes.....</b>	<b>23</b>
19	<b>6. IMPORTER RESPONSIBILITIES .....</b>	<b>24</b>
20	<b>6.1 Introduction .....</b>	<b>24</b>
21	<b>6.2 Information Retention.....</b>	<b>24</b>
22	<b>6.3 Working Meta-model .....</b>	<b>24</b>
23	<b>6.4 Meta-attributes.....</b>	<b>24</b>
24	<b>7. META-META-MODEL OVERVIEW .....</b>	<b>25</b>
25	<b>7.1 Introduction.....</b>	<b>25</b>
26	<b>7.2 Meta-meta-model Overview Format.....</b>	<b>25</b>
27	<b>7.3 AttributableMetaObject Hierarchy .....</b>	<b>25</b>
28	7.3.1 Introduction .....	25
29	7.3.2 Hierarchy .....	26
30	<b>7.4 Meta-meta-model Description.....</b>	<b>26</b>
31	7.4.1 Description.....	26
32	7.4.2 Uniqueness of Names .....	27
33	<b>7.5 Meta-meta-model Diagram .....</b>	<b>28</b>
34	<b>7.6 Meta-meta-object Summary Format .....</b>	<b>28</b>
35	<b>7.7 Meta-meta-entity Summary .....</b>	<b>29</b>
36	<b>7.8 Meta-meta-relationship Summary .....</b>	<b>31</b>
37	<b>8. META-META-MODEL DEFINITIONS.....</b>	<b>32</b>
38	<b>8.1 Meta-meta-object Definition Format.....</b>	<b>32</b>
39	<b>8.2 Meta-meta-entity and Meta-meta-attribute Definitions .....</b>	<b>35</b>
40	8.2.1 AttributableMetaObject .....	35
41	8.2.2 CollectableMetaObject .....	37
42	8.2.3 MetaAttribute.....	38
43	8.2.4 MetaEntity .....	41
44	8.2.5 MetaObject .....	43

1	8.2.6 MetaRelationship.....	46
2	8.2.7 SubjectArea.....	49
3	<b>8.3 Meta-meta-relationship Definitions .....</b>	<b>50</b>
4	8.3.1 AttributableMetaObject.HasSubtype.AttributableMetaObject .....	50
5	8.3.2 CollectableMetaObject.IsUsedIn.SubjectArea .....	51
6	8.3.3 MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject .....	52
7	8.3.4 MetaRelationship.HasDestination.MetaEntity .....	53
8	8.3.5 MetaRelationship.HasSource.MetaEntity .....	54
9	<b>APPENDIX A META-MODEL EXAMPLE .....</b>	<b>65</b>
10	<b>A1 Introduction .....</b>	<b>65</b>
11	<b>A2 The Use of the Meta-meta-model Definitions .....</b>	<b>65</b>
12	A2.1 Instance of SubjectArea .....	65
13	A2.2 Instances of MetaEntities .....	65
14	A2.3 Instances of MetaRelationships .....	65
15	A2.4 AttributableMetaObjects and MetaAttributes .....	66
16	A2.5 Instances of MetaEntity Subtypes .....	66
17	A3. Reading the Example Subject Area .....	67
18	<b>A4. Subject Area Object Definitions .....</b>	<b>68</b>
19	A4.1 Subject Area called <i>ProjectSpecification</i> .....	68
20	A4.2 Meta-entity called <i>Project</i> .....	69
21	A4.3 Meta-relationship Project.IsComposedOf.Task .....	71
22	<b>APPENDIX B QUESTIONS AND ANSWERS .....</b>	<b>73</b>
23	<b>B1. Introduction .....</b>	<b>73</b>
24	<b>B3. Subject Areas .....</b>	<b>76</b>
25	<b>B4. Relationships .....</b>	<b>79</b>
26	<b>B5. Identifiers and Keys.....</b>	<b>81</b>
27	<b>B6. Other Points .....</b>	<b>82</b>
28	<b>GLOSSARY .....</b>	<b>87</b>
29	<b>REFERENCES.....</b>	<b>95</b>
30	<b>INDEX.....</b>	<b>97</b>
31	<b>RELATED EIA STANDARDS.....</b>	<b>INSIDE BACK COVER</b>

---

## FIGURES

1	
2	Figure 1 Position in the CDIF Family of Standards..... 1
3	Figure 2 Multiple Inheritance ..... 8
4	Figure 3 Multiple Inheritance - Invalid Meta-attribute, <i>ZI</i> ..... 8
5	Figure 4 Example of CDIF Graphical Notation ..... 11
6	Figure 5 The Meta-entity Symbol..... 11
7	Figure 6 The Meta-relationship Symbol..... 12
8	Figure 7 A Reflexive Meta-relationship ..... 12
9	Figure 8 The Meta-relationship Cardinality Symbol..... 13
10	Figure 9 The Meta-entity Subtype Hierarchy Symbol ..... 13
11	Figure 10 The Meta-entity Subtype Hierarchy Symbol - Multiple Inheritance ..... 14
12	Figure 11 Mutual Exclusivity of Meta-relationships..... 15
13	Figure 12 The CDIF Meta-meta-model ..... 28
14	Figure A1 Example Meta-model for Project Specification ..... 67
15	Figure B1 Audit Meta-relationships ..... 74
16	Figure B2 Inserting a General Supertype..... 75
17	Figure B3 Adding a Third Participant to a Meta-relationship..... 80
18	

## TABLES

19	
20	Table 1 Inheritance of Meta-meta-attribute Values by Subtype Meta-entities..... 7
21	Table 2 Inheritance of Meta-meta-attribute Values by Subtype Meta-relationships ..... 9
22	Table 3 Data Types Supported by CDIF ..... 18
23	Table 4 Contents of Meta-meta-entity Definition..... 33
24	Table 5 Contents of Meta-meta-attribute Definition ..... 34
25	Table 6 Contents of Meta-meta-relationship Definition..... 34

This page is intentionally blank



## 1. SCOPE

The CDIF Family of Standards is primarily designed to be used as a description of a mechanism for transferring information between CASE tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF. The language that is defined for the Transfer Format also has applicability as a general language for Import/Export from repositories. The CDIF Integrated Meta-model defined for CASE also has applicability as the basis of standard definitions for use in repositories.

The standards that form the complete family of CDIF Standards are documented in *EIA/IS-106 CDIF - CASE Data Interchange Format - Overview*. These standards cover the overall framework, the transfer format and the CDIF Integrated Meta-model. The diagram in Figure 1 depicts the various standards that comprise the CDIF Family of Standards. The shaded box depicts this Standard and the its position in the CDIF Family of Standards.

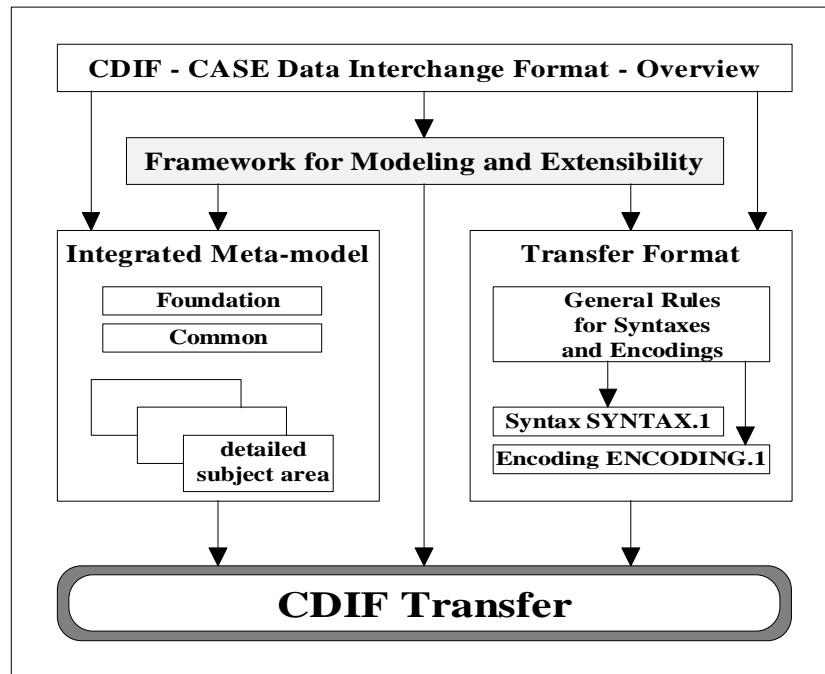


Figure 1  
Position in the CDIF Family of Standards

This document (*EIA/IS-107 CDIF - Framework for Modeling and Extensibility*) includes the definition of the CDIF Meta-meta-model and describes the rules and notations used throughout the CDIF Family of Standards. The rules for extending the CDIF Integrated Meta-model are also defined.

## 2. INTRODUCTION

### 2.1 Audience

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides a definition of the framework of the entire CDIF Family of Standards. It is suitable for:

- those evaluating CDIF
- those who wish to understand the principles and concepts of a CDIF transfer
- those developing importers and exporters.

The document *EIA/IS-106 CDIF - CASE Data Interchange Format Overview* and this document, *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*, should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF Family of Standards.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

- Entity-Relationship-Attribute modeling
- CASE tools
- Information repositories
- Data dictionaries
- Multiple meta-layer modeling.

### 2.2 Structure of this Document

This document is organized into the following sections:

#### Section 1 'Scope'

This describes the CDIF Family of Standards and positions the current document within the set of documents.

#### Section 2 'Introduction'

This describes the intended audience, structure and conventions used in the document.

#### Section 3 'Meta-meta-model Concepts'

This defines the Meta-meta-model, modeling rules and graphical conventions that are used. The data types supported in meta-models are introduced. The concept of the Working Meta-model is also explained.

---

**Section 4 'Meta-model Extensibility'**

This describes the types of extensibility that may be used when extending the CDIF Integrated Meta-model.

**Section 5 'Exporter Responsibilities'**

This describes the responsibilities of a tool with regard to overlapping semantics and maximum output.

**Section 6 'Importer Responsibilities'**

This describes the responsibilities of a tool with regard to the retention of information.

**Section 7 'Meta-meta-model Overview'**

This contains an overview of the CDIF Meta-meta-model. This section includes an AttributableMetaObject Hierarchy, a Meta-meta-model Diagram, a Meta-meta-entity Summary and a Meta-meta-relationship Summary.

**Section 8 'Meta-meta-model Definitions'**

This contains the full definitions of all the objects in the CDIF Meta-meta-model.

**Appendix A 'Meta-model Example'**

This shows how the modeling approach is applied.

**Appendix B 'Questions and Answers'**

This includes detailed responses to questions that have been raised by those reviewing this document. It is intended that the inclusion of these will aid the reader in understanding this document, and related issues concerning CDIF.

**Glossary**

The Glossary contains definitions of terms used within this Standard.

**References**

This contains details of other standards referenced within this standard.

**Index**

A comprehensive index is provided.

## 1 **2.3 Conventions**

2 The typographical and naming conventions defined below are used throughout the CDIF Family  
3 of Standards.

- 4       • Double quotes are used to introduce new terms (e.g., "model layer")
- 5       • Bold type is used for emphasis (e.g., this is **important**)
- 6       • All meta-objects and meta-meta-objects in CDIF (in meta-models and meta-meta-  
7 meta-models) are named by concatenating all the words that name the meta-object or  
8 meta-meta-object; the first letter of each word is upper-case, the rest are lower-  
9 case (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*)

10 Full details of the CDIF Graphical Notation used in the Meta-model and the Meta-meta-model  
11 can be found in Section 3.3, CDIF Graphical Notation.

## 3. META-META-MODEL CONCEPTS

### 3.1 Introduction

This section describes the concepts of the CDIF Meta-meta-model. An overview and a diagram of the CDIF Meta-meta-model are included in Section 7, Meta-meta-model Overview. The formal definition can be found in Section 8, Meta-meta-model Definitions

The CDIF Meta-meta-model is defined using the same facilities that are used to define the CDIF Meta-model (and could be used to define models). It is, therefore, quite common to get the names of the objects at the different meta-layers confused. To minimize this problem, the reader should remember the following:

- The components of the Meta-meta-model are known as meta-meta-objects (i.e., meta-meta-entities, meta-meta-relationships and meta-meta-attributes)
- The components of the Meta-model are known as meta-objects (i.e., meta-entities, meta-relationships and meta-attributes)
- The components of models are known as objects (e.g., processes, entities, flows)
- The names of the components in the Meta-meta-model usually have a single meta- as a prefix (e.g., there is a meta-meta-entity, in the Meta-meta-model, known as *MetaAttribute*)
- The names of the components in the meta-model usually have no meta prefix.

The CDIF Meta-meta-model is the definition of the concepts used to develop CDIF-conformant meta-models. This includes the Entity-Relationship-Attribute diagramming technique and information definition requirements pertinent to the meta-model development process.

### 3.2 Modeling Approach

#### 3.2.1 Introduction

Because the CDIF Meta-meta-model is used to define meta-models, it must be capable of expressing a complete description of a meta-model. A CDIF-conformant meta-model contains three types of objects: entities, relationships and attributes; these objects in the meta-model are properly known as meta-entities, meta-relationships and meta-attributes. These objects are used to model the information content of the meta-model, and in fact, it is a rule in CDIF, that any single concept, or set of information may only be represented one way; that is, although there may be several ways of modeling a concept, only one will be included in the model.

The Meta-meta-model is composed of two things:

- 1           •     An Entity-Relationship-Attribute model that is used to describe the CDIF Meta-  
2                     model. This Entity-Relationship-Attribute model is also self-describing (i.e., is  
3                     defined using itself).
  
- 4           •     A graphical notation for representing occurrences (i.e., instances) of the Meta-  
5                     meta-model. This notation is used in CDIF in two ways; to depict the contents of  
6                     subject areas in the CDIF Meta-model, and to depict the CDIF Meta-meta-model  
7                     itself.

8 It is important for the reader to remember that the rules in this section apply only to the meta-  
9 models defined using the CDIF Meta-meta-model. In particular, these rules do not relate to  
10 Entity-Relationship-Attribute models that are transferred during a CDIF Transfer (see *EIA/IS-114*  
11 *CDIF - Integrated Meta-model - Data Modeling Subject Area* for further information).

12 This section describes some general modeling rules that are part of the definition of the CDIF  
13 Meta-meta-model. They are described with a meta- prefix. The rules apply to the CDIF  
14 Integrated Meta-model and all extensions defined using the extensibility mechanism. The rules  
15 are also adhered to by the CDIF Meta-meta-model<sup>1</sup>.

### 16 **3.2.2 Naming Conventions**

17 All meta-entities, meta-relationships and meta-attributes in CDIF (in either the Meta-model or the  
18 Meta-meta-model) are named by concatenating together all the words that name the object. The  
19 first letter of each word in a name is upper-case and the rest are lower case. This convention can  
20 be relaxed in those cases where the "words" that the object name is composed of, already have  
21 capitalization or word-separator information within them. This is especially common when one of  
22 the "words" is an acronym (e.g., *VCRModelName*). Other examples of names that conform to this  
23 convention include: *Person*, *HasSubtype*, *IsUsedToRepresent*.

24 Names shall conform to the Identifier data type, see Section 3.5.7, Identifier. In addition, the  
25 name shall have a maximum of 32 characters. It shall consist of the characters A-Z, a-z, 0-9,  
26 underscore (\_) and hyphen (-). It must end with an alphabetic character (i.e., A-Z or a-z) or a  
27 numeric character (i.e., 0-9).

### 28 **3.2.3 Many-to-many Meta-relationships**

29 The CDIF Meta-meta-model supports many-to-many meta-relationships.

### 30 **3.2.4 Attributed Meta-relationships**

31 The CDIF Meta-meta-model supports attributed meta-relationships.

### 32 **3.2.5 Subtyping of Meta-entities**

33  
34 The CDIF Meta-meta-model provides support for subtyping (sometimes known as specialization)  
35 and its converse, supertyping (sometimes known as generalization).

---

<sup>1</sup> The CDIF modeling rules may be used at any meta-level; they may be used to create models.

In particular, a meta-entity may be defined to be a "subtype" of another meta-entity. The subtype meta-entity inherits all of the meta-attributes and meta-relationships of its immediate and indirect supertype meta-entities.

Supertyping is used as a classification mechanism to group meta-objects with a substantial level of common semantics; an importer can use this classification as an aid to deciding how to represent any meta-object that is not directly representable within the importing tool.

Subtype meta-entities have unique *CDIFMetaIdentifiers*; that is, the *CDIFMetaIdentifier* is not inherited from the parent. The *Description*, *Constraints*, *Usage*, and *Aliases* that were defined for the parent meta-entity are inherited. These textual fields may also be expanded in the local definition of the subtype meta-entity. Subtype meta-entities inherit all the meta-attributes that have been defined for the parent meta-entity, and all those of its antecedents. The characteristics of the meta-attribute (for example, *Name*) are unchanged when inherited by a subtype meta-entity. Additional, local, meta-attributes may be defined specific to the subtype meta-entity.

In the same way that meta-attributes are inherited, any meta-relationships that are defined with the parent meta-entity as the source or destination are inherited by subtype meta-entities. None of the characteristics of the meta-relationship are changed by its use by a subtype; that is, the *Name*, *Cardinality*, *CDIFMetaIdentifier*, *Aliases*, *Constraints*, *Description*, and *Usage*, are unchanged. Notice that even the full name of the meta-relationship is unchanged and will contain the name of the supertype meta-entity rather than that of the subtype.

A summary of the inheritance rules for subtype meta-entities is shown in Table 1.

Table 1  
Inheritance of Meta-meta-attribute Values by Subtype Meta-entities

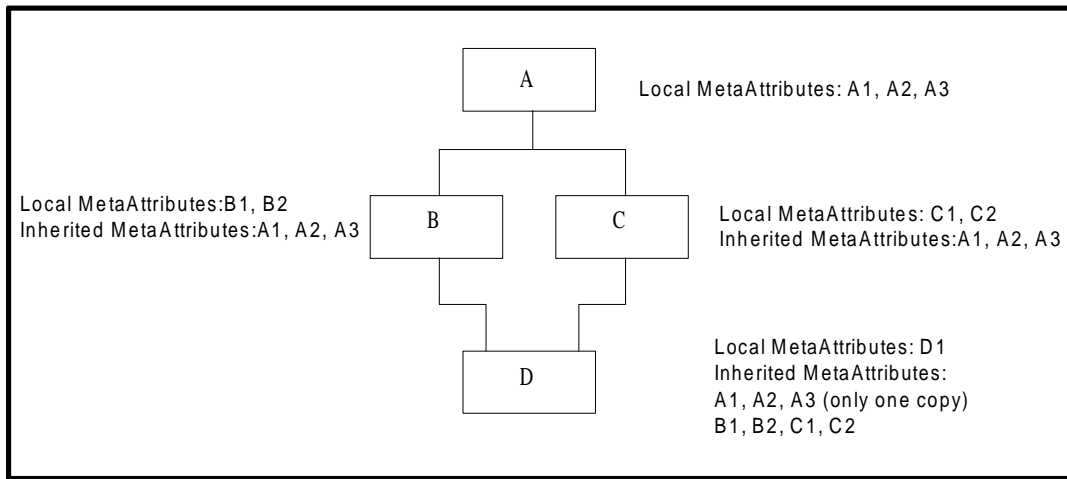
Meta-meta-attribute	Inherited by Subtype Meta-entity
Name	No
CDIFMetaIdentifier	No
Description	Yes, may be extended
Usage	Yes, may be extended
Aliases	Yes, may be extended
Constraints	Yes, may be extended
Type	No

### 3.2.6 Multiple Inheritance in Meta-entities

A meta-entity may have multiple supertypes, allowing for "multiple inheritance". In this case, the subtype inherits all of the meta-attributes and meta-relationships of all of its supertypes (and their supertypes, etc.).

1 There are several situations that occur when using multiple-inheritance that either seem to, or do,  
 2 present inheritance conflicts.

3 When several inheritance threads of a meta-entity converge at a common supertype, the inheriting  
 4 meta-entity does not inherit multiple copies of the meta-attributes and meta-relationships from  
 5 that point to the top of the hierarchy; it only inherits one copy. Figure 2 shows meta-entity **A**  
 6 having two subtype meta-entities (**B** and **C**), which are both supertype meta-entities of meta-entity  
 7 **D**. Both local and inherited meta-attributes for all meta-entities are shown.



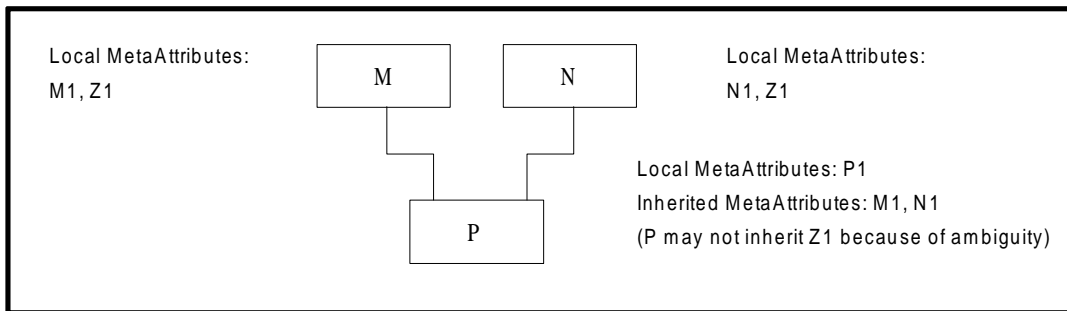
8

9

10

Figure 2  
 Multiple Inheritance

11 The other potential conflict occurs when a meta-entity inherits meta-attributes from two  
 12 supertypes, both having a meta-attribute with the same name that are, in fact, different meta-  
 13 attributes. The behavior for CDIF is undefined in these circumstances. To guarantee proper  
 14 (predictable) behavior from importers and exporters, this situation must be avoided. The CDIF  
 15 Integrated Meta-model and CDIF Meta-meta-model are defined ensuring that this does not occur.



16

17

18

Figure 3  
 Multiple Inheritance - Invalid Meta-attribute, **Z1**

19 Figure 3 shows an invalid inheritance example, where the meta-entity **P** has two supertype meta-  
 20 entities (**M** and **N**) that both have an identically named meta-attribute, **Z1**.

### 3.2.7 Subtyping of Meta-relationships

A meta-relationship may be defined to be a subtype of another meta-relationship.

A subtype meta-relationship may be defined at the 'level' in the meta-entity hierarchy where the supertype meta-relationship was defined (that is, with the same source and destination meta-entities as the supertype). Alternatively, the source meta-entity or destination meta-entity, or both, may be subtypes of the source and destination meta-entities in the supertype meta-relationship.

Each subtype meta-relationship has a unique *CDIFMetaIdentifier*; that is, it does not inherit the *CDIFMetaIdentifier* from the supertype meta-relationship.

The cardinalities of the subtype meta-relationship are defined locally; however, they may only be as constrained, or more constrained than at the supertype level. Therefore, the minimum cardinality of the subtype must be greater than, or equal to, that of the supertype, and the maximum cardinality must be less than, or equal to, that of the supertype.

The other, textual, characteristics of the subtype meta-relationship are inherited from the supertype meta-relationship but may be further refined for the subtype.

The immediate and indirect subtype meta-relationship inherits any meta-attributes that have been defined for the supertype meta-relationship. None of the characteristics of the meta-attributes are changed. Additional meta-attributes may be created for the subtype meta-relationship.

A summary of the inheritance rules for subtype meta-relationships is shown in Table 2.

Table 2  
Inheritance of Meta-meta-attribute Values by Subtype Meta-relationships

Meta-meta-attribute	Inherited by Subtype Meta-relationship
Name	No
CDIFMetaIdentifier	No
MinSourceCard	No, must be locally defined
MaxSourceCard	No, must be locally defined
MinDestCard	No, must be locally defined
MaxDestCard	No, must be locally defined
Description	Yes, may be extended
Usage	Yes, may be extended
Aliases	Yes, may be extended
Constraints	Yes, may be extended

1

2 Between any pair of meta-entities, there may be both a supertype meta-relationship (possibly due  
3 to inheritance by a subtype meta-entity) and a subtype meta-relationship defined. There may not  
4 be an instance of both the supertype meta-relationship and the subtype meta-relationship between  
5 the same pair of meta-entity instances. However, in most cases it is expected that the supertype  
6 meta-relationship will be abstract (not instantiable), which means that the condition of an instance  
7 of both supertype and subtype meta-relationships between the same instance pair will not arise.

### 8 **3.2.8 Multiple Inheritance of Meta-relationships**

9 Meta-relationships may inherit from multiple supertypes in the same way that meta-entities may  
10 inherit from multiple supertypes. The discussion in Section 3.2.6, Multiple Inheritance in Meta-  
11 entities, that describes potential inheritance conflicts arising from multiple inheritance applies  
12 equally to multiple inheritance of meta-relationships, and the diagrams, although depicting meta-  
13 entities (in boxes) may be read as though the boxes represent meta-relationships.

### 14 **3.2.9 Arity of Meta-relationships<sup>2</sup>**

15 The CDIF Integrated Meta-model, and any extensions, must be defined using binary meta-  
16 relationships rather than n-ary meta-relationships. (Note that the functionality of n-ary  
17 relationships is covered in the Data Modeling subject area in the CDIF Integrated Meta-model,  
18 see *EIA/IS-114 CDIF - Integrated Meta-model - Data Modeling Subject Area.*)

### 19 **3.2.10 Mutual Exclusivity of Meta-relationships**

20 In meta-models mutual exclusivity of meta-relationships is documented as a constraint on the  
21 meta-relationships that are in the exclusivity group.

## 22 **3.3 CDIF Graphical Notation**

### 23 **3.3.1 Introduction**

24 The same graphical notation is used for both the CDIF Meta-meta-model and CDIF Meta-model.  
25 The notation is described in this section with regard to meta-models, but applies equally to the  
26 Meta-meta-model defined in this document<sup>3</sup>. For example, the meta-entity symbol is defined;  
27 however, the definition is equally valid for the meta-meta-entity symbol.

---

<sup>2</sup> The arity of a relationship may also be called the degree of the relationship. A binary relationship has an arity of two; a ternary relationship has an arity of three, etc.

<sup>3</sup> CDIF Graphical Notation can be used for any meta-level. This particular example, shown in Figure 5, is of a model.

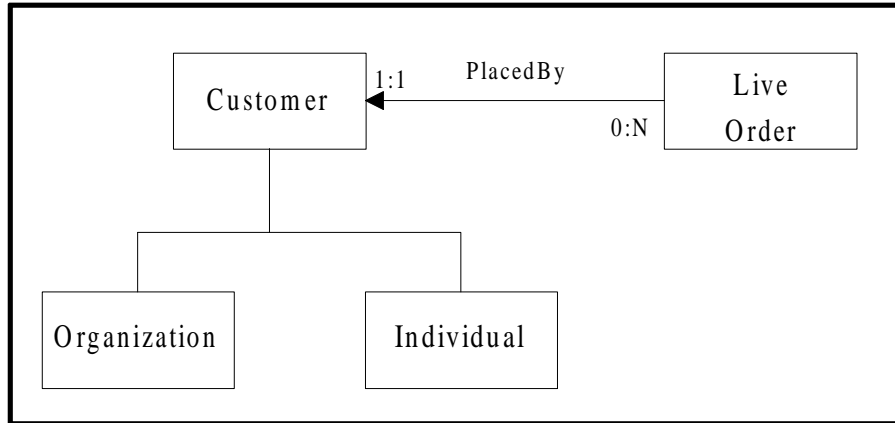


Figure 4  
Example of CDIF Graphical Notation

Figure 4 is an example model that uses the CDIF Graphical Notation. A further example is shown in Appendix A.

### 3.3.2 Meta-entitySymbol

Meta-entities (that is, instances of the meta-meta-entity called *MetaEntity*) used in a subject area are drawn as rectangular boxes, with the name of the meta-entity on a white background. Examples are '*Meta-entity A*' in Figure 5, and *Customer* in Figure 4.

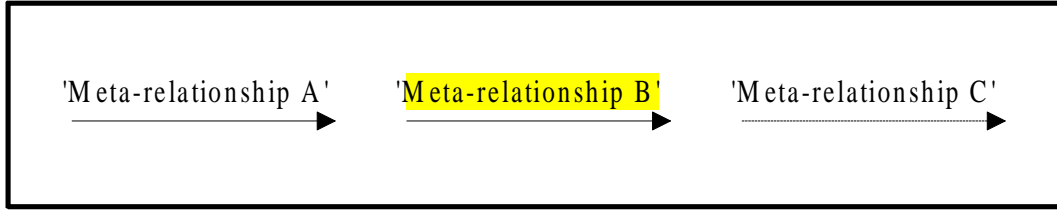
Additionally, a special notations may be used to represent meta-entities on diagrams of meta-models. If the meta-entity is not used in a subject area but is used in another subject area and is included in the diagram for clarity, the box shall be shaded. An example is '*Meta-entity B*' in Figure 5.



Figure 5  
The Meta-entity Symbol

### 3.3.3 Meta-relationship Symbol

This symbol names and depicts an instance of the meta-meta-entity *MetaRelationship* which exists between two meta-entities. It is usually shown as a straight line (possibly multiple segments) joining two meta-entities/rectangles. A single arrowhead must appear somewhere along the meta-relationship line to indicate the direction in which the meta-relationship name shall be read. The two meta-entities participating in a meta-relationship are referred to as the "source" and "destination" as indicated by the arrowhead. Examples of meta-relationships are '*Meta-relationship A*' in Figure 6, and *PlacedBy* Figure 4.



1

2

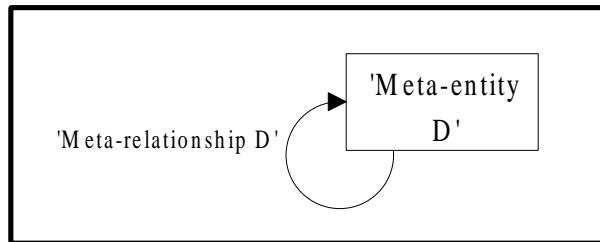
3

Figure 6  
The Meta-relationship Symbol

4 Additionally, three special notations may be used to represent meta-relationships on diagrams of  
5 meta-models.

6 If a meta-relationship is included in the diagram for clarification, but it is not used in the subject  
7 area, the text of the meta-relationship name shall be back-shaded. This is shown for '*Meta-*  
8 *relationship B*' in Figure 6.

9 If a supertype meta-entity that participates in a meta-relationship is not included in the diagram,  
10 the inherited meta-relationship may be shown on the diagram as a broken line to/from the  
11 inheriting subtype meta-entity. This is shown for '*Meta-relationship C*' in Figure 6.



12

13

14

Figure 7  
A Reflexive Meta-relationship

15 Where a reflexive (recursive) meta-relationship is drawn from a meta-entity to itself, it may be  
16 drawn as an arc (often referred to as a pig's ear or scorpion) as shown in Figure 7.

### 17 3.3.4 Meta-relationship Cardinality Symbol

18 The cardinality and optionality of each meta-relationship is defined by colon-separated labels (x:y)  
19 at both ends of the meta-relationship line symbol. The labels indicate the minimum (x) and the  
20 maximum (y) number of instances of the meta-entity nearest the label from the perspective of a  
21 single instance of the other meta-entity participating in the meta-relationship. The cardinalities of  
22 the *PlacedBy* meta-relationship are shown in Figure 8,

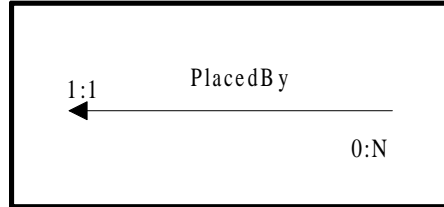


Figure 8  
The Meta-relationship Cardinality Symbol

The expansion of the notation into text for the *PlacedBy* meta-relationship shown in Figure 8 can be read:

"Each *LiveOrder* must be *placed by* one (and only one) *Customer*."

"must be" because the minimum cardinality at the *Customer* end of the meta-relationship is 1, and "one (and only one)" because the maximum cardinality is also 1.

Reading the meta-relationship in the other direction gives us:

"Each *Customer* may be *placer of* one or more *Live Orders*."

"may be" because the minimum cardinality at the *Customer* end is zero, and "one or more" because the maximum cardinality at *LiveOrder* is N, for many.

### 3.3.5 Meta-entity Subtype Hierarchy Symbol

A special type of symbol is used to show that a meta-entity is a subtype of another meta-entity. It is shown as a hierarchy from the supertype meta-entity to each subtype. The supertype will always occur closer to the top of the diagram than its subtypes. The lines that connect the supertype to its subtypes are **not** labeled and are constrained to vertical and horizontal. Figure 9 shows an example of a single supertype meta-entity with two subtype meta-entities; that is, customers may either be organizations or individuals.

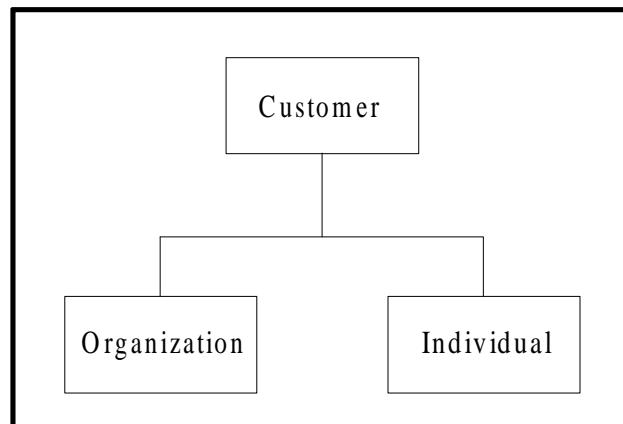
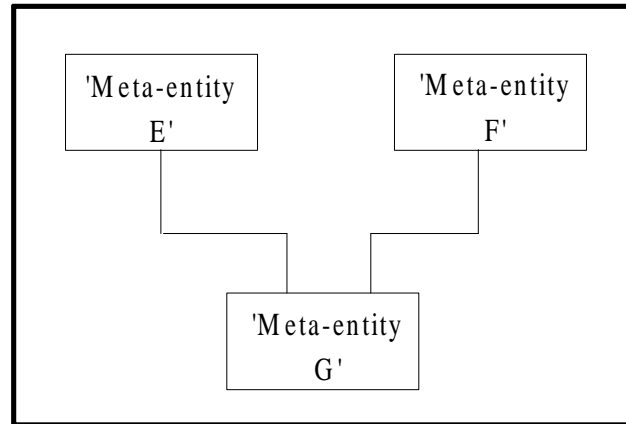


Figure 9  
The Meta-entity Subtype Hierarchy Symbol

1 Figure 10 shows an example where a single subtype meta-entity is inheriting characteristics from  
 2 two supertype meta-entities; an example of multiple inheritance.



3

Figure 10

4

5 The Meta-entity Subtype Hierarchy Symbol - Multiple Inheritance

6

### 6 3.3.6 Meta-relationship Subtype Hierarchy

7 The meta-relationship subtype hierarchy is not represented diagrammatically. The inheritance of  
 8 meta-relationships is shown in the *AttributableMetaObject* hierarchy that is provided for each  
 9 subject area. For example the following meta-relationship hierarchy is defined in the Common  
 10 subject area (*EIA/IS-112 CDIF - Integrated Meta-model - Common Subject Area*):

11 RootEntity.IsRelatedTo.RootEntity  
 12     ProjectionComponent.IsProjectionOf.SemanticObject  
 13         ProjectionComponent.IsFullProjectionOf.DataObject  
 14         ProjectionComponent.IsProjectionOf.Attribute

15

### 16 3.3.7 Mutual Exclusivity of Meta-relationships

17 Mutual exclusivity of meta-relationships may be shown diagrammatically by placing an arc across  
 18 the mutually exclusive meta-relationships.

19 Each instance of the meta-entity on which the mutual exclusivity group is specified can be  
 20 connected to instances of only one of the meta-entities at the far end of all the included meta-  
 21 relationships. If a connection to an instance of a meta-entity at the far end is required for each  
 22 meta-entity instance at the near end, then the CDIF convention will be to indicate this with a  
 23 minimum cardinality value of one (1:x) at the far end of all the included meta-relationships. This  
 24 does not mean an instance of each is required; only one instance is required across the entire  
 25 mutual exclusivity group. More than one instance can participate for the same meta-entity if the  
 26 maximum cardinality is greater than one on the connecting meta-relationship.

27 Figure 11 shows that each *ContactName* must be for either an *Organization* or an *Individual*,  
 28 but that a single instance of *ContactName* would not be for both an *Organization* and an  
 29 *Individual*. (The "must be" can be deduced because of the minimum cardinality of one at the  
 30 destination end of the *For* meta-relationship.)

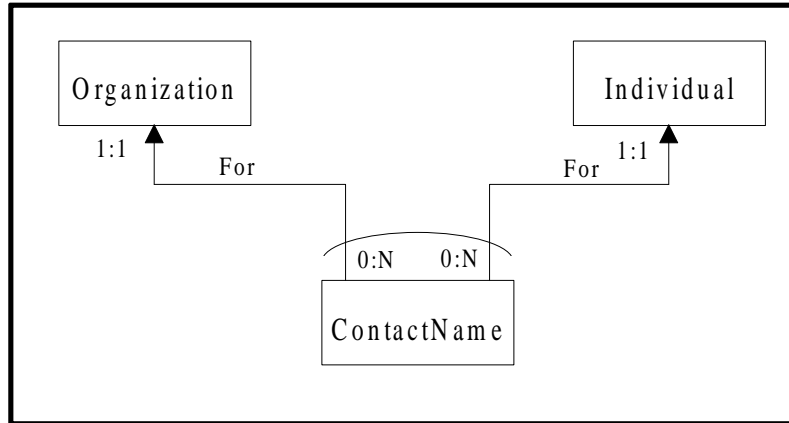


Figure 11  
Mutual Exclusivity of Meta-relationships

## 3.4 Subject Areas

### 3.4.1 Definition

The CDIF Integrated Meta-model is a single integrated model that is a description of the set of concepts and notations used to define an information model. This model is huge; therefore, it is partitioned into manageable sections by building views of the underlying meta-model. These partitions, or "subject areas" as they are known, group together collectable meta-objects (meta-entities, meta-relationships and meta-attributes) that address a particular aspect within the Systems Development Life Cycle (e.g., Data Flow Modeling), or that provide similar functionality (e.g., Data Definition). Meta-objects may be shared by many subject areas.

### 3.4.2 Working Meta-model

A CDIF Transfer may contain several, possibly overlapping subject areas. The actual subject areas, both standard and those created using extensibility, are identified in the transfer. An importer can then create what is called the Working Meta-model for a transfer by integrating the objects from standard subject areas and any additional objects created using extensibility.

When a transfer uses a standard subject area, the reference to that subject area is sufficient to define both explicitly and implicitly used meta-objects. This means that it is not necessary to give a subject area reference for the subject area where supertype meta-objects are defined purely to support those supertypes. If other meta-objects from that subject area are used in their own right, then the subject area reference must also be defined. In a CDIF Transfer where the meta-model is being defined entirely using extensibility, a subject area reference for the Foundation Subject Area as defined in *EIA/IS-111 CDIF - Integrated Meta-model - Foundation Subject Area* must be used since it contains the definitions for the eventual roots of all meta-entity and meta-relationship hierarchies used in CDIF.

### 3.4.3 Explicit and Implicit Use of *CollectableMetaObjects* in Subject Areas

Subject areas may contain some *CollectableMetaObjects* that are explicitly used and, may contain some that are implicitly used through inheritance.

1 In the standardized CDIF subject areas, which are published as standards, the explicitly used  
2 *CollectableMetaObjects* are defined in Section 5, Subject Area Detailed Definitions, of each  
3 subject area standard. The implicitly used *CollectableMetaObjects* are defined in Section 6,  
4 Referenced Meta-object Definitions, of each subject area standard; these are both supertype meta-  
5 entities whose meta-attributes and meta-relationships are inherited by meta-entities that are  
6 explicitly used, and supertype meta-relationships for those meta-relationships that are explicitly  
7 used.

8 When a new subject area is being created using extensibility, instances of the  
9 *CollectableMetaObject.IsUsedIn.SubjectArea* meta-relationship must be created for all explicitly  
10 used *CollectableMetaObjects*. Instances of meta-relationships should not be created for implicitly  
11 used *CollectableMetaObjects* since inheritance of meta-attributes and meta-relationships is  
12 assumed all the way up the inheritance hierarchies to *RootObject*.

### 13 3.4.4 Rules for Defining Subject Area Scope

14 Subject area views are built by instantiating the *CollectableMetaObject.IsUsedIn.SubjectArea*  
15 meta-meta-relationship for any *MetaEntity*, *MetaRelationship*, and *MetaAttribute* that is explicitly  
16 used. The following rules should be observed when defining the scope of a subject area:

- 17 • Where a meta-relationship is explicitly used in a subject area, both the source and  
18 destination meta-entities must also be used.
- 19 • Where both source and destination meta-entities for a meta-relationship happen to  
20 be explicitly used in a subject area, there is no obligation to use the meta-  
21 relationship.
- 22 • Where a meta-entity is explicitly used in a subject area and it participates in the  
23 integrated meta-model in a mandatory meta-relationship (at the mandatory end),  
24 the meta-relationship need not be used in the subject area.
- 25 • Where a meta-attribute is explicitly used in a subject area, the meta-entity or meta-  
26 relationship that the meta-attribute describes must also be explicitly used in the  
27 subject area.
- 28 • Where a meta-entity or meta-relationship is explicitly used in a subject area, any  
29 local meta-attributes that describe it that are required by the subject area must be  
30 explicitly used.
- 31 • Where a meta-entity or meta-relationship is explicitly used in a subject area, there  
32 is no obligation to use any of its meta-attributes, even mandatory ones.
- 33 • All inherited meta-attributes for both meta-entities and meta-relationships will be  
34 implicitly used in the subject area all the way up the inheritance hierarchy. The  
35 *CollectableMetaObject.IsUsedIn.SubjectArea* meta-meta-relationship should not  
36 be instantiated for inherited meta-attributes.

- 
- 1           •     For subtype meta-entities, any meta-relationships in which the supertype meta-  
2                     entity participates will be implicitly used in a subject area, provided the other meta-  
3                     entity, or a subtype of it, is explicitly used.

4  
5                     Note that where two subject areas are used in a transfer, a meta-relationship  
6                     between two implicitly used supertype meta-entities (one in each subject area) will  
7                     be inherited between the appropriate subtype meta-entities in the Working Meta-  
8                     model for the transfer.

### 9     **3.4.5 Subject Area Diagrams**

10     Diagrams of subject areas will show explicitly used meta-entities. Meta-entities that are included  
11     in a diagram for clarification shall be shaded. Explicit meta-relationships will be shown in  
12     diagrams. Implicit meta-relationships (i.e. inherited meta-relationships) may be shown on the  
13     diagram even if the supertype meta-entity is not used explicitly. (See Section 3.3, CDIF Graphical  
14     Notation for further details.)

## 3.5 Data Types

### 3.5.1 Introduction

This section describes the data types that are available for the meta-attributes of meta-entities and meta-relationships, in all instances of the CDIF Meta-meta-model (i.e., the CDIF meta-model, the CDIF Meta-meta-model itself, and any additional meta-models built using the extensibility facilities of CDIF). The reader is reminded that the encoding of the information represented by these data types is defined in the specific syntax and encoding being used and is not defined here (e.g., a Boolean is defined here to hold either true or false, not *I* or *O*, not *Y* or *N*). The explicit limits for some of the data types is defined by the syntax being used.

Although some of the data types are defined as structured types, none of the components are optional. There are no defaults defined for any data types or for any of the components.

Table 3 shows the data types that are available for meta-meta-attributes in the Meta-meta-model and for meta-attributes in meta-models.

Table 3  
Data Types Supported by CDIF

Data Type	Description
<b>Bitmap</b>	A bounded matrix of RGB <sup>4</sup> triples
<b>Boolean</b>	TRUE or FALSE
<b>Date</b>	A date (relative or absolute)
<b>Enumerated</b>	A list of valid values
<b>Float</b>	A floating point number
<b>Identifier</b>	A name of an object
<b>Integer</b>	A signed integer
<b>IntegerList</b>	A list of integers
<b>Point</b>	A location in three dimensions
<b>PointList</b>	A list of points
<b>String</b>	A group of printable characters
<b>Text</b>	A group of characters, not necessarily printable
<b>Time</b>	A time (relative or absolute)

<sup>4</sup> RGB stands for Red, Green, Blue

---

1 Where a domain is not specified for an attribute, it shall be assumed to be the full range of values  
2 valid within the specified data type.

### 3 **3.5.2 Bitmap**

4 The bitmap data type is used to represent a two-dimensional array of graphical point  
5 specifications. Each graphical point is a RGB triple composed of an 8-bit unsigned red value, an  
6 8-bit unsigned green value and an 8-bit unsigned blue value. The bitmap has a stated unsigned  
7 height and width (h and w).

### 8 **3.5.3 Boolean**

9 The Boolean data type is used to represent a single item that can hold one of two values: TRUE  
10 or FALSE.

### 11 **3.5.4 Date**

12 The date data type is used to represent a date to the unit of granularity of a day. The class is  
13 specified to indicate whether the date is absolute or relative. An absolute date is given as  
14 *Absolute*. Relative dates are either positive or negative to their (implied) point of reference; this  
15 is indicated with the *RelativePositive* and *RelativeNegative* classes.

### 16 **3.5.5 Enumerated**

17 An enumerated data type specifies a list of possible values for a meta-attribute. There is no  
18 implicit ordering or sequence. Each value must conform to the rules of a valid identifier data  
19 type.

### 20 **3.5.6 Float**

21 A decimal number with up to 16 decimal digits of precision within the range  $10^{-1023}$  to  $10^{1023}$ .

### 22 **3.5.7 Identifier**

23 The identifier data type is used to represent a name. It shall be between one and 128 characters  
24 long. If the character set identified in the Transfer Envelope contains multibyte characters, they  
25 may be used in the names of model objects being transferred.

26 When identifiers are used as the names of CDIF meta-objects, special rules apply. See Section  
27 3.2.2, Naming Conventions, for more details.

### 28 **3.5.8 Integer**

29 An integer is a signed number capable of holding values in the range from -2147483648 to  
30 2147483647 ( $-2^{31}$  to  $2^{31}-1$ ).

### 31 **3.5.9 IntegerList**

32 An integer list is an ordered list of values conforming to the integer data type.

### 1 **3.5.10 Point**

2 A point is a triple of (x, y, z) coordinates and represents an address in the Coordinate Frame.  
3 Each coordinate is represented using an integer. (See *EIA/IS-118 CDIF - Integrated Meta-model*  
4 *- Presentation Location and Connectivity Subject Area* for more information about the CDIF  
5 Transfer Coordinate Frame.)

### 6 **3.5.11 PointList**

7 A point list is an ordered list of points. Each point is a triple of (x, y, z) coordinates and  
8 represents an address in the Coordinate Frame. (See *EIA/IS-118 CDIF - Integrated Meta-model*  
9 *- Presentation Location and Connectivity Subject Area* for more information about the CDIF  
10 Transfer Coordinate Frame).

### 11 **3.5.12 String**

12 The string data type is represented as a group of printable characters of some maximum length,  
13 where  $1 \leq \text{maximum length} \leq 1024$ .

### 14 **3.5.13 Text**

15 The text data type is represented as a group of characters, not necessarily printable, of any pre-  
16 determined maximum length.

### 17 **3.5.14 Time**

18 The time data type is used to represent a time to the granularity of fractional seconds with a class  
19 to indicate whether the time is absolute or relative. Absolute times are given either as  
20 ***AbsoluteUTC***<sup>5</sup> or ***AbsoluteLocal***. Relative times are either positive or negative relative to their  
21 (implied) point of reference; this is indicated with the ***RelativePositive*** and ***RelativeNegative***  
22 classes.

23 Times recorded in a CASE tool are exported as the values recorded by the tool. No conversions  
24 are specified by CDIF for time values where more than one time zone is involved.

---

<sup>5</sup> UTC stands for Universal Coordinated Time (English translation of French term) as defined in Recommendation 460 of the Consultative Committee on International Radio (CCIR) [1].

## 4. META-MODEL EXTENSIBILITY

### 4.1 Introduction

Extensibility is an important aspect of the CDIF Family of Standards. In general, extensibility means allowing two (or more) tools that support CDIF to exchange information that is not defined in the CDIF Integrated Meta-model. Specifically, this involves having the exporter warn the importer about extensions to the standard CDIF Integrated Meta-model before transferring information that uses those extensions, see Section 5, Exporter Responsibilities. CDIF makes no statement about whether or not the importer needs to be able to retain the contents of information transferred (especially where extensibility has been used). For a more detailed discussion of the responsibilities of an importer, see Section 6, Importer Responsibilities. The following rules must be adhered to when using extensibility:

- Standardized subject areas may not be extended. This rule ensures that a consistent basis for importers to interpret is retained.
- A new, additional, subject area must be created to contain the extended meta-objects.
- The valid values of enumerated types are invariant by subject area. When an additional enumerated value is added it is added across the working meta-model, not for a specific subject area.
- Standardized meta-objects may also be used in exporter-defined subject areas.
- When reusing an existing meta-object in extensibility, the transfer must contain a reference to a subject area that contains the full definition of the meta-object.
- When any meta-object is added (i.e., a meta-entity, meta-relationship, meta-attribute, or subject area) a value must be defined for the meta-meta-attribute called *CDIFMetaIdentifier*. This identifier may have up to 32 characters but shall start with an alphabetic character and must be unique. The *CDIFMetaIdentifiers* of meta-objects in the CDIF Integrated Meta-model begin with a numeric character. No further structure to the identifier shall be given.

### 4.2 Extensibility

#### 4.2.1 Add SubjectArea

An exporter may define a new subject area for the meta-model, provided the *CDIFMetaIdentifier* given to the new subject area is not already used by another attributable meta-object that has been created through extensibility in the transfer. The subject area may use both existing collectable meta-objects (i.e., meta-entities, meta-relationships, meta-attributes) and newly defined ones. (See Section 3.4.4, Rules for Defining Subject Area Scope for further details.)

## 4.2.2 Add CollectableMetaObject.IsUsedIn.SubjectArea

An exporter must define the collectable meta-objects that are used explicitly in a subject area. Inherited meta-attributes and meta-relationships are implicitly used and therefore this meta-meta-relationship shall not be defined for them. (See 3.4.4, Rules for Defining Subject Area Scope for further details.)

## 4.2.3 Add Meta-entities

An exporter may add a meta-entity, provided the *CDIFMetaIdentifier* given to the new meta-entity is not already used by another attributable meta-object that has been created through extensibility. All meta-entities must have *RootObject* (as defined in the CDIF Integrated Meta-model) as their eventual supertype, and therefore inherit the meta-attribute called *CDIFIdentifier*.

## 4.2.4 Add Meta-relationships

An exporter may add a meta-relationship, provided the *CDIFMetaIdentifier* given to the new meta-relationship is not already used by another attributable meta-object that has been created through extensibility. All meta-relationships must have *RootObject* (as defined in the CDIF Integrated Meta-model) as their eventual supertype and therefore inherit the meta-attribute called *CDIFIdentifier*. When adding a meta-relationship, the rules defined in Section 3.4.4, Rules for Defining Subject Area Scope, regarding the inclusion of meta-entities in subject areas must be observed.

## 4.2.5 Add AttributableMetaObject.HasSubtype.AttributableMetaObject

An exporter may define additional supertype associations for existing meta-entities and meta-relationships. Both subtype and supertype *AttributableMetaObjects* must be defined before the association may be specified. These *AttributableMetaObjects* may either be existing *AttributableMetaObjects* from the CDIF Integrated Meta-model, or may be created using extensibility. The supertype association is defined by an instance of the *AttributableMetaObject.HasSubtype.AttributableMetaObject* meta-meta-relationship. A new supertype association shall not be added to *RootObject*.

## 4.2.6 Add Meta-attribute to a Meta-entity or Meta-relationship

An exporter may add additional meta-attributes to a meta-entity or a meta-relationship.

## 4.2.7 Add Legal Enumerated Value to a Meta-attribute

An exporter may define additional legal values for a meta-attribute whose data type is enumerated. Enumerated values are not defined within the context of a subject area. New values, along with all the existing values are available to all subject areas in the working meta-model.

## 1 **5. EXPORTER RESPONSIBILITIES**

### 2 **5.1 Introduction**

3 This section describes the responsibilities of a CDIF exporter. Note that this section does not  
4 describe any syntax or encoding-specific responsibilities; these may be found in the definition of  
5 the syntax and encoding itself.

### 6 **5.2 Extensibility**

7 When an exporter extends the CDIF Integrated Meta-model, the extensions should not overlap  
8 semantically with the definitions contained in the Integrated Meta-model. The exporter shall  
9 minimize the use of extensibility by using standard definitions from the Integrated Meta-model  
10 where they exist. Wherever extensibility is used the tool user can not be assured that transfer of  
11 information between the exporter and the importer will provide meaningful results.

### 12 **5.3 Maximum Output**

13 All CDIF exporters should export as much information about a model as is possible. The  
14 exporter should not try to anticipate the capabilities of the importer (other than to include, or  
15 exclude, informed extensions to the standardized meta-model, that may be understood by some  
16 importers). This strategy ensures that the maximum amount of information will be transferred  
17 between the exporter and the importer.

### 18 **5.4 Meta-attributes**

19 For the instances of any meta-entities transferred by an exporter, values for all the mandatory  
20 (*IsOptional: False*) meta-attributes used in the subject area must be provided, even where no data  
21 is readily available.

## 6. IMPORTER RESPONSIBILITIES

### 6.1 Introduction

This section describes the responsibilities of a CDIF importer. Note that this section does not describe any syntax or encoding-specific responsibilities; these may be found in the definition of the syntax and encoding itself.

### 6.2 Information Retention

Any importer must be able to read all information contained in a CDIF Transfer even if it doesn't do anything with the information once it has been read. This means that an importer must, for example, always parse a CDIF Transfer correctly - even extensions. However, this does not mean the importer needs to do anything specific (e.g., retain) with the information once it has been accepted. The importer need not understand all the semantics defined within the CDIF Integrated Meta-model.

### 6.3 Working Meta-model

While a CDIF Transfer is being imported, a CDIF importer builds a "working meta-model", see Section 3.4.2, Working Meta-model. The initial Working Meta-model contains a single meta-object called *RootObject*. The Working Meta-model is then populated by the importer as it encounters and processes certain types of information in the transfer. Examples of transfer contents that populate the Working Meta-model include references to versions of the predefined CDIF Integrated Meta-model subject areas, the addition of a new *MetaEntity*, and so on.

When a CDIF importer encounters a reference to a meta-object in the meta-model, that meta-object must already exist in the current Working Meta-model. Forward references in the Meta-model Section of a CDIF Transfer are not permitted.

Note that the Working Meta-model may be only a conceptual device and not implemented explicitly in an importer. In some cases, where the Integrated Meta-model has been used as the internal schema by the tool, mapping the components of a CDIF Transfer directly to a tools' internal schema will be possible.

### 6.4 Meta-attributes

There are no default values specified in the CDIF Integrated Meta-model. Therefore, when an importer encounters an optional meta-attribute for which no value is given, it should interpret this lack of data as meaning that there was no data available to the exporter.

## 7. META-META-MODEL OVERVIEW

### 7.1 Introduction

This section provides an overview to the CDIF Meta-meta-model. The detailed definitions of the meta-meta-objects that form the Meta-meta-model can be found in Section 8, Meta-meta-model Definitions.

### 7.2 Meta-meta-model Overview Format

The following topics are presented in this section:

- `AttributableMetaObject` Hierarchy
- Entity-Relationship Diagram of the CDIF Meta-meta-model
- Meta-meta-entity Summary
- Meta-meta-relationship Summary

### 7.3 `AttributableMetaObject` Hierarchy<sup>1</sup>

#### 7.3.1 Introduction

The `AttributableMetaObject` Hierarchy in this section contains all the meta-meta-entities and meta-meta-relationships defined in the CDIF Meta-meta-model.

The `AttributableMetaObject` Hierarchy is presented as an indented list of meta-meta-entity and meta-meta-relationship names. The indentation indicates the subtyping; all names indented below a given name are subtypes of that meta-meta-object. For example, *AttributableMetaObject* and *MetaAttribute* are both subtypes of *CollectableMetaObject*, which is shown as follows:

```
CollectableMetaObject
  AttributableMetaObject
  MetaAttribute
```

---

<sup>1</sup> The term "*AttributableMetaObject*" rather than "Meta-meta-object" can be used here since, as has been said, the CDIF Meta-meta-model is defined in terms of itself; that is, meta-meta-entities and meta-meta-relationships are *AttributableMetaObjects*.

## 7.3.2 Hierarchy

MetaObject

    SubjectArea

    CollectableMetaObject

        AttributableMetaObject

            MetaEntity

            MetaRelationship

        MetaAttribute

AttributableMetaObject.HasSubtype.AttributableMetaObject

CollectableMetaObject.IsUsedIn.SubjectArea

MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject

MetaRelationship.HasDestination.MetaEntity

MetaRelationship.HasSource.MetaEntity

## 7.4 Meta-meta-model Description

### 7.4.1 Description

The CDIF Meta-meta-model contains the definition and structure of data that may be included in CDIF-compliant meta-models.

*MetaObject* is the abstract name given to any thing that may appear in a meta-model, or in a meta-meta-model since it is defined in terms of itself. *MetaObject* has several meta-meta-attributes, including *CDIFMetaIdentifier* which must be unique for all *MetaObjects*. These meta-meta-attributes are inherited by all the meta-meta-entities in the Meta-meta-model. *MetaObject* will never be directly instantiated; only as one of its direct or indirect subtypes. The two subtypes of *MetaObject* are *SubjectArea* and *CollectableMetaObject*.

*SubjectAreas* are groupings of *MetaObjects* that address either or both of the following:

- some functional area or technique found within the Systems Development Life Cycle
- areas that provide similar functionality.

*CollectableMetaObjects* are those *MetaObjects* that may be grouped into a *SubjectArea*; they may either be *MetaAttributes*, or *AttributableMetaObjects*.

*AttributableMetaObjects* (that is *MetaEntities* and *MetaRelationships*) may have *MetaAttributes*. *MetaAttributes* are the properties, or characteristics of *AttributableMetaObjects*. Both *MetaEntities* and *MetaRelationships* may have subtypes for inheritance.

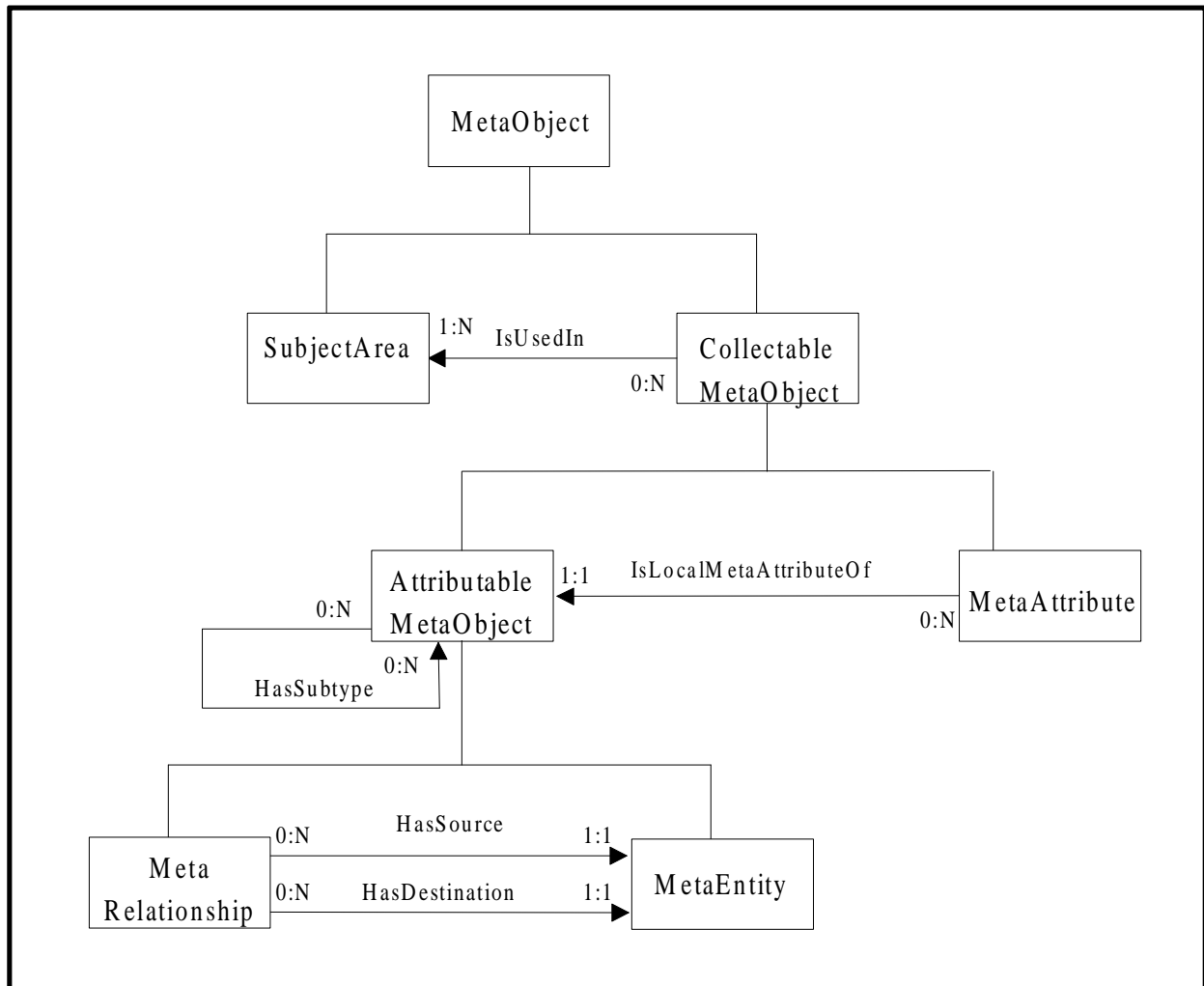
## 7.4.2 Uniqueness of Names

Names are not necessarily unique across all *MetaObjects*. Uniqueness of names within the Working Meta-model is considered for *SubjectAreas*, *MetaEntities*, *MetaRelationships* and *MetaAttributes* separately:

- The name given to each subject area must be unique within a Working Meta-model.
- The name given to each meta-entity must be unique within a Working Meta-model.
- The full name of each meta-relationship must be unique within a Working Meta-model, where the full name is the concatenation of the source meta-entity name, the meta-relationship name, and the destination meta-relationship name.
- The name of a meta-attribute concatenated with the name of the *AttributableMetaObject* it defines must be unique within a Working Meta-model.

## 1 7.5 Meta-meta-model Diagram

2 Figure 12 shows a diagram of the CDIF Meta-meta-model.



3

4

5

Figure 12  
The CDIF Meta-meta-model

## 6 7.6 Meta-meta-object Summary Format

7 For each meta-meta-entity in the CDIF Meta-meta-model all the meta-meta-attributes, both  
 8 inherited and local, are listed. The inherited meta-meta-attributes are shown first, in italics,  
 9 sequenced alphabetically within parent meta-meta-entity. The local meta-meta-attributes are then  
 10 listed alphabetically in normal font. The optionality of each meta-meta-attribute is given.

11 Each meta-meta-relationship in the CDIF Meta-meta-model is listed in alphabetical order.

---

## 1 7.7 Meta-meta-entity Summary

---

### 2 **Meta-meta-entity Name**

3	<i>Inherited Meta-meta-attribute Name</i>	
4	Local Meta-meta-attribute Name	Mandatory/Optional

---

### 5 **AttributableMetaObject**

6	<i>Aliases</i>	<i>Optional</i>
7	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
8	<i>Constraints</i>	<i>Optional</i>
9	<i>Description</i>	<i>Mandatory</i>
10	<i>Name</i>	<i>Mandatory</i>
11	<i>Usage</i>	<i>Optional</i>

### 12 **CollectableMetaObject**

13	<i>Aliases</i>	<i>Optional</i>
14	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
15	<i>Constraints</i>	<i>Optional</i>
16	<i>Description</i>	<i>Mandatory</i>
17	<i>Name</i>	<i>Mandatory</i>
18	<i>Usage</i>	<i>Optional</i>

### 19 **MetaAttribute**

20	<i>Aliases</i>	<i>Optional</i>
21	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
22	<i>Constraints</i>	<i>Optional</i>
23	<i>Description</i>	<i>Mandatory</i>
24	<i>Name</i>	<i>Mandatory</i>
25	<i>Usage</i>	<i>Optional</i>
26	<i>DataType</i>	<i>Mandatory</i>
27	<i>Domain</i>	<i>Optional</i>
28	<i>IsOptional</i>	<i>Mandatory</i>
29	<i>Length</i>	<i>Optional</i>

---

1	<b>Meta-meta-entity Name</b>	
2	<i>Inherited Meta-meta-attribute Name</i>	
3	Local Meta-meta-attribute Name	Mandatory/Optional
4	<b>MetaEntity</b>	
5	<i>Aliases</i>	<i>Optional</i>
6	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
7	<i>Constraints</i>	<i>Optional</i>
8	<i>Description</i>	<i>Mandatory</i>
9	<i>Name</i>	<i>Mandatory</i>
10	<i>Usage</i>	<i>Optional</i>
11	Type	Optional
12		
13	<b>MetaObject</b>	
14	<b>Aliases</b>	<b>Optional</b>
15	CDIFMetaIdentifier	Mandatory
16	Constraints	Optional
17	Description	Mandatory
18	Name	Mandatory
19	Usage	Optional
20		
21	<b>MetaRelationship</b>	
22	<i>Aliases</i>	<i>Optional</i>
23	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
24	<i>Constraints</i>	<i>Optional</i>
25	<i>Description</i>	<i>Mandatory</i>
26	<i>Name</i>	<i>Mandatory</i>
27	<i>Usage</i>	<i>Optional</i>
28	MaxDestCard	Mandatory
29	MaxSourceCard	Mandatory
30	MinDestCard	Mandatory
31	MinSourceCard	Mandatory
32		
33	<b>SubjectArea</b>	
34	<i>Aliases</i>	<i>Optional</i>
35	<i>CDIFMetaIdentifier</i>	<i>Mandatory</i>
36	<i>Constraints</i>	<i>Optional</i>
37	<i>Description</i>	<i>Mandatory</i>
38	<i>Name</i>	<i>Mandatory</i>
39	<i>Usage</i>	<i>Optional</i>
40	VersionNumber	Mandatory

---

---

## 1 7.8 Meta-meta-relationship Summary

---

### 2 Meta-meta-relationship Name<sup>2</sup>

---

4 **AttributableMetaObject.HasSubtype.AttributableMetaObject**

5

6 **CollectableMetaObject.IsUsedIn.SubjectArea**

7

8 **MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject**

9

10 **MetaRelationship.HasDestination.MetaEntity**

11

12 **MetaRelationship.HasSource.MetaEntity**

---

<sup>2</sup> None of the meta-meta-relationships in the CDIF Meta-meta-model are attributed.

## 1 **8. META-META-MODEL DEFINITIONS**

### 2 **8.1 Meta-meta-object Definition Format**

3 The meta-meta-object definitions in this section are the formal definitions of the meta-meta-  
4 objects in the CDIF Meta-meta-model. These take precedence over any descriptive or  
5 explanatory text in other parts of the document.

6 The definition of each meta-meta-entity in the Meta-meta-model is structured according to the  
7 classes of information listed in Table 4.

8 For each meta-meta-entity:

- 9       • Inherited meta-meta-attributes and inherited meta-meta-relationships are listed by  
10       parent meta-meta-entity in a top-down hierarchical order. Within each sub-list,  
11       meta-meta-attributes and meta-meta-relationships are listed alphabetically.
- 12       • Local meta-meta-attributes are listed alphabetically.
- 13       • Detailed meta-meta-attribute descriptions are listed alphabetically. These  
14       definitions are structured according to the classes of information listed in Table 5.

1  
2

Table 4  
Contents of Meta-meta-entity Definition

<b>8.N.N</b>	<b>Meta-meta-entity Name</b>
<b>NAME .....</b>	<b>meta-entity name</b>
<b>SUBTYPEOF.....</b>	List of direct supertypes of the meta-meta-entity...
<b>SUPERTYPEOF .....</b>	List of direct subtypes of the meta-meta-entity
<b>DESCRIPTION.....</b>	
<b>USAGE.....</b>	
<b>ALIASES .....</b>	
<b>CONSTRAINTS.....</b>	
<b>TYPE .....</b>	
<hr/>	
<b>INHERITED META-META-ATTRIBUTES</b>	
List of meta-attributes inherited from all supertypes of the meta-meta-entity, with the name of the meta-meta-entity for which the meta-meta-attribute is defined...	
meta-meta-attribute name      from supertype meta-meta-entity	
<hr/>	
<b>LOCAL META-META-ATTRIBUTES</b>	
List of meta-attributes defined for the meta-meta-entity...	
meta-meta-attribute name	
<hr/>	
<b>INHERITED META-META-RELATIONSHIPS</b>	
List of meta-meta-relationships inherited from all supertypes of the meta-meta-entity...	
meta-meta-relationship name	
<hr/>	
<b>LOCAL META-META-RELATIONSHIPS</b>	
List of meta-meta-relationships defined for the meta-meta-entity...	
meta-meta-relationship name	
<hr/>	
Full definitions of all local meta-meta-attributes appear at this point. See Table 5 for format of local meta-meta-attribute definitions.	

1  
2

Table 5  
Contents of Meta-meta-attribute Definition

<b>META-META-ATTRIBUTE NAME .....</b> meta-meta-attribute name
DESCRIPTION .....
USAGE.....
ALIASES .....
CONSTRAINTS.....
DATA TYPE .....
DOMAIN .....
LENGTH.....
ISOPTIONAL.....

3

4 The definition for each meta-meta-relationship in the Meta-meta-model is structured according to  
5 the classes of information listed in Table 6.<sup>1</sup>

6  
7

Table 6  
Contents of Meta-meta-relationship Definition

<b>8.N.N</b>	<b>Full Meta-meta-relationship Name</b>
<b>NAME .....</b>	<b>meta-meta-relationship name</b>
MINSOURCECARD .....	Minimum source meta-meta-entity cardinality
MAXSOURCECARD.....	Maximum source meta-meta-entity cardinality
MINDESTCARD .....	Minimum destination meta-meta-entity cardinality
MAXDESTCARD .....	Maximum destination meta-meta-entity cardinality
DESCRIPTION.....	
USAGE.....	
ALIASES .....	
CONSTRAINTS.....	

8

---

<sup>1</sup> Although the meta-meta-model permits inheritance in meta-meta-relationships, there are no instances in the meta-meta-model. Therefore, the definition does not include lines for *SubtypeOf* or *SupertypeOf*.

## 8.2 Meta-meta-entity and Meta-meta-attribute Definitions

### 8.2.1 **AttributableMetaObject**

NAME	<b>AttributableMetaObject</b>
SUBTYPEOF	CollectableMetaObject
SUPERTYPEOF	MetaRelationship
METAENTITY	
DESCRIPTION	Defines an abstract supertype for those meta-meta-objects (i.e., <i>MetaEntities</i> and <i>MetaRelationships</i> ) that may have meta-attributes.
USAGE	Used to allow meta-entities and meta-relationships to have meta-attributes in meta-models. This supertype shall not normally be instantiated.
ALIASES	
CONSTRAINTS	
TYPE	Kernel

---

#### INHERITED META-META-ATTRIBUTES

Aliases	from MetaObject
CDIFMetaIdentifier	from MetaObject
Constraints	from MetaObject
Description	from MetaObject
Name	from MetaObject
Usage	from MetaObject

---

#### LOCAL META-META-ATTRIBUTES

---

#### INHERITED META-META-RELATIONSHIPS

CollectableMetaObject.IsUsedIn.SubjectArea

---

- 1 LOCAL META-META-RELATIONSHIPS
  - 2     AttributableMetaObject.HasSubtype.AttributableMetaObject
  - 3     MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
  - 4
- 
-

---

## 1 8.2.2 CollectableMetaObject

NAME	<b>CollectableMetaObject</b>
SUBTYPEOF	MetaObject
SUPERTYPEOF	AttributableMetaObject MetaAttribute
DESCRIPTION	Defines an abstract supertype for those meta-meta-objects (i.e., <i>AttributableMetaObjects</i> and <i>MetaAttributes</i> ) that may be grouped into subject area views.
USAGE	Used to group those meta-objects that may be used in subject areas. This supertype shall not be instantiated.
ALIASES	
CONSTRAINTS	
TYPE	Kernel

2

---

### 3 INHERITED META-META-ATTRIBUTES

4	Aliases	from MetaObject
5	CDIFMetaIdentifier	from MetaObject
6	Constraints	from MetaObject
7	Description	from MetaObject
8	Name	from MetaObject
9	Usage	from MetaObject

10

---

### 11 LOCAL META-META-ATTRIBUTES

12

---

### 13 INHERITED META-META-RELATIONSHIPS

14

---

### 15 LOCAL META-META-RELATIONSHIPS

16 CollectableMetaObject.IsUsedIn.SubjectArea

17

---

## 1 8.2.3 MetaAttribute

**NAME** ..... **MetaAttribute**  
**SUBTYPEOF**..... CollectableMetaObject  
**SUPERTYPEOF** .....  
**DESCRIPTION**..... Defines a specific property of a meta-entity or meta-relationship  
 (i.e., an *AttributableMetaObject*).  
**USAGE**.....  
**ALIASES** ..... Property, Characteristic.  
**CONSTRAINTS**..... The names of all *MetaAttributes* must be unique within their  
 parent object within the working meta-model. If the attribute is an  
 attribute of a subtype, this uniqueness rule applies all the way up  
 that particular inheritance thread Within the CDIF Integrated  
 Meta-model, all meta-attributes that are semantically identical are  
 given the same names.  
**TYPE** ..... Kernel

2

---

### 3 INHERITED META-META-ATTRIBUTES

4	Aliases	from MetaObject
5	CDIFMetaIdentifier	from MetaObject
6	Constraints	from MetaObject
7	Description	from MetaObject
8	Name	from MetaObject
9	Usage	from MetaObject

10

---

### 11 LOCAL META-META-ATTRIBUTES

12	DataType
13	Domain
14	IsOptional
15	Length

16

---

### 17 INHERITED META-META-RELATIONSHIPS

18	CollectableMetaObject.IsUsedIn.SubjectArea
----	--

19

---

### 20 LOCAL META-META-RELATIONSHIPS

21	MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
----	---

22

---

**META-META-ATTRIBUTE NAME .....** **DataType**  
DESCRIPTION..... The type of data that can be stored in a meta-attribute.  
USAGE.....  
ALIASES ..... Usage, Type  
CONSTRAINTS.....  
DATATYPE ..... Enumerated  
DOMAIN ..... Identifier, Integer, String, Text, Boolean, Date, Time,  
BitMap, Float, Enumerated, Point, PointList, IntegerList  
LENGTH.....  
ISOPTIONAL..... False

1

---

**META-META-ATTRIBUTE NAME .....** **Domain**  
DESCRIPTION..... The pool of valid values for a meta-attribute.  
USAGE..... This is not a computable field. When the type of the meta-  
attribute is *Enumerated*, the domain shall contain a  
comma-separated list of unique, valid values for the meta-  
attribute. When the type of the meta-attribute is not  
*Enumerated*, the domain shall be expressed in  
unambiguous English. Where domain is not specified, the  
full range of values for the specified data type shall be  
assumed  
ALIASES .....  
CONSTRAINTS..... The Domain must be specified where the data type is  
enumerated.  
DATATYPE ..... Text  
DOMAIN .....  
LENGTH.....  
ISOPTIONAL..... True

2

---

**META-META-ATTRIBUTE NAME .....** **IsOptional**  
DESCRIPTION..... Indicates whether or not a value must be supplied for a  
given meta-attribute for every instance of its  
*AttributableMetaObject*.  
USAGE.....  
ALIASES ..... NotRequired, NotMandatory, Nullable  
CONSTRAINTS.....  
DATATYPE ..... Boolean  
DOMAIN .....  
LENGTH.....  
ISOPTIONAL..... False

3

---

<b>META-META-ATTRIBUTE NAME .....</b>	<b>Length</b>
DESCRIPTION.....	The maximum length allowed for a meta-attribute of data type String.
USAGE.....	Used only for Strings.
ALIASES .....	Size.
CONSTRAINTS.....	This should only be specified for a meta-attribute of data type String.
DATATYPE.....	Integer
DOMAIN.....	1<=n<=1024
LENGTH.....	
ISOPTIONAL.....	True

---

---

1

---

## 8.2.4 MetaEntity

1 **NAME** ..... **MetaEntity**  
SUBTYPEOF..... **AttributableMetaObject**  
SUPERTYPEOF.....  
DESCRIPTION..... *MetaEntity* defines the characteristics of all meta-entities that occur in a meta-model.  
USAGE.....  
ALIASES .....  
CONSTRAINTS..... The names of all *MetaEntities* must be unique within the meta-layer that contains them within the working meta-model. All *MetaEntities* must have *RootObject* as their eventual supertype. For subtype meta-entities, the Description, Usage, Aliases, and Constraints described local to this meta-entity are in addition to those of the supertype meta-entity. The meta-relationships in which the supertype meta-entity participated will be inherited by the subtype meta-entity. The meta-attributes of the supertype meta-entity are also inherited.  
TYPE ..... Kernel

---

### INHERITED META-META-ATTRIBUTES

4 Aliases from MetaObject  
5 CDIFMetaIdentifier from MetaObject  
6 Constraints from MetaObject  
7 Description from MetaObject  
8 Name from MetaObject  
9 Usage from MetaObject

---

### LOCAL META-META-ATTRIBUTES

12 Type

---

### INHERITED META-META-RELATIONSHIPS

15 AttributableMetaObject.HasSubtype.AttributableMetaObject  
16 CollectableMetaObject.IsUsedIn.SubjectArea  
17 MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject

---

### LOCAL META-META-RELATIONSHIPS

21 MetaRelationship.HasDestination.MetaEntity  
22 MetaRelationship.HasSource.MetaEntity

---

---

**META-META-ATTRIBUTE NAME ..... Type**

DESCRIPTION.....	A classification for the meta-entity that may convey additional semantic meaning which may be of use to the importer.
USAGE.....	Is used to indicate which modeling semantic is being represented by the meta-entity. Kernel: This classification is used for a meta-entity whose instances are meaningful without the occurrences of other meta-entities. Characteristic: Each instance of a characteristic meta-entity is logically only related to one instance of one other meta-entity. Therefore an importer could incorporate the meta-attributes of a characteristic meta-entity with those of the 'owning' meta-entity, where the owning meta-entity is the one to which the characteristic meta-entity is related with a cardinality of 1:1. Associative: An associative meta-entity is used to represent a meta-relationship between other meta-entities. It is used when a meta-relationship does not otherwise provide sufficient mechanisms.
ALIASES .....	Classification, RepresentedSemantic
CONSTRAINTS.....	
DATATYPE .....	Enumerated
DOMAIN .....	Kernel, Characteristic, Associative
LENGTH.....	
ISOPTIONAL.....	True

---

1 **8.2.5 MetaObject**

NAME ..... **MetaObject**  
SUBTYPEOF..... -  
SUPERTYPEOF ..... CollectableMetaObject  
SUBJECTAREA .....  
DESCRIPTION..... Defines shared meta-meta-attributes of all meta-meta-entities. It is  
the root of the type hierarchy in the CDIF Meta-meta-model.  
ALIASES .....  
USAGE..... This supertype shall not be instantiated.  
CONSTRAINTS.....  
TYPE ..... Kernel

2

---

3 INHERITED META-META-ATTRIBUTES

4

---

5 LOCAL META-META-ATTRIBUTES

6 Aliases  
7 CDIFMetaIdentifier  
8 Constraints  
9 Description  
10 Name  
11 Usage  
12

---

13 INHERITED META-META-RELATIONSHIPS

14

---

15 LOCAL META-META-RELATIONSHIPS

16

---

**META-META-ATTRIBUTE NAME ..... Aliases**  
DESCRIPTION..... Specifies alternate names for a given meta-object by which  
the meta-object may be more normally known in any  
particular CASE environment.  
USAGE..... Aliases are defined in CDIF meta-models to help the reader  
identify the semantics of a given meta-object. Note that  
aliases do not provide the same degree of precision as a  
meta-object's name. Aliases are not used in the CDIF  
transfer as an alternative to *Name*.  
ALIASES ..... Synonyms, AlternateNames, AKAs  
CONSTRAINTS..... The value of this meta-meta-attribute must be a comma  
separated list of names that conform to the rules for CDIF  
identifiers.  
DATATYPE..... String  
DOMAIN.....  
LENGTH..... 1024  
ISOPTIONAL..... True

17

**META-META-ATTRIBUTE NAME ..... CDIFMetaIdentifier**

DESCRIPTION..... A unique identifier for a *MetaObject*.  
 USAGE..... *CDIFMetaIdentifiers* are required in a CDIF transfer to allow meta-objects to be referenced in the Meta-model Section. They are not pertinent to repositories.

ALIASES .....

CONSTRAINTS..... Meta-objects in the standardized meta-model shall have a value for *CDIFMetaIdentifier* starting with a numeric character. Meta-objects added using extensibility must start with an alphabetic character.

DATATYPE..... Identifier  
 DOMAIN.....  
 LENGTH.....  
 ISOPTIONAL..... False

1

**META-META-ATTRIBUTE NAME ..... Constraints**

DESCRIPTION..... Specifies restrictions on a meta-object which can involve its interrelationships with one or more other objects.  
 USAGE..... Constraints shall be expressed unambiguously. Constraints are non-computable. For meta-relationships, mutual exclusivity constraints should be defined, specifying the other meta-relationships in the mutual exclusivity set.

ALIASES ..... Restriction, Rule, Obligation, Policy  
 CONSTRAINTS.....  
 DATATYPE..... Text  
 DOMAIN.....  
 LENGTH.....  
 ISOPTIONAL..... True

2

**META-META-ATTRIBUTE NAME ..... Description**

DESCRIPTION..... The description of a meta-object.  
 USAGE..... Used to explain the inherent nature of the meta-object. Not used to convey constraints, usage or information about valid values (i.e., domain information).

ALIASES ..... Definition  
 CONSTRAINTS.....  
 DATATYPE..... Text  
 DOMAIN.....  
 LENGTH.....  
 ISOPTIONAL..... False

3

---

<b>META-META-ATTRIBUTE NAME .....</b>	<b>Name</b>
DESCRIPTION.....	The name used to refer to a meta-object.
USAGE.....	Names should be concise but not at the expense of precision or clarity.
ALIASES .....	Title, Identifier
CONSTRAINTS.....	The first letter of each of the component words in the name shall be capitalized. All other letters in the name shall be lowercase (unless one of the component words is naturally uppercase - e.g., an acronym). Uniqueness constraints are specified at the level of instantiation.
DATATYPE.....	Identifier
DOMAIN.....	
LENGTH.....	
ISOPTIONAL.....	False

1

---

<b>META-META-ATTRIBUTE NAME .....</b>	<b>Usage</b>
DESCRIPTION.....	Comments to clarify how a given meta-object should be used by an importer/exporter writer.
USAGE.....	Can describe the purpose of the meta-object. The usage for abstract supertype <i>MetaEntities</i> shall state that the <i>MetaEntity</i> may not be instantiated.
ALIASES .....	Examples, Hints, Explanation
CONSTRAINTS.....	
DATATYPE.....	Text
DOMAIN.....	
LENGTH.....	
ISOPTIONAL.....	True

---

2

---

## 1 8.2.6 MetaRelationship

NAME .....	<b>MetaRelationship</b>
SUBTYPEOF .....	AttributableMetaObject
SUPERTYPEOF .....	
DESCRIPTION .....	Defines an association (i.e., a meta-relationship) between two meta-entities, one of which is called the source and one of which is called the destination, or between a meta-entity and itself.
USAGE .....	
ALIASES .....	
CONSTRAINTS .....	The full names of all <i>MetaRelationships</i> must be unique within the meta-layer that contains them within the working meta-model. The full name of a meta-relationship is formed by concatenating the name of the source meta-entity, the name of the meta-relationship and the name of the destination meta-entity (e.g., <i>Box.Contains.Part</i> , where source meta-entity= <i>Box</i> , meta-relationship= <i>Contains</i> and destination meta-entity= <i>Part</i> ). For subtype meta-relationships, the Description, Usage, Aliases, and Constraints defined local to this meta-relationship are considered to be in addition to those of the supertype meta-relationship. The cardinalities of the subtype meta-relationship may only be the same, or more constraining than those of the supertype meta-relationship. The meta-attributes of the supertype meta-relationship are inherited by the subtype. Subtype meta-relationships may be defined between the same meta-entities as the supertype, or may take subtypes of those meta-entities as the source and destination.
TYPE .....	Associative

2

---

### 3 INHERITED META-META-ATTRIBUTES

4	Aliases	from MetaObject
5	CDIFMetaIdentifier	from MetaObject
6	Constraints	from MetaObject
7	Description	from MetaObject
8	Name	from MetaObject
9	Usage	from MetaObject

10

---

### 11 LOCAL META-META-ATTRIBUTES

12	MaxDestCard
13	MaxSourceCard
14	MinDestCard
15	MinSourceCard

16

---

### 17 INHERITED META-META-RELATIONSHIPS

18	AttributableMetaObject.HasSubtype.AttributableMetaObject
19	CollectableMetaObject.IsUsedIn.SubjectArea
20	MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject

21

1 LOCAL META-META-RELATIONSHIPS  
2       MetaRelationship.HasDestination.MetaEntity  
3       MetaRelationship.HasSource.MetaEntity  
4

---

---

**META-META-ATTRIBUTE NAME ..... MaxDestCard**

DESCRIPTION..... Specifies the maximum occurrences of the destination meta-entity that have an association with any one occurrence of the source meta-entity via the meta-relationship.

USAGE.....

ALIASES .....

CONSTRAINTS..... Must be greater than or equal to the *MinDestCard*.

DATATYPE..... String

DOMAIN..... Positive integers from 1 up to the maximum number representable in a CDIF Integer. May also be the single character 'N' (case-insignificant) to indicate that the maximum cardinality is unbounded.

LENGTH..... 10

ISOPTIONAL..... False

5

---

---

**META-META-ATTRIBUTE NAME ..... MaxSourceCard**

DESCRIPTION..... Specifies the maximum occurrences of the source meta-entity that have an association with any one occurrence of the destination meta-entity via the meta-relationship.

USAGE.....

ALIASES .....

CONSTRAINTS..... Must be greater than or equal to the *MinSourceCard*.

DATATYPE..... String

DOMAIN..... Positive integers from 1 up to the maximum number representable in a CDIF Integer. May also be the single character 'N' (case-insignificant) to indicate that the maximum cardinality is unbounded.

LENGTH..... 10

ISOPTIONAL..... False

6

---

**META-META-ATTRIBUTE NAME ..... MinDestCard**

DESCRIPTION..... Specifies the minimum occurrences of the destination meta-entity that have an association with any one occurrence of the source meta-entity via the meta-relationship.

USAGE..... Value of 0 means optional, other values mean non-optional.

ALIASES .....

CONSTRAINTS..... Must be less than or equal to the *MaxDestCard*.

DATATYPE..... String

DOMAIN..... Positive integers from 0 up to the maximum number representable in a CDIF Integer.

LENGTH..... 10

ISOPTIONAL..... False

1

---

**META-META-ATTRIBUTE NAME ..... MinSourceCard**

DESCRIPTION..... Specifies the minimum occurrences of the source meta-entity that have an association with any one occurrence of the destination meta-entity via the meta-relationship.

USAGE..... Value of 0 means optional, other values mean non-optional.

ALIASES .....

CONSTRAINTS..... Must be less than or equal to the *MaxSourceCard*.

DATATYPE..... String

DOMAIN..... Positive integers from 0 up to the maximum number representable in a CDIF Integer.

LENGTH..... 10

ISOPTIONAL..... False

3

4

## 8.2.7 SubjectArea

**Name**..... **SubjectArea**  
SUBTYPEOF..... MetaObject  
SUPERTYPEOF .....

DESCRIPTION..... Defines a view of the underlying integrated CDIF meta-model containing those *CollectableMetaObjects* that are of significance to a single subject area. *CollectableMetaObjects* may be used in more than one subject area, but are defined only once in the CDIF Integrated Meta-model. The name of a subject area must be unique within the working meta-model.

USAGE..... A *SubjectArea* will be created for each distinct subject area of the meta-model. *SubjectAreas* will be created as the definition for separate standards within the CDIF Family of Standards and may also be created using extensibility.

ALIASES ..... View  
CONSTRAINTS..... The name of a subject area shall be unique.  
TYPE ..... Kernel

---

### INHERITED META-META-ATTRIBUTES

Aliases	from MetaObject
CDIFMetaIdentifier	from MetaObject
Constraints	from MetaObject
Description	from MetaObject
Name	from MetaObject
Usage	from MetaObject

---

### LOCAL META-META-ATTRIBUTES

VersionNumber

---

### INHERITED META-META-RELATIONSHIPS

---

### LOCAL META-META-RELATIONSHIPS

CollectableMetaObject.IsUsedIn.SubjectArea

---

### META-META-ATTRIBUTE NAME ..... **VersionNumber**

DESCRIPTION..... The version of the definition of the subject area.  
USAGE..... The standardized subject areas will have a version number with the form nn.nn.

ALIASES .....

CONSTRAINTS.....

DATATYPE..... String  
DOMAIN.....  
LENGTH..... 8  
ISOPTIONAL..... False

1 **8.3 Meta-meta-relationship Definitions**

2 **8.3.1 *AttributableMetaObject.HasSubtype*.*AttributableMetaObject***

NAME .....	<b>HasSubtype</b>
MINSOURCECARD .....	0
MAXSOURCECARD .....	N
MINDESTCARD .....	0
MAXDESTCARD .....	N
DESCRIPTION .....	Indicates that <i>AttributableMetaObjects</i> , that is meta-entities and meta-relationships, can have subtypes.
USAGE .....	The subtype attributable meta-object will inherit the Description, Constraints, Usage, and Aliases of the supertype. The Type of a meta-entity is not inherited by a subtype meta-entity. Subtype attributable meta-objects inherit the meta-attributes of the supertype attributable meta-object and may have additional, locally defined meta-attributes. Subtype meta-entities inherit any meta-relationships in which the supertype meta-entity participated. Subtype meta-relationships may only have cardinalities defined that are the same, or more constraining than those of the supertype meta-relationship. Subtype meta-relationships may be defined between the same meta-entities as the supertype meta-relationship, or have subtypes of those meta-entities as the source or destination.
ALIASES .....	
CONSTRAINTS .....	An <i>AttributableMetaObject</i> cannot be an immediate supertype of itself, nor can it through a chain of supertypes be an indirect supertype of itself (i.e., recursive supertypes are not allowed). This meta-meta-relationship must be instantiated for every <i>AttributableMetaObject</i> in the meta-model except <i>RootObject</i> , since <i>RootObject</i> must be the eventual supertype of every <i>AttributableMetaObject</i> .

3

---

---

### 1 8.3.2 CollectableMetaObject.IsUsedIn.SubjectArea

NAME .....	<b>IsUsedIn</b>
MINSOURCECARD .....	0
MAXSOURCECARD .....	N
MINDESTCARD .....	1
MAXDESTCARD .....	N
DESCRIPTION .....	Indicates that a <i>CollectableMetaObject</i> , (that is an <i>AttributableMetaObject</i> or a <i>MetaAttribute</i> ) may be defined as being used within a <i>SubjectArea</i> .
USAGE.....	The meta-meta-relationship will be instantiated for a <i>CollectableMetaObject</i> when subject area views of the underlying integrated meta-model are being developed.
ALIASES .....	
CONSTRAINTS.....	Where this is created for a <i>MetaAttribute</i> , an instance must also be created for the <i>AttributableMetaObject</i> that the <i>MetaAttribute</i> describes. For example, the meta-attribute called <i>IsAbstract</i> may not be used in the Data Modeling subject area without the meta-entity called <i>Entity</i> also being used.

2

---

### 1 8.3.3 MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject

NAME ..... **IsLocalMetaAttributeOf**

MINSOURCECARD..... 0

MAXSOURCECARD..... N

MINDESTCARD ..... 1

MAXDESTCARD ..... 1

DESCRIPTION..... Indicates that a meta-attribute belongs to a single *MetaEntity* or a *MetaRelationship*. There is no sharing of meta-attributes between *AttributableMetaObjects*.

A subtype meta-object inherits the meta-attributes that have been defined for the supertype meta-object. Any additional meta-attributes are defined locally to the subtype meta-object and will require this meta-meta-relationship.

USAGE..... All *AttributableMetaObjects* will have a meta-attribute called *CDIFIdentifier*. Meta-entities inherit *CDIFIdentifier* from *RootObject*. *CDIFIdentifier* is explicitly defined as a meta-attribute for each meta-relationship.

ALIASES .....

CONSTRAINTS.....

2

---

---

### 1 8.3.4 MetaRelationship.HasDestination.MetaEntity

**NAME** ..... **HasDestination**  
**MINSOURCECARD**..... 0  
**MAXSOURCECARD**..... N  
**MINDESTCARD** ..... 1  
**MAXDESTCARD** ..... 1  
**DESCRIPTION**..... Indicates that a *MetaRelationship* must have a destination  
*MetaEntity*.  
**USAGE**.....  
**ALIASES** .....  
**CONSTRAINTS**..... A subtype meta-relationship shall have the same destination meta-  
entity as the supertype meta-relationship, or shall have an  
immediate or indirect subtype meta-entity as the destination meta-  
entity.  
The cardinalities for subtype meta-relationships must either be the  
same as, or be more constrained than, those of the supertype  
meta-relationship.

2

1 **8.3.5 MetaRelationship.HasSource.MetaEntity**

**NAME** ..... **HasSource**  
**MINSOURCECARD**..... 0  
**MAXSOURCECARD**..... N  
**MINDESTCARD** ..... 1  
**MAXDESTCARD** ..... 1  
**DESCRIPTION**..... Indicates that a *MetaRelationship* must have a source *MetaEntity*.  
**USAGE**.....  
**ALIASES** .....  
**CONSTRAINTS**..... A subtype meta-relationship shall have the same source meta-  
entity as the supertype meta-relationship, or shall have an  
immediate or indirect subtype meta-entity as the source meta-  
entity.  
The cardinalities for subtype meta-relationships must either be the  
same as, or be more constrained than, those of the supertype  
meta-relationship.

2

---

1  
2

This page is intentionally blank



## APPENDIX A META-MODEL EXAMPLE <sup>A</sup>

### A1 Introduction

This appendix contains a fragment of a meta-model that has been constructed using the rules defined in the CDIF Meta-meta-model. The subject area contained here is purely hypothetical in nature and **is not part of the CDIF Integrated Meta-model**. These rules are applied to a meta-model for a Project Specification subject area<sup>9</sup>. The graphical depiction of the subject area is shown in Figure A1. Detailed definitions for the subject area, the meta-entity called *Project* and the meta-relationship called *Project.IsComposedOf.Task* are included in Section A4, Subject Area Object Definitions, as examples.

### A2 The Use of the Meta-meta-model Definitions

#### A2.1 Instance of SubjectArea

In this example, *ProjectSpecification* is an instance of the meta-meta-entity called *SubjectArea*.

#### A2.2 Instances of MetaEntities

The following meta-objects in the subject area diagram are instances of the meta-meta-entity called *MetaEntity* and are therefore shown in boxes:

- ProjectSpecificationObject
- Person
- Project
- Skill
- Task
- WorkActual
- WorkAllocation
- WorkEstimate

Notice how the CDIF naming conventions produce names for the meta-entities like *ProjectSpecificationObject*.

#### A2.3 Instances of MetaRelationships

There are nine examples of instances of the meta-meta-entity called *MetaRelationship*. The meta-relationship name, shown by the meta-relationship line, must be placed between the name of the source meta-entity and destination meta-entity to give the following full, unique names:

---

<sup>a</sup>This appendix is not a formal part of the attached EIA standard, but is included for purposes of information only.

<sup>9</sup> A set of subject areas to support Project Management is being developed by the CDIF Technical Committee.

1           Person.Has.Skill  
 2           Person.IsAllocatedTo.WorkAllocation  
 3           WorkAllocation.ResultsIn.WorkActual  
 4           WorkAllocation.IsAgainst.WorkEstimate  
 5           WorkEstimate.Requires.Skill  
 6           WorkEstimate.IsAllocatedTo.Task  
 7           Project.IsComposedOf.Task  
 8           Task.DecomposesInto.Task  
 9           Task.DependsUpon.Task

10          The arrowhead on the meta-relationship line implies the source and destination meta-entity and  
 11          therefore the sequence to use when constructing the full name.

## 12          **A2.4           AttributableMetaObjects and MetaAttributes**

13          The meta-entities and meta-relationships listed above are all instances of the meta-meta-entity  
 14          called *AttributableMetaObject* and so they may have meta-attributes defined. For example,  
 15          *TimeUnits* is an instance of the meta-meta-entity called *MetaAttribute* that describes *Project*, an  
 16          instance of the meta-meta-entity called *MetaEntity*.

## 17          **A2.5           Instances of MetaEntity Subtypes**

18          *AttributableMetaObjects* (that is both *MetaEntities* and *MetaRelationships*) may have subtypes.  
 19          In this example, the meta-entity called *ProjectSpecificationObject* has four subtype meta-entities;  
 20          *Person*, *Skill*, *Project* and *Task*. These are meta-entities connected to the supertype meta-entity  
 21          by instances of the meta-meta-relationship called *MetaEntity.HasSubtype.MetaEntity*.

22          All meta-entities and meta-relationships in a CDIF subject area must have the  
 23          *AttributableMetaObject* called *RootObject* as their eventual supertype. The subtype meta-entities  
 24          and meta-relationships of *RootObject* are shown in the *AttributableMetaObject Hierarchy* below:

25          RootObject  
 26                  RootEntity  
 27                          ProjectSpecificationObject  
 28                                  Person  
 29                                  Project  
 30                                  Skill  
 31                                  Task  
 32                          WorkActual  
 33                          WorkAllocation  
 34                          WorkEstimate  
 35          RootEntity.IsRelatedTo.RootEntity  
 36                  Person.HasSkill.Skill  
 37                  Person.IsAllocatedTo.WorkAllocation  
 38                  WorkAllocation.ResultsIn.WorkActual  
 39                  WorkAllocation.IsAgainst.WorkEstimate  
 40                  WorkEstimate.Requires.Skill  
 41                  WorkEstimate.IsAllocated.Task  
 42                  Project.IsComponentOf.Task

Task.DependsOn.Task  
 Task.DecomposesInto.Task

### A3. Reading the Example Subject Area

The Project Specification subject area defines *Projects* which are composed of *Tasks*. These *Tasks*, that depend on other *Tasks*, can be decomposed and at their lowest level can be allocated *WorkEstimates* that record the amount of effort of a specific *Skill* (measured in a unit of time as defined in the *TimeUnits* meta-attribute of the meta-entity *Project*) required to complete them. There can be a number of *Persons*, possessing these and other *Skills*, who can be allocated to one or more of these *WorkEstimates*, through a *WorkAllocation*. Actual work completed by these *Persons* can then be recorded against one or more of these *WorkAllocations* with instances of the *WorkActual* meta-entity.

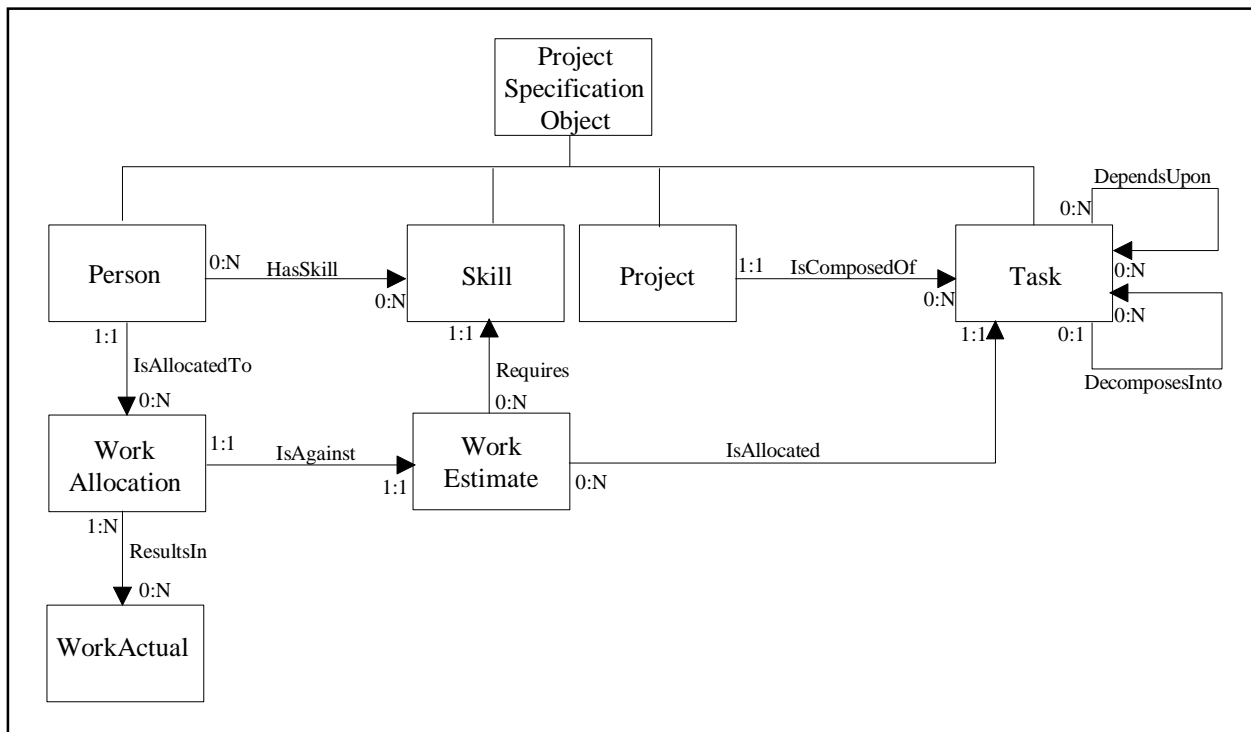


Figure A1  
 Example Meta-model for Project Specification

1 **A4. Subject Area Object Definitions**

2 **A4.1 Subject Area called *ProjectSpecification***

NAME ..... **ProjectSpecification**

CDIFMETAI DENTIFIER ..... PROJSPEC

DESCRIPTION ..... The Project Specification Subject Area provides the meta-entities and meta-relationships to allow the definition of models for the specification of project data.

USAGE.....

ALIASES .....

CONSTRAINTS.....

VERSION ..... 01.00

---

## A4.2 Meta-entity called *Project*

The full definition of the meta-entity called *Project* and one of its meta-attributes called *TimeUnits* is included as an example.

NAME ..... **Project**  
SUBTYPEOF..... ProjectSpecificationObject  
SUPERTYPEOF .....

CDIFMETAIDENTIFIER ..... PROJ0010  
DESCRIPTION..... A means of grouping sets of tasks necessary to achieve a predefined goal.  
USAGE..... Also used to record project-wide parameters such as *TimeUnits*.  
ALIASES ..... Program, Campaign  
CONSTRAINTS.....  
TYPE ..... Kernel

---

### INHERITED META-ATTRIBUTES

CDIFIdentifier	from RootObject
DateCreated	from RootObject
DateUpdated	from RootObject
TimeCreated	from RootObject
TimeUpdated	from RootObject
[Others	from ProjectSpecificationObject]

---

### LOCAL META-ATTRIBUTES

TimeUnits  
[Others not included in this example]

---

### INHERITED META-RELATIONSHIPS

RootEntity.IsRelatedTo.RootEntity

---

### LOCAL META-RELATIONSHIPS

Project.IsComposedOf.Task

---

1

**META-ATTRIBUTE NAME..... TimeUnits**

CDIFMETAIDENTIFIER ..... PROJ0020

DESCRIPTION..... The units used to measure effort when estimates and actuals are recorded.

USAGE.....

ALIASES .....

CONSTRAINTS..... Cannot be changed once *WorkEstimates* have been allocated to *Tasks*.

DATATYPE ..... Enumerated

DOMAIN ..... Hours, Days, Weeks, Months

LENGTH.....

ISOPTIONAL..... False

2

3

---

---

---

### 1 **A4.3 Meta-relationship *Project.IsComposedOf.Task***

2 This meta-relationship is included as an example.

**NAME** ..... **IsComposedOf**  
**CDIFMETAIDENTIFIER** ..... PROJ0030  
**SUBTYPEOF** ..... RootEntity.IsRelatedTo.RootEntity  
**SUPERTYPEOF** .....  
**MINSOURCECARD** ..... 1  
**MAXSOURCECARD** ..... 1  
**MINDESTCARD** ..... 0  
**MAXDESTCARD** ..... N  
**DESCRIPTION** ..... Relates a project and its component tasks.  
**USAGE** .....  
**ALIASES** ..... ConsistsOf  
**CONSTRAINTS** .....

---

#### 5 INHERITED META-ATTRIBUTES

6	CDIFIdentifier	from RootObject
7	DateCreated	from RootObject
8	DateUpdated	from RootObject
9	TimeCreated	from RootObject
10	TimeUpdated	from RootObject

---

#### 13 LOCAL META-ATTRIBUTES

---



## APPENDIX B QUESTIONS AND ANSWERS<sup>10</sup>

### B1. Introduction

The following questions and answers are presented to help the reader's understanding of the standard. They have been derived from discussions held within the CDIF Technical Committee and during presentations to outside observers.

The questions are presented in the following sections:

- Extensibility
- Subject Areas
- Relationships
- Identifiers and Keys
- Other Points.

### B2. Extensibility

#### Question 1:

**In what ways may the CDIF Integrated Meta-model be extended?**

#### Answer:

A number of operations are available for defining extensions to the CDIF Integrated Meta-model. The additional objects must be defined in a new subject area. The following operations are supported by the extensibility mechanism:

- Add subject areas
- Add *CollectableMetaObject.IsUsedIn.SubjectArea*
- Add meta-entities
- Add meta-relationships
- Add attributable meta-object supertype associations
- Add meta-attributes to meta-entities or meta-relationships
- Add a new legal enumerated value to an enumerated meta-attribute

Further details of these operations can be found in Section 4, Meta-model Extensibility.

---

<sup>10</sup> This appendix is not a formal part of the attached EIA standard, but is included for purposes of information only.

1 It is not possible to remove things from a meta-model since existing importers that are relying on  
 2 the pre-defined CDIF Meta-model would fail. This also means that subtype/supertype links may  
 3 not be broken once established (i.e., one can't remove a supertype from a *MetaEntity*). It is also  
 4 not possible to modify the definition of a meta-attribute of a meta-entity or meta-relationship.  
 5 The only exception to this rule is that additional enumerated values for enumeration-typed meta-  
 6 attributes may be provided. It is possible to define a user-specific subject area that uses some of  
 7 the objects in the CDIF Integrated Meta-model; thus effectively excluding some of the things in  
 8 the standardized meta-model. However, it should be remembered that the use of non-standard  
 9 subject areas reduces the power of using CDIF since the importer must understand a new set of  
 10 definitions.

### 11 Question 2:

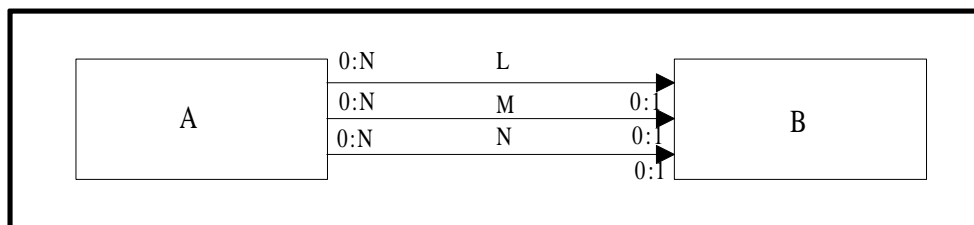
12 **Can an exporter use extensibility to extend the list of valid values for an enumerated-type**  
 13 **attribute?**

### 14 Answer:

15 All CDIF syntaxes provide a mechanism to extend the valid values for an enumerated type. For  
 16 details about the general mechanism, see *EIA/IS-108 CDIF - Transfer Format - General Rules*  
 17 *for Syntaxes and Encodings*. For an example of the specific mechanism used in the CDIF  
 18 Transfer Format, see *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1* and *EIA/IS-110*  
 19 *CDIF - Transfer Format - Encoding ENCODING.1*

### 20 Question 3:

21 **Given that all meta-relationships in the meta-model are subtyped (possibly indirectly) from**  
 22 ***RootEntity.IsRelatedTo.RootEntity*, how can a meta-relationship be added to an existing**  
 23 **meta-relationship to indicate some form of generalization using extensibility?**



24

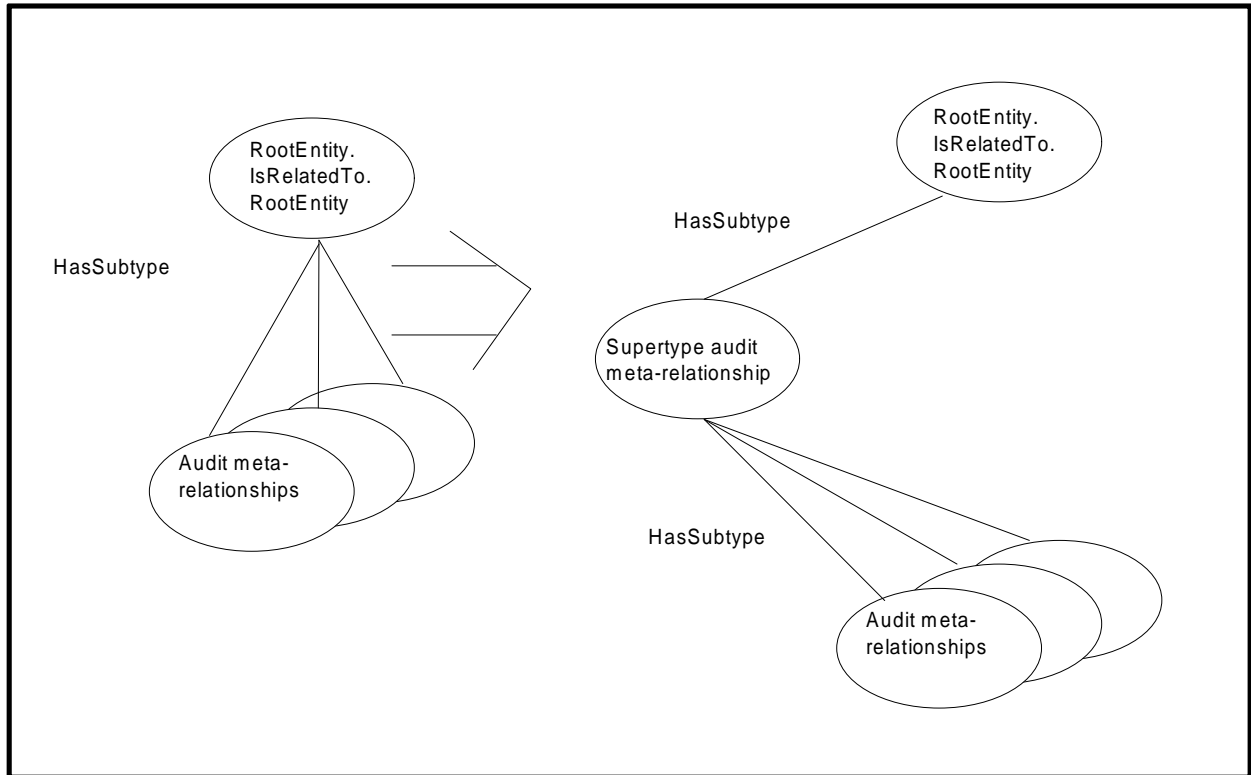
25

26

Figure B1  
Audit Meta-relationships

1 **Answer:**

2 Assume that the three meta-relationships between meta-entities **A** and **B** shown in Figure B1 are  
3 to be classified as audit meta-relationships and given a common meta-attribute. The additional  
4 supertype meta-relationship cannot be inserted into the existing meta-relationship hierarchy; a new  
5 leg in the inheritance hierarchy must be created using multiple inheritance as shown in Figure B2.



6

7

8

Figure B2  
Inserting a General Supertype

9 Where the requirement is to add the generalized meta-relationship, multiple inheritance must be  
10 used, since we can not change the existing meta-model by breaking the existing *HasSubtype* links.

11 If such an extension was made, an exporter should instantiate both the standard, specialised,  
12 meta-relationships and the extended ones in accordance with the Principle of Maximum  
13 Information Transfer. The importer may then ignore the standard data in an input file.

1 **Question 4:**

2 **If an extended meta-relationship is added to *SemanticObject*, is a subject area reference**  
3 **required to the Common subject area (*EIA/IS-112 CDIF - Integrated Meta-model -***  
4 ***Common Subject Area*) even though *SemanticObject* itself will never be instantiated in the**  
5 **Transfer?**

6 **Answer:**

7 When reusing an existing meta-object in extensibility, the transfer must contain a reference to a  
8 subject area that contains the full definition of the meta-object.

9 For the example given in the question, the extended meta-relationship must be explicitly used in a  
10 new subject area added for the purposes of defining the extensions. *SemanticObject*, as the  
11 source or destination of the meta-relationship, must also be explicitly used in the new subject area.  
12 A reference to a subject area that contains the full definition of *SemanticObject* is also required  
13 within the transfer.

14 **Question 5:**

15 **How can a meta-entity subtype be added?**

16 **Answer:**

17 Both subtype and supertype attributable meta-objects must first have been defined either in the  
18 CDIF Integrated Meta-model or using extensibility. An instance of the meta-meta-relationship  
19 called *AttributableMetaObject.HasSubtype.AttributableMetaObject* is then created to define the  
20 subtype association.

21 **B3. Subject Areas**

22 **Question 6:**

23 Why were subject areas formally introduced?

24 **Answer:**

25 As the CDIF Integrated Meta-model grows it is important to be able to partition to help control  
26 the development. By defining subject areas formally, the CDIF Technical Committee can develop  
27 and deliver single standards for each subject area.

28 Tools are unlikely to support the entire CDIF Integrated Meta-model and so by introducing  
29 subject areas, it will allow claims of compliance to be made against a single subject area. Tools  
30 that do not support the entire systems development life cycle will not be forced to consider these  
31 aspects of the underlying model.

1 **Question 7:**

2 **How can a meta-object be renamed for use in a new subject area?**

3 **Answer:**

4 Where an existing meta-object is useful to a new subject area but bears an inappropriate name, a  
5 new subtype should be created with the chosen name. In this way the properties and behavior of  
6 the original meta-object are maintained.

7 **Question 8:**

8 **CDIF mandates the use of ALL inherited meta-attributes since they are implicitly used.**  
9 **Why is there no way of excluding them?**

10 **Answer:**

11 Subtype meta-entities are created as a specialization of a supertype meta-entity. This means that  
12 the subtype meta-entity has all the properties of the supertype, and then additional properties  
13 (meta-attributes or meta-relationships) are defined. If a subtype entity, as used in a subject area,  
14 requires to exclude the use of inherited meta-attributes, it is questionable whether the meta-entity  
15 is really a subtype at all, and should probably be created as a separate meta-entity.

16 **Question 9:**

17 **If more than one subject area is included in a transfer, what does the Working Meta-model**  
18 **look like? Could it contain conflicting definitions?**

19 **Answer:**

20 Subject areas are views of the underlying CDIF Integrated Meta-model. The CDIF Integrated  
21 Meta-model is developed under a principle of upward compatibility and, therefore, where multiple  
22 subject areas are used, the Working Meta-model will contain those meta-objects from the  
23 integrated model used by each of the subject areas.

24 When developing a new subject area and reusing the existing definitions, or at any time, an error  
25 is found in the integrated model, all subject areas that use the meta-object will be re-issued. It will  
26 therefore be sensible to use the most current version of a subject area when configuring a transfer.

27 The CDIF Technical Committee may consider a major upgrade of the CDIF Integrated Meta-  
28 model at some point when all subject areas will be re-released to sort out any ambiguities that may  
29 arise.

1 **Question 10:**

2 **If a component of a standard subject area is deemed to be in conflict with the requirements**  
3 **of an exporter, how can it be overridden?**

4 **Answer:**

5 A new subject area will have to be created using only those standard objects from the underlying  
6 meta-model that are suitable to the exporter who can then extend the subject area as requires.  
7 However, exporters are deterred from doing this, unless absolutely necessary, because in doing so  
8 many of the benefits of using a standardized meta-model are lost.

9 **Question 11:**

10 **Why are nested subject areas not supported?**

11 **Answer:**

12 There is currently no functional requirement for nested subject areas. Where it may be considered  
13 useful to implementors to provide guide-lines about the use of a group of subject areas, a separate  
14 publication may be produced; for example, in the use of the presentation subject areas.

15 Functional requirements may be identified as a result of harmonization with other standards  
16 activities. In this case, the requirements may be incorporated structurally in the Meta-meta-  
17 model.

18 **Question 12:**

19 **In a CDIF Transfer, can a meta-entity be instantiated without instantiating the subject**  
20 **area reference for the subject area in which it is explicitly used?**

21 **Answer:**

22 In a transfer, the subject area reference for subject areas to be instantiated in the transfer must be  
23 stated. (Refer to the transfer format standards for full details.) Thereafter, instance of meta-  
24 entities that are explicitly used in that subject area may follow.

25 **Question 13:**

26 **Can implicitly inherited meta-attributes be instantiated in a CDIF Transfer without a**  
27 **subject area reference to the subject area where they are explicitly used?**

28 **Answer:**

29 Yes. Provided the subject area reference for the subject area where a meta-attribute is implicitly  
30 used (inherited) is given, the full definition for the meta-attribute as it is implicitly used can be  
31 determined.

1 **Question: 14:**

2 **When some of the meta-attributes of a meta-entity that is explicitly used in a subject area**  
3 **are not themselves explicitly used, which meta-attributes will be used in a subject area if**  
4 **subtypes for that meta-entity are explicitly used?**

5 **Answer:**

6 A subtype meta-entity that is explicitly used in a subject area will inherit all the meta-attributes of  
7 its supertype, regardless of whether they have been stated as being explicitly used locally for the  
8 supertype meta-entity.

9 **B4. Relationships**

10 **Question 15:**

11 **Why does CDIF support many-to-many relationships?**

12 **Answer:**

13 The modeling concepts (i.e., the CDIF Meta-meta-model) used to develop the CDIF Meta-model  
14 support many-to-many relationships because many-to-many relationships are very common in the  
15 CASE world. The absence of many-to-many relationships would increase the complexity of the  
16 CDIF Meta-model and also result in a loss of clarity.

17 **Question 16:**

18 **What happens when, in an extended meta-model, a third participant in a meta-relationship**  
19 **would cause a meta-relationship to "become" an associative meta-entity?**

20 **Answer:**

21 Figure B3 contains two diagrams that show the original state and the new state of a meta-model  
22 fragment. In the original fragment, the meta-relationship *AppearsOn* is used to show that a  
23 *ScreenField* appears on a *Screen*. Assume that the *AppearsOn* meta-relationship has a single  
24 meta-attribute called *Coords* that describes the location of the given field on the screen (this  
25 assumes that the field may be used on many different screens, each time occurring at a different  
26 location). Additionally, assume that these two meta-entities and the meta-relationship are part of  
27 the CDIF Integrated Meta-model (they aren't - they are hypothetical).

28 In the new fragment, CDIF's extensibility has been used to extend the CDIF Integrated Meta-  
29 model. The binary (two-way) meta-relationship is being "extended" into a three-way meta-  
30 relationship. This extension is being modeled using an associative meta-entity (i.e., a *MetaEntity*  
31 whose type is *Associative*). The third participant in the meta-relationship contains formatting and  
32 display information about the field's use on the screen and is known as *FormatInfo*.

33 **Question 17a:**

34 **How can an exporter override meta-relationships?**

1 **Answer:**

2 When exporting the new associative meta-entity and its associated formatting information, an  
 3 exporter must also export the original binary meta-relationship. If additional attributes are added  
 4 to the original binary relationship using extensibility, they must also be added to the new  
 5 associative entity.

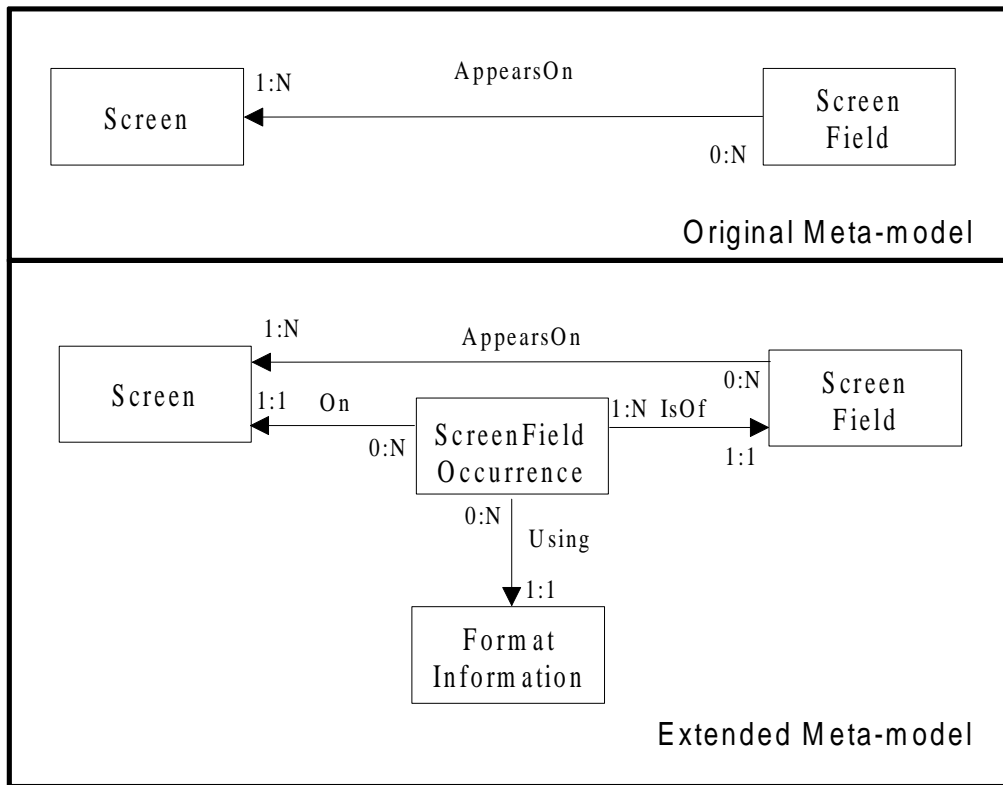
6 **Question 17b:**

7 **How should an importer process overridden meta-relationships?**

8 **Answer:**

9 If the importer doesn't understand the extension, the importer simply processes the original binary  
 10 meta-relationship and ignores the additional semantics associated with the extension. If it does  
 11 understand the extension, it may choose to ignore the original meta-relationship.

12 Figure B3 shows an example of an extended meta-model.



13

14

15

Figure B3  
 Adding a Third Participant to a Meta-relationship

1 **Question 18:**

2 **How does CDIF support n-ary meta-relationships?**

3 **Answer:**

4 N-ary meta-relationships can be represented in CDIF by replacing the n-ary meta-relationship with  
5 a meta-entity and binary meta-relationships in place of the roles associated with the n-ary meta-  
6 relationship.

7 **Question 19:**

8 **Why is minimum source cardinality and maximum source cardinality handled differently**  
9 **in the Data Modeling subject area of the CDIF Integrated Meta-model than in the Meta-**  
10 **meta-model?**

11 **Answer:**

12 The approach in the Data Modeling subject area of the CDIF Integrated Meta-model is more  
13 comprehensive than the modeling technique used to define the Integrated Meta-model and allows  
14 the expression of cardinality constraints such as "zero or 2, to many", which may be required  
15 when modeling end user applications. This flexibility is not required by the CDIF Meta-meta-  
16 model.

17 **B5. Identifiers and Keys**

18 **Question 20:**

19 **Does CDIF utilize keys in its meta-modeling?**

20 **Answer:**

21 The CDIF Meta-model does not support the definition of a "(prime) key" (i.e., a unique identifier)  
22 for each type of *MetaEntity*. This is because different tools have different requirements with  
23 regard to uniqueness and referential integrity. Keys are not relevant in the CDIF Meta-model  
24 since it merely represents the structure of a static transfer.

25 *CDIFIdentifiers* are provided, in place of unique keys defined from other meta-attributes, solely  
26 for the purposes of maintaining uniqueness of entries in the transfer for cross-referencing. All  
27 *MetaEntities* and *MetaRelationships* inherit the meta-attribute *CDIFIdentifier* as a result of being  
28 subtypes (directly or indirectly) of the foundation meta-entity called *RootObject*.

29 The CDIF Meta-model is not defined using foreign or candidate keys. These are outside the  
30 scope of CDIF.

1 **Question 21:**

2 **Why are there *CDIFIdentifiers* *CDIFIdentifier* for *Entities* and *Relationships*?**

3 **Answer:**

4 *CDIFIdentifiers* have been defined for the instances of meta-entities and meta-relationships to  
5 support cross referencing of instances in the Model Section of a transfer. For example, when an  
6 instance of a meta-relationship is created, it is necessary to define the source and destination  
7 instances of meta-entities for that meta-relationship instance. Note, however, that forward  
8 referencing is not allowed; therefore it is necessary to include the instances of source and  
9 destination meta-entities, prior to the instance of the meta-relationship.

10 The unique identifier for a meta-attribute is the *CDIFIdentifier* of the parent object concatenated  
11 with the name of the meta-attribute.

12 **Question 22:**

13 **Why are there *CDIFMetaIdentifiers* for *MetaEntities*, *MetaAttributes* and**  
14 ***MetaRelationships*?**

15 **Answer:**

16 *CDIFMetaIdentifiers* allow instances in the Presentation Subject Areas to cross-reference meta-  
17 objects in the Meta-model Section of a transfer.

18 **B6. Other Points**

19 **Question 23:**

20 **Could a CDIF transfer be used as an archiving mechanism?**

21 **Answer:**

22 Yes. Some customers may want to use CDIF as an archiving mechanism and therefore importers  
23 should be able to import 'previous/old' versions of CDIF.

1 **Question 24:**

2 **How should time attributes in multinational organizations be handled?**

3 **Answer:**

4 If time zones are significant to an organization, the user shall be responsible for ensuring a suitable  
5 approach is adopted when transferring data between tools.

6 **Question 25:**

7 **Why are Class Hierarchies for meta-relationships not supported?**

8 **Answer:**

9 Single inheritance of meta-relationships is required to allow us to inherit *CDIFIdentifier* for meta-  
10 relationships from *RootEntity.IsRelatedTo.RootEntity*.

11 Multiple inheritance is required to allow a supertype meta-relationship to be added without  
12 breaking the rules of extensibility that also apply when developing a new subject area.

13 The requirements of class hierarchies and multiple inheritance of meta-relationships are in conflict.  
14 The requirement for multiple inheritance is described above but no clear requirement for class  
15 hierarchies has been identified. If CDIF had implemented anything, it might have made it difficult  
16 to change the functionality should a clearer requirement become identified.

17 **Question 26:**

18 **Can supertype meta-entities be instantiated in CDIF?**

19 **Answer:**

20 Yes and No. In some cases a supertype meta-entity in the CDIF Integrated Meta-model may be  
21 instantiated where further specialization to one of the subtypes cannot be determined. There are  
22 some supertype meta-entities in the CDIF Integrated Meta-model where it is specified in the  
23 Usage section of the definition that the supertype should not be instantiated.

24 **Question 27:**

25 **Does the meta-relationship *AttributableMetaObject.HasSubtype.AttributableMetaObject***  
26 **imply that a meta-entity may be the supertype of a meta-relationship and vice versa?**

27 **Answer:**

28 Yes. At the top of the inheritance tree, *RootObject* (an *AttributableMetaObject*) is the supertype  
29 of *RootEntity* (a *MetaEntity*) and *RootEntity.IsRelatedTo.RootEntity* (a *MetaRelationship*).  
30 Subtype meta-entities of *RootEntity* may indeed have subtype meta-relationships. In CDIF meta-  
31 entities may also be subtypes of meta-relationships. As yet the CDIF Integrated meta-model has  
32 no instances of these kinds of subtypes.

1 **Question 28:**

2 **Can *MetaObjects* with different *CDIFMetaIdentifiers* have the same name, specifically**  
3 **meta-entities and subject areas?**

4 **Answer:**

5 Yes. Names are not unique across all *MetaObjects*. The rules for uniqueness are specified at the  
6 lowest level of subtype; namely, for subject areas, meta-entities, meta-relationships and meta-  
7 attributes. The rules are fully defined in Section 7.4.2,

1 Uniqueness of Names.

2 **Question 29:**

3 **Is subtyping of meta-relationships being used to support heterogeneous collections?**

4 **Answer:**

5 Yes. For example, in the Data Modeling subject area (*EIA/IS-114 CDIF - Integrated Meta-model*  
6 *- Data Modeling Subject Area*) *RoleConstraint.Incorporates.RoleConstraint* and  
7 *RoleConstraint.Incorporates.RolePlayer* are subtypes of  
8 *RoleConstraint.Incorporates.SemanticObject*. This structure is the mechanism used for providing  
9 heterogeneous *RoleConstraints* that contain both *RoleConstraints* and *RolePlayers*.

1  
2

This page is intentionally blank

## GLOSSARY<sup>1</sup>

<b>ARITY</b>	The arity (of a relationship) is the number of roles that participate in a relationship. A binary relationship has an arity of two. An n-ary relationship has an arity of n. (n>=2) Sometimes known as the "degree" of a relationship.
<b>ASSOCIATIVE ENTITY</b>	An entity used to represent a relationship between other entities. An Associative Entity is used when a relationship does not otherwise provide sufficient mechanisms.
<b>ATTRIBUTE</b>	A single-valued characteristic of an entity or relationship.
<b>ATTRIBUTED RELATIONSHIP</b>	A relationship that has attributes.
<b>CARDINALITY</b>	Describes the constraint on the number of entity instances that are related to the subject entity through a relationship. Cardinality is represented for each entity participating in a relationship by indicating the minimum and maximum number of its instances that may be associated with one particular instance of the related entity.
<b>CDIF</b>	See <i>CDIF Family of Standards</i> .

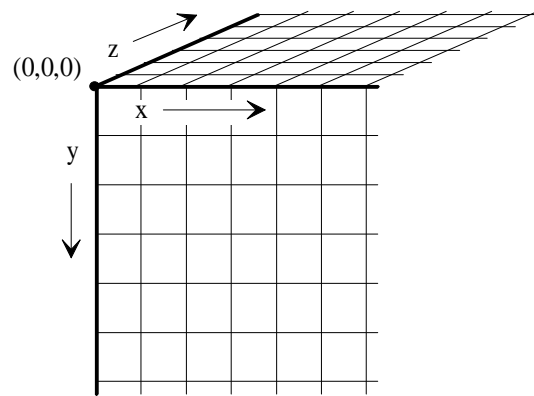
---

<sup>1</sup> This glossary is not a formal part of the attached EIA standard, but is included for information purposes only.

**CDIF COORDINATE FRAME**

The system of graphics addressing used by CDIF. Graphics space is defined as a coordinate system that is a three dimensional rectilinear grid. The intersections of grid lines define addresses (CDIF Points). The intersection of any two grid lines defines a coordinate plane. Each grid cell in a coordinate plane is a CDIF Pixel. All grid coordinates are integers and all grid lines are infinitely thin.

The figure below describes the CDIF Coordinate Frame.



**CDIF FAMILY OF STANDARDS**

The CDIF Family of Standards is composed of a set of standards that, when used together, provide a standard definition for the interchange of information between CASE tools.

**CDIF GRAPHICAL NOTATION**

The set of rules governing the projection of CDIF modeling concepts in diagrams.

**CDIF IDENTIFIER**

An attribute that uniquely identifies an object in the Model Section of a transfer.

**CDIF INTEGRATED META-MODEL**

The description of the set of concepts and notations used to define a model. The CDIF Integrated Meta-model defines an Entity-Relationship-Attribute model that is used to construct and define models used in systems development.

**CDIF METAIDENTIFIER**

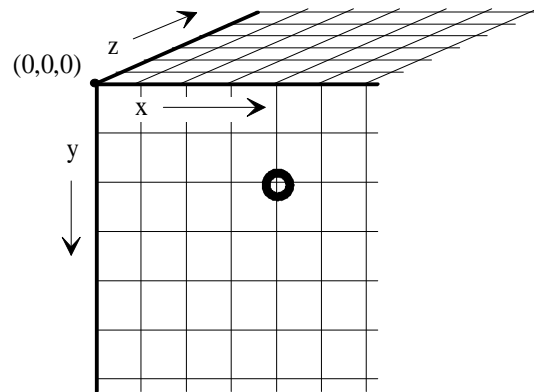
A meta-meta-attribute that uniquely identifies a meta-object in the Meta-model Section of a transfer.

**CDIF META-META-MODEL**

The description of the set of concepts and notations used to define a meta-model. Specifically, the CDIF Meta-meta-model defines an Entity-Relationship-Attribute model that is used to construct and define both meta-models and the CDIF Meta-meta-model itself.

**CDIF POINT**

The location of a point in 3-dimensional space as specified by the x,y,z coordinates. The circle in the figure below encloses a point located at (4,2,0).

**CDIF TRANSFER**

A combination of a particular syntax, a particular encoding of that syntax, and a meta-model. In other words, a complete definition of the format and contents of a transfer.

**CDIF TRANSFER FORMAT**

A combination of a particular syntax and a particular encoding of that syntax which together provides a complete definition of the transfer format.

**CHARACTERISTIC ENTITY**

A meta-entity that provides additional attribution for another meta-object. Other common names for Characteristic Entity are: Attributive Entity and Dependent Entity. Each instance of a characteristic meta-entity is logically only related to one instance of one other meta-object, therefore an importer could incorporate the meta-attributes of a characteristic meta-entity with those of the 'owning' meta-object, where the owning meta-object is the one to which the characteristic meta-entity is related with a cardinality of 1:1.

**CLEAR TEXT**

A form of encoding where the resulting physical file is human-readable.

**CONSTRAINT**

A constraint is generally a restriction on the value of an attribute or the existence of any object based on the value or existence of one or more others. In CDIF a constraint may be classed as computable (i.e., code may be built to implement the constraint) or non-computable (e.g., a constraint expressed in natural language). An example of a non-computable constraint -- "No two instances of *DFMProcess* may have the same value for the meta-attribute *Name*." An example of a computable *textual constraint* "*ForAny (DFMProcess,c) Unique (c.Name)*."

**COORDINATE PLANE**

See *CDIF Coordinate Frame*.

**DATA MODEL**

A model about data by which a reasonable interpretation of the data can be obtained. In the CASE industry a data model is one that may be encoded and manipulated by a computer.

**ENCODING**

An encoding defines how the elements of a syntax are physically represented. Details of representation of the various terminal symbols and data types in the syntax's grammar are provided.

<b>ENCODING.1</b>	The CDIF Family of Standards supports multiple transfer formats, each composed of a syntax and an encoding. ENCODING.1 is the primary encoding defined within the CDIF Family of Standards.
<b>ENTITY</b>	An object (i.e., thing, event or concept) that occurs in a model (i.e., transfer).
<b>INFORMATION CONTENT</b>	The Information Content is the set of meta-model and model instances found in a CDIF Transfer.
<b>INSTANCE</b>	<p>An individual occurrence of a type (e.g., the diagram <i>MyDiagram</i> is an instance of the <i>Diagram</i> type).</p> <p>Usually, an instance is an object which occurs one meta-level beneath its (type's) definition. This is not always true, however (e.g., within the CDIF presentation subject areas where icon type definitions and their instances are both contained within the same meta-level).</p>
<b>INTEGRATED META-MODEL</b>	See <i>CDIF Integrated Meta-model</i> .
<b>KERNEL ENTITY</b>	This classification is used for a meta-entity whose instances are meaningful without the occurrences of other meta-entities. For example, an instance of the meta-entity called <i>Attribute</i> , having a name, full description and brief description, is significant without the knowledge of the <i>DataObject</i> it describes.
<b>META</b>	<p>According to Webster's Ninth New Collegiate Dictionary: <i>more comprehensive: transcending - used with the name of a discipline to designate a new but related discipline designed to deal critically with the original one.</i></p> <p>Meta- is used in CDIF generally as a prefix to a concept to imply definition information about the concept. Specifically, used to designate the location of an object in the three model layers.</p>

<b>META-ATTRIBUTE</b>	A definition of a characteristic of a meta-entity or meta-relationship. Instances of a meta-attribute occur in a model as data values.
<b>META-ENTITY</b>	A definition of a type of data object that occurs in CDIF models. Specifically, a meta-entity represents a set of zero or more meta-attributes, stored together to represent a thing, event or concept that has instances in a model.
<b>META-LAYERS</b>	See <i>Model Layers</i> .
<b>META-META-ATTRIBUTE</b>	A definition of a characteristic of a meta-meta-entity or meta-meta-relationship. Instances of a meta-meta-attribute occur in a meta-model as meta-data values.
<b>META-META-ENTITY</b>	A definition of the behavior and structure of meta-entities, meta-relationships, meta-attributes, or subject areas (i.e., a definition of the meta-object definitions used to describe information in models).
<b>META-META-MODEL</b>	See <i>CDIF Meta-meta-model</i> .
<b>META-META-RELATIONSHIP</b>	A definition of a type of data object that occurs in CDIF meta-models. Specifically, a meta-meta-relationship represents the definition of a relationship between instances of meta-meta-entities.
<b>META-MODEL</b>	<p>A meta-model contains detailed definitions of the meta-entities, meta-relationships and meta-attributes whose instances represent an actual CDIF transfer.</p> <p>The CDIF Integrated Meta-model (as defined in the set of standards that comprise the CDIF Family of Standards) is a definition of all of the types of information that can be transferred in a CDIF Transfer without using the CDIF extensibility mechanism.</p>
<b>META-OBJECT</b>	A meta-object is a generic term for meta-entities and meta-relationships

---

<b>META-RELATIONSHIP</b>	A definition of a type of data object that occurs in CDIF models. Specifically, a meta-relationship represents the definition of a relationship between meta-entities that has instances in a model. A meta-relationship may also define a set of zero or more meta-attributes, stored together to represent characteristics of a relationship between meta-entities.
<b>MODEL</b>	<p>A specific collection of software engineering data. This is called a model because it usually represents a model of a software system under development.</p> <p>A collection of instances of meta-objects.</p>
<b>MODEL LAYERS</b>	<p>The different layers of definition (or abstraction) used in defining the CDIF Family of Standards. The four model layers in CDIF are: user data, model, meta-model, meta-meta-model.</p> <p>Any given model layer provides an accurate and complete definition of all the instances that may occur one layer below the given layer. For example, the meta-meta-model provides a set of definitions that are used to construct and understand the meta-model; the meta-model provides a set of definitions that are used to construct and understand a model.</p>
<b>N-ARY RELATIONSHIP</b>	A relationship with arity (degree) $n > 2$ . A relationship that has more than two participating entities. (Note that a single entity may participate several times in a single relationship.)
<b>RELATIONSHIP</b>	A real-world association among one or more entities. Where the association is between an entity and itself, the relationship is said to be recursive.
<b>ROLE</b>	The participation of an entity in a relationship. Each instance of a role has a minimum and maximum cardinality, direction, and may be attributed.

---

<b>SUBJECT AREA</b>	A related collection of meta-object instance definitions. Subject areas are used to define scoped areas of interest. Subject areas overlap to ensure the integration of the overall Meta-model, but a tool need only use those subject areas relevant to the data to be exported/imported.
<b>SYNTAX</b>	A syntax is a definition of the format of information in a CDIF transfer. The definition is in the form of a grammar. It is specified with regard to ordering and repetition, down to the level of terminals, but does not specify the representation of any of the terminal objects in the grammar. Those decisions are defined in the encoding of the syntax.
<b>SYNTAX.1</b>	The CDIF Family of Standards supports multiple transfer formats, each composed of a syntax and an encoding. SYNTAX.1 is the primary syntax defined within the CDIF Family of Standards.
<b>TRANSFER</b>	See <i>CDIF Transfer</i> .
<b>TRANSFER FORMAT</b>	See <i>CDIF Transfer Format</i> .
<b>WORKING META-MODEL</b>	The working meta-model is the definition of the specific meta-objects that may be instantiated in the model section of a CDIF Transfer. The working meta-model comprises the meta-objects in the CDIF Integrated Meta-model that are used by the subject areas referenced in the meta-model section of the transfer, and the meta-objects defined as extensions in the meta-model section.

## REFERENCES

- [1] Recommendation 460 of the Consultative Committee on International Radio (CCIR)

1  
2

This page is intentionally blank

1

## INDEX

2

### A

- 3 Aliases
- 4     IsAttributeOf MetaObject, 43
- 5 Arc, 14
- 6 Archiving mechanism, 82
- 7 Arity, 10
- 8     glossary entry, 87
- 9 Associative, 42
- 10 Associative Entity
- 11     glossary entry, 87
- 12 Associative meta-entity
- 13     adding participants to meta-relationships, 79
- 14 AttributableMetaObject, 26, 35
- 15     HasSubtype.AttributableMetaObject, 50
- 16     hierarchy, 25
- 17 AttributableMetaObject.HasSubtype.AttributableMetaObj
- 18     ect, 22, 50
- 19 Attribute
- 20     glossary entry, 87
- 21 Attributed meta-relationship, 6
- 22 Attributed relationship
- 23     glossary entry, 87

24

### B

- 25 Binary meta-relationships, 10
- 26 Bitmap, 19
- 27 Bold type, 4
- 28 Boolean, 19
- 29 Broken line, 12

30

### C

- 31 Cardinality
- 32     glossary entry, 87
- 33     handled in Meta-meta-model and Meta-model, 81
- 34     symbol, 12
- 35 CDIF
- 36     evaluation, 2
- 37     glossary entry, 87
- 38 CDIF Coordinate Frame
- 39     glossary entry, 88

- 40 CDIF Family of Standards, 1
- 41     glossary entry, 88
- 42 CDIF Graphical Notation, 10
- 43     glossary entry, 88
- 44 CDIF Identifier
- 45     glossary entry, 88
- 46 CDIF Integrated Meta-model, 15
- 47     glossary entry, 88
- 48 CDIF MetaIdentifier
- 49     glossary entry, 88
- 50 CDIF Meta-meta-model, 5, 26
- 51     glossary entry, 89
- 52     modeling approach, 5
- 53     overview, 25
- 54 CDIF Point
- 55     glossary entry, 89
- 56 CDIF Transfer
- 57     glossary entry, 89
- 58 CDIF Transfer Format
- 59     glossary entry, 89
- 60 CDIFIdentifier, 22, 82
- 61 CDIFMetaIdentifier, 26, 44, 82
- 62 Characteristic, 38, 42
- 63 Characteristic Entity
- 64     glossary entry, 90
- 65 Class hierarchies
- 66     meta-relationships, 83
- 67 Clear Text
- 68     glossary entry, 90
- 69 CollectableMetaObject, 26, 37
- 70     add, 22
- 71     IsUsedIn.SubjectArea, 51
- 72 CollectableMetaObject.IsUsedIn.SubjectArea, 51
- 73 Common supertype, 8
- 74 Constraint
- 75     glossary entry, 90
- 76     IsAttributeOf MetaObject, 44
- 77 Conventions, 4
- 78 Coordinate Frame, 20
- 79 Coordinate Plane
- 80     glossary entry, 90

81

### D

- 82 Data type, 18
- 83     bitmap, 19

1	Boolean, 19	48	questions and answers, 73
2	date, 19		
3	enumerated, 19		
4	float, 19	49	<b>F</b>
5	identifier, 19		
6	integer, 19	50	Float, 19
7	integer list, 19	51	Formal definitions, 32
8	point, 20		
9	point list, 20		
10	string, 20	52	<b>G</b>
11	text, 20		
12	time, 20	53	Graphical notation, 6, 10
13	Data-model	54	Graphical point, 19
14	glossary entry, 90		
15	DataType		
16	IsAttributeOf MetaAttribute, 39	55	<b>H</b>
17	Date, 19		
18	Defaults for data types, 18	56	Hierarchy, 13
19	Description		
20	IsAttributeOf MetaObject, 44		
21	Destination meta-entity, 11	57	<b>I</b>
22	Diagram		
23	subject areas, 17	58	Identifier, 19
24	Diagram Notation, 11	59	Implicit
25	Diagraming, 5	60	meta-object, 16
26	Document structure, 2	61	use of meta-objects, 15, 22
27	Domain	62	Importer, 23
28	IsAttributeOf MetaAttribute, 39	63	IMPORTER RESPONSIBILITIES, 24
29	<b>E</b>	64	Indented list, 25
		65	Information Content
		66	glossary entry, 91
		67	Information retention, 24
30	Encoding	68	Inheritance, 83
31	data types, 18	69	meta-meta-attributes, 32
32	glossary entry, 90	70	meta-meta-relationships, 32
33	ENCODING.1	71	Inherited meta-attribute, 8
34	glossary entry, 91	72	Inherited meta-meta-attribute, 28
35	Entity	73	Instance
36	glossary entry, 91	74	glossary entry, 91
37	Entity-Relationship-Attribute, 5	75	Integer, 19
38	Enumerated, 19	76	IntegerList, 19
39	Enumerated-type attribute, 74	77	Integrated Meta-model
40	Example, 65	78	glossary entry, 91
41	Explicit	79	IsOptional
42	meta-object, 16	80	IsAttributeOf MetaAttribute, 39
43	use of MetaAttribute, 79		
44	use of MetaEntity, 79		
45	use of meta-objects, 15, 22		
46	Exporter responsibilities, 23		
47	Extensibility, 21, 23		

1	<b>J</b>	39	MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaO
		40	bject, 52
		41	Meta-attributes, 23
2	<b>K</b>	42	Meta-entity, 5, 11
		43	explicit and implicit use, 15
3	Kernel, 42	44	glossary entry, 92
4	Kernel Entity	45	instantiation of supertypes, 83
5	glossary entry, 91	46	shaded, 11
6	Keys, 81	47	special notation, 11
		48	subject area scope, 16
		49	subtype symbol, 13
7	<b>L</b>	50	subtyping, 6
		51	supertype symbol, 13
8	Length	52	MetaEntity, 26, 27, 41
9	IsAttributeOf MetaAttribute, 40	53	add, 22
10	Local meta-attribute, 8	54	add supertype associations, 22
11	Local meta-meta-attribute, 28, 32	55	example, 65
		56	explicit, 79
		57	unique names, 41
12	<b>M</b>	58	Meta-layers
		59	glossary entry, 92
13	Many-to-many meta-relationships	60	Meta-meta-attribute
14	support of, 6	61	descriptions, 32
15	Many-to-many relationships	62	glossary entry, 92
16	support of, 79	63	inherited, 28, 32
17	MaxDestCard	64	local, 28, 32
18	IsAttributeOf MetaRelationship, 47	65	Meta-meta-entity
19	Maximum output, 23	66	definitions, 32
20	MaxSourceCard	67	glossary entry, 92
21	IsAttributeOf MetaRelationship, 47	68	hierarchy, 25
22	Meta-	69	subtypes, 25
23	glossary entry, 91	70	META-META-MODEL
24	MetaAttribute, 26, 38	71	components, 5
25	add, 22	72	CONCEPTS, 5
26	add legal enumerated value, 22	73	diagram, 28
27	explicit, 79	74	glossary entry, 92
28	inheritance, 77	75	Meta-meta-object
29	IsLocalMetaAttributeOf.AttributableMetaObject, 52	76	definitions, 32
30	unique names, 27, 38	77	name, 4
31	Meta-attribute, 5, 24	78	Meta-meta-relationship
32	data types, 18	79	definitions, 32
33	defaults, 24	80	glossary entry, 92
34	explicit and implicit use, 15	81	hierarchy, 25
35	glossary entry, 92	82	inherited, 32
36	inherited, 8	83	subtypes, 25
37	local, 8	84	Meta-model
38	subject area scope, 16	85	components, 5
		86	extensibility, 21
		87	glossary entry, 92
		88	MetaObject, 26, 43

1	unique names, 84	48	<b>N</b>
2	Meta-object, 15, 77		
3	glossary entry, 92		
4	implicit use, 15	49	Name, 84
5	name, 4	50	identifier, 19
6	reuse in extensions, 76	51	IsAttributeOf MetaObject, 45
7	Meta-relationship	52	meta-meta-object, 4
8	subtyping, 9	53	meta-object, 4
9	Meta-relationship, 5, 11	54	uniqueness, 27
10	arc, 14	55	Names
11	broken line, 12	56	meta- prefix, 5
12	cardinality, 12	57	Naming
13	explicit and implicit use, 15	58	meta-objects, 5
14	glossary entry, 93	59	Naming conventions, 6
15	mutual exclusivity, 14	60	N-ary
16	optionality, 12	61	meta-relationship, 79
17	recursive, 12	62	meta-relationships, 81
18	shaded, 12	63	N-ary relationship
19	special notation, 12	64	glossary entry, 93
20	subject area scope, 16	65	Nested subject areas, 78
21	<i>MetaRelationship</i> , 11, 26, 46	66	Notation, 10
22	add, 22		
23	add supertype associations, 22	67	<b>O</b>
24	example, 65		
25	HasDestination.MetaEntity, 53	68	Optional data types, 18
26	HasSource.MetaEntity, 54	69	Optionality
27	unique names, 27, 46	70	meta-relationship, 12
28	Meta-relationship Subtyping, 9		
29	MetaRelationship.HasDestination.MetaEntity, 53		
30	MetaRelationship.HasSource.MetaEntity, 54	71	<b>P</b>
31	MinDestCard		
32	IsAttributeOf MetaRelationship, 48	72	Point, 20
33	MinSourceCard	73	PointList, 20
34	IsAttributeOf MetaRelationship, 48	74	Prerequisites, 2
35	Model	75	Project Specification, 65
36	components, 5	76	Property, 38
37	glossary entry, 93		
38	Model layers		
39	glossary entry, 93	77	<b>Q</b>
40	Modeling Approach, 5		
41	Modeling rules, 6	78	Quotes, 4
42	Multiple inheritance		
43	meta-entities, 7		
44	meta-relationship, 75	79	<b>R</b>
45	Multiple subject areas, 77		
46	Mutual exclusive meta-relationships, 14	80	Recursive meta-relationship, 12
47	Mutual exclusivity of meta-relationships, 10	81	Reflexive meta-relationship, 12
		82	Relationship
		83	glossary entry, 93

1	Role	40	<b>T</b>
2	glossary entry, 93		
3	RootObject, 22		
4	and identifiers, 81		
5	<b>S</b>		
6	Shaded		
7	meta-relationship, 12		
8	Shaded Box, 11		
9	Shaded meta-entities, 11		
10	Source meta-entity, 11		
11	Specialization, 6		
12	String, 20		
13	Subject area, 15, 26, 65, 76		
14	add, 21		
15	diagrams, 17		
16	extensibility, 16		
17	glossary entry, 94		
18	multiple, 77		
19	nesting, 78		
20	reference, 78		
21	rules, 16		
22	unique names, 27, 49		
23	SubjectArea, 49		
24	CollectableMetaObject		
25	IsUsedIn, 51		
26	Subtype, 13		
27	Subtyping		
28	meta-meta-entity hierarchy, 25		
29	meta-meta-relationship hierarchy, 25		
30	Subtyping of meta-entities		
31	used in CDIF, 6		
32	Subtyping of meta-relationships		
33	used in CDIF, 9		
34	Syntax		
35	data types, 18		
36	glossary entry, 94		
37	SYNTAX.1	61	<b>Z</b>
38	glossary entry, 94		
39	Systems Development Life Cycle, 15		
		41	Text, 20
		42	Time, 20, 83
		43	Transfer
		44	glossary entry, 94
		45	Transfer Format
		46	glossary entry, 94
		47	Type
		48	IsAttributeOf MetaEntity, 42
		49	<b>U</b>
		50	Usage
		51	IsAttributeOf MetaObject, 45
		52	<b>V</b>
		53	VersionNumber
		54	IsAttributeOf SubjectArea, 49
		55	<b>W</b>
		56	Working meta-model, 21, 24, 77
		57	glossary entry, 94
		58	unique names, 27
		59	<b>X</b>
		60	<b>Y</b>

1

This page is intentionally blank

## **RELATED EIA STANDARDS**

*EIA/IS-106 CDIF - CASE Data Interchange Format - Overview*

*EIA/IS-107 CDIF - Framework for Modeling and Extensibility*

*EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*

*EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*

*EIA/IS-110 CDIF - Transfer Format - Encoding ENCODING.1*

*EIA/IS-111 CDIF - Integrated Meta-model - Foundation Subject Area*