



TECHNICAL REPORT

EIA-IS-118™

Issued 1996-12
Stabilized 2015-07

CDIF Integrated Meta-model Presentation Location and Connectivity Subject Area

NOTICE

This document has been taken directly from the original TechAmerica document and contains only minor editorial and format changes required to bring it into conformance with the publishing requirements of SAE Technical Standards. The release of this document is intended to replace the original with the SAE International document. Any numbers established by the original document remain unchanged.

The original document was adopted as an SAE publication under the provisions of the SAE Technical Standards Board (TSB) Rules and Regulations (TSB 001) pertaining to accelerated adoption of specifications and standards. TSB rules provide for (a) the publication of portions of unrevised specifications and standards without consensus voting at the SAE committee level, and (b) the use of the existing specification or standard format.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2016 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE WEB ADDRESS:

Full document details: SAE EIAIS118

SAE values your input. To provide feedback on this Technical Report, please visit <http://www.sae.org/technical/standards/EIAIS118>

Documents

Document Sharing Hub

Interim Standard

EIA/IS-118

CDIF

CASE Data Interchange Format
A Division of Electronic Industries Association

**CDIF Integrated Meta-model
Presentation Location and
Connectivity Subject Area**

December 1996
Electronic Industries Association
CDIF Technical Committee



Documents
Document Sharing Hub

NOTICE

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

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

EIA INTERIM STANDARDS

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

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

This EIA standard is considered to have international standardization implication.

COPYRIGHT 1996

Published by
ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2500 Wilson Blvd.
Arlington, VA 22201

PRICE: Please refer to the current
Catalog of EIA & JEDEC STANDARDS & ENGINEERING PUBLICATIONS
or contact EIA Electronic Information Group
Phone: (703) 907-2545, FAX: (703) 907-7501

All rights reserved
Printed in U.S.A.

ISBN 0-7908-0024

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.

***ELA/IS-106 CDIF - CASE Data Interchange Format - Overview and ELA/IS-107 CDIF - Framework for Modeling and Extensibility* should be read first when initially exploring CDIF.** The first explains the overall CDIF Architecture and how the family of standards fits together. The second explains the scope, and modeling approach in CDIF. The CDIF Meta-meta-model and extensibility mechanism are also defined in that document.

This standard, *ELA/IS-118 CDIF - Integrated Meta-model - Presentation Location and Connectivity Subject Area*, explains one of the subject areas of the CDIF Integrated Meta-model, which is used to ensure that the information held by tools communicating using CDIF is transferred with an agreed meaning. It covers the information required to express presentation concepts such as nodes, edges, and annotations.

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 was formulated under the cognizance of the CDIF Technical Committee.

CONTENTS

1.	SCOPE	1
2.	INTRODUCTION	2
2.1	Audience.....	2
2.2	Structure of this Document	2
2.3	Conventions	4
3.	STRUCTURE OF SUBJECT AREA DEFINITION.....	5
3.1	Introduction.....	5
3.2	Structure of Subject Area Overview	5
3.2.1	Summary	5
3.2.2	Description of Major Concepts	5
3.2.3	Diagram(s)	5
3.2.4	AttributableMetaObject Hierarchy	8
3.2.5	MetaEntity and MetaRelationship Summaries	8
3.3	Structure of Subject Area Detailed Definitions.....	9
3.3.1	Summary	9
3.3.2	Subject Area Definition	9
3.3.3	MetaEntity Definitions.....	10
3.3.4	MetaRelationship Definitions	12
3.4	Structure of Referenced Meta-object Definitions	14
4.	SUBJECT AREA OVERVIEW	15
4.1	Description.....	15
4.1.1	Introduction.....	15
4.1.2	Transfer Coordinate Frame	17
4.1.3	Diagrams.....	17
4.1.4	Points	18
4.1.5	PositionedElements, Center and Extent.....	22
4.1.6	Nodes	22
4.1.7	Annotations	23
4.1.8	Edges and EdgeElements.....	24
4.1.9	Connection of Presentation and Semantic Information	25
4.1.9.1	Presentation information for meta-entity instances	26
4.1.9.2	Presentation information for meta-relationship instances.....	27
4.1.9.3	Presentation information for values of meta-attributes.....	28
4.1.10	Computable Languages.....	29
4.2	Diagrams.....	30
4.3.	AttributableMetaObject Hierarchy	31
4.4.	MetaEntity Summary	32
4.5.	MetaRelationship Summary	36

5.	SUBJECT AREA DETAILED DEFINITIONS	38
5.1	Introduction.....	38
5.2	SubjectArea Definition	39
5.3	MetaEntity Definitions	40
5.3.1	AbsolutePoint.....	40
5.3.2	Annotation.....	42
5.3.3	AnnotationArgument	45
5.3.4	Diagram.....	48
5.3.5	Edge	52
5.3.6	EdgeElement	54
5.3.7	GraphicalElement	55
5.3.8	Node.....	56
5.3.9	Point	58
5.3.10	PositionedElement	60
5.3.11	PresentationInformationObject	63
5.3.12	RelativePoint.....	65
5.3.13	SemanticObjectReference.....	71
5.4	MetaRelationship Definitions	73
5.4.1	Annotation.Uses.AnnotationArgument.....	73
5.4.2	Edge.ConsistsOf.EdgeElement	75
5.4.3	Edge.IsAttachedTo.GraphicalElement	77
5.4.4	EdgeElement.HasEnd.Point.....	78
5.4.5	GraphicalElement.AppearsOn.Diagram	79
5.4.6	Point.IsDependentUpon.PositionedElement.....	80
5.4.7	Point.IsLocatedOn.Diagram	81
5.4.8	PositionedElement.HasCenter.Point	82
5.4.9	PresentationInformationObject.Represents.SemanticObjectReference.....	83
5.4.10	RelativePoint.IsRelativeTo.Point.....	84
6.	REFERENCED OBJECT DEFINITIONS	85
6.1	AttributableMetaObject Definitions	85
6.1.1	RootObject.....	85
6.2	MetaEntity Definitions	89
6.2.1	RootEntity	89
6.3	MetaRelationship Definitions	91
6.3.1	RootEntity.IsRelatedTo.RootEntity	91
APPENDIX A	EXAMPLE.....	93
A1.	Introduction.....	93
A2.	Details of Example	93
APPENDIX B	QUESTIONS AND ANSWERS	103

GLOSSARY	106
INDEX.....	112
RELATED STANDARDS.....	INSIDE BACK COVER

FIGURES

Figure 1	Position in the CDIF Family of Standards.....	1
Figure 2	Conventions for Representing Meta-entities	6
Figure 3	Conventions for Representing Meta-relationships	6
Figure 4	Conceptual Diagram and an Instance Diagram	7
Figure 5	Connectivity information in a transfer that does not have Shape information	16
Figure 6	CDIF Coordinate Frame	17
Figure 7	CDIF Points and CDIF Pixels.....	18
Figure 8	A transformed coordinate system	20
Figure 9	A Node in a transformed coordinate system.....	21
Figure 10	Center and Extent	22
Figure 11	Example of an Annotation	23
Figure 12	Instance diagram for the Annotation example of Figure 11	24
Figure 13	Points, Edges and EdgeElements.....	25
Figure 14	Presentation for a DFMPProcess	26
Figure 15	Representation of both presentation and semantic information for a DFMPProcess	26
Figure 16	Presentation of a SubtypeSet in an inheritance situation.....	27
Figure 17	Representation of both semantics and presentation for a SubtypeSet	27
Figure 18	Presentation of a label denoting the name of an Entity.	28
Figure 19	Representing both semantic and presentation of a name of an Entity	28
Figure 20	Presentation Location and Connectivity Subject Area Overview.....	30
Figure 21	Node and Annotation Sub-model.....	30
Figure A1	Presentation Location and Connectivity Example.....	94

TABLES

Table 1	Contents of Subject Area Definition.....	9
Table 2	Contents of Meta-entity Definition	11
Table 3	Contents of Local Meta-attribute Definition.....	12
Table 4	Contents of Meta-relationship definition	13

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 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.

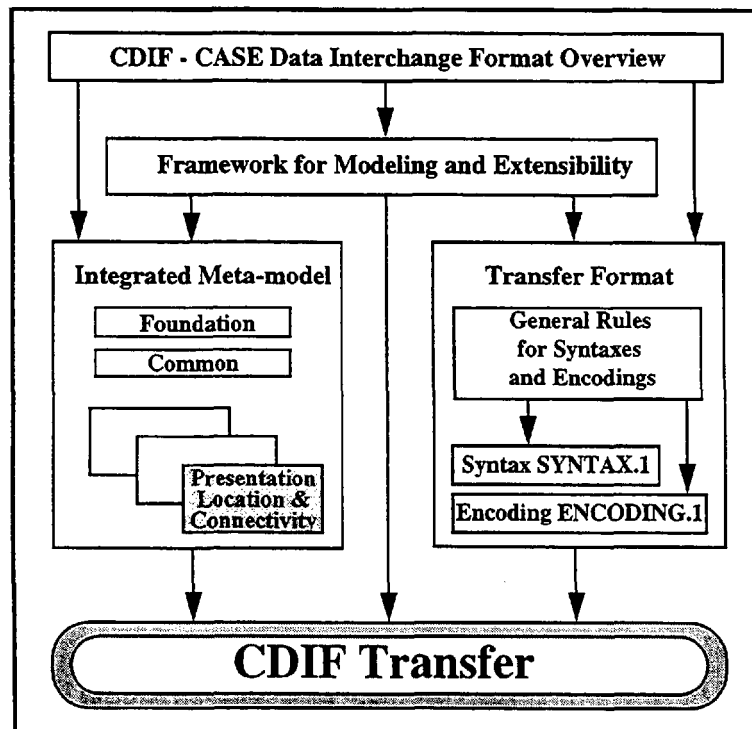


Figure 1
Position in the CDIF Family of Standards

The diagram in Figure 1 depicts the various standards that comprise the CDIF Family of Standards. The shaded box depicts this Standard and its position in the CDIF Family of Standards.

This standard, *EIA/IS-118 CDIF - Integrated Meta-model - Presentation Location and Connectivity Subject Area*, defines the portion of the CDIF Integrated Meta-model that supports presentation concepts such as nodes, edges, and annotations.

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 a single subject area of the CDIF Integrated Meta-model. 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 the 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 'Structure of Subject Area Definition'

This describes in detail the structure of the subsequent sections of this standard including the notation and layout of the diagrams and the detailed definitions.

Section 4 'Subject Area Overview'

This gives an overview of the coverage of this subject area.

Section 5 'Subject Area Detailed Definitions'

This gives the detailed definitions of all the objects used in the subject area.

Section 6 'Referenced Meta-object Definitions'

This gives the detailed definitions of objects that are not used in the subject area defined in this Standard, but are referenced by objects that are. These definitions are included to avoid the need to reference multiple standards to obtain the complete definition of the current subject area.

Appendix A 'Examples'

This gives one or more examples of how information is represented using this CDIF Subject Area. Where there are different technique variants used within the coverage of a single Subject Area, the examples attempt to show how some of these different variants are represented in the CDIF Integrated Meta-model.

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.

Index

A comprehensive index is provided.

References

The References section contains references to other standards and documents used within this Standard.

2.3 Conventions

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

- Double quotes are used to introduce new terms (e.g., "model layer")
- Bold type is used for emphasis (e.g., this is **important**)
- All meta-objects and meta-meta-objects in CDIF (in meta-models and meta-meta-models) are named by concatenating all the words that name the meta-object or meta-meta-object; the first letter of each word is upper-case, the rest are lower-case. These names are shown in italic (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*)
- The names of objects and values for meta-attributes that appear in examples and instance diagrams are shown in bold italic (e.g., ***Customer***, ***True***, ***100***)

Full details of the CDIF Graphical Notation used in the Meta-model and Meta-meta-model can be found in *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*. A summary of the major concepts is given in Section 3.2.3, Diagram(s).

3. STRUCTURE OF SUBJECT AREA DEFINITION

3.1 Introduction

The following subsections describe how the subject area is defined in the remainder of this document. The subject area is first introduced in the Subject Area Overview Section, and the detailed definitions of the objects contained in the subject area are given in the Subject Area Definitions Section.

3.2 Structure of Subject Area Overview

3.2.1 Summary

The Subject Area Overview Section consists of the following sub-sections:

- Description of Major Concepts
- Diagram(s)
- AttributableMetaObject Hierarchy
- MetaEntity Summary
- MetaRelationship Summary

3.2.2 Description of Major Concepts

This provides a textual description of the major concepts in the subject area. It explains the coverage of this subject area and outlines what it is capable of representing.

3.2.3 Diagram(s)

Included in the Subject Area Overview are various diagrams. These diagrams are intended to graphically present an overview of the meta-entities and meta-relationships used in this subject area. All CDIF subject area diagrams use the following diagramming notation conventions.

Meta-entities used in the subject area are shown as rectangular boxes, with the name of the meta-entity on a white background. An example is the meta-entity *A* in Figure 2. An additional special notation may be used to represent meta-entities on diagrams of meta-models. If the meta-entity is not used in the subject area, but is used in another subject area and is included for clarity, the box shall be shaded. An example is the meta-entity *B* in Figure 2.

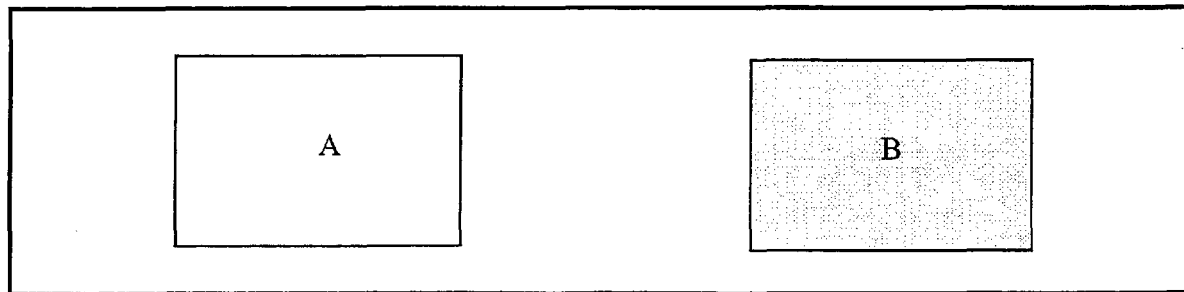


Figure 2
Conventions for Representing Meta-entities

Meta-relationships are normally represented by a line joining the source and destination entities, with the name of the meta-relationship given next to the line. An arrow on the line indicates the direction in which to read the name. The meta-relationship *C* in Figure 3 is an example of this. Additionally, two special notations may be used to represent meta-relationships on diagrams of meta-models.

If a meta-relationship is included in the diagram for clarification, but it is not used in the subject area, the text of the meta-relationship name shall be back-shaded. This is shown for the meta-relationship *D* in Figure 3.

If a supertype meta-entity that participates in a meta-relationship is not included in the diagram, the inherited meta-relationship shall be shown on the diagram as a broken line to/from the inheriting subtype meta-entity. This is shown for the meta-relationship *E* in Figure 3.

Subtype meta-relationships are shown with the meta-relationship name in italics. This is shown for the meta-relationship *F* in Figure 3.

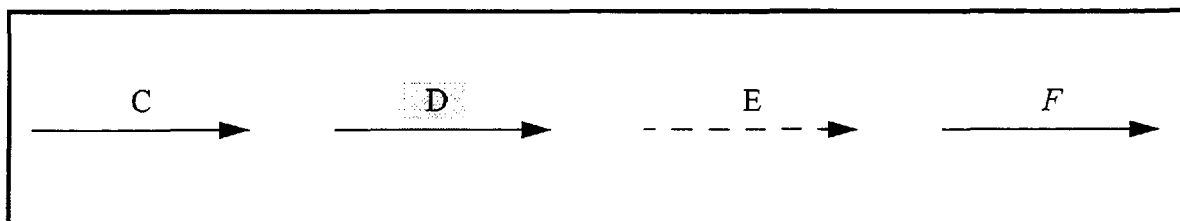


Figure 3
Conventions for Representing Meta-relationships

Full details of the CDIF Graphical Notation used in the Meta-model and the Meta-meta-model can be found in *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*.

To aid in the understanding of some of the concepts modeled in this subject area, instance diagrams have been used. These diagrams differ from those described above in that an instance diagram depicts one or more occurrences or instances of meta-entities and meta-relationships depicted in the diagrams previously described. Where one of the diagrams previously described may show that the meta-model includes an *Automobile* meta-entity, an instance diagram might depict *Honda*, *Ford*, *Mercedes*, and *Fiat* as instances of *Automobile*.

In an instance diagram, instances of meta-entities are depicted as boxes with a horizontal divider. Above the divider is the name of the meta-entity that is instantiated. Below the divider may be a name or a value assigned to a significant meta-attribute of that meta-entity. Where the name is inherited, inferred, or otherwise not significant to the example (e.g., some associative meta-entity instances), no name or value may be present.

Meta-relationships are depicted as arrows, just as in the previous diagrams described. They do not, however, have cardinality or exclusivity indicators, since none are necessary.

Meta-attributes of both meta-entities and meta-relationships are represented as:

[*meta-attribute name* :] *value*

where the name of the meta-attribute may be omitted depending on the ease of understanding the example with or without the name (e.g., *Name* might be omitted, *MinLength* might not).

When an instance of a meta-entity or meta-relationship not in this Subject Area is included for clarity, the box/arrow shall be shaded just as in the E-R Diagrams.

Figure 4 shows both the conceptual diagram and an instance diagram.

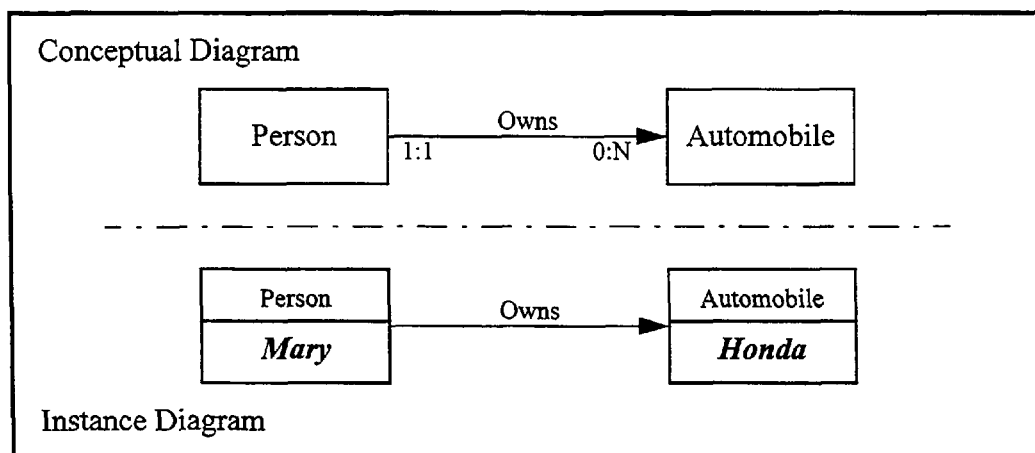


Figure 4
Conceptual Diagram and an Instance Diagram

3.2.4 AttributableMetaObject Hierarchy

The AttributableMetaObject Hierarchy in the Subject Area Overview section contains all the meta-entities and meta-relationships used in the subject area. It also identifies all supertypes required to relate the meta-objects used back to *RootObject*. Other subtypes of objects shown in the hierarchy, where neither they nor any of their subtypes are used in this subject area, are not shown.

The AttributableMetaObject Hierarchy is presented as an indented list of meta-object names. The indentation indicates the subtyping; all names indented below a given name are subtypes of that meta-object. For example, if *Car* and *Truck* are both subtypes of *Vehicle*, this would be shown as follows:

```
Vehicle
  Car
  Truck
```

A meta-entity or meta-relationship and all its subtypes can be duplicated in the hierarchy due to multiple inheritance. Where this occurs, the duplicate entries are shown in italics.

The overall AttributableMetaObject supertype, *RootObject*, defined in *EIA/IS-111 CDIF - Integrated Meta-model - Foundation Subject Area*, is shown at the top of the hierarchy.

Where a page break occurs in the list, all the supertypes of the first meta-entity or meta-relationship on the new page are repeated for clarity with a shaded background.

3.2.5 MetaEntity and MetaRelationship Summaries

Summaries are given of all the meta-entities and meta-relationships used in the Subject Area. The meta-entity definitions are given first, in alphabetical order, followed by the meta-relationship definitions, also in alphabetical order.

For each meta-entity used in the subject area, all the meta-attributes, both inherited and local, are listed. The inherited meta-attributes are shown first, in italics, sequenced alphabetically within parent meta-entity. The local meta-attributes are then listed alphabetically in normal font. The optionality, whether mandatory or optional, of each meta-attribute is given.

For each meta-relationship all the meta-attributes are listed sequenced alphabetically. Inherited and local meta-attributes are shown in the same way as for meta-entities. The optionality of each meta-attribute is given.

3.3 Structure of Subject Area Detailed Definitions

3.3.1 Summary

The Subject Area Detailed Definitions section consists of the following sub-sections:

- Subject Area Definition
- MetaEntity Definitions
- MetaRelationship Definitions

3.3.2 Subject Area Definition

This is the formal definition of the instance of the meta-meta-entity called SubjectArea (SA). It contains the information listed in Table 1.

Table 1
Contents of Subject Area Definition

5.2	SubjectArea Definition
NAME.....	subject area name
CDIFMETAIDENTIFIER	
DESCRIPTION.....	
USAGE.....	
ALIASES	
CONSTRAINTS	
VERSIONNUMBER	

3.3.3 MetaEntity Definitions

The definition of each meta-entity used in the subject area is structured according to the classes of information listed in Table 2.

For each meta-entity:

- Inherited meta-attributes are listed by supertype meta-entity in a top-down hierarchical order. Within each sub-list, meta-attributes are listed alphabetically.
- Inherited meta-relationships are grouped by source meta-entity (listed in alphabetical order). Within each group, the inherited meta-relationships are sorted alphabetically by "verb" name and then alphabetically by destination meta-entity name. Inherited abstract meta-relationships are not listed. Local meta-attributes are listed alphabetically.
- Detailed local meta-attribute descriptions are listed alphabetically. These definitions are structured according to the classes of information listed in Table 3.

Table 2
Contents of Meta-entity Definition

5.N.N	Meta-entity Name
NAME	meta-entity name
CDIFMETAIDENTIFIER.....	
SUBTYPEOF	list of direct supertypes of the meta-entity ... meta-entity [not] in this SA
SUPERTYPEOF	list of direct subtypes of the meta-entity within this subject area ... meta-entity
DESCRIPTION	
USAGE	
ALIASES.....	
CONSTRAINTS	
TYPE	
INHERITED META-ATTRIBUTES	
List of meta-attributes inherited from all supertypes of the meta-entity, with the name of the meta-object for which the meta-attribute is defined...	
	meta-attribute name from meta-entity name [not] in this SA
LOCAL META-ATTRIBUTES	
List of meta-attributes defined for the meta-entity...	
	meta-attribute name
INHERITED META-RELATIONSHIPS	
List of instantiable meta-relationships inherited from all supertypes of the meta- entity...	
	meta-relationship name [not] in this SA
LOCAL META-RELATIONSHIPS	
List of meta-relationships defined for the meta-entity...	
	meta-relationship name
Full definitions of all local meta-attributes appear at this point. See Table 3 for format of local meta-attribute definitions.	

Table 3
Contents of Local Meta-attribute Definition

META-ATTRIBUTE NAME meta-attribute name
CDIFMETAIDENTIFIER.....
DESCRIPTION
USAGE
ALIASES.....
CONSTRAINTS
DATATYPE.....
DOMAIN.....
LENGTH
ISOPTIONAL.....

3.3.4 MetaRelationship Definitions

The definition of each meta-relationship used in the subject area is structured according to the classes of information listed in Table 4.

For each meta-relationship:

- inherited meta-attributes are listed by supertype meta-relationship in a top-down hierarchical order. Within each sub-list, meta-attributes are listed alphabetically.
- Local meta-attributes are listed alphabetically.
- Detailed local meta-attribute descriptions are listed alphabetically.

Table 4
Contents of Meta-relationship definition

5.N.N	Full Meta-relationship Name
NAME	meta-relationship name
CDIFMETAIDENTIFIER	
SUBTYPEOF	list of direct supertypes of the meta-relationship meta-entity [not] in this SA
SUPERTYPEOF	list of direct subtypes of the meta-relationship within this subject area ... meta-entity
MINSOURCECARD	Minimum source meta-entity cardinality
MAXSOURCECARD	Maximum source meta-entity cardinality
MINDESTCARD	Minimum destination meta-entity cardinality
MAXDESTCARD	Maximum destination meta-entity cardinality
DESCRIPTION	
USAGE	
ALIASES	
CONSTRAINTS	
INHERITED META-ATTRIBUTES	
	List of meta-attributes inherited from all supertypes of the meta-relationship, with the name of the meta-object for which the meta-attribute is defined...
	meta-attribute name from meta-entity name [not] in this SA
LOCAL META-ATTRIBUTES	
	List of meta-attributes defined for the meta-relationship...
	meta-attribute name
Full definitions of all local meta-attributes appear at this point. See Table 3 for format of local meta-attribute definitions.	

3.4 Structure of Referenced Meta-object Definitions

This section contains the formal definitions of meta-objects referenced from the definitions of meta-objects used in the subject area.

These will include the *AttributableMetaObject RootObject*, since this is the root of the *AttributableMetaObject* Hierarchy, and it may also include other meta-entities and meta-relationships.

The definitions are given in the following order: *AttributableMetaObject*, meta-entities, then meta-relationships. Within each section, they are listed in alphabetical order. The definition layout for the *AttributableMetaObject* is the same as that of the meta-entity; both meta-entity and meta-relationship definitions follow the same layout as that defined for the Subject Area Detailed Definitions section.

4. SUBJECT AREA OVERVIEW

4.1 Description

4.1.1 Introduction

The Presentation Location and Connectivity Subject Area supports the representation of annotated node-edge diagrams found in modeling tools. This subject area defines presentation information only; if semantic information is to be represented, one or more of those subject areas within the CDIF Integrated Meta-model that define semantic content shall be used.

The information defined by this subject area is divided into location and connectivity information. Location information describes relative or absolute positions for a graphical object, e.g. a line's end point is at coordinates (5,4,3). Connectivity information identifies where graphical objects are connected to each other. An example of connectivity information is that a line connects two objects, rather than just being a line with a location. In a CDIF transfer, the line might represent a *Relationship* and the two objects might represent *Entities*. Thus, connectivity describes the presentation information that the two *Entities* are connected by a *Relationship*.

Other graphical information such as shape, or line style and font, is not supported by this subject area.

Both location and connectivity information are required to faithfully reproduce the topology of a diagram when transferring presentation information of a model. The location information provides the placement of all graphic objects. Connectivity information permits the importing tool to move objects or change their shape and still maintain the graphic representation of the connectivity.

The effect of using this subject area is illustrated in the following example (see Figure 5).

Assume modeling tool A uses squares to represent *Processes*, and tool B uses circles to represent *Processes*, and tool B is the receiver in a CDIF transfer that only uses the Presentation Location and Connectivity Subject Area to represent presentation information. After importing by tool B, *Processes* are represented as circles with the relative placement of the objects maintained.

Figure 5 shows an example diagram in tool A on the left hand side. If only location information and no connectivity information is exported by tool A, tool B will likely preserve the location information for the start and end points of the connecting line, and create the diagram shown on the upper right hand side of Figure 5.

If tool A exports the diagram with connectivity information, though, tool B is able to change the coordinates of the start and end points of the line upon importing using its own algorithm to show that the line continues to connect the two nodes. This is shown on the lower right hand side of Figure 5.

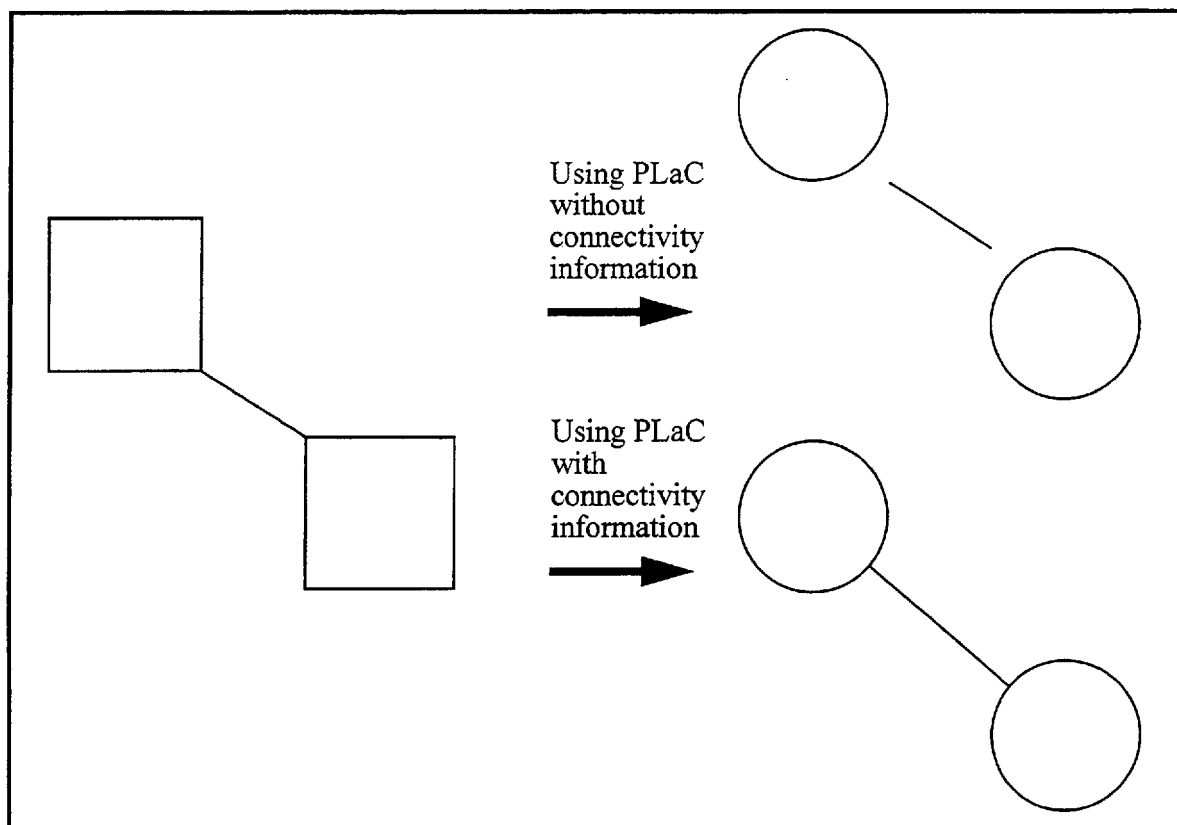


Figure 5
Connectivity information in a transfer that does not have Shape information

Similarly, if tool B exported data corresponding to the upper part of Figure 5 that is subsequently imported by tool A, the connectivity information allows tool A to determine that the line between the nodes is not attached and should be kept separate. Tool A could do this, for example, by modifying the start and end coordinates of the line.

If tool B exported the lower part of Figure 5 without connectivity information, tool A would most likely assume that the lines end within the node, as opposed to at the boundary of the node. With connectivity information it can adjust the coordinates in a way that presents the connectivity information correctly.

The following sub-sections explain the major concepts supported in this subject area.

4.1.2 Transfer Coordinate Frame

The first criterion for specifying location is the establishment of a coordinate frame. The CDIF Coordinate Frame for a CDIF Transfer is the same for all transfers. It is a three dimensional system with the origin at (0,0,0), assumed to be at the top left front, with X, Y, Z coordinate limits of minimum to the maximum of a CDIF integer. The positive X-axis extends to the right, the positive Y-axis downwards, and the positive Z-axis to the back.

At this time most modeling tools use a two dimensional coordinate system, however to permit upward compatibility a three dimensional coordinate system is specified. CDIF requires that the Z coordinate have a value of 0 in all transfers that use only a two dimensional system¹. Figure 6 illustrates the CDIF Coordinate Frame.

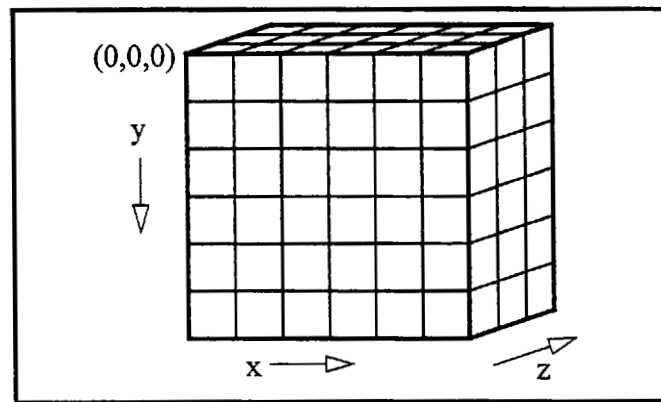


Figure 6
CDIF Coordinate Frame

This coordinate system is a grid, addressed in terms of CDIF Points, that are defined as the intersection of grid lines on the coordinate plane. A CDIF Point is the unit of graphical addressing. Figure 7 shows the relationships between grid lines, Points and CDIF Pixels.

Note that the grid lines are infinitely thin and thus CDIF Points are infinitely small.

4.1.3 Diagrams

The *Diagram* meta-entity is provided as the container to which all other presentation information objects are attached. This attachment is represented by the meta-relationships *GraphicalElement.AppearsOn.Diagram* and *Point.IsLocatedOn.Diagram*.

-
1. Throughout this document examples will be provided in two dimensions only. It is assumed that the examples will have more relevance to the current user community in this form.

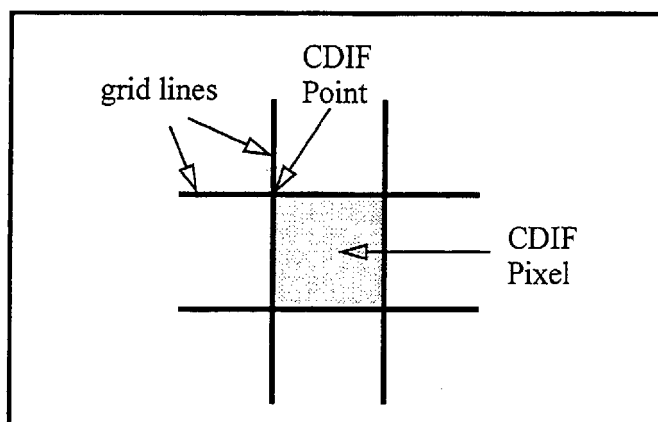


Figure 7
CDIF Points and CDIF Pixels

Diagram can represent screens, sheets, paper, a three-dimensional model space, or any other means that can carry presentation information. Meta-attributes *ExtentX*, *ExtentY* and *ExtentZ* specify the size of the *Diagram*.

The *DotsPerUnit* and *Unit* meta-attributes provide the mapping from CDIF coordinates within the *Diagram* into 'real world' coordinates. For instance, in a *Diagram* whose meta-attribute *DotsPerUnit* specifies 72, and whose meta-attribute *Unit* specifies *Inch*, a line from (0,0) to (0,72) would be one inch long.

4.1.4 Points

In order to specify the location of a graphical element on a diagram, a subtype of meta-entity *Point* is used. A *Point* is always located on a *Diagram*. This is expressed using meta-relationship *Point.IsLocatedOn.Diagram*.

Meta-entity *AbsolutePoint* represents a location on a *Diagram* which is fixed relative to the origin of the *Diagram*. For example, if a *Node* uses meta-relationship *PositionedElement.HasCenter.Point*, the coordinate represented by meta-entity *AbsolutePoint* with value (20,10,0) means that the center of the *Node* is found 20 pixels to the right of the upper-left corner of the diagram, and 10 pixels to the bottom.

Meta-entity *RelativePoint* represents a point whose location is relative to the location of another *Point*. This allows one graphical object, or a group of graphical objects, to be positioned relative to another graphical object or group of objects.

This association is achieved using meta-relationship *RelativePoint.IsRelativeTo.Point*. The relative offset of the *RelativePoint* to the *Point* is specified using meta-attributes *DeltaX*, *DeltaY* and *DeltaZ*. To find the absolute coordinates of a *RelativePoint* which is relative to an *AbsolutePoint*, the relative coordinates are added to the absolute coordinates of the *AbsolutePoint*.

This can be used, for example, if an *EdgeElement* has a *RelativePoint* as its start point, and this *RelativePoint* is relative to a location of a *Node*, this represents the “stickiness” of the starting point of the *EdgeElement* relative to the *Node*.

RelativePoints may also rotate and stretch the relative coordinate system they define relative to another coordinate system. In order to do this, nine optional meta-attributes are provided on meta-entity *RelativePoint*: *Transform11*, *Transform12*, ... *Transform33*. These meta-attributes define the values for a matrix representing the coordinate transformation. This transformed coordinate system is used for all coordinates specified relative to the *RelativePoint* described by the corresponding meta-attributes. The main purpose of this is to rotate objects or groups of objects.

In order to determine the coordinates of a location *L* in the coordinate system of a *Point B*, if only the coordinates of the location in the coordinate system of a *RelativePoint A* that is relative to *Point B* are given, the following transformation is applied:

$$L_B = T \cdot L_A + \Delta$$

where:

L_A is the location in coordinate system defined by *RelativePoint A*.

L_B is the location in coordinate system defined by *Point B*.

T is the transformation matrix formed by the *Transform11* ... *Transform33* meta-attributes (abbreviated as $t_{11} \dots t_{33}$ in the following equation) with

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

Δ is the translation vector defined by *DeltaX*, *DeltaY*, *DeltaZ*.

If values for the meta-attributes *Transform11* ... *Transform33* are not given, the matrix is assumed to be the unity matrix. If values for the meta-attributes *Transform11* ... *Transform33* are given, the base vectors of the matrix need to be orthogonal, i.e. $t_{ik} \cdot t_{jk} = 0$ for all $i, j, k \in [1, 2, 3]$, $i \neq j$.

Figure 8 shows an example for this. *RelativePoint A* is relative to *B* by an offset of 5 in the x direction, and an offset of -2 in the y direction. Note that the positive y direction in the CDIF Coordinate Frame points downwards.

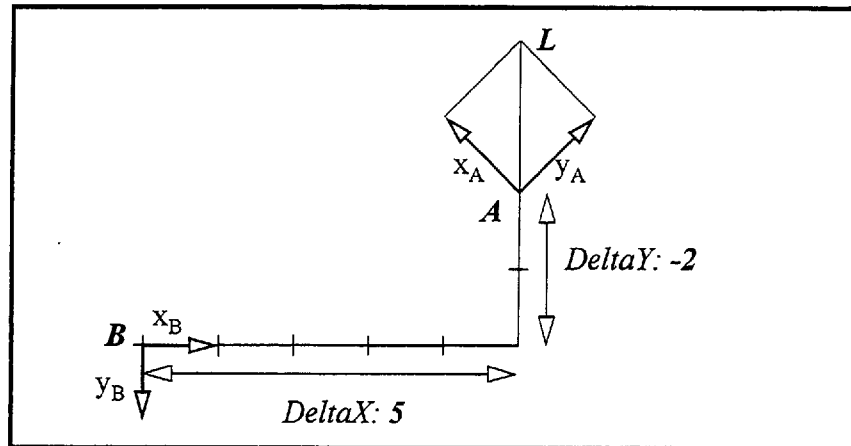


Figure 8
A transformed coordinate system

In addition, the local coordinate system defined by *A* is a transformation of the coordinate system for *B*. This transformation is described by the following matrix:

$$T = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In a CDIF Transfer, this matrix would be represented as meta-attributes of the *RelativePoint*:

<i>Transform11</i>	-1
<i>Transform12</i>	-1
<i>Transform13</i>	0 (optional in the 2-dimensional case)
<i>Transform21</i>	-1
<i>Transform22</i>	1
<i>Transform23</i>	0 (optional in the 2-dimensional case)
<i>Transform31</i>	0 (optional in the 2-dimensional case)
<i>Transform32</i>	0 (optional in the 2-dimensional case)
<i>Transform33</i>	1 (optional in the 2-dimensional case)

Location L in the coordinate system defined by A is $(1,1,0)$. Location L in the coordinate system defined by B is $(5,-5,0)$. Location L could be defined within this subject area as either an *AbsolutePoint*, or a *RelativePoint* that is relative to B , or a *RelativePoint* that is relative to A . If it was relative to B , its meta-attribute values would be $\Delta X=5$, $\Delta Y=-5$, $\Delta Z=0$. If it was relative to A , its meta-attributes would be $\Delta X=1$, $\Delta Y=1$, $\Delta Z=0$.

There is a significant difference for *Nodes* attached to a *Point L*, depending on whether L is absolute, or relative to A or B . If it is absolute, the *Node's* x and y directions will be aligned to the x and y directions of the enclosing *Diagram*. If it is relative to B , the *Node's* x direction would be parallel to the x direction of the coordinate system defined by B . If it is relative to A , though, a *Node's* x direction would be parallel to the x direction of the coordinate system defined by A .

Additionally the absolute distances could be different because the different coordinate systems do not necessarily have the same norm. In the example, the absolute distances are by a factor of $\sqrt{2}$ larger than the *Extent* meta-attributes seem to indicate. This is because the matrix has a norm different from 1.

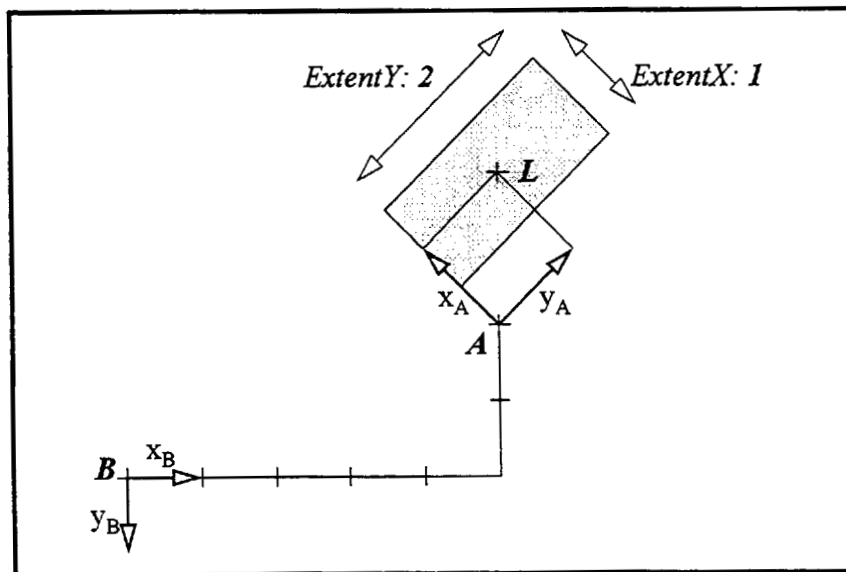


Figure 9
A *Node* in a transformed coordinate system

Figure 9 shows a *Node* which is related to L using meta-relationship *Node.HasCenter.Point* for the case of L being a *RelativePoint* relative to A .

4.1.5 PositionedElements, Center and Extent

Abstract meta-entity *PositionedElement* is provided as a supertype for all concrete meta-entities representing concepts that occupy space or area on a *Diagram*.

All *PositionedElements* are characterized by a location and an extent. The location of any *PositionedElement* is specified by a *Point* which is considered the center of the *PositionedElement*. The center of a *PositionedElement* is considered the center of the bounding box, adjusted to the local X,Y,Z coordinate frame defined by the chosen subtype of *Point* (see Section 4.1.4, Points).

The extent of a *PositionedElement* is described by the meta-attributes *ExtentX*, *ExtentY* and *ExtentZ*. They represent the side lengths of the enclosing rectangle of the shape. Figure 10 shows examples of locations and extents.

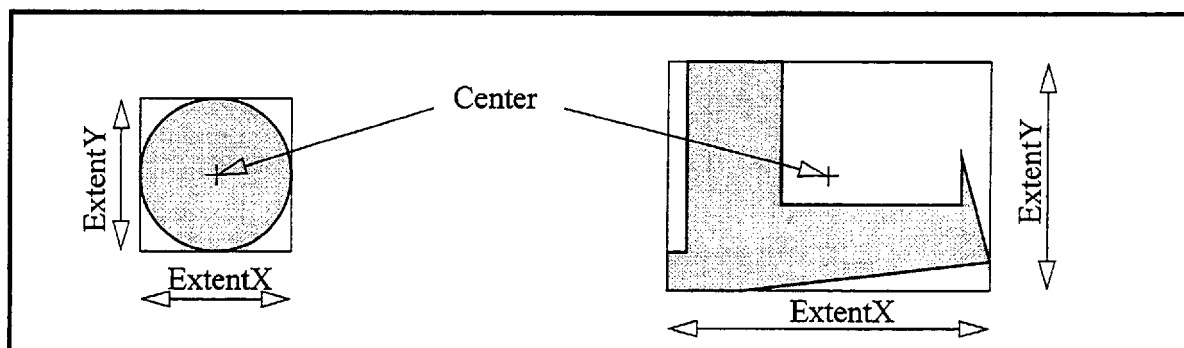


Figure 10
Center and Extent

Meta-relationship *Point.IsDependentUpon.PositionedElement* relates *PositionedElements* to those *Points* whose location could potentially be changed if the shape of the *PositionedElement* changes.

This is illustrated in the example in Figure 5. *Points* determining the location of *EdgeElements* may depend on the shape of a *Node*. Only the extent of a *Node*, and not its shape can be specified using this subject area. The *Points* whose location potentially changes due to the lack of shape information in this subject area, are related to the *Node* using meta-relationship *Point.IsDependentUpon.PositionedElement*.

4.1.6 Nodes

Any 2-dimensional or 3-dimensional graphical element in a diagram but not a label or text, is represented using meta-entity *Node*.

4.1.7 Annotations

The concept of an *Annotation* is provided in order to allow graphical, textual or other adornments of node-edge diagrams. *Annotation* is a subtype of *PositionedElement*, and as such its position is specified using a center location and extent into the three dimensions, potentially rotated as specified by the *Point* which acts as its center.

The actual value for the *Annotation* is constructed from the *AnnotationArguments* related to the *Annotation* using meta-relationship *Annotation.Uses.AnnotationArgument*. In order to form this value, the inherited meta-attributes *DerivationLanguage* and *DerivationText* can be used. If no derivation is specified, the value for the *Annotation* is the concatenation of the *AnnotationArguments* in the sequence specified by the meta-attribute *SequenceNumber* on meta-relationship *Annotation.Uses.AnnotationArgument*.

For example, the label associated with an *Edge* representing a *Relationship* in a *Diagram* could be represented using one *Annotation* with five *AnnotationArguments*: the first one could represent the name of the source *Entity*, the second and fourth a '.', the third the short name of the *Relationship*, and the fifth the name of the destination *Entity*. Figure 11 and Figure 12 illustrate this example.

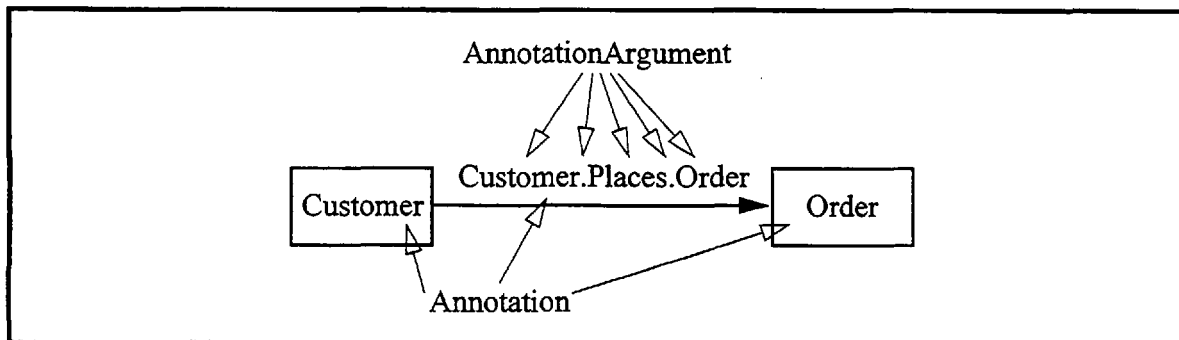


Figure 11
Example of an *Annotation*

Using the inherited meta-relationship

PresentationInformationObject.Represents.SemanticObjectReference, the value of individual *AnnotationArguments* can be linked to values of meta-attributes in the semantic part of the model (see Section 4.1.9, Connection of Presentation and Semantic Information).

Alternatively, the meta-attributes *MIMETYPE*, *MIMESubtype* and *DataBlock* carried by meta-entity *AnnotationArgument* can be used to specify information that does not depend on semantic information. *MimeType* and *MimeSubtype* describe the data type for the multimedia information contained in meta-attribute *DataBlock*, as defined in *Internet RFC 1521 Mime Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*.

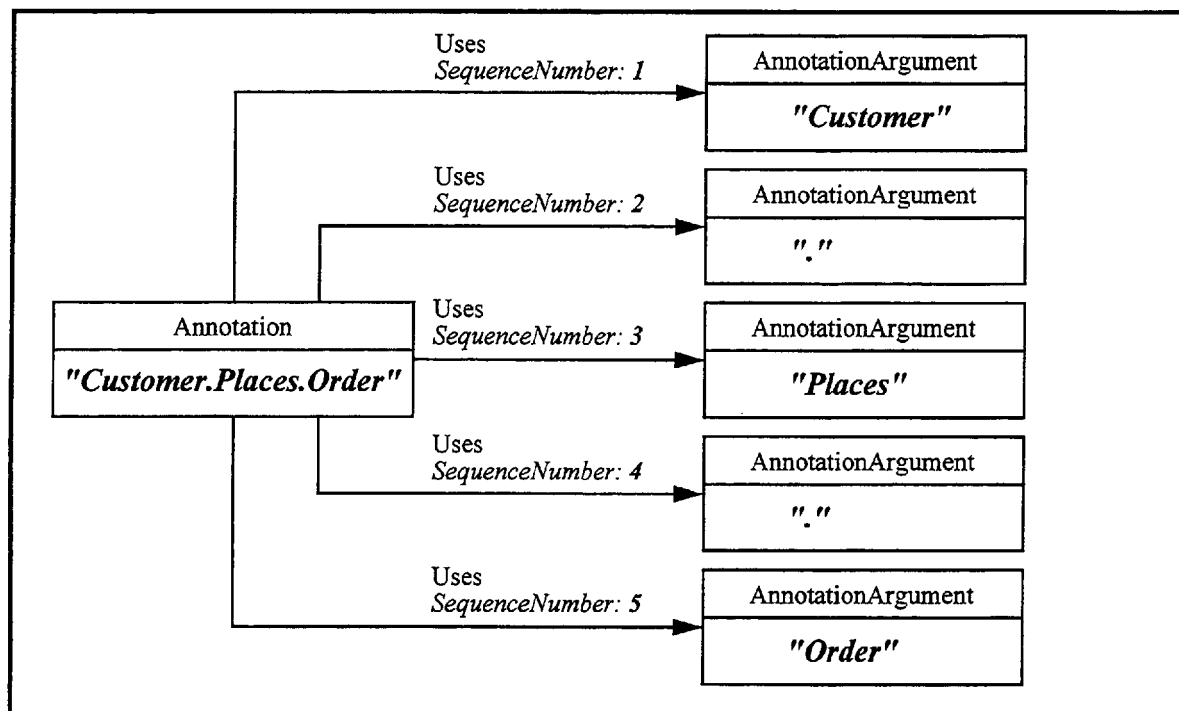


Figure 12
Instance diagram for the *Annotation* example of Figure 11

For example, an instance of *AnnotationArgument* carrying the values **image** for meta-attribute *MimeType*, **jpeg** for meta-attribute *MimeSubtype* expresses that meta-attribute *DataBlock* contains a JPEG raster image which is the value of the *AnnotationArgument*.

4.1.8 Edges and EdgeElements

Any undirected or directed line on a diagram is represented using the meta-entities *Edge* and *EdgeElement*. The meta-entity *Edge* provides the connectivity information, while *EdgeElements* are used to represent individual line segments with one source and one destination *Point*.

Many *EdgeElements* can be used to build up a single *Edge*. Meta-relationship *Edge.ConsistsOf.EdgeElement* is used to represent this aggregation. Figure 13 shows an example for this.

The bent line between the two boxes is represented using meta-entity *Edge*. By instantiating meta-relationship *Edge.IsAttachedTo.GraphicalElement* for the source and destination *Nodes*, the connectivity information is represented.

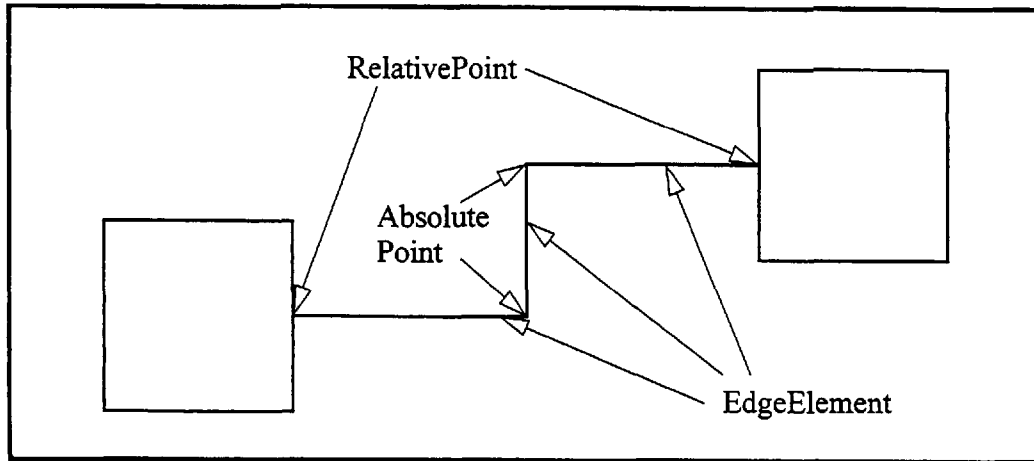


Figure 13
Points, Edges and EdgeElements

The actual graphical representation of the *Edge* is performed using *EdgeElements*. An *EdgeElement* is an individual segment of an *Edge*, characterized by one start and one end point. These start and end points, each identified by an instance of meta-relationship *EdgeElement.HasEnd.Point*, represent the locations in the diagram where the edge element has its start and end points. Often the start point of one segment is identical to the end point of another segment of the same edge.

The specification of the shape of arrowheads is not supported in this subject area.

4.1.9 Connection of Presentation and Semantic Information

Semantic information may be referenced by a *PresentationInformationObject*. For example, in Figure 14, a *Node* (defined in this subject area) is used to represent the semantics of a *DFMProcess* (from EIA/IS-115 CDIF - Integrated Meta-model - Data Flow Model Subject Area).

To express this relationship, meta-entity *SemanticObjectReference* and meta-relationship *PresentationInformationObject.Represents.SemanticObjectReference* are provided.

Instances of meta-entity *SemanticObjectReference* act as references to

- a) instances of subtypes of *SemanticInformationObject* (Figure 14 is an example);
- b) instances of subtypes of *RootEntity.IsRelatedTo.RootEntity*, where the source and destination meta-entity instances are instances of subtypes of *SemanticInformationObject* (Figure 16 is an example);
- c) values of specific meta-attributes in such an instance of a subtype of *SemanticInformationObject*, or an instance of *RootEntity.IsRelatedTo.RootEntity*, where the source and destination meta-entity instances are instances of subtypes of *SemanticInformationObject* (Figure 18 is an example);

In each of these cases, the instance of *SemanticObjectReference* acts as a proxy for the actual instance data to be represented. The instance data is identified by the *CDIFIdentifier* meta-attribute which has a value assigned during a CDIF transfer.

4.1.9.1 Presentation information for meta-entity instances

Figure 14 shows an example for a). Figure 15 is the corresponding instance diagram.

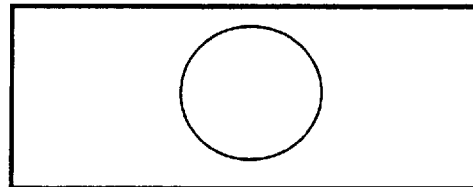


Figure 14
Presentation for a *DFMPProcess*

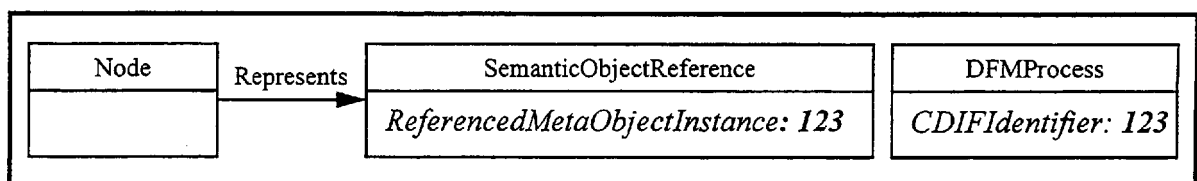


Figure 15
Representation of both presentation and semantic information for a *DFMPProcess*

In order to represent the *DFMPProcess* both semantically and graphically, meta-entity *DFMPProcess* and meta-entity *Node* need to be instantiated. The *Node* represents the fact that this *DFMPProcess* is represented as a box at a specific location with a certain extent in a specific *Diagram*.

In order to represent that the *Node* and the *DFMProcess* are related to each other in a way that makes the *Node* a graphical representation of the semantic concepts associated with the *DFMProcess*, meta-entity *SemanticObjectReference* and meta-relationship *PresentationInformationObject.Represents.SemanticObjectReference* are instantiated.

In order to express the association, the *SemanticObjectReference* carries the *CDIFIdentifier* value of the instance of the semantic concept referenced (i.e. the *CDIFIdentifier* of the meta-entity *State*) as the value of meta-attribute *ReferencedMetaObjectInstance*.

4.1.9.2 Presentation information for meta-relationship instances

Figure 16 shows an example for b). Figure 17 is the corresponding instance diagram.

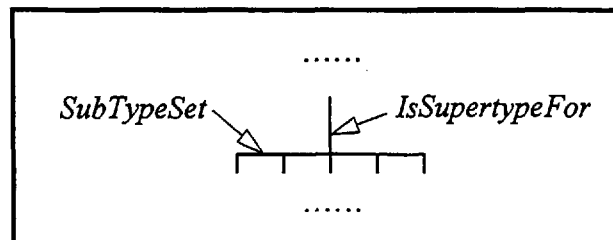


Figure 16
Presentation of a *SubTypeSet* in an inheritance situation

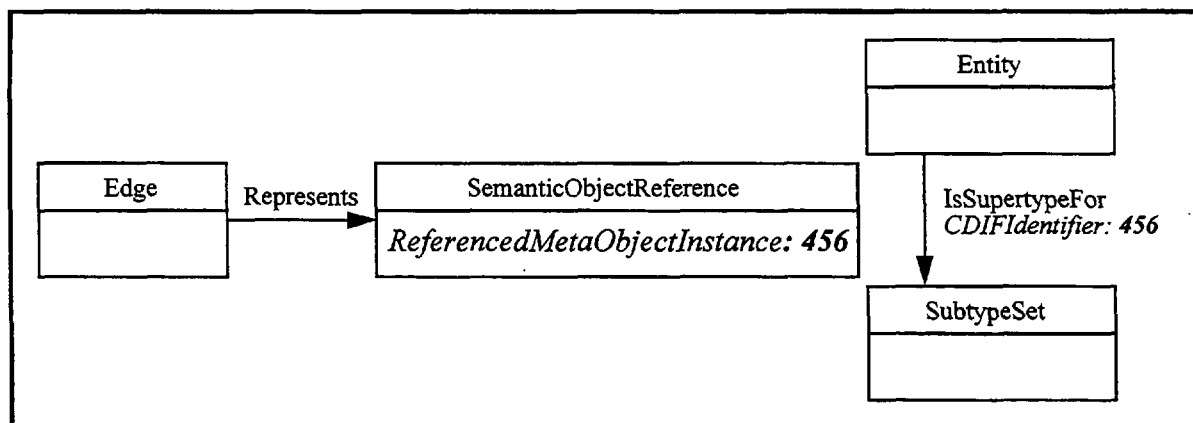


Figure 17
Representation of both semantics and presentation for a *SubtypeSet*

Here, an instance of a subtype of *PresentationInformationObject*, *Edge*, represents an instance of a subtype of *RootEntity.IsRelatedTo.RootEntity*, the meta-relationship *Entity.IsSupertypeFor.SubtypeSet* (as found in *EIA/IS-114 CDIF - Integrated Meta-model - Data Modeling Subject Area*).

Similarly to a), meta-entity *SemanticObjectReference* is instantiated. The *ReferencedMetaObjectInstance* meta-attribute contains the value of the *CDIFIdentifier* for the referenced meta-relationship instance.

4.1.9.3 Presentation information for values of meta-attributes

Figure 18 shows an example for c). Figure 19 is the corresponding instance diagram.

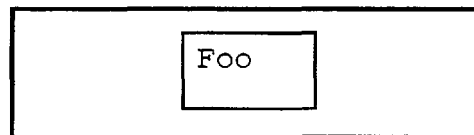


Figure 18
Presentation of a label denoting the name of an *Entity*.

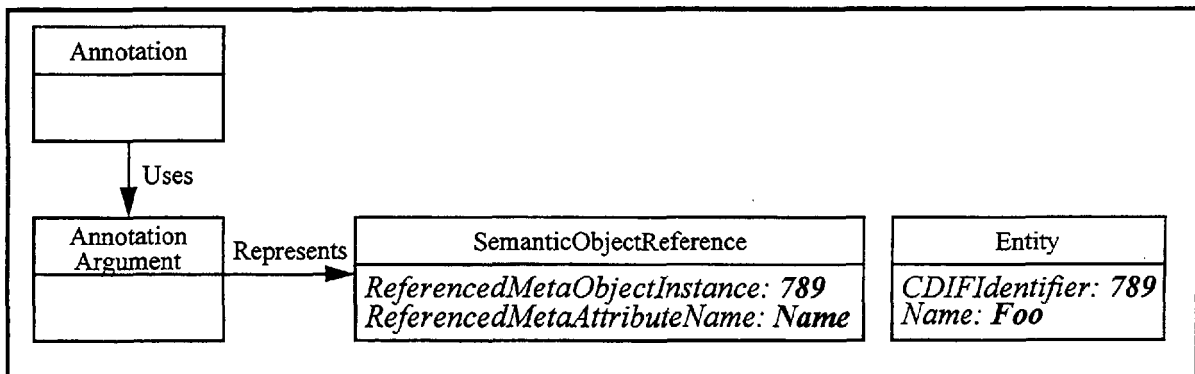


Figure 19
Representing both semantic and presentation of a name of an *Entity*

In this example, an instance of a subtype of *PresentationInformationObject*, *AnnotationArgument*, represents the value of a meta-attribute in an instance of a subtype of *SemanticInformationObject*, the value of the *Name* meta-attribute of an instance of meta-entity *Entity*.

In order to reference the value of a specific meta-attribute for a given meta-entity or meta-relationship in such an instance, the meta-attribute *ReferencedMetaAttributeName* on meta-entity *SemanticObjectReference* is used. This meta-attribute, if specified, denotes the name of the meta-attribute whose value the *SemanticObjectReference* represents, in addition to the *CDIFIdentifier* of the parent meta-entity or meta-relationship instance.

4.1.10 Computable Languages

The set of computable languages that can be supported is defined in Section 4 of *EIA/IS-112*
CDIF - Integrated Meta-model - Common Subject Area.

4.2 Diagrams

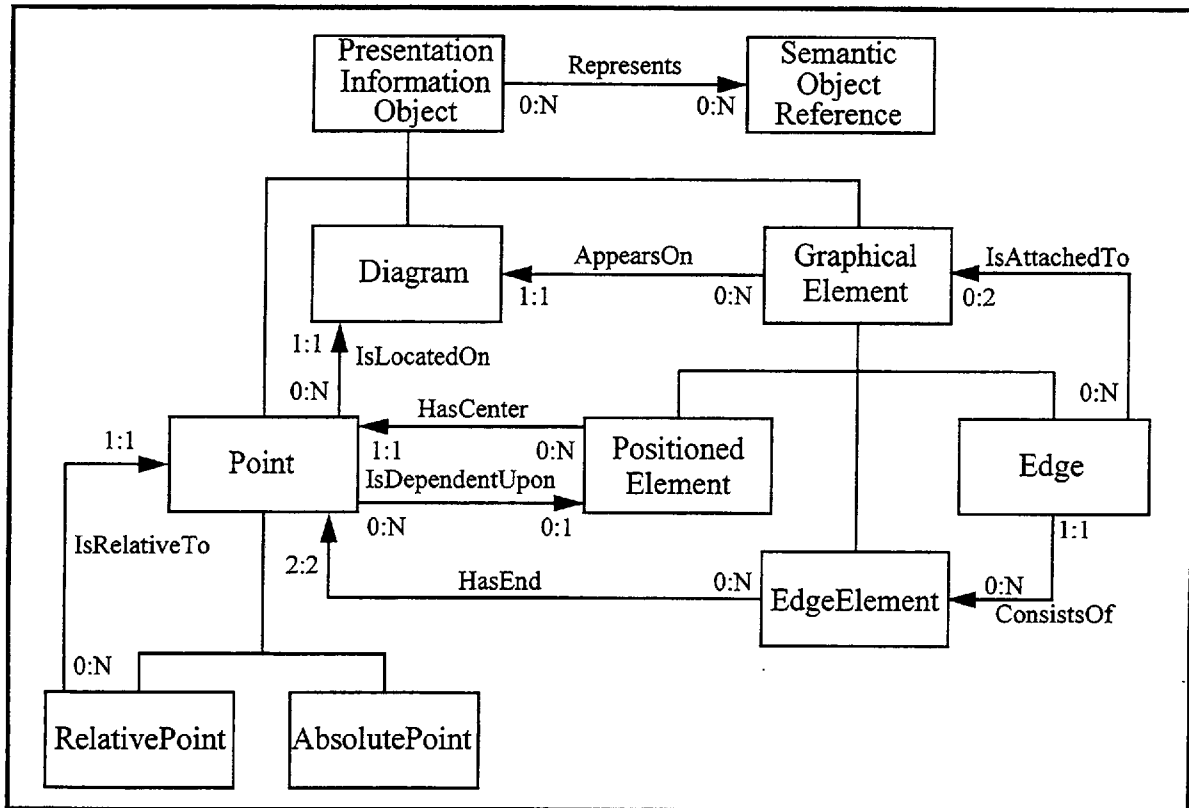


Figure 20
 Presentation Location and Connectivity Subject Area Overview

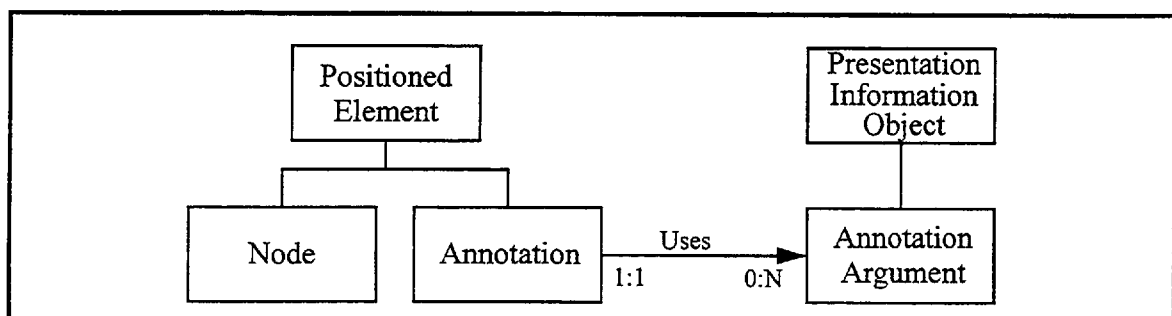


Figure 21
 Node and Annotation Sub-model