IEEE Standard for Information Technology—Software Reuse—Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM)

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Developed in cooperation with the

Reuse Library Interoperability Group (RIG)

Approved December 12, 1995

IEEE Standards Board

Abstract: The minimal set of information about assets that reuse libraries should be able to exchange to support interoperability is provided**.**
Keywords: reuse libraries; software, interoperability

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Print: ISBN 1-55937-584-1, SH94365
PDF: ISBN 0-7381-0424-8, SS94365

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

[This introduction is not part of IEEE Std 1420.1-1995, IEEE Standard for Information Technology—Software Reuse—Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM).]

This standard was developed as the result of a collaboration between the Reuse Library Interoperability Group (RIG) and the Software Engineering Standards Committee (SESC) of the IEEE Computer Society. The RIG is a volunteer organization of experts from the industrial, government, and academic communities working together to create consensus-based specifications for the interoperation of software reuse libraries. Under the terms of a Memorandum of Understanding with the IEEE, the RIG develops specifications and other documents on this subject and submits them to the SESC. The RIG then responds to the IEEE Sponsor Balloting process in order to improve the documents and widen the community of consensus. Following normal IEEE rules and procedures, the documents are finally published as IEEE standards.

This document was originally developed as a RIG Proposed Standard RPS-0001 (1993), *A Basic Interoperability Data Model (BIDM)*. It was prepared by the RIG Technical Committee on Asset Exchange Interfaces. At the time the proposed standard was approved, membership of the committee was as follows:

Pamela K. Arya, *Chair*
Elizabeth T. Hobbs, *Past Chair*

Peter Berggren
Scott Clark
Ted Davis
Karen Dennis
Marcelle Hajjar
Stephen Huang
Barry Jacobs
Jeffrey Kay
Roy Lloyd
Thomas Loggia
Jim Moore
Tim Niesen
Steven Parker
Jeff Poulin
John Shockro
Bruce Soileau
Jim Solderitsch
Wilson Tan
Claire Tozier
Jack Tupman
Newman Vosbury
Frank Walker

A balloting group formed of members of the Combined Technical Committee approved the document for submission to the RIG Executive Board. Members of the balloting group were as follows:

Eric Beser; *Chair, Combined Technical Committee*

Mary Armstrong
Rose Armstrong
Pam Arya
Peter Berggren
Barry Bounds
Linda Braun
Sherrie Chubin
Scott Clark
Ted Davis
Karen Dennis
Dave Dikel
David Dymm
Manuel Faria
Barbara Fleming
Karen Fleming
Marcelle Hajjar
Al Hankinson
Terri Hobbs
Stephen Huang
George Jackelen
Lawrence Jacowitz
Jeffrey Kay
Debby LaForme
Gertrude Levine
Roy Lloyd
Thomas Loggia
Jim Moore
David Nassau
Stephen Parker
Teri Payton
Jeffrey Poulin
Donald J. Reifer
Jack Rothrock
Robert Saisi
John Shockro
Bruce Soileau
Jim Solderitsch
Claire Tozier
Newman Vosbury
Jim Wilson

The RIG Executive Board approved this document on April 1, 1993, with the following officers:

James W. Moore, *Chair*
Eric Beser, *Vice Chair*
Peter Berggren, *Secretary*

When this standard was balloted in the IEEE, the balloting committee had the following membership:

Syed Ali
Pamela K. Arya
Man K. Au
Mordechai Ben-Menachem
Peter A. Berggren
H. Ronald Berlack
Christos Bezirtzoglou
Richard E. Biehl
William J. Boll
Fletcher Buckley
Michael Caldwell
Lawrence Catchpole
Betty P. Chao
John P. Chihorek
Antonio M. Cicu
Theo Clarke
Sharon Cobb
Virgil Lee Cooper
Geoffrey Darnton
Bostjan K. Derganc
Harpal Dhama
Dave Dikel
Leo G. Egan
John W. Fendrich
Julian Forster
Adel N. Ghannam
Julio Gonzalez Sanz
Eric Grosse
Lawrence M. Gunther
David A. Gustafson
Rob Harker
Myron Hecht
Rick Hefner
Mark Heinrich
Umesh P. Hiriyannaiah
Helmut Hummel
Fabrizio Imelio
Frank Jay
David Johnson
Martha Johnson
Vladan V. Jovanovic
William S. Junk
Chris F. Kemerer
Judy Kerner
Robert J. Kierzyk
Motti Y. Klein
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Fang Ching Lim
Gianluca Marcellino
Robert A. Martin
Tomoo Matsubara
Patrick McCray
Jacques Meekel
Glen A. Meldrum
Celia H. Modell
James W. Moore
Dennis E. Nickle
Mike Ottewill
Gerald L. Ourada
Indradeb P. Pal
Lalit Mohan Patnaik
Donald J. Pfeiffer
John G. Phippen
Peter T. Poon
Jeffrey S. Poulin
Lawrence S. Przybylski
John Reddan
Dennis Rilling
James R. Roberts
R. Waldo Roth
Terence P. Rout
Andrew P. Sage
Stephen R. Schach
Peter E. Schilling
Gregory D. Schumacher
Katsutoshi Shintani
David M. Siefert
Raghu P. Singh
Nancy M. Smith
Melford E. Smyre
Jag Sodhi
Fred J. Strauss
Toru Takeshita
Booker Thomas
Patricia Trellue
Leonard L. Tripp
Mark-Rene Uchida
Theodore J. Urbanowicz
Glenn D. Venables
Dolores Wallace
Camille S. White-Partain
Scott A. Whitmire
Charles D. Wilson
Paul R. Work
Kathryn P. Yglesias
Weider D. Yu
Geraldine Zimmerman
Peter F. Zoll

When the IEEE Standards Board approved this standard on December 12, 1995, it had the following membership:

E. G. "Al" Kiener, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman
Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
Ivor N. Knight
Joseph L. Koepfinger*
D. N. "Jim" Logothetis
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
John W. Pope
Arthur K. Reilly
Gary S. Robinson
Ingo Rusch
Chee Kiow Tan
Leonard L. Tripp
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Steve Sharkey
Robert E. Hebner
Chester C. Taylor

Kristin M. Dittmann
IEEE Standards Project Editor

Contents

IEEE Standard for Information Technology—Software Reuse—Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM)

1. Overview

1.1 Scope

This document describes the Basic Interoperability Data Model (BIDM) standard as developed by the Reuse Library Interoperability Group (RIG).

1.2 Purpose

The purpose of the BIDM is to define the minimal set of information about assets that reuse libraries should be able to exchange to support interoperability. Because the BIDM describes a minimal set, other data that would be useful for interoperability, including library data model information, is not included in the BIDM. Furthermore, the BIDM does not include communication protocol or related standards. Other RIG standards will address these aspects of interoperability. More information about the RIG and about some of these future standards is included in annex A.

Important terms used in the BIDM are defined in clause 2. The core of the BIDM standard is contained in clause 3, which includes a description of the meta-model of the BIDM and a definition of the data model for basic interoperability.

2. Definitions

This clause contains some key terms used in this standard. Other terms are defined in the RIG Glossary [B4].[1]

2.1 asset: (1) The items of interest which are stored in a reuse library, such as design documentation, specifications, source code, documentation, test suites, etc., or any other unit of information of potential value to a reuser [B4]. (2) A class in the BIDM. *Asset* will always be capitalized when referring to the class Asset.

[1]Numbers in brackets correspond to those of the bibliography in clause 5.

2.2 attribute: A predefined characteristic that provides a property of a class. Properties are inherited from a class to its subclasses.

2.3 Basic Interoperability Data Model (BIDM): Defines the minimal set of information that reuse libraries should be able to exchange about assets in order to interoperate.

2.4 class: A grouping of similar objects that have the same structure but potentially different attribute and/or relationship values.

2.5 class hierarchy: An ordering of classes, in which a subclass is a specialization of its superclass. A class inherits attributes and relationships from its superclass and can define additional attributes and relationships of its own.

2.6 library data model: The organizing principles and concepts underlying structured data in a reuse library and the means of representing that structure.

2.7 object: A representation of a real-world entity. An object is an instance of a class and has values for the attributes and relationships defined for that class.

2.8 relationship: An association between two classes.

3. The basic interoperability data model

This clause describes the BIDM developed by the RIG. The principal sources of data for the model were the Common Data Model defined in the Asset Library Open Architecture Framework (ALOAF) [B1], the Repository Guidelines and Standards [B3], and the RIG Technical Committee 2 members. Figure 1 gives a pictorial overview of the BIDM. The * in figure 1 identifies a local relationship.

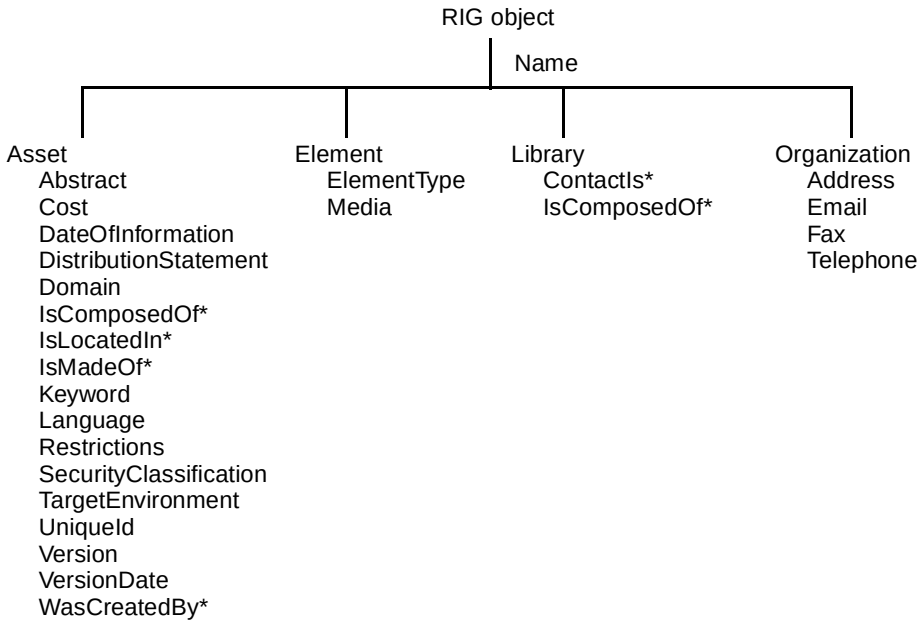

Figure 1—BIDM class hierarchy

The meta-model (see RIG Glossary [B4]) used to describe the BIDM is described in 3.1. The data types that will be used in specifying the attributes in the BIDM classes are defined in 3.2. A description of the classes that make up the BIDM is contained in 3.3. An alphabetized list of attributes and their definitions appears in 3.4. Finally, an alphabetized list of the relationships and their definitions is included in 3.5.

3.1 The BIDM meta-model

Techniques from the ALOAF [B1] and the CASE Data Interchange Format (CDIF) [B2] meta-models were incorporated into the BIDM meta-model. The BIDM meta-model uses the following entities:

a) Classes

b) A class hierarchy

c) Class attribute

d) Bidirectional relationships between classes

e) Relationship attributes

Classes are defined with the template shown in figure 2. Not all data in the class template need be supplied. If no data is present for a field in the template, the field is empty.

Name	:
SubClassOf	:
SuperClassOf	:
Description	:
Usage	:
Inherited attributes	:
Local attributes for class	:
Inherited relationships	:
Local relationships	:

Figure 2—BIDM class template

The definition of the terms in figure 2 follow:

— **Name:** Name of a grouping of similar objects that have the same structure but potentially different attribute and relationship values.

— **SubClassOf:** Name of immediate superclass, the more generalized or abstract version of this class.

— **SuperClassOf:** Name of immediate subclasses, a specialization, or a refined version of this class.

— **Description:** Describes the intent or purpose of the class.

— **Usage:** Additional guidance in determining values for the class in the data model.

— **Inherited attributes:** A characteristic defined by a superclass that provides a property of this class.

— **Local attributes for class:** A characteristic defined by this class that provides a property of this class and its subclass.

— **Inherited relationships:** An association between two classes where the source of the association is a superclass of this class.

— **Local relationships:** An association between two classes where the source of the association is this class.

The classes have attributes. These attributes are defined using the template shown in figure 3. Not all data for this template need be supplied. If the data for a template field is not applicable, the word “None” is used to show this fact. AttributeName, Description, Class, Single/Multiple, DataType, and IsOptional are required.

AttributeName	:
Description	:
Class	:
Single/Multiple	:
DataType	:
IsOptional	:
Constraints	:

Figure 3—BIDM attribute template

The definition of the terms in figure 3 follow:

— **AttributeName:** A descriptive name assigned to the attribute.
— **Description:** Informal definition of the attribute and its usage.
— **Class:** The name of the origin class, the topmost class for which the attribute is specified.
— **Single/Multiple:** Indicates whether only one or many values may be given for an attribute.
— **Datatype:** Category that indicates whether the values for the attribute are date, integer, string, or text.
— **IsOptional:** Indicator that an attribute value is not required.
— **Constraints:** Text description of additional limitations of appropriate attribute values.

Relationships may exist between classes. All relationships are bidirectional; that is, all relationships have a source and a destination. Cardinality is given for both the source and destination. The template shown in figure 4 describes a relationship in one direction (for example, the template uses the terms *source* and *destination*). Nonetheless, the relationship is also automatically defined in the reverse direction.

Name	:
Description	:
SourceClass	:
DestinationClass	:
MinDestinationCard	:
MaxDestinationCard	:
MinSourceCard	:
MaxSourceCard	:

Figure 4—BIDM relationship template

The definition for the terms in figure 4 follow:

— **Name:** Name of the relationship from the source class to the destination class.
— **Description:** Description of the intent and purpose of the relationship from the source class to the destination class.
— **SourceClass:** Origin of the relationship.
— **DestinationClass:** Destination of the relationship.
— **MinDestinationCard:** Minimum number of instances of the relationship allowed from the objects of the source class to objects of the destination class.
— **MaxDestinationCard:** Maximum number of instances of the relationship allowed from the objects of the source class to objects of the destination class.
— **MinSourceCard:** Minimum number of instances of the relationship allowed from objects of the destination class to objects of the source class.

— **MaxSourceCard:** Maximum number of instances of the relationship allowed from objects of the destination class to objects of the source class.

Relationships are given names, as can be seen in the template in figure 4. To be precise, however, the full name of the relationship consists of the source class, the relationship name, and the destination class separated with dots. For example: Asset.WasCreatedBy.Organization is a full relationship name; Asset is the originating class, Organization the destination class, and WasCreatedBy is the name of the relationship.

The destination cardinalities show the minimum and maximum number of instances of the relationship that are allowed from objects of the source class to objects of the destination class. Using the “Asset.WasCreatedBy.Organization” example, MinDestinationCard and MaxDestinationCard give the minimum and maximum number of relationship instances going from objects of the Asset class to objects of the Organization class. In other words, for any object in the Asset class, the number of WasCreatedBy relationships must be in the range given by the MinDestinationCard and MaxDestinationCard.

The source cardinalities show the reverse of the destination cardinalities. They define the minimum and maximum number of instances of the relationship that are allowed from objects of the destination class to objects of the source class. In the “Asset.WasCreatedBy.Organization” example, MinSourceCard and MaxSourceCard give the range of the number of WasCreatedBy relationship instances that are allowed from objects of the Organization class to objects of the Asset class.

A minimum cardinality of 0 means that the relationship is optional in that direction. A maximum cardinality of N indicates that the number of relationship instances an object can have in that direction is unlimited.

3.2 Data types

This subclause describes the data types that will be used to describe attributes in the BIDM. These types are taken from [B2]. The formats of these data types will be defined in a RIG standard addressing data representation.

3.2.1 Date

The date format will be determined by the transport protocol. Dates past the year 2000 shall be supported.

3.2.2 Integer

An integer shall be capable of holding values from –2**31 to 2**31–1.

3.2.3 String

A string is a series of printable characters with maximum length of 1023.

3.2.4 Text

A series of characters, each of which may be or may not be printable. This group may be of any length.

3.3 The classes

The BIDM consists of five classes composed in a class hierarchy with one parent class and four subclasses. The local attributes defined in the parent RIG_Object class are inherited by the subclasses. The four sub-

classes represent the entities modeled in the BIDM, i.e., assets, elements (which include files), libraries, and organizations (which include people).

Name	:	RIG_Object
SubClassOf	:	None
SuperClassOf	:	Asset, Element, Library, Organization
Description	:	The root class of the BIDM model.
Usage	:	This class provides the basis for the other classes in the BIDM model. It is not intended to be used for instantiating its own objects.
Inherited attributes None		
Local attributes for RIG_Object Name		
Inherited relationships None		
Local relationships None		

Name	:	Asset
SubClassOf	:	RIG_Object
SuperClassOf	:	None
Description	:	Provides the descriptive information about a reusable entity.
Usage	:	A reusable entity consists of at least one Asset object.
Inherited attributes Name from RIG_Object		
Local attributes Abstract Cost DateOfInformation DistributionStatement Domain Keyword Language Restrictions SecurityClassification TargetEnvironment UniqueID Version VersionDate		
Inherited relationships None		
Local relationships Asset.IsComposedOf.Asset Asset.IsLocatedIn.Library Asset.IsMadeOf.Element Asset.WasCreatedBy.Organization		

Name	:	Element
SubClassOf	:	RIG_Object
SuperClassOf	:	None
Description	:	Models the discrete pieces that make up an asset. Example pieces are documents, requirement specifications, test cases, source code, installation information, and read me files.
Usage	:	Asset objects consist of at least one element object. In some libraries, documents, requirement specifications, test cases, etc., might be assets themselves and not elements of an asset.
Inherited attributes		
	Name from RIG_Object	
Local attributes		
	ElementType Media	
Inherited relationships		
	None	
Local relationships		
	None	

Name	:	Library
SubClassOf	:	RIG_Object
SuperClassOf	:	None
Description	:	Models, repositories, or libraries that contain reusable entities. Only the information about libraries that is needed for the exchange of assets is provided.
Usage	:	Library objects participate in IsLocatedIn relationships with Asset objects.
Inherited attributes		
	Name from RIG_Object	
Local attributes		
	None	
Inherited relationships		
	None	
Local relationships		
	Library.ContactIs.Organization Library.IsComposedOf.Library	

Name	:	Organization
SubClassOf	:	RIG_Object
SuperClassOf	:	None
Description	:	Describes entities like people, companies, and committees. Information about organizations that is not needed for the exchange of assets is not included.
Usage	:	Organization objects participate in ContactIs relationships with Library objects. They also participate in WasCreatedBy relationships with Asset objects.
Inherited attributes		
	Name from RIG_Object	

Local attributes	
	Address
	Email
	Fax
	Telephone
Inherited relationships	
	None
Local relationships	
	None

3.4 The attributes

This subclause contains an alphabetical listing of the attributes for the classes defined in 3.3. The definitions of the data types used in the attribute definitions are contained in 3.1.

AttributeName	:	Abstract
Description	:	The general definition and/or explanation of the asset.
Class	:	Asset
Single/Multiple	:	S
DataType	:	Text
IsOptional	:	False

AttributeName	:	Address
Description	:	The mailing address of the person or organization. Includes street number and name, city, state, zip code, and country.
Class	:	Organization
Single/Multiple	:	S
DataType	:	String
IsOptional	:	False

AttributeName	:	Cost
Description	:	The type and amount of fees a subscriber must pay to obtain the right to reuse the asset. This information includes the currency that the fees shall be paid in and the pricing policy for maintenance support and upgrades.
Class	:	Asset
Single/Multiple	:	S
DataType	:	Text
IsOptional	:	False

AttributeName	:	DateOfInformation
Description	:	The date the information about the asset (that is, the meta-data defined in the BIDM) was last updated.
Class	:	Asset
Single/Multiple	:	S
DataType	:	Date
IsOptional	:	False

AttributeName	:	DistributionStatement
Description	:	An indication of who may receive the asset.
Class	:	Asset
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	Domain
Description	:	Broad area of interest (application area). Area of activity or knowledge.
Class	:	Asset
Single/Multiple	:	M
DataType	:	String
IsOptional	:	False

AttributeName	:	ElementType
Description	:	The kind of element; for example, requirements specification, test suite and make file.
Class	:	Element
Single/Multiple	:	M
DataType	:	String
IsOptional	:	False

AttributeName	:	Email
Description	:	An electronic mail address for the organization. Example: librarian@source.asset.com
Class	:	Organization
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	Fax
Description	:	A fax telephone number for the organization.
Class	:	Organization
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	Keyword
Description	:	A word or phrase that describes an aspect of the asset, such as the function performed by the asset or the object manipulated by the asset, and is useful in generating classification information. This attribute provides an uncontrolled description of the asset, as compared to the Domain attribute.
Class	:	Asset
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	Language
Description	:	A language used to express one or more elements of the asset. For example, Ada, German, and IDEF.
Class	:	Asset
Single/Multiple	:	M
DataType	:	String
IsOptional	:	False

AttributeName	:	Media
Description	:	An indication of the media through which the element can be obtained. For example, CD-ROM, hardcopy, tape, and electronic.
Class	:	Element
Single/Multiple	:	M
DataType	:	String
IsOptional	:	False

AttributeName	:	Name
Description	:	A name or title for the object (Element, Asset, RIG_Object, etc.). Note that name is repeatable, so that Stack Package, Stack, and Stack Data Structure Package can all be names for the same object.
Class	:	RIG_Object, Asset, Element, Library, Organization
Single/Multiple	:	M
DataType	:	String
IsOptional	:	False

AttributeName	:	Restrictions
Description	:	Legal information governing the use of the asset. The following legal aspects should be considered: **Copyright:** A copyright statement for this asset, including the date of the copyright. **Data rights:** Copy release or other proprietary release information. **Disclaimers:** Indicate any warnings or limitations associated with the asset. **Export restrictions:** For example, Export Administration regulations or International Traffic in Arms restrictions. **Licenses:** A statement by the owner that reports the terms and conditions under which an asset may be used. It will normally specify what the user is allowed to do (for example, make a single-use copy), and what the user is restricted from doing (for example, reverse engineer the asset if it contains only object code).

Class	:	Asset
Single/Multiple	:	S
DataType	:	Text
IsOptional	:	True

AttributeName	:	SecurityClassification
Description	:	The highest security classification assigned to the asset (any part of it).
Class	:	Asset
Single/Multiple	:	S
DataType	:	String
IsOptional	:	False

AttributeName	:	TargetEnvironment
Description	:	Name of a computer system, an operating system, and/or a compiler for which the asset is developed.
Class	:	Asset
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	Telephone
Description	:	A (voice) telephone number for the organization.
Class	:	Organization
Single/Multiple	:	M
DataType	:	String
IsOptional	:	True

AttributeName	:	UniqueID
Description	:	A name or handle that is used to refer to the set of data that makes up the identity of an asset. This UniqueID, unlike the Identifier attribute, is intended to be unique not only in the context of an asset exchange, but in all contexts and in all libraries.
Class	:	Asset
Single/Multiple	:	S
DataType	:	String
IsOptional	:	False

AttributeName	:	Version
Description	:	The designation of the version of the asset (usually a number).
Class	:	Asset
Single/Multiple	:	S
DataType	:	String
IsOptional	:	True

AttributeName	:	VersionDate
Description	:	The date on which the version of the given asset was completed. More specifically, the date the library considers this information available for public use. A publication date.
Class	:	Asset
Single/Multiple	:	S
DataType	:	Date
IsOptional	:	True

3.5 The relationships

The following contains a list of the five relationships defined in the BIDM classes.

Name	:	ContactIs
Description	:	Relates Library and Organization. It shows an organization that is the point of contact for the library. People can be organizations.
SourceClass	:	Library
DestinationClass	:	Organization
Alias	:	IsContactFor
MinDestinationCard	:	1
MaxDestinationCard	:	N
MinSourceCard	:	0
MaxSourceCard	:	N

Name	:	IsComposedOf
Description	:	Relates Asset to Asset or relates Library to Library. It indicates an asset that is a subasset or part of another asset. For a library, it indicates a library that is part of a library. The composition of assets is library dependent.
SourceClass	:	Asset, Library
DestinationClass	:	Asset, Library
MinDestinationCard	:	0
MaxDestinationCard	:	N
MinSourceCard	:	0
MaxSourceCard	:	N

Name	:	IsLocatedIn
Description	:	Relates Asset and Library. It indicates the library or libraries where an asset is located. The library contains both a description of the asset and the asset itself.
SourceClass	:	Asset
DestinationClass	:	Library
MinDestinationCard	:	1
MaxDestinationCard	:	N
MinSourceCard	:	0
MaxSourceCard	:	N

Name	:	IsMadeOf
Description	:	Relates Asset and Element. It indicates what elements make up an asset.
SourceClass	:	Asset
DestinationClass	:	Element
MinDestinationCard	:	1
MaxDestinationCard	:	N
MinSourceCard	:	1
MaxSourceCard	:	N

Name	:	WasCreatedBy
Description	:	Relates Asset and Organization. It shows what organizations originated or produced an asset.
SourceClass	:	Asset
DestinationClass	:	Organization
MinDestinationCard	:	1
MaxDestinationCard	:	N
MinSourceCard	:	0
MaxSourceCard	:	N

4. Bibliography

[B1] Asset Library Open Architecture Framework (ALOAF), v1.2, Contract No. F19628-88-D0031, Publication No. GR-07670-1317, prepared for Electronic Systems Division, Air Force Systems Command, USAF, by the Boeing Company, IBM Federal Sector Division, and Unisys Defense Systems, Inc., August 1992.

[B2] CASE Data Interchange Format (CDIF)—Framework for Modeling and Extensibility, EIA/IS-81, July 1991.

[B3] Repository Guidelines and Standards for the Software Technology for Adaptable, Reliable Systems, Contract No. F19628-88-D-0032, CDRL No.0460, prepared for Electronic Systems Division, Air Force Systems Command, USAF, by IBM Systems Integration Division, March 1989.

[B4] RIG Glossary, RTR-0001, Reuse Library Interoperability Group, April, 1993.

Annex A
Background

(informative)

The RIG is a volunteer, consensus-based organization composed of members from government, academia, and private industry. Its job is to draft standards for the interoperability of reuse libraries in the areas such as nomenclature, interchange protocols, and software component exchange formats. The RIG's proposed standards will be submitted to IEEE, ANSI, or other standards organizations.

The technical committees of the RIG are investigating how components represented in one library's mechanism can be used by other libraries that have a different mechanism. These groups are examining the types of data required to achieve a basic level of expressibility, a standard means of describing library data models, and the best method for transmitting this data. Such information could support exchange of assets between cooperating libraries, and allow users of one library (using the browsing paradigm of that library) to interact with, view, and retrieve components from another remote library as if the remote library were an extension of the user's library.

The technical approach currently under investigation by the RIG is based on the three layer data model described in [B1] that includes the meta-model layer, the model layer, and data layer. The meta-model provides a set of basic constructs and rules that are used in the creation and modification of data models. The data model describes the structure of the data maintained on assets within the library system. The data layer is the actual data that is organized by and conforms to the data model. The RIG is developing possible standards for the meta-model layer and the data model layer.

Data models of existing library systems vary from one another, and will likely continue to vary, if for no other reason than to capture domain-specific characteristics of the assets in their respective libraries. The variation between library data models and the methods used to describe them make the exchange of library data between libraries difficult, reducing interoperability. A possible solution is a meta-model layer for individual libraries to define their library data model.

Technical Committee 2 (TC2) of the RIG has developed the Basic Interoperability Data Model (BIDM). The purpose of the BIDM is to define the minimal set of information that reuse libraries should be able to exchange about assets in order to interoperate. TC2 views the minimal set of information as that information that would enable reuse library users to make quick, intelligent decisions about which assets in other reuse libraries will likely meet their needs.

The BIDM described in this standard represents one of many steps toward evolving a solution to the reuse library interoperability problem. The RIG, through its members and technical committees, is committed to taking these steps and developing draft standards that will enable reuse libraries to interoperate.