erwin Data Modeler

Metamodel Overview

Release 12.0

Legal Notices

This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by Quest Software, Inc and/or its affiliates at any time. This Documentation is proprietary information of Quest Software, Inc and/or its affiliates and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Quest Software, Inc and/or its affiliates

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Quest Software, Inc and/or its affiliates copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Quest Software, Inc and/or its affiliates that all copies and partial copies of the Documentation have been returned to Quest Software, Inc and/or its affiliates or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, QUEST SOFTWARE, INC. PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL QUEST SOFTWARE, INC. BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF QUEST SOFTWARE, INC. IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Quest Software, Inc and/or its affiliates.

Contact erwin

Understanding your Support

Review [support maintenance programs and offerings](#).

Registering for Support

Access the [erwin support](#) site and click **Sign in** or **Sign up** to register for product support.

Accessing Technical Support

For your convenience, erwin provides easy access to "One Stop" support for all editions of [erwin Data Modeler](#), and includes the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- erwin Support policies and guidelines
- Other helpful resources appropriate for your product

For information about other erwin products, visit http://erwin.com/products.

Provide Feedback

If you have comments or questions, or feedback about erwin product documentation, you can send a message to techpubs@erwin.com.

erwin Data Modeler News and Events

Visit www.erwin.com to get up-to-date news, announcements, and events. View video demos and read up on customer success stories and articles by industry experts.

Contents

Diagrams

This section includes diagrams that illustrate some of the key portions of the metamodel. These diagrams are provided as a visual counterpart to the textual descriptions of the metadata.

The diagrams are presented in pairs, one using UML notation and the other using IDEF1X notation. The UML notation presents a raw view of the metamodel; the IDEF1X notation presents the view exposed using the ODBC interface.

> Some of the diagrams include abstract object types. Abstract object types are colored light blue in the diagrams. Although the definitions of these object types exist in the metamodel (termed 'M1'), no instances of them exist within any data model (termed 'M0'). Because no instances exist in any data model, abstract object types are unavailable for M0 queries using the ODBC inter-face.

Some of the IDEF1X diagrams show vector properties (properties with more than one value). Vector properties are shown as tables colored orange. For more information about the trans-formations between the raw metamodel view and the ODBC metamodel view, see the *erwin   Data Modeler ODBC Reporting Guide*.

This section contains the following topics

Entity and View Diagram

Model and Subject Area Diagram

Entity, Attribute, and Key_Group Diagram

Entity, Attribute, Key_Group, and Relationship Diagram

Attribute, Domain, Default, and Validation Rule Diagram

Entity and View Diagram

This diagram shows the object type structure for Entities, Views, and Cached Views. This diagram makes the following assertions:
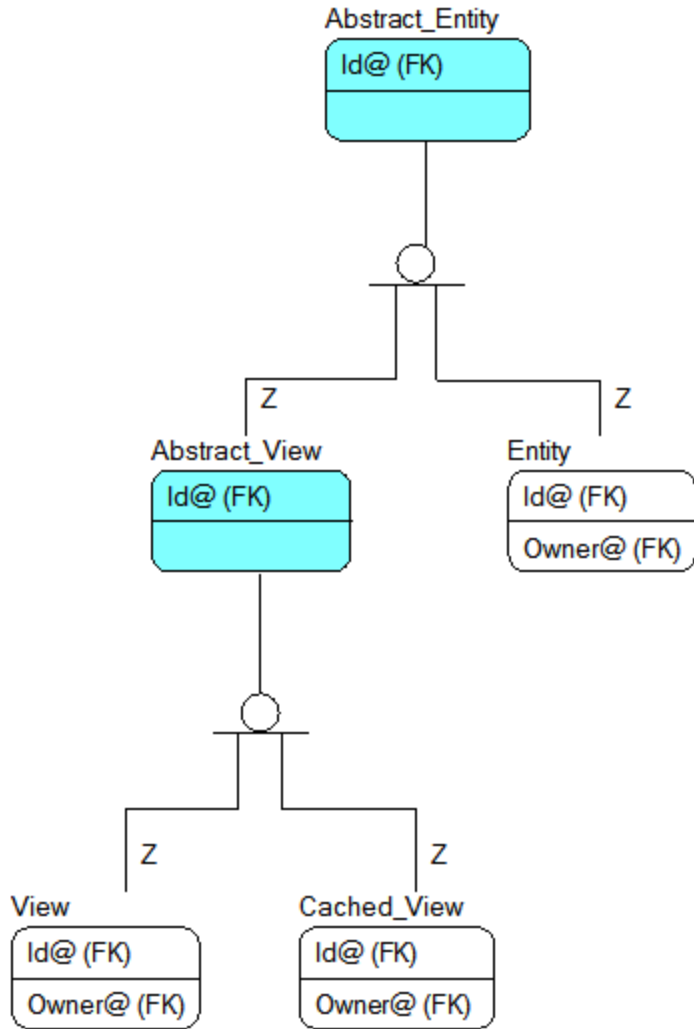
- *Entity* is a type of *Abstract_Entity*

- *Abstract_View* is a type of *Abstract_Entity*

- *Cached_View* is a type of *Abstract_Entity*

- *View* is a type of *Abstract_Entity*

Entity and View Diagram--UML Notation:
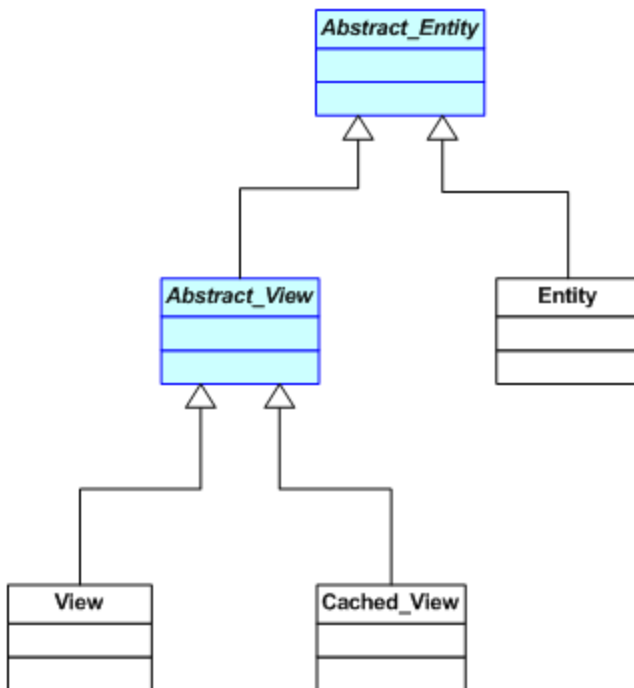
Entity and View Diagram--IDEF1X Notation:

Model and Subject Area Diagram

This diagram shows the storage of Subject Area membership information and makes the following assertions:

- *Abstract_ERwin_Object* is a type of *Abstract_Object*

- The object types *Abstract_Model_Node*, *Model*, *Subject_Area*, and *Relationship* are all types of *Abstract_Objects*

- The object types *Subtype_Symbol* and *Abstract_Entity* are both types of *Abstract_Model_Nodes*

- *Abstract_Model_Nodes* are owned by *Model*

- *Relationships* are owned by *Model*

- *Subject_Areas* are owned by *Model*

- *Subject_Areas* hold their collection of member *Abstract_Model_Nodes* in *Referenced_Entities_Ref*. When viewed using SCAPI, this is held as a vector reference property on the *Subject_Area*. When viewed using ODBC, the associative table *Referenced_Entities_Ref* stores this membership information.

- *Subject_Areas* hold their collection of member *Relationships* in *Referenced_Relationships_Ref*. When viewed using SCAPI this is held as a vector reference property on the *Subject_Area*. When viewed using ODBC, the associative table *Referenced_Relationships_Ref* stores the membership information.

> This data is maintained by erwin   Data Modeler based on the data in *Referenced_Entities_Ref*. Do not attempt to directly change this data.

Model and Subject Area Diagram--UML Notation:
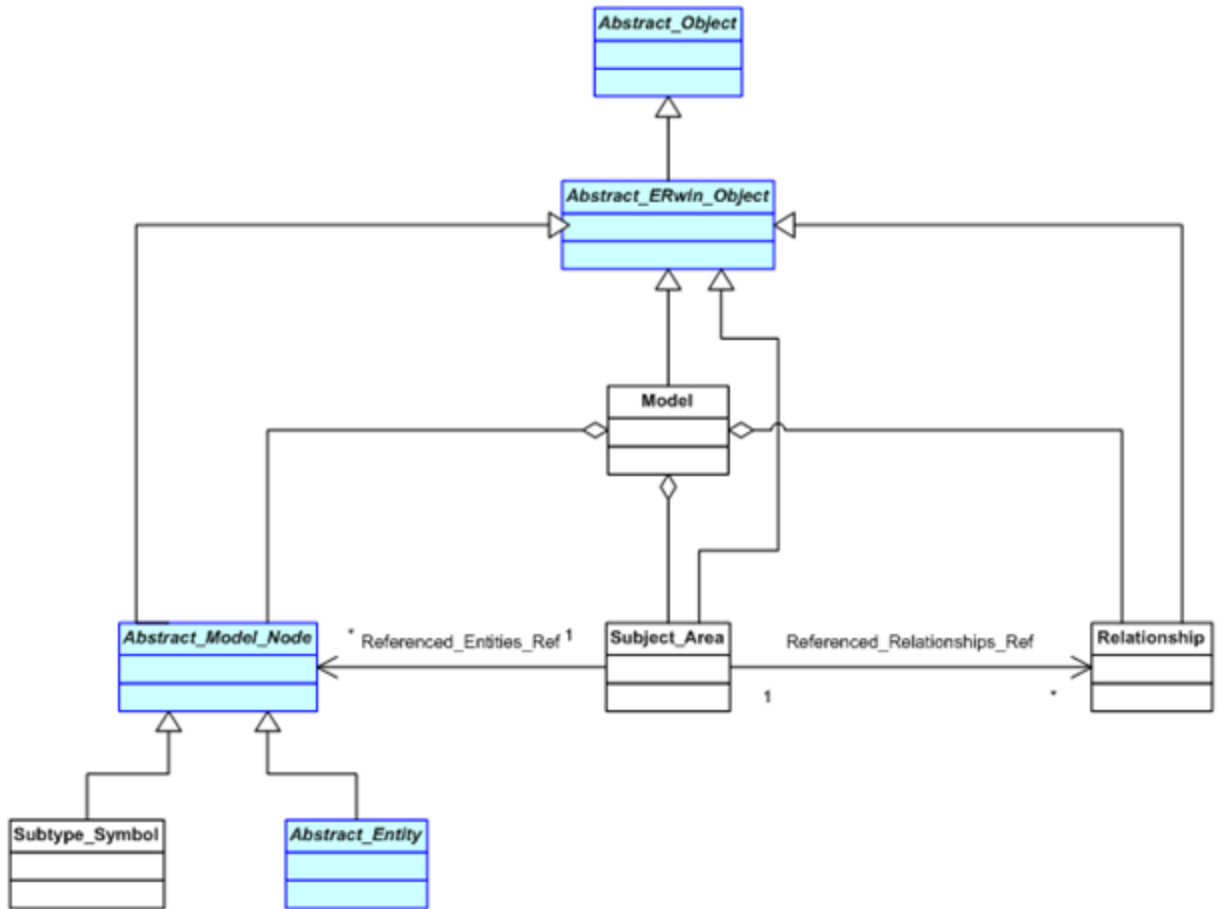
Model and Subject Area Diagram--IDEF1X Notation:

Entity, Attribute, and Key_Group Diagram

This diagram shows the structure of Entities and Attributes and their associated Key_Groups and Key_Group_Members. This diagram makes the following assertions:

- *Attributes* are owned by *Entities*

- *Key_Groups* are owned by *Entities*

- *Key_Group_Members* are owned by *Key_Groups*

- *Key_Group_Members* are linked to their associated *Attribute* using the *Attribute_Ref* property

- *Key_Group_Members* are linked to their parent *Key_Group_Member* using the *Parent_Key_Group_Member_Ref* property

- *Attributes* are linked to their parent *Attribute* using the *Parent_Attribute_Ref* property

Entity, Attribute, and Key_Group Diagram--UML Notation:
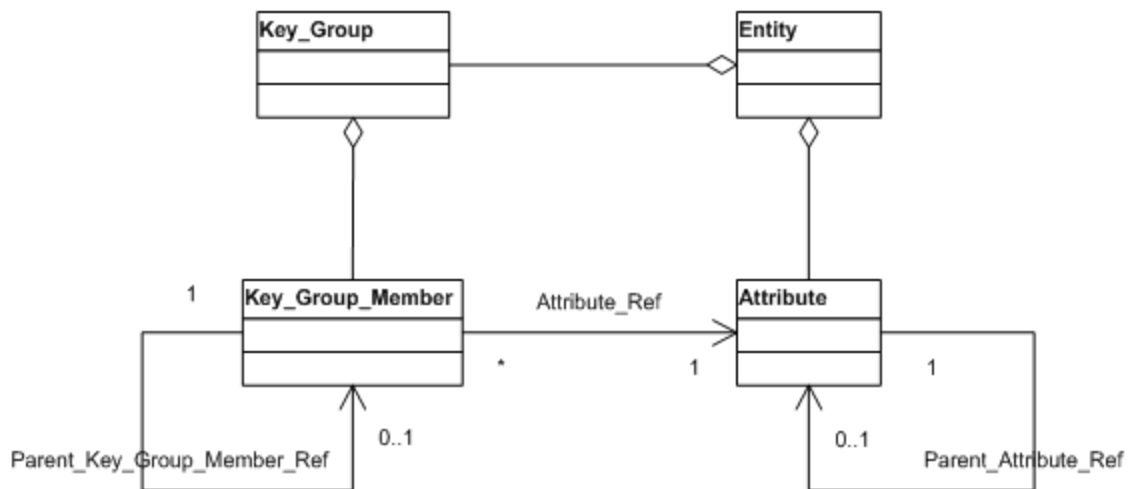
Entity, Attribute, and Key_Group Diagram--IDEF1X Notation:

Entity, Attribute, and Key_Group Diagram
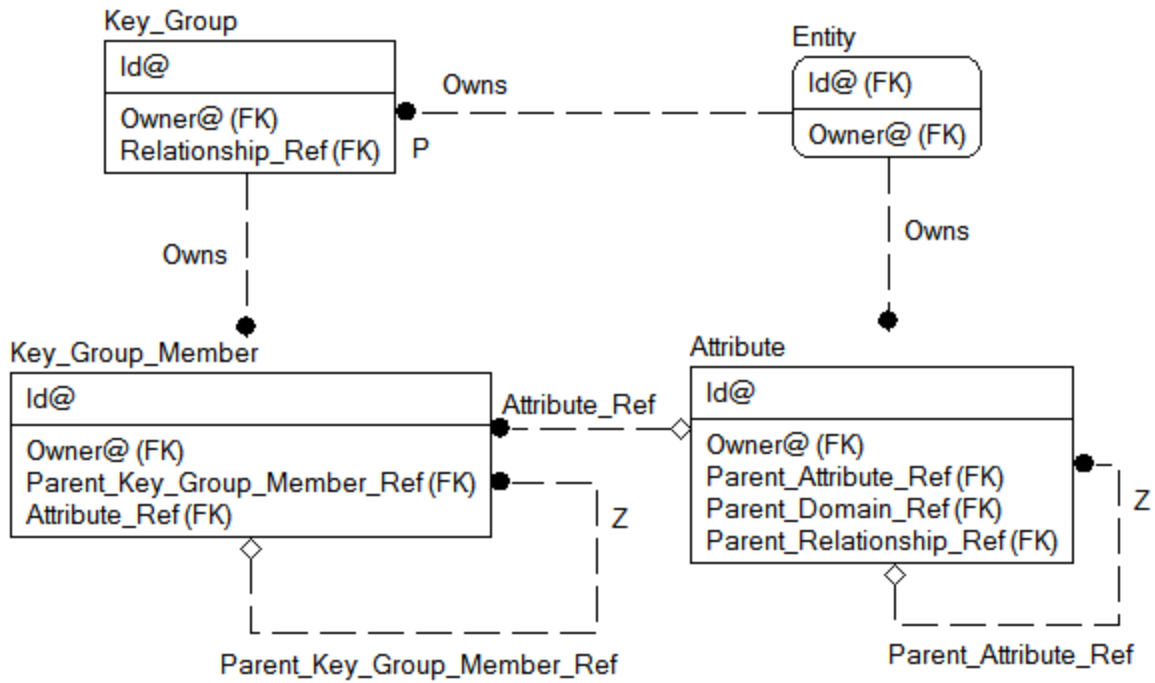
Entity, Attribute, Key_Group, and Relationship Diagram

This diagram extends the prior diagram by adding Relationships. The following assertions are added:

- *Key_Groups* reference their associated *Relationships* using the *Key_Group_Ref* property

- *Relationships* reference their associated *Key_Group* using the *Key_Group_Ref* property

- *Relationships* reference their parent *Entity* using the *Parent_Entity_Ref* property

- *Relationships* reference their child *Entity* using the *Child_Entity_Ref* property

- The collection of *Relationships* in which an *Entity* participates as the parent is stored in the *Parent_Relationships_Ref* property. When viewed using SCAPI, this is held as a vector reference property on *Entity*. When viewed using ODBC, the associative table *Parent_Relationships_Ref* stores this information.

- The collection of *Relationships* in which an *Entity* participates as the child is stored in the *Child_Relationships_Ref* property. When viewed using SCAPI, this is held as a vector reference property on *Entity*. When viewed using ODBC, the associative table *Child_Relationships_Ref* stores this information.

- *Attributes* reference their parent *Attribute* using the *Parent_Relationship_Ref* property.

Entity, Attribute, Key_Group, and Relationship Diagram--UML Notation:
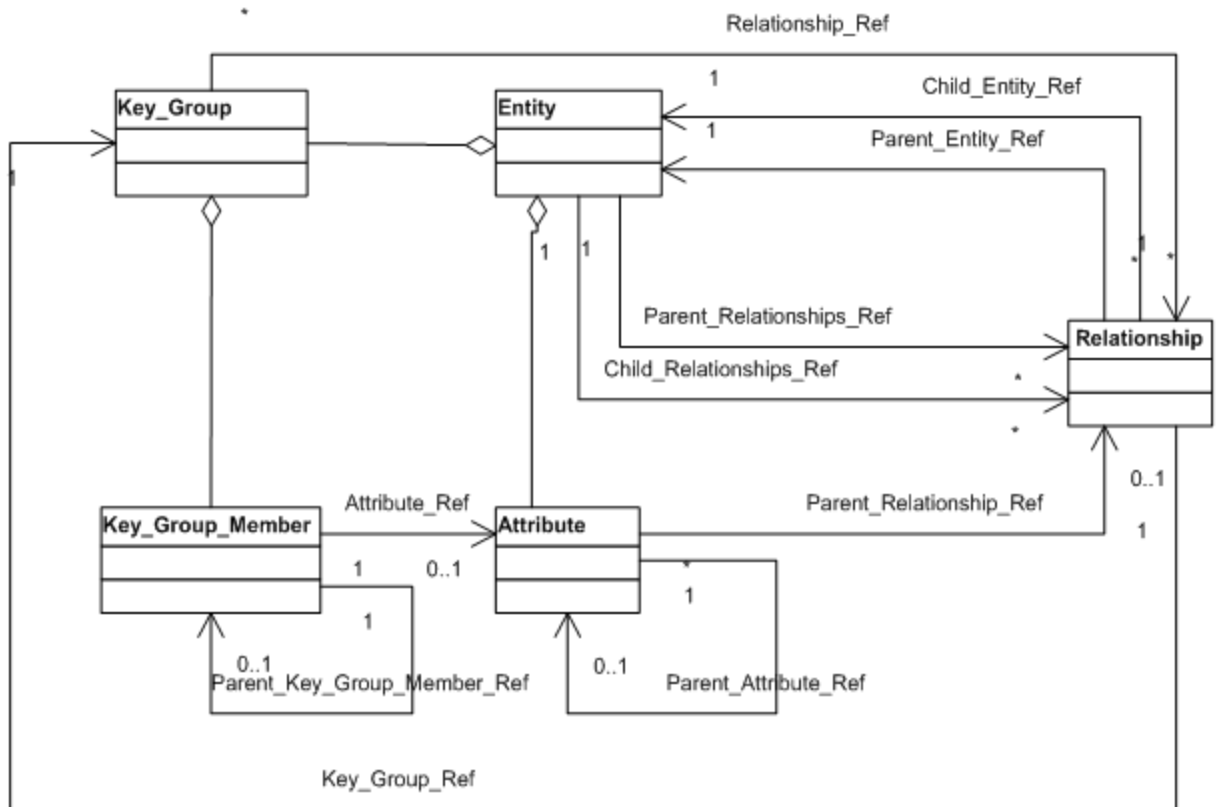
Entity, Attribute, Key_Group, and Relationship Diagram--IDEF1X Notation:

Entity, Attribute, Key_Group, and Relationship Diagram

Key_Group_Ref

Key_Group

Id@
Owner@ (FK) Relationship_Ref (FK)

Owns — P

Entity

Id@ (FK)
Owner@ (FK)

Child_Entity_Ref

Parent_Entity_Ref

Owns

Owns

Parent_Relationships_Ref

Id@ (FK) Value@ (FK)

1   1   1

Relationship

Id@
Owner@ (FK) Parent_Entity_Ref (FK) Child_Entity_Ref (FK) Key_Group_Ref (FK)

Child_Relationships_Ref

Id@ (FK) Value@ (FK)

Relationship_Ref

Key_Group_Member

Id@
Owner@ (FK) Parent_Key_Group_Member_Ref (FK) Attribute_Ref (FK)

Attribute_Ref

Attribute

Id@
Owner@ (FK) Parent_Attribute_Ref (FK) Parent_Domain_Ref (FK) Parent_Relationship_Ref (FK)

Parent_Relationship_Ref

Z

Z

Z

Parent_Key_Group_Member_Ref

Parent_Attribute_Ref

Attribute, Domain, Default, and Validation Rule Diagram

This diagram shows the links between Attributes, Domains, Defaults, and Validation_Rules. This diagram makes the following assertions:

▪ *Attributes* reference their parent *Attribute* using the *Parent_Attribute_Ref* property

▪ *Attributes* reference their parent *Domain* using the *Parent_Domain_Ref* property

▪ *Domains* reference their parent *Domain* using the *Parent_Domain_Ref* property

▪ *Default_Constraint_Usage* objects can be owned by *Attributes* or *Domains*

▪ *Default_Constraint_Usage* objects reference their parent using the *Parent_Default_ Constraint_Usage_Ref* property

▪ *Default_Constraint_Usage* objects reference their associated *Default* object using the *Default_Ref* property

▪ *Check_Constraint_Usage* objects can be owned by *Attributes* or *Domains*

▪ *Check_Constraint_Usage* objects reference their parent using the *Parent_Check_Con- straint_Usage_Ref* property

▪ *Check_Constraint_Usage* objects reference their associated *Validation_Rule* object using the *Validation_Rule_Ref* property

Attribute, Domain, Default, and Validation Rule Diagram--UML Notation:

Attribute, Domain, Default, and Validation Rule Diagram

Parent_Attribute_Ref
1

Parent_Domain_Ref
1

Attribute

Parent_Domain_Ref

Domain

0..1

0..1

*

0..1

Default_Constraint_Usage

Check_Constraint_Usage

1

1

*

0..1
Parent_Default_Constraint_Usage_Ref

*

0..1
Parent_Check_Constraint_Usage_Ref

Default_Ref

Validation_Rule_Ref

0..1

0..1

Default
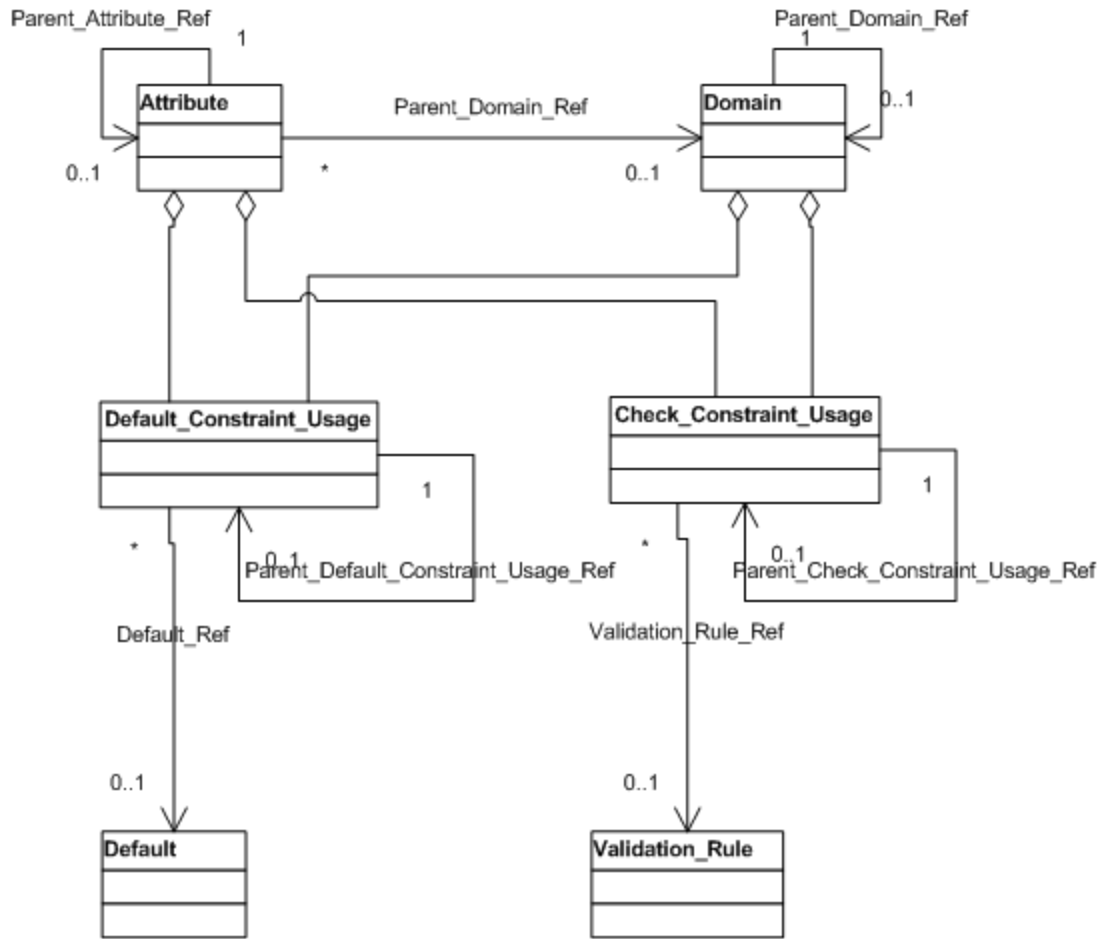
Validation_Rule

Attribute, Domain, Default, and Validation Rule Diagram--IDEF1X Notation:

Attribute, Domain, Default, and Validation Rule Diagram

Attribute

Id@
Owner@ (FK) Parent_Attribute_Ref (FK) Parent_Domain_Ref (FK) Parent_Relationship_Ref (FK)

Domain

Id@
Owner@ (FK) Parent_Domain_Ref (FK)

Parent_Domain_Ref

Z Parent_Attribute_Ref

Z

Z

Parent_Domain_Ref

Owns

Owns

Owns

Owns

Owns

Default_Constraint_Usage

Id@
Owner@ (FK) Parent_Default_Constraint_Usage_Ref (FK) Default_Ref (FK)

Z

Z

Parent_Default_Constraint_Usage_Ref

Default_Ref

Check_Constraint_Usage

Id@
Owner@ (FK) Parent_Check_Constraint_Usage_Ref (FK) Validation_Rule_Ref (FK)

Z

Z

Parent_Check_Constraint_Usage_Ref

Validation_Rule_Ref

Default

Id@
Owner@ (FK)

Validation_Rule

Id@
Owner@ (FK)