

## METAMODEL FOR INFORMATION MODELS

### INTRODUCTION

The metamodel that follows covers the key concepts of the following information modelling methods:

- The notation developed by Harry Ellis and Richard Barker and later adopted by Oracle for its CASE\*Method and by the UK's CCTA for SSADM (Structured Systems Analysis and Design Method).
- IDEF1X, a US standard.
- Information Engineering, as described by James Martin and Clive Finkelstein.
- Entity-Relationship Modelling, as described by Peter Chen.
- Express-G, an ISO standard.
- UML Class Diagrams.

The Ellis-Barker notation does not show foreign keys. It does not recognise the concept of a primary key. Instead it recognises the concept of a unique identifier, where each element of the identifier is either an attribute or a relationship. Two link phrases are used to name relationships. It includes the 'exclusive arc' concept to represent mutually exclusive relationships. It only permits one specialisation hierarchy per entity type, although there may then be further specialisations of each of the subtypes. All subtypes must be mutually exclusive and their instances must form the complete set of the supertype.

IDEF1X includes all foreign keys, and where these foreign keys form part of the primary key the resulting relationship is known as an identifying relationship. Relationships are named with a single name. There is no equivalent of the 'exclusive arc'. It allows multiple specialisation hierarchies for any one entity type, and each hierarchy may be complete or incomplete. Each specialisation hierarchy has an attribute which is the category discriminator.

There are many 'flavours' of Information Engineering but generally they exhibit a mixture of the concepts from the Ellis-Barker notation and IDEF1X.

Chen's Entity-Relationship Modelling had a separate notation for relationships and allowed n-ary relationships. Some relationships are allowed attributes, which would need to be considered as 'entity types' within this metamodel.

Although an Express-G diagram looks very different to other modelling notations, its concepts are very similar. The only notable exception to this is the ability to record whether a 'many' relationship is a set, a bag, a list or an array.

A UML Class Diagram is often used to express business information requirements, with the object class being the equivalent of the entity type. There is no equivalent of the unique identifier or primary key, but there are special notations for composition and aggregation. Object classes can 'play roles' with associations (relationships). Later versions include a notation for the exclusivity of associations. Multiple class specialisation hierarchies are allowed. Each hierarchy may be complete or incomplete and the members of the hierarchy may be mutually exclusive or may overlap. The Information Management Metamodel (IMM) Specification, currently under development by OMG, will introduce an additional notation for a conceptual (or platform independent) model.

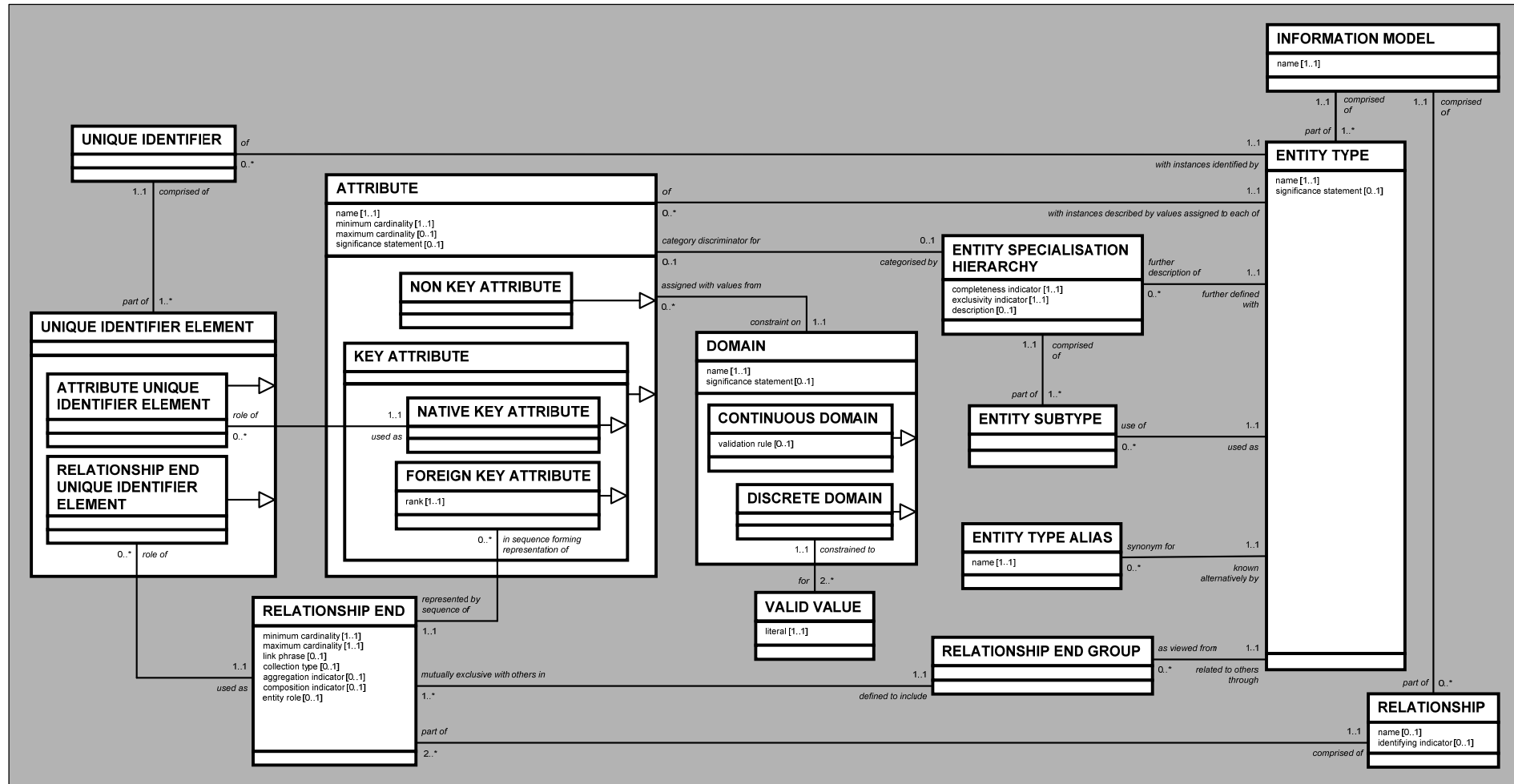
Database schemas were determined to be out of scope. At the November 2010 Interim Meeting ORM (Object-Role Modelling) was considered to be in scope but it is unclear how it can be accommodated in this model.

The metamodel diagram is drawn using the draft OMG notation for UML conceptual models.

All comments or contributions welcome.

*Keith Gordon*  
January 2011

# THE METAMODEL DIAGRAM



## **SPECIFICATION OF THE METAMODEL**

### **1. Object Class Specifications**

#### **1.1. Information Model**

##### **1.1.1. Description**

Each Information Model is a set of definitions for things of significance to an organisation, about which information needs to be held, and the relationships between them.

##### **1.1.2. Alias**

Data Model, Conceptual Data Model, Logical Data Model, Entity Relationship Model, Object Class Diagram

##### **1.1.3. Superclass**

Model (from MFI Core)

##### **1.1.4. Subclasses**

None

##### **1.1.5. Attributes**

###### **1.1.5.1. name**

Definition: A unique name by which this model is known.

Multiplicity: [1..1]

#### **1.2. Entity Type**

##### **1.2.1. Description**

Each Entity is a concept or thing of significance about which information may need to be held in support of business operations.

##### **1.2.2. Alias**

Entity, Object Class

##### **1.2.3. Superclass**

Model Component (from MFI Core)

##### **1.2.4. Subclasses**

None

##### **1.2.5. Attributes**

###### **1.2.5.1. name**

Definition: A unique name by which this Entity Type is primarily known.

Multiplicity: [1..1]

### **1.2.5.2. significance statement**

Definition: A statement that explains the significance of this Entity Type to the business and or organisation that is the subject of this Information Model. May also be known as a description.

Multiplicity: [0..1]

## **1.3. Entity Type Alias**

### **1.3.1. Description of *Entity Type Alias***

Each Entity Type Alias is one of a set of alternative names for an Entity Type.

### **1.3.2. Alias**

Entity Alias, Entity Type Synonym, Entity Synonym, Object Class Alias, Object Class Synonym

### **1.3.3. Superclass**

Model Component (from MFI Core)

### **1.3.4. Subclasses**

None

### **1.3.5. Attributes**

#### **1.3.5.1. name**

Definition: A name by which the associated Entity Type is alternatively known.

Multiplicity: [1..1]

## **1.4. Entity Specialisation Hierarchy**

### **1.4.1. Description**

Each Entity Specialisation Hierarchy is a means by which an Entity Type may be classified.

### **1.4.2. Alias**

Entity Generalisation Hierarchy, Entity Subtype Hierarchy, Entity Type Classification, Entity Classification

### **1.4.3. Superclass**

Model Component (from MFI Core)

### **1.4.4. Subclasses**

None

### **1.4.5. Attributes**

#### **1.4.5.1. completeness indicator**

Definition: An indicator that specifies whether the instances of the associated Entity Subtypes that form this particular Entity Specialisation Hierarchy are the complete set of the instances of the Entity Type that is the supertype or not.

Multiplicity: [1..1]

#### **1.4.5.2. exclusivity indicator**

Definition: An indicator that specifies whether the instances of the associated Entity Subtypes that form this particular Entity Specialisation Hierarchy are mutually exclusive or not.

Multiplicity: [1..1]

#### **1.4.5.3. description**

Definition: A statement that describes the purpose or the classification of this particular Entity Specialisation Hierarchy.

Multiplicity: [0..1]

### **1.5. Entity Subtype**

#### **1.5.1. Description**

Each Entity Subtype is representation of a subset of the instances of an Entity Type (the supertype) that share common Attributes and/or Relationships distinct from other subsets.

#### **1.5.2. Alias**

Subtype, Object Subclass, Subclass

#### **1.5.3. Superclass**

Model Component (from MFI Core)

#### **1.5.4. Subclasses**

None

#### **1.5.5. Attributes**

None

### **1.6. Attribute**

#### **1.6.1. Description**

Each Attribute is detail that serves to qualify, identify, classify, quantify or express the state of an instance of an Entity Type.

#### **1.6.2. Alias**

None

#### **1.6.3. Superclass**

Model Component (from MFI Core)

#### **1.6.4. Subclasses**

Key Attribute, Non Key Attribute

## **1.6.5. Attributes**

### **1.6.5.1. name**

Definition: A unique name for this Attribute. In some methods this will be unique within the Entity Type (in which case the Entity Type Name must be concatenated with the Attribute Name to gain model uniqueness) whilst in other methods this will be unique within the Information Model.

Multiplicity: [1..1]

### **1.6.5.2. minimum cardinality**

Definition: A statement of the minimum number of occurrences of values of this Attribute for any particular instance of the associated Entity Type. In most circumstances this will be '0' (indicating that the Attribute is optional) or '1' (indicating that the Attribute is mandatory).

Multiplicity: [1..1]

### **1.6.5.3. maximum cardinality**

Definition: A statement of the maximum number of occurrences of values of this Attribute for any particular instance of the associated Entity Type. In most methods this is not specified.

Multiplicity: [0..1]

### **1.6.5.4. significance statement**

Definition: A statement that explains the significance of this Attribute to the business and or organisation that is the subject of this Information Model. May also be known as a description.

Multiplicity: [0..1]

## **1.7. Key Attribute**

### **1.7.1. Description**

Each Key Attribute is an Attribute whose value contributes in some way to the identification of individual instances of the host Entity Type or of some related Entity Type.

### **1.7.2. Alias**

None

### **1.7.3. Superclass**

Attribute

### **1.7.4. Subclasses**

Native Key Attribute, Foreign Key Attribute

### **1.7.5. Attributes**

None

## **1.8. Native Key Attribute**

### **1.8.1. Description**

Each Native Key Attribute is a Key Attribute whose value contributes in some way to the identification of individual instances of the host Entity Type.

### **1.8.2. Alias**

None

### **1.8.3. Superclass**

Key Attribute

### **1.8.4. Subclasses**

None

### **1.8.5. Attributes**

None

## **1.9. Foreign Key Attribute**

### **1.9.1. Description**

Each Foreign Key Attribute is a Key Attribute whose value contributes in some way to the identification of the one related instance of that Entity Type involved in the associated Relationship End. It provides or contributes to an alternative representation of the relationship concerned. Its value must be drawn from the Domain of the corresponding Key Attribute of the related Entity Type.

### **1.9.2. Alias**

None

### **1.9.3. Superclass**

Key Attribute

### **1.9.4. Subclasses**

None

### **1.9.5. Attributes**

#### **1.9.5.1. rank**

Definition: A statement of the position of this Foreign Key Attribute in the foreign key of which it is a part.

Multiplicity: [1..1]

## **1.10. Non Key Attribute**

### **1.10.1. Description**

Each Non Key Attribute is an Attribute whose value is fully independent of all Relationships or other Attributes.

### **1.10.2. Alias**

None

### **1.10.3. Superclass**

Attribute

### **1.10.4. Subclasses**

None

### **1.10.5. Attributes**

None

## **1.11. Domain**

### **1.11.1. Description**

Each Domain is a pool of values from which an Attribute must take its value. A Domain provides a set of business validation rules, format constraints and other properties for one or more Attributes.

### **1.11.2. Alias**

None

### **1.11.3. Superclass**

*[To be determined]*

### **1.11.4. Subclasses**

Continuous Domain, Discrete Domain

### **1.11.5. Attributes**

#### **1.11.5.1. name**

Definition: A unique name by which this Domain is known.

Multiplicity: [1..1]

#### **1.11.5.2. significance statement**

Definition: A statement that explains the significance of this Domain to the business and or organisation that is the subject of this Information Model. May also be known as a description.

Multiplicity: [0..1]

## **1.12. Continuous Domain**

### **1.12.1. Description**

Each Continuous Domain is a Domain whose values are not drawn from an explicit list of Valid Values

### **1.12.2. Alias**

Non-enumerated Domain

### **1.12.3. Superclass**

Domain

### **1.12.4. Subclasses**

None



### **1.12.5. Attributes**

#### **1.12.5.1. validation rule**

Definition: A statement of the validation that may be applied to this domain. At its simplest it may just be a statement of the data type that may be applied to Attributes. It might show upper and lower bounds of a range of values. It might be a 'format mask'. Or, it may be any combination of these.

Multiplicity: [1..1]

### **1.13. Discrete Domain**

#### **1.13.1. Description**

Each Discrete Domain is a Domain whose permitted values consist of an explicit list of Valid Values.

#### **1.13.2. Alias**

Enumerated Domain

#### **1.13.3. Superclass**

Domain

#### **1.13.4. Subclasses**

None

#### **1.13.5. Attributes**

None

### **1.14. Valid Value**

#### **1.14.1. Description**

Each Valid Value is one of the explicit set of permitted values that comprise a Discrete Domain.

#### **1.14.2. Alias**

Permitted Value

#### **1.14.3. Superclass**

*[To be determined]*

#### **1.14.4. Subclasses**

None

#### **1.14.5. Attributes**

##### **1.14.5.1. literal**

Definition: This is the actual permitted value. In a conceptual (or platform independent) model it will probably be the concept, for example, 'Male'. In a logical (or platform specific) model it will probably be the code, for example, 'M' or '0'.

Multiplicity: [1..1]

## **1.15. Relationship**

### **1.15.1. Description**

Each Relationship is an association between two or more Entity Types, or between one Entity Type and itself.

### **1.15.2. Alias**

Association

### **1.15.3. Superclass**

Model Component (from MFI Core)

### **1.15.4. Subclasses**

None

### **1.15.5. Attributes**

#### **1.15.5.1.name**

Definition: A unique name by which this Relationship is known. Some methods do not provide such a name.

Multiplicity: [0..1]

#### **1.15.5.2. identifying indicator**

Definition: If this relationship is a binary 'one-to-many' Relationship, an indicator that specifies whether this Relationship provides part (or all) of the primary Unique Identifier for the Entity Type that is at the 'many' end of the Relationship or not. Not all methods recognise this concept.

Multiplicity: [0..1]

## **1.16. Relationship End**

### **1.16.1. Description**

Each Relationship End is part of the definition of a Relationship as seen from a given Entity Type (the host).

### **1.16.2. Alias**

Association End

### **1.16.3. Superclass**

Model Component (from MFI Core)

### **1.16.4. Subclasses**

None

### **Attributes**

#### **1.16.4.1.minimum cardinality**

Definition: A statement of the minimum number of instances of the associated Entity Type (through the associated Relationship End Group) that must participate in the Relationship of which this Relationship End is a part. In most circumstances this will be '0' (indicating that the Entity Type has optional participation) or '1' (indicating that the Entity Type has mandatory participation).

Multiplicity: [1..1]

#### **1.16.4.2. maximum cardinality**

Definition: A statement of the maximum number of instances of the associated Entity Type (through the associated Relationship End Group) that may participate in the Relationship of which this Relationship End is a part. In most circumstances this will be '1' (indicating that one and only one Entity Type may participate) or '\*' (indicating that an unspecified number of Entity Types may participate).

Multiplicity: [1..1]

#### **1.16.4.3. link phrase**

Definition: A statement that explains the nature of the Relationship of which this Relationship End is a part from the perspective of the associated Entity Type (through the associated Relationship End Group). This is normally expressed in business terms. Not all methods recognise this concept.

Multiplicity: [0..1]

#### **1.16.4.4. entity role**

Definition: A statement that explains the role that the associated Entity Type (through the associated Relationship End Group) is playing in the associated Relationship. Not all methods recognise this concept.

Multiplicity: [0..1]

#### **1.16.4.5. collection type**

Definition: A statement as whether the instances of the associated Entity Type (through the associated Relationship End Group) are considered to be a 'set', a 'bag' (or 'multiset'), a 'list' or an 'array'. Most methods do not recognise this concept.

Multiplicity: [0..1]

#### **1.16.4.6. aggregation indicator**

Definition: An indicator that specifies whether the instance of the associated Entity Type (through the associated Relationship End Group) is considered to be an aggregation of the instances of the other Entity Type participating in the Relationship (identified through the associated Relationship and Relationship End Group) or not. Most methods do not recognise this concept.

Multiplicity: [0..1]

#### **1.16.4.7. composition indicator**

Definition: An indicator that specifies whether the instance of the associated Entity Type (through the associated Relationship End Group) is considered to be a composition of the instances of the other Entity Type participating in the Relationship (identified through the associated Relationship and Relationship End Group) or not. Most methods do not recognise this concept.

Multiplicity: [0..1]

### **1.17. Relationship End Group**

#### **1.17.1. Description**

Each Relationship End Group is a statement that links one or more Relationship Ends to their host Entity Type such that they are mutually exclusive. The most common case is where the 'group' comprises just one Relationship End.

#### **1.17.2. Alias**

Association End Group

#### **1.17.3. Superclass**

Model Component (from MFI Core)

#### **1.17.4. Subclasses**

None

#### **1.17.5. Attributes**

None

### **1.18. Unique Identifier**

#### **1.18.1. Description**

Each Unique Identifier is a statement that the values of a specified set of Attributes and/or Relationship Ends are sufficient to uniquely identify an instance of an Entity Type.

#### **1.18.2. Alias**

Key, Unique Key

#### **1.18.3. Superclass**

Model Component (from MFI Core)

#### **1.18.4. Subclasses**

None

#### **1.18.5. Attributes**

None

### **1.19. Unique Identifier Element**

#### **1.19.1. Description**

Each Unique Identifier Element is a statement that a particular Attribute or a particular Relationship End is a part of a particular Unique Identifier.

#### **1.19.2. Alias**

None

#### **1.19.3. Superclass**

Model Component (from MFI Core)

#### **1.19.4. Subclasses**

Attribute Unique Identifier Element, Relationship End Unique Identifier Element

#### **1.19.5. Attributes**

None

### **1.20. Attribute Unique Identifier Element**

#### **1.20.1. Description**

Each Attribute Unique Identifier Element is a Unique Identifier Element that is a statement that a particular Attribute is a part of a particular Unique Identifier.

### **1.20.2. Alias**

None

### **1.20.3. Superclass**

Unique Identifier Element

### **1.20.4. Subclasses**

None

### **1.20.5. Attributes**

None

## **1.21. Relationship End Unique Identifier Element**

### **1.21.1. Description**

Each Relationship End Unique Identifier Element is a Unique Identifier Element that is a statement that a particular Relationship End is a part of a particular Unique Identifier.

### **1.21.2. Alias**

None

### **1.21.3. Superclass**

Unique Identifier Element

### **1.21.4. Subclasses**

None

### **1.21.5. Attributes**

None

## **2. Associations**

### **2.1. Information Model – Entity Type**

Each Information Model is comprised of one or more Entity Types  
Each Entity Type is part of one and only one Information Model

### **2.2. Information Model – Relationship**

Each Information Model is comprised of zero, one or more Relationships  
Each Relationship is part of one and only one Information Model

### **2.3. Entity Type – Entity Type Alias**

Each Entity Type is known alternatively by zero, one or more Entity Type Aliases  
Each Entity Type Alias is synonym for one and only one Entity Type

#### **2.4. Entity Type – Entity Specialisation Hierarchy**

Each Entity Type is further defined with zero, one or more Entity Specialisation Hierarchies  
Each Entity Specialisation Hierarchy is further description of one and only one Entity Type

#### **2.5. Entity Specialisation Hierarchy – Entity Subtype**

Each Entity Specialisation Hierarchy is comprised of one or more Entity Subtypes  
Each Entity Subtype is part of one and only one Entity Specialisation Hierarchy

#### **2.6. Entity Type – Entity Subtype**

Each Entity Type is used as zero, one or more Entity Subtypes  
Each Entity Subtype is use of one and only one Entity Type

#### **2.7. Entity Type – Attribute**

Each Entity Type is with instances described by values designed to each of zero, one or more Attributes  
Each Attribute is of one and only one Entity Type

#### **2.8. Attribute – Entity Specialisation Hierarchy**

Each Attribute is category discriminator for zero or one Entity Specialisation Hierarchy  
Each Entity Specialisation Hierarchy is categorised by zero or one Attribute

#### **2.9. Attribute – Domain**

Each Attribute is assigned with values from one and only one Domain  
Each Domain is constraint on zero, one or more Attribute

#### **2.10. Discrete Domain – Valid Value**

Each Discrete Domain is constrained to two or more Valid Values  
Each Valid Value is for one and only one Discrete Domain

#### **2.11. Relationship – Relationship End**

Each Relationship is comprised of two or more Relationship Ends  
Each Relationship End is part of one and only one Relationship

#### **2.12. Relationship End – Foreign Key Attribute**

Each Relationship End is represented by sequence of zero, one or more Foreign Key Attributes  
Each Foreign Key Attribute is in sequence forming representation of one and only one Relationship End

#### **2.13. Relationship End Group – Relationship End**

Each Relationship End Group is defined to include one or more Relationship Ends  
Each Relationship End is mutually exclusive with others in one and only one Relationship End Group

**2.14. Entity Type – Relationship End Group**

Each Entity Type is related to others through zero, one or more Relationship End Groups  
Each Relationship End Group is as viewed from one and only one Entity Type

**2.15. Entity Type – Unique Identifier**

Each Entity Type is with instances identified by zero, one or more Unique Identifiers  
Each Unique Identifier is of one and only one Entity Type

**2.16. Unique Identifier – Unique Identifier Element**

Each Unique Identifier is comprised of one or more Unique Identifier Elements  
Each Unique Identifier Element is part of one and only one Unique Identifier

**2.17. Attribute Unique Identifier Element – Native Key Attribute**

Each Attribute Unique Identifier Element is role of one and only one Native Key Attribute  
Each Native Key Attribute is used as zero, one or more Attribute Unique Identifier Elements

**2.18. Relationship End Unique Identifier Element – Relationship End**

Each Relationship End Unique Identifier Element is role of one and only one Relationship End  
Each Relationship End is used as zero, one or more Relationship End Unique Identifier Elements