ANNEX I. SUMMARY OF IDEF1X DATA MODELLING METHODOLOGY AND NOTATION

I1 Introduction

I1.1 Whenever data structures and business rules required to support a business area need to be specified, it is convenient to build a data model in order to capture that information. A data model is a description of the organisation of data in a manner that reflects the information structure of an enterprise. It encompasses the entity definitions, relationships, and the integrity constraints through which the information created and used by the functional activity is managed, and from which standard data are created. [DoD 8020.1 1992].

I1.2 Having identified what a data model is, one still needs a structured syntax to begin expressing the information structure of the business. IDEF1X, a methodology created to help design data, provides such a structured environment, with special focus on relational constructs.

I1.3 The following sections provide a brief description of the IDEF1X syntax as discussed in Thomas A. Bruce’s book *Designing Quality Databases with IDEF1X Information Models* [Bruce 1992].

I2 Entities and Attributes

I2.1 An entity is anything about which information is stored in a database. In a conceptual schema language, any concrete or abstract thing of interest, including associations among things.

I2.2 IDEF1X distinguishes between independent and dependent entities. Figure I-1 shows the symbols associated with independent and dependent entities. The kind of information stored in the database is, loosely speaking, the attributes or properties that describe the entity. For instance, if PERSON is an entity in a given data model, then person-name, person-social-security-number, person-address, etc., may all be properties or attributes of that entity for the purposes of that enterprise. Attributes are divided into key-attributes and non-key-attributes, i.e. those used to uniquely identify the entity and those properties of the entity not used for that purpose.

![Figure I-1. IDEF1X Symbols for Independent and Dependent Entities](image)

Note: The area above the line is reserved for the identifying keys.

I2.3 The IDEF1X syntax further categorises attributes according to its diverse uses in either the key-area or the data area of the entity. Table I-1 summarises these different usages.
Table I-1. IDEF1X Attribute Notation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute (FK)</td>
<td>Foreign Key: Primary key of another entity contributed by a relationship</td>
</tr>
<tr>
<td>role.name.attribute (FK)</td>
<td>Role Name: New name for a foreign key connoting its use.</td>
</tr>
<tr>
<td>attribute (AKn)</td>
<td>Alternate Key: Alternate unique identifier of the entity</td>
</tr>
<tr>
<td>attribute (IEn)</td>
<td>Inversion Entry: Non-unique access identifier of the entity</td>
</tr>
<tr>
<td>group.(c1,c2,c3)</td>
<td>Group Attribute: Attribute is a group containing the listed constituents.</td>
</tr>
<tr>
<td>attribute(fk1,fk2,fk3)(FK)</td>
<td>Unified Foreign Key: Listed foreign keys are unified to a single foreign key attribute</td>
</tr>
</tbody>
</table>

I3 Category Notation

I3.1 A data model may contain a series of entities that share one or more attributes. IDEF1X provides a method for aggregating these common attributes into a base entity, while retaining the subtypes with their unique properties. This avoids unnecessary duplication of attributes and helps with the management of the model.

I3.2 Figure I-2 shows the two types of category supported by IDEF1X. If the listing of the subtypes is exhaustive, the category is complete and the double line is used to indicate this fact. If the subtypes depicted are only a fraction of the complete set then the category is incomplete and only one line is used in the symbol. The subtypes of the generic parent inherit all the attributes of that parent, but are not limited to spawning their own unique relationships and subtypes if necessary.

![Figure I-2. IDEF1X Syntax for Entity Categories](image)

I4 Relationship Notation

IDEF1X allows three main types of relationship, namely, identifying relationships, non-identifying relationships and non-specific relationships. (See Figure I-3.)
A further aspect of a relationship is its cardinality. The first two relationships shown in Figure I-3 (above) were one-to-many, that is, where at least one parent entity has zero or more child entities associated to it. There are, however, situations in which zero or one parent entity may have zero or more child entities associated to it, or where it is guaranteed that there is either at least one parent or one child present in the relationship in combination with zero or more of the other kind. Figure I-4 depicts all these combinations diagrammatically.
**Figure I-4. IDEF1X Cardinality Notation**

**Mandatory Relationships**
- One-to-zero-or-more
- One-to-one-or-more
- One-to-zero-or-one
- One-to-exactly-N

**Optional Relationships**
- Zero-to-zero-or-more
- Zero-to-one-or-more
- Zero-to-zero-or-one
- Zero-to-exactly-N