BDA Data Architecture Modelling Standards

Version: 13.0

Issue Date: 19/12/2016

Document Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Issue Date</th>
<th>Modified by</th>
<th>BDA Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>Draft</td>
<td>18/5/2012</td>
<td>Stephen Palmer</td>
<td></td>
<td>Next version following release of previous version with BDA</td>
</tr>
<tr>
<td>3.0</td>
<td>Draft</td>
<td>28/1/2013</td>
<td>Stephen Palmer</td>
<td></td>
<td>New version following release of version 4 Business Data Architecture</td>
</tr>
<tr>
<td>4.0</td>
<td>Draft</td>
<td>3/7/2013</td>
<td>Stephen Palmer</td>
<td></td>
<td>Modified to remove copyright material</td>
</tr>
<tr>
<td>5.0</td>
<td>Final</td>
<td>20/09/2013</td>
<td>Martyn Grandy</td>
<td></td>
<td>Transferred into new ISB document template + minor editorial changes.</td>
</tr>
<tr>
<td>6.0</td>
<td>Final</td>
<td>04/06/2014</td>
<td>Stephen Palmer</td>
<td></td>
<td>Updated version for new release of the Business Data Architecture</td>
</tr>
<tr>
<td>7.0</td>
<td>Final</td>
<td>08/12/2014</td>
<td>Stephen Palmer</td>
<td></td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1) Standardise terms used and to provide additional information on the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>standards used</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2) Data Types handling modifications</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3) Subject areas changes to minimise Change control impact on associated BDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4) Change Register</td>
</tr>
<tr>
<td>Version</td>
<td>Stage</td>
<td>Date</td>
<td>Author</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td>Final</td>
<td>09/03/2015</td>
<td>Stephen Palmer</td>
<td>Updated for version 9 BDA.</td>
<td></td>
</tr>
<tr>
<td>9.0</td>
<td>Final</td>
<td>27/03/2015</td>
<td>Stephen Palmer</td>
<td>Updated to 1. Reflect modifications to abbreviations list 2. Amend Data Model and XSD Change Management section to reflect introduction of an Access Data Base to replace previous change log</td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td>Final</td>
<td>10/07/2015</td>
<td>Stephen Palmer</td>
<td>Updated to 1. Cover Aggregated Data Standards 2. Include Primary Key rolenameing 3. Additional abbreviations</td>
<td></td>
</tr>
<tr>
<td>11.0</td>
<td>Final</td>
<td>26/10/2015</td>
<td>Stephen Palmer</td>
<td>Updated to cover: 1. Additional abbreviations 2. Minor formatting and text changes</td>
<td></td>
</tr>
<tr>
<td>12.0</td>
<td>Final</td>
<td>26/04/2016</td>
<td>Stephen Palmer</td>
<td>Updated to cover: 1. Additional abbreviations 2. Removal of rolename and original attribute display</td>
<td></td>
</tr>
<tr>
<td>13.0</td>
<td>Final</td>
<td>19/12/2016</td>
<td>Stephen Palmer</td>
<td>Updated to cover 1) upgrade from Erwin 7.3.12 to Erwin 9.6 2) inclusion of summary model and conceptual model to facilitate automation of Erwin to ISB web site production 3) Embedding Controlled List values into Erwin data model</td>
<td></td>
</tr>
</tbody>
</table>
Contents

1 Introduction .................................................................................................................. 5

2 Data Model Types ...................................................................................................... 5

3 Business Data Architecture Definitions .................................................................... 7
   3.1 Entity ..................................................................................................................... 7
   3.2 Attribute ............................................................................................................... 7
   3.3 Relationship ......................................................................................................... 7
      3.3.1 Parent-child relationship .............................................................................. 7
      3.3.2 Non-identifying relationship ...................................................................... 7
      3.3.3 Identifying relationship .............................................................................. 7
      3.3.4 Subtype relationship .................................................................................... 7
   3.4 Keys (Identifier) ................................................................................................. 8
      3.4.1 Primary key (Identifier) ............................................................................ 8
      3.4.2 Foreign or secondary keys ......................................................................... 8
   3.5 Supertype and Subtype entities .......................................................................... 8
   3.6 Cardinality and optionality ............................................................................... 9
   3.7 Recursive Relationship ...................................................................................... 10
   3.8 Link or associative entities .............................................................................. 10
   3.9 Concatenated or Compound Keys .................................................................... 11
   3.10 Primary Key Role naming (rolename) .............................................................. 11
   3.11 Reference Data .................................................................................................. 12
   3.12 Aggregated Data ............................................................................................... 13
   3.13 Subject Areas ..................................................................................................... 13
   3.14 Notation ............................................................................................................ 15
   3.15 Diagram Colour Coding .................................................................................... 17

4 Tool Management ...................................................................................................... 19
   4.0 ERwin ................................................................................................................. 19
   4.0.1 ERwin Version ............................................................................................... 19
   4.0.2 Data Model Version ....................................................................................... 19
   4.0.3 Stored Display Tabs ....................................................................................... 19
   4.0.4 Meta Data ....................................................................................................... 20
   4.0.5 Attribute Update Window .............................................................................. 21
   4.0.6 Entity Update Window .................................................................................... 22
   4.0.7 Defaults .......................................................................................................... 22
   4.0.8 Model type ..................................................................................................... 22
   4.0.9 Logical Attribute Data Types ....................................................................... 23
   4.0.10 Primary Key Propagation Rolenaming ......................................................... 23
   4.0.11 Reports ....................................................................................................... 24
   4.0.12 Controlled Lists in Erwin ............................................................................ 26
      4.0.12.1 Entity level meta data .......................................................................... 26
      4.0.12.2 CL Values level metadata ..................................................................... 26
      4.0.12.3 Validation Rule metadata ..................................................................... 27
      4.0.12.4 Creating a Validation Rule to hold the CL Values ......................... 27
      4.0.12.5 Importing the CL Values and associated metadata ....................... 27
      4.0.12.6 Import CL Values only via csv file ....................................................... 27
      4.0.12.7 Import CL Values and metadata via Erwin API ................................ 27
      4.0.12.7.1 Step A – Create the Validation Rule in Erwin ......................... 28
4.0.12.7.2 Step B – update Erwin using the Erwin API spreadsheet ........................................ 28
4.0.12.7.3 Step C – Link the Validation rule to the attributes in each applicable entity that the
CL attribute is a foreign key ........................................................................................................... 28
4.0.13 Abacus Interface ............................................................................................................ 29
4.0.14 ERwin Support ............................................................................................................. 29

5 Data Model and XSD Change Management .......................................................... 30

6 Naming Standards .......................................................................................................... 31

6.1 Objective ......................................................................................................................... 31
6.2 Entity Name ....................................................................................................................... 31
6.3 Entity Description .............................................................................................................. 32
6.4 Associative Entity ............................................................................................................ 32
6.5 Attribute Description ....................................................................................................... 32
6.6 Attribute Name ................................................................................................................ 32
6.7 Reference Data ............................................................................................................... 33
6.8 Aggregated Data ............................................................................................................. 33
6.9 Entity Relationships ........................................................................................................ 33
6.10 Master Data .................................................................................................................... 34

7 Approved Abbreviations ............................................................................................. 34

8 Modelling Concepts ....................................................................................................... 39

8.0 Objective ......................................................................................................................... 39

9 Changes from previous version ................................................................................ 39

10 References ....................................................................................................................... 39

11 Copyright Notice ............................................................................................................. 40
1 INTRODUCTION

This document describes the data modelling standards applied to the ISB Business Data Architecture.

The standards used consist of

- General modelling notation standards for Entity Relationship Diagram modelling using Information Engineering (IE) notation
- Specific standards applicable to the development of a Canonical data model as used for the ISB Business Data Architecture

This document is intended to be used by persons with a data modelling background or exposure to data models. Therefore, it explains the specific standards used such that a data modeller who uses, for example, UML may understand the equivalent IE notation. It also covers such terms as parent/child entities and also super and sub types. These terms are generally used in data modelling but some methodologies use alternative terms and so the reader may not have been exposed to them. This document does not intend to teach someone how to data model.

2 DATA MODEL TYPES

The Business Data Architecture needs to be distinguished from logical data models, physical data models and databases or physical data stores:

- **Logical data models** provide an abstract view from a business perspective of the data that is stored in and used by a particular information system. Logical data models define data in a way that is independent of any underlying data repository.
  In a Greenfield site, where an organisation designs and builds its own information systems, the logical view of the data in a particular information system would be identical to that contained in the Business Data Architecture. However, we are not in a Greenfield site and therefore existing systems, which have been delivered at different times, will often contain data that is duplicated in other systems. The definition of the data in each system may be similar, but not identical and therefore make integration and data sharing more difficult. The same considerations apply where an organisation uses commercial off the shelf (COTS) software systems provided by different vendors.

- **The Business Data Architecture** provides a business view of the data needed by an “enterprise” that is independent of applications and physical implementations. An “enterprise” is understood to be the collection of or parts of organisations that are the scope of the enterprise data architecture.

- **Physical data models** are the basis for physical database or file design and will typically introduce features that are required to optimise performance that are not inherently required to provide a logical and coherent view of the data in the system from a business perspective.
• **Databases** or physical data stores are the physical databases or files that implement the physical data model.

From an Enterprise Architecture perspective physical data stores, physical data models, logical data models and the Business Data Architecture must all be integral, in that the physical data store is derived from the physical data model, the physical data model is derived from the logical data model and the logical data model is derived from, or at least mapped to, the Business Data Architecture. All of the models can be represented in the Enterprise Architecture, although not all of the lower level models will always be of interest from an Enterprise Architectural perspective. Where they are, it will usually be at a higher level of abstraction than that required by the project teams that are implementing solutions.
3 BUSINESS DATA ARCHITECTURE DEFINITIONS

3.1 Entity
An entity is a real world thing which is recognised as being capable of an independent existence and which can be uniquely identified. Entities can be thought of as nouns.

An entity of interest to a Data Architecture is a “Thing, Person, Place or Event” about which the organisation records data.

Where the data architecture refers to an entity within definitions or text it is shown in upper case eg the definition of the PARTY entity is “A PERSON or ORGANISATION who is known to and recorded by the education, skills, and children's services 'system'.”

An entity when converted to a table in a data base will contain one or more rows of data for that entity. At the logical level as there are no tables then each “row” of data in an entity is referred to as an “entity instance”.

3.2 Attribute
Attributes are properties of an entity. They provide more detail about the entity they are associated with. Examples of attributes of [PERSON] are date of birth, national identity, country type, etc.

Attributes must not contain any embedded logic, which is where two or more items of information are encoded into a single attribute. An example of embedded logic would be, for example, where the first two characters are used to indicate the country of origin and the next eight characters are used to indicate the date of birth.

3.3 Relationship
A relationship captures how two entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. There are a number of relationship types.

3.3.1 Parent-child relationship
Where an identifying relationship exists between two entities, the entity from which the primary keys are propagated from is known as the Parent entity and the entity to which the primary keys propagate to is known as the Child entity.

3.3.2 Non-identifying relationship
A non-identifying relationship is one where the primary keys of one entity are propagated to another entity but not as part of the identifiers of the second entity. This allows the second entity to refer to the first entity but it is not a child of the first entity.

3.3.3 Identifying relationship
An identifying relationship is one where the child entity cannot exist without the parent and it is identified using the primary key attribute(s) of the parent.
3.3.4 Subtype relationship

A subtype (or sub-category) relationship is a form of abstraction that specifies that two or more entities share common attributes and can be generalised into a supertype. Subtypes can either be mutually exclusive or inclusive. An example of a mutually exclusive subtype is a Vehicle type where the subtype may be car or lorry. One vehicle can either be a car or a lorry but not both. An example of an inclusive subtype is a Person Role type where a Person can be both a customer and employee. Generally inclusive subtypes reflect a lack of rigour in the data model as a Person is a customer as part of a customer relationship and an employee as part of an employment relationship. The BDA only uses exclusive subtypes.

3.4 Keys (Identifier)

3.4.1 Primary key (Identifier)

A primary key is the set of attributes that between them uniquely identify an entity. There are two types of primary key that can be used and these are Natural Keys or Surrogate Keys. A Natural Key is a key made from a value that exists naturally in the data and can be used usually with other Natural Keys to uniquely identify the entity eg an Order Number/Organisation. A Surrogate Key is an identifier that is artificially assigned to the entity instance and does not exist within the data itself. An example is a Party Id. Where possible Natural Keys are used rather than surrogate because:

- Natural Keys limit the possibility of data content being swapped to the wrong key – easy to do with a Surrogate Key.
- Natural Keys provide additional validation of the entity- natural keys describe the entity, surrogates do not.
- Natural Keys provide built in data integrity when loading data or assembling messages.

3.4.2 Foreign or secondary keys

In order to support the relationship between two entities the primary key of one entity (the parent entity) needs to be included as an attribute of the entity to which it relates (the child entity). This is sometimes referred to as primary key propagation or migration.

Inclusion of a primary key as an attribute of another entity is called a secondary or foreign key since it identifies the entity to which it is related.

3.5 Supertype and Subtype entities

Where a number of entities share a similar behaviour and many of the same attributes, it is possible to combine them into a supertype/subtype design. This has a number of benefits:

- The common attributes can be held in the supertype
- The subtypes hold only the attributes specific to their type thereby reducing the number of unused attributes for a specific instance of data
- The identifiers in the supertype are inherited by the subtype
- The total attributes for a type are a combination of the supertype plus the individual subtype
- The specific subtype is identified in the supertype by an identifying attribute that denotes the subtype that the supertype instance relates to
- Allows other entities to relate to the supertype thereby allowing the number of subtypes to be expanded as required

Some examples:
A [PARTY] supertype contains two subtypes- [PERSON] and [ORGANISATION]
The date when a [PARTY] first became known to the ‘system’ would be an example of an attribute that applies to [PARTY] and therefore by definition is inherited by both [PERSON] and [ORGANISATION]. However [PERSON] would have attributes of date of birth, gender, ethnicity which would not apply to [ORGANISATION].
Likewise, some relationships would apply to [PERSON] that would not apply to [ORGANISATION] and some would apply to [PARTY] because they could apply to both [ORGANISATION] and [PERSON].
The relationship between a supertype and its subtypes together with the symbols used is illustrated in the following diagram.

3.6 Cardinality and optionality
Cardinality defines the numeric relationship between instances of the entities on either end of the relationship line.
The permitted combinations are:
- One to many (1:M) - One instance of entity A may be related to one or more instances of entity B
- Many to Many (M:M) - one or more instances of entity A may be related to one or more instances of entity B
Optionality defines whether an instance of an entity may or must have a relationship with an instance of another entity. Optionality can be defined at both ends of a relationship between two entities.
The permitted values are:

- **Optional/Optional** - Instances of entity A or B can be created without the need to create the other. Therefore, an instance of entity A can be created without creating a relationship to an instance of entity B and an instance of entity B can be created without creating a relationship with entity A. An example is that a Person and an Address can both be created but do not have to be related to each other. However, they can be related when the person lives at that address.

- **Mandatory/Optional** - An instance of entity B cannot be created unless an instance of entity A already exists. Therefore, an instance of entity A can be created without creating a relationship with entity B but, if an instance of entity B is created, it must be related to entity A. In this situation entity A is the parent of entity B with B being the child of A. An example is a Person and a Name. The person can be created without a name being assigned immediately (new born baby). However, once a name is created it must be assigned to a person.

- **Mandatory/Mandatory** - an instance of entity A can be created only if it is related to an instance of entity B, an instance of entity B can only be created if it relates to an instance of entity A. An example is Car and VIN number (Vehicle Identification Number). A car must have a VIN number and a VIN number must exist for one vehicle. However, this normally never happens at the same time and so very seldom used.

Combining cardinality and optionality rules ensures that data integrity is maintained. The notation that has been used for cardinality and optionality is illustrated in Section 3.14 below.

### 3.7 Recursive Relationship

A recursive relationship is where instances of an entity can be related to other instances of the same entity and where the relationship is always a hierarchical one. This is sometimes referred to as a “pig’s ear”. The Business Data Architecture supports recursive relationships at the conceptual stage but, at the logical stage, it resolves them such that they future proof as the pure recursion method only allows the recused primary key(s) of the entity to be non-identifying. This method does not allow multiple concurrent hierarchies with the same object to be defined.

### 3.8 Link or associative entities

Where a relationship is many to many, the many to many nature of the relationship is resolved by the introduction of a link or associative entity. This ensures that the model is normalised and eliminates the repeating groups of foreign keys that would be required in each of the entities being related without the introduction of a link entity.
3.9 **Concatenated or Compound Keys**

Where a child entity has multiple parents, the keys from each of the parents will be joined together in the child entity creating a key that has multiple attributes. In this event the primary key of the child entity is a concatenation of the foreign or secondary keys. Sometimes this is called a compound key.

3.10 **Primary Key Role naming (rolename)**

There are situations where the name of the primary key attribute in a parent entity, when propagated to a child entity needs to be modified. This can be for one of the following reasons:

- The name would be a duplicate of a similar named attribute propagated to the same entity. This can occur where there are dual parent entities that both have the same attribute name in their primary key but the data instance contained within each parent entity is different.
- Where the child entity needs to clarify the use of that attribute eg Party_Id when propagated to the Learner entity becomes Learner_Party_Id.
- Where there is a dual relationship from the Parent entity to the child eg Party_ID becomes Party_Id_1st and Party_Id_2nd when propagated to the child entity from the Party entity using a dual relationship.
To identify where this has taken place and what the original attribute name was, the Rolename attribute name and the migrated attribute name are shown on the attribute diagram

To identify the original attribute name and its source entity, the data standards document will include against a foreign key attribute that has been role named:

- **FK Source Entity** – the name of the source entity from which the attribute has been migrated
- **FK Source Attribute** – the name of the attribute within the source entity

An example is as follows from the Organisation Post Fulfilment entity

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Definition</th>
<th>PK</th>
<th>FK</th>
<th>M</th>
<th>Datatype</th>
<th>FK Source Entity</th>
<th>FK Source Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Party_Id_Employer</td>
<td>A value that denotes and distinguishes the PARTY.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Unicode_String(1,32)</td>
<td>Organisation Post</td>
<td>Organisation_Party_Id</td>
</tr>
</tbody>
</table>

The above attribute Party_Id_Employer originated in the entity Organisation Post and had the original name Organisation_Post_Party_Id and so has been role named on the identifying relationship from Organisation Post to Organisation Post Fulfilment.

### 3.11 Reference Data

Reference data is data that defines a set of constrained values and descriptions that can be used to categorise objects.

In data models, entities that contain reference data are often called ‘Type’ entities.

Some examples of reference data would be:

- [Learning Provider Type], which might categorise learning providers depending on the age group of [Learners] to whom they provide learning opportunities.
- [National Partners Type], which might categorise organisations depending on their legal status eg Department of State, Executive Agency, Non-Departmental Public Body.
- In the Business Data Architecture and related Business Data Standards the reference data entities will only contain the reference value attribute. Other reference data, eg metadata, will be required to use the reference data entity in the wider context.

On the data model, reference data ‘Type’ entities are always coloured orange, and have the suffix ‘Type’ in their name.
3.12 Aggregated Data

Aggregated data is data that is non-atomic and has been aggregated from atomic data. It is used for a specific purpose and requires rules to define how to aggregate atomic data. An example is ‘Total Sales’. This will be built up from individual sales and any rules of what atomic records are included/excluded must accompany the entity/attributes.

On the data model aggregated data entities are always coloured grey and accompanied by the prefix ‘Agg’ immediately followed by an abbreviation of the data standard name in capitals eg the data standard Agg Leavers Destination would result in the prefix AggLD being applied to the entity and attributes within the data standard.

AggLD Working Hours A Week

<table>
<thead>
<tr>
<th>AggLd_Working_Hours_Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>AggLd_Working_Hours_Eff_Date</td>
</tr>
<tr>
<td>AggLd_Working_Hours_Min</td>
</tr>
<tr>
<td>AggLd_Working_Hours_Max</td>
</tr>
</tbody>
</table>

All attributes within the aggregated data entity(s) will also begin with the above defined prefix as applicable to the entity.

All reference data entity(s) defined for the Aggregated Data Standard that are only applicable to the specific Aggregated Data Standard will also have the above prefix applied to their entity and attribute name eg

AggLd Current Activity Type

| AggLd_Current_Activity_Type |

3.13 Subject Areas

Because of the size and complexity of the data model, various Subject Area models have been produced, which provide a particular perspective on the overall Business Data Architecture.
The Subject Areas are described in the Business Data Architecture. Here it is sufficient to note that entities that are particular to a Subject Area are enclosed within a box. Entities outside this box are included for contextual understanding and show the entities in other Subject Areas to which the entities in the Subject Area are related. (See above Party Contact model)

As part of the upgrade from Erwin version 7.3 to Erwin version 9.6, the previously separate summary model and domain model have both been incorporated into the one BDS model. This is to aid in the future automated production of the ISB BDA web site direct from the Erwin data model.

There are 5 types of Subject area used in the Business Data Architecture

1) The main Domain Model subject area that identifies all of the sub-domains – this is prefixed with a C) and is called the Domain Model

2) The sub-domain subject areas – these are the high level breakdown of the model into domains such as Government and Administration – Party

3) The Summary Domain model subject area is a model that only includes the primary entity from each Business Data Standard. Its purpose is to show the overall navigation of the Business Data Standards and how they interrelate to each other – this is prefixed with C) and is called the Summary Domain Model. To support the Summary Domain Model, additional Summary Domain Model specific primary entities from each Business Data Standard have been added. These are identified by being in upper case

4) The Business Data Standards (BDS) – these are the entities that form the Business Data Standard. All BDS subject areas are prefixed with “A)” eg A) Party. All entities in a BDS or ADS are identified by the use of CamelCase

5) The Aggregated Data Standards (ADS) – these are the entities that form the Business Data Standard. All ADS subject areas are prefixed with “A) Agg” eg A) Agg Leavers Destination. All entities in a BDS or ADS are identified by the use of CamelCase

The main subject area – this contains all of the entities in the model – this is optional in Erwin version 9 but has been recreated to support forward engineering of the schema Data Definition Language

Subject areas for the Domain models use the BDA stored display (see stored display later in this document)

Subject areas for the Business Data Standards models and Aggregated Data Standards use the Entity Def View and Attribute View stored displays (see stored display later in this document)

For the Business Data Standards/ Aggregated Data Standards subject area’s Attribute View, the related entities (those outside of the box) will be manually amended in size to only show the primary key attributes. This will reduce the change release impact on associated BDS of an addition to non-identifying attributes in an entity. If the whole set of attributes is shown, then any change to an associated entity non-primary key attribute would have to be reflected in the release document plus new BDS models would have to be produced even though the actual entity(s) within the BDS box had not been amended
3.14 Notation

The business data architecture is represented as an entity relationship diagram (ERD) using the information engineering (IE) notation.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Graphic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>A thing, person, place or event which is recognised as being capable of an</td>
<td>Stakeholder</td>
</tr>
<tr>
<td></td>
<td>independent existence and which can be uniquely identified. Subtyped entities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>have rounded corners whereas supertype entities have square corners, see</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subtypes below. Entity colour can vary – see below for colour coding.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The name of the entity is above the box, the definition is inside the box.</td>
<td></td>
</tr>
<tr>
<td>Non-Identifying Relationship</td>
<td>A relationship where the child entity can be identified independently of the parent.</td>
<td>results in</td>
</tr>
<tr>
<td>Identifying Relationship</td>
<td>A relationship where the child entity cannot be uniquely identified without the parent. i.e. the parent can exist without the child, but the child cannot exist without the parent.</td>
<td>results in</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Graphic</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Subtype</td>
<td>A form of abstraction that specifies that two or more entities share common attributes and can be generalised into a supertype. Subtypes can either be mutually exclusive or overlapping. The BDA only uses mutually exclusive subtypes as overlapping types are usually an alternative entity.</td>
<td><img src="image" alt="Subtype Diagram" /></td>
</tr>
<tr>
<td>0, 1 or Many cardinality and optionality</td>
<td>The relationship may or may not occur and if it does may have 0, 1 or more occurrences.</td>
<td><img src="image" alt="0, 1 or Many Diagram" /></td>
</tr>
<tr>
<td>1 or Many cardinality and optionality</td>
<td>The relationship always occurs and there may be many related items for the primary object.</td>
<td><img src="image" alt="1 or Many Diagram" /></td>
</tr>
<tr>
<td>Many to many cardinality and optionality</td>
<td>The relationship occurs any number of times in both directions.</td>
<td><img src="image" alt="Many to many Diagram" /></td>
</tr>
<tr>
<td>Recursive Relationship (pig’s ear)</td>
<td>The relationship is from and to the same entity as the entity can contain other instances of itself.</td>
<td><img src="image" alt="Recursive Diagram" /></td>
</tr>
<tr>
<td>Link or associate entity</td>
<td>Required to resolve a many-to-many relationship and identified by its colour.</td>
<td><img src="image" alt="Link Diagram" /></td>
</tr>
<tr>
<td>Reference Data</td>
<td>Data that defines a set of controlled values that can be used to categorise objects.</td>
<td><img src="image" alt="Reference Diagram" /></td>
</tr>
</tbody>
</table>
3.15 Diagram Colour Coding

The following colour coding is used to assist in navigating the model:

- Entity Fill – Light Blue – Supertype
- Entity Fill - Yellow – Subtype
- Entity Fill - Orange - Reference Data
- Entity Fill – Dark Blue - Associative or Link entity
- Entity Fill - White – Standard entity
- Entity Fill – Grey – Aggregate Data
- Shape - Grey boxing - entities particular to a Subject Area
- Attributes - Black
- Entity Outline/Relationship Line/Attribute - Green – new, awaiting review
- Entity Fill/Attribute/Relationship Line - Red – deleted, awaiting review
- Entity Fill/Attribute/Relationship Line - Pink – amended, awaiting review
- Entity Outline and Relationship Line – Black if entity or relationship have not yet been analysed and used as part of a Business Data Standard

For example:

- Entity Outline and Relationship Line – Blue if entity and child relationship have been analysed and used as part of an approved Business Data Standard otherwise black
For example:

- Colours used for a new, deleted or amended entity/relationship will be set to the correct colour once the BDA release is complete and the model is ready for the next version changes to be made.
4 TOOL MANAGEMENT

This format is supported by the modelling tool used by the Department, ‘CA’s ERwin data modeller’.

4.0 ERwin

4.0.1 ERwin Version
The current model is maintained in Version 9.64.02

4.0.2 Data Model Version
The current versioning consists of a prefix and suffix number:
- The prefix relates to the major version and is incremented each time the Business Data Architecture document is released.
- The suffix is incremented for each new project or significant change to the model.

The suffix versioning is primarily to allow roll-back in the event of a change no longer being required.

4.0.3 Stored Display Tabs
Two stored displays are used as defined above in Subject Areas section:
- Entity Def View – this displays the entity definition
- Attribute View – this displays the attributes

Stored display settings for the above are:
- For Entity Def View
  - Display Level – Definition
  - Relationship Option – Verb Phrase, Cardinality and Subtype Discriminator
  - Entity Option – Rolename/Attribute, Foreign Key Designator, Alternate Key Designator and Show Migrated Attribute
- For Attribute View
  - Display Level – Attribute
  - Relationship Option – Verb Phrase and Subtype Discriminator
  - Entity Option – Primary Key Designator, Foreign Key Designator, Alternate Key Designator and Show Migrated Attribute
  - Note – to reduce the impact of changes in a related Business Data Standard (BDS) where the entity appears in another BDS for reference purposes only ie outside of the boxed area, only the primary keys of the referenced entity must be visible to reduce related impacts where non-identifying attributes only change in the referenced entity. This is achieved by selecting the entity to only have its primary keys shown and then pressing the display option of
“Primary Key Display Level” from the Display tool bar. An example is:

These can be set up from the menu option Format>Stored Display Settings – or by a right click in the model window.

4.0.4 Meta Data

Erwin supports metadata for most types of Erwin objects – such as Entity, Attribute and Relationship. The following table identifies the metadata that is used and also any User Defined Properties that have been added.
<table>
<thead>
<tr>
<th>Logical Object</th>
<th>Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Definition</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes</td>
</tr>
<tr>
<td>Attribute</td>
<td>Datatype</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Notes</td>
</tr>
<tr>
<td></td>
<td>User defined properties</td>
</tr>
<tr>
<td></td>
<td>• Def QA Checked</td>
</tr>
<tr>
<td></td>
<td>• ISB Logical Data Type</td>
</tr>
<tr>
<td>Relationship</td>
<td>Parent-to-Child description</td>
</tr>
<tr>
<td></td>
<td>Cardinality</td>
</tr>
<tr>
<td></td>
<td>Role Name</td>
</tr>
</tbody>
</table>

4.0.5 Attribute Update Window

The following tabs should be used in the above window

- General – contains attribute name and also primary key indicator
- Data type – the Erwin standard data types as applicable to the User Defined Property/ISB Logical Data Type – see Business Data Architecture Data type for the applicable data type to use
- Definition – enter the description of the attribute. This should be clear and unambiguous as to the meaning and use of the attribute and should contain examples if required.
• Note – enter the Standard Erwin logical data type equivalent to the ISB Logical data type list (see “Business Data Architecture Data Types”)
• UDP tab (see Defaults) – this must contain the following property/value combinations
  o Def QA Checked – the date the attribute definition was approved by the vocabulary manager
  o ISB Logical Data Type – the data types as per the Business Data Architecture Data type document

4.0.6 Entity Update Window

The following tabs should be used in the above window:
• Definition – enter the description of the attribute. This should be clear and unambiguous as to the meaning and use of the attribute and should contain examples if required.
• Note – Event = User Notes, Comment – enter the definition source such as a URL, name of person, book title etc.
• UDP tab (see Defaults) – this should contain the following property/value combinations
  o Def QA Checked
  o Controlled List Y/N
  o CL Ref Number
  o CL Version
  o CL Effective Date
  o CL End Effective Date
  o CL Status
  o CL Maintained by

4.0.7 Defaults

Set up Attribute User Defined Properties (UDP). Available via MODEL>UDP Dictionary. Open window and enter the above Attribute UDPS against Class/Attribute (in drop down list).


Colours should be set up as defined in the section 3.15 Diagram Colour Coding. These need to be maintained via the Themes or via Override Fonts and Colours.

4.0.8 Model type

The model uses the Logical type primarily but, when creating a new model, select Logical and Physical.
Note – select a database type such as Oracle that allows a maximum of 30 long attribute and entity names to prevent name truncation.

4.0.9 Logical Attribute Data Types

The Data Type tab in the Attribute properties shows a list of predefined data types available to the model. The BDA model uses a set of Logical Data Types defined in the following document Business Data Architecture Data Types available from the ESCS ISB web site.

Due to the inbuilt ability by Erwin to automatically adjust the physical data types used based on the particular Erwin Logical Data Type and database repository selected, the base attribute data type in Erwin must not be amended from the inbuilt Erwin data types. To support the ISB data types a UDP is added to the attribute to store the ISB Logical Data Type as outlined in the previous section “Attribute Update Window”

The equivalent Erwin logical data type to ISB Logical data type is as follows:

<table>
<thead>
<tr>
<th>ESCS Logical Data Type – Attribute UDP</th>
<th>Erwin Standard Logical Data Type – Attribute data type tab</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple_String (a,b)</td>
<td>VARCHAR(b)</td>
<td>Where b=maxlength</td>
</tr>
<tr>
<td>Unicode_String(a,b)</td>
<td>NVARCHAR(b)</td>
<td>Where b=maxlength</td>
</tr>
<tr>
<td>Simple_Integer(a,b)</td>
<td>INTEGER</td>
<td>No parameters</td>
</tr>
<tr>
<td>Simple_Decimal(p,s)</td>
<td>DECIMAL(p,s)</td>
<td>P=number of digits, s = Number of digits to right of dec place</td>
</tr>
<tr>
<td>Simple_Flag</td>
<td>CHAR(1)</td>
<td>Value Y or N</td>
</tr>
<tr>
<td>Simple_Date</td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>Simple_Time</td>
<td>TIME</td>
<td></td>
</tr>
<tr>
<td>Simple_Date_Time</td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>Simple_Binary(a,b)</td>
<td>BINARY(b)</td>
<td>Where b=maxlength</td>
</tr>
</tbody>
</table>

4.0.10 Primary Key Propagation Rolenaming

To support the requirement detailed above to rolename and attribute when propagated from the parent entity to the child entity, Erwin provides the following functionality.
1. Right click on a relationship line
2. Select Relationship Properties
3. Select Rolename tab
4. Enter the name that should be propagated to the child entity

Eg to rolename Party_Id to Party_Id_1st on a relationship from Party to Party Relationship:

![Diagram showing relationship Editor](image)

### 4.0.11 Reports

The standard reports required for conversion to data standards and Business Data Architecture documents is called “Official BDS (Vs2.2+CL). The report requires the following:
4.0.12 Controlled Lists in Erwin

Controlled list values for any Reference entity within the data model have until recently been maintained outside of Erwin within PDF documents on the ISB website. As part of the Erwin upgrade to version 9.6, the potential via Erwin 9.6 and the Erwin API, to automatically extract data from Erwin and load it up to a dynamic ISB Web site requires as much model data as possible to be in Erwin. This has necessitated loading the CL values directly into Erwin so that they can be extracted automatically. It also ensures that as much of the BDA data is within the Erwin model, so that the places required to hold the various information required to support the BDA is minimised.

The method used to hold the Controlled List data is via the Validation Rules and Valid Values constraints in Erwin. There are additional meta data entries required and whilst the actual additional fields can be added via the User Defined Properties editor, it is not possible to display the UDPS on the Valid Values tab. Therefore, a custom spreadsheet using the Erwin API has been written by the ISB TSS. This facilitates loading the Valid Values plus their metadata to the spreadsheet and then uploading the data via the Erwin API that the spreadsheet has coded into it.

The method for creation and population of the CL data is as follows:

4.0.12.1 Entity level metadata

Add following UDP to the Entity via the UDP editor

1) CL Ref Number
2) CL Version
3) CL Effective Date – start date of the CL list – Note, set this data type as text rather than date as, when selecting the UDP, Erwin locks up
4) CL End Effective Date – end date of the CL list - Note, set this data type as text rather than date as, when selecting the UDP, Erwin locks up
5) CL Status – set to default of Approved:Recommended
6) CL Maintained by – set to default of ISB TSS

To add the metadata to the entity, open the entity properties and go to the UDP tab- manually add the values

4.0.12.2 CL Values level metadata

Add the following UDP to the Valid Value UDP via the UDP editor

1) CL Derivation (Aggregations only)
2) CL Value Source
3) CL Value Version
4) CL Value Status – set to default of Approved:Recommended
5) CL Value Effective Date – start date of the CL list – Note, set this data type as text rather than date as, when selecting the UDP, Erwin locks up
6) CL Value End Effective Date – end date of the CL list - Note, set this data type as text rather than date as, when selecting the UDP, Erwin locks up
7) CL Value Notes

The actual values for the controlled list and the UDP metadata is added later on in the process
4.0.12.3 Validation Rule metadata

Add the following UDP to the Validation Rule UDP via the UDP editor

1) Date CL Values Imported - Note, set this data type as text rather than date as, when selecting the UDP, Erwin locks up

4.0.12.4 Creating a Validation Rule to hold the CL Values

To store the CL values, use the Erwin Validation Rule via Models>Validation Rules

Open the validation rules and create a new rule with the same name as the CL attribute name, eg Absence_Authorisation_Type. The Rule type should be set to “Valid Values”

Select the UDP tab and the “Date CL Values Imported” should be set to the current date. Only change this when updating the full CL values list via an import

Select the Definition tab and set the description to the same definition as the definition of the reference entity of the same name.

Selecting the General tab at the lower portion of the UI will show the Valid Value, Display Value and Definition fields. It does not show the additional UDPs as there is a bug in Erwin 9.64. Therefore, it is necessary to import the CL values metadata via the Erwin API, and display the metadata using the query tool or report as shown later.

4.0.12.5 Importing the CL Values and associated metadata

As mentioned in the previous section, the UDPs for the Valid Values do not display in Erwin. Therefore, the Values can either be added manually or via the import button, but the metadata has to be imported via the Erwin API.

4.0.12.6 Import CL Values only via csv file

The import via CSV will only import the

1) Valid Value
2) Display Value
3) Definition

To construct a csv file, the fields are put in the above order and separated by a comma between each field

To display the Valid_Value in the metadata use the query “Select * from Valid_Value”.

4.0.12.7 Import CL Values and metadata via Erwin API

To load the CL Value meta data into the non-visible (on the Valid Values tab of the validation rule), the Erwin API can be used as this accesses the underlying Erwin meta model directly. The ISB TSS have created an excel spreadsheet with embedded Visual Basic code that adds or updates individual CL Values and the metadata entered onto the spreadsheet template directly into the Erwin meta model. The Excel spreadsheet can also delete all Valid Values for a Validation
Rule. Once loaded the meta data can be seen via either an Erwin query or using Erwin report designer.

The Excel Erwin API CL spreadsheet is currently called SJP_API_Create_CL_Values_Er9_vs3.0

4.0.12.7.1 Step A – Create the Validation Rule in Erwin

1) Create the validation rule in Erwin as in the above section “Creating a Validation Rule to hold the CL Values”

4.0.12.7.2 Step B – update Erwin using the Erwin API spreadsheet,

1) Make sure the Erwin data model is not open in Erwin
2) Open the Excel Erwin API CL spreadsheet file
3) Select the worksheet “Valid.Values”
4) Copy and paste the CL values and meta data from the Data Dictionary “Controlled List reference data” worksheet for the CL list onto the API template. Note the values in the Valid Value Name and Display Name fields need to be both populated from the data dictionary Code Full Name field
5) Update the Validation Rule Name cell using the field on the data dictionary spreadsheet called “Reference to Attribute” (ie the name with underscores between each word).
6) Select the worksheet “Update Erwin”
7) Click on the “Update Valid Values” button
8) When requested, select the Erwin data model file to be updated
9) Confirm the update
10) The program will run and advise how many rows were added or updated

4.0.12.7.3 Step C – Link the Validation rule to the attributes in each applicable entity that the CL attribute is a foreign key

1) Open the data model that has been updated via the API spreadsheet in the previous step
2) Applying the Validation Rule to the CL reference entity
   o Using the Erwin Explorer
     o Open the reference entity that the control list relates to and click on the primary key that you want to attach the validation rule to. Open the attribute Properties
     o To assign the validation rule on the attribute edit window go to the Constraint tab.
     ▪ In the Check Constraints windows click the “new” icon
     ▪ In the Validation Rule column click the drop down box and select the validation rule created above in Step A
     ▪ The rule will now appear in the Constraint Name but you will note it has added a number to the end. Edit the name in the Name column back to the original name but with a suffix of “-CL” eg if adding the Validation Rule to the reference entity of Absence Authorisation Type and its attribute Absence_Authorisation_Type, name the constraint as Absence_Authorisation_Type-CL
Click Close and the rule is assigned
  
  In Erwin validation rules are locally applied and are not global. Therefore, you need to apply the validation rule to the attribute in every entity it is used. To Identify where the CL attribute is used in the model, open the CL reference entity attribute properties and click on the “where used” tab. This will show what entities and any role named attributes inherit the reference attribute.

3) Applying the Validation Rule to every use of the CL reference attribute
  
  For each entity identified in the previous CL attribute “Where Used”, open the entity and click on the attribute that you want to attach the rule to. Open the attribute Properties
  
  To assign the validation rule on the attribute edit window go to the Constraint tab.
    
    In the Check Constraints windows click the “new” icon
    
    In the Validation Rule, column click the drop down box and select the validation rule created above in Step A
    
    The rule will now appear in the Constraint Name but you will note it has added a number to the end. Edit the name in the Name column back to the original name. However, each check constraint must be unique in the model and as the same name has already been used for the reference entity check constraint, suffix the name in the current entity with the abbreviation of the entity in capitals eg if the Absence_Reason_Type validation rule is being added to the same name attribute in the Party Time Period Attendance, create the check constraint name as Absence_Reason_Type-PTPA. If there are multiple uses in the same entity, add a sequence number to the end of the name.
    
    Click Close and the rule is assigned

The data model can now be tested to confirm that the CL values are linked to the entity attribute(s) by creating a report that lists an entity, its attributes and then any validation rules for each attribute. The report designer sees all the Valid Values meta data UDP fields added and so can print out the contents

4.0.13 Abacus Interface

The ERwin data models can be transferred to Abacus using a bespoke interface in Abacus.

To export from ERwin in suitable format, simply save the model as type XML Standard File (*.xml). The Abacus interface is written to accept the xml output format of ERwin.

4.0.14 ERwin Support

At this time two CA Erwin Data Modeler standard edition licences and support for ERwin are purchased through

Sandhill Consultants Ltd, Brewery Hill, Grantham, Lincolnshire NG31 6DW, UK
5 DATA MODEL AND XSD CHANGE MANAGEMENT

It is important to maintain a list of changes to the Business Data Architecture and associated xsd as this will assist with:

- Parties who use the BDA and XSD understanding any changes since the last release version
- Identifying the reason for a change made historically
- Alignment of other models that conform to the BDA

It is also important when making a change to the model to identify any associated document impact. The associated documents are:

- Business Data Standard Document
- Technical Data Standard Document
- Data Block Design Document
- Architecture Standards Document
- Business Data Architecture document

There is an Access Database called the BDAChangeLog that is used to track changes because using a database has the advantage of providing tailored information for managing the BDA and xsd changes and providing input for the consultation and release documents. Please email ISB Secretariat to obtain details.
6 NAMING STANDARDS

6.1 Objective

It is important to define a set of standards for naming entities, attributes, relationships, primary keys and descriptions so that future development of the architecture maintains a level of consistency.

The approach taken in the Business Data Architecture is to define standards that reflect the business level at which the architecture is developed rather than a physical design.

The names used in the Business Data Architecture must be used unmodified in the physical implementation to ensure interoperability.

6.2 Entity Name

- Each entity name must be unique in the model
- Only nouns may be used
- The Entity name must be singular
- The name must be business based and reflect a business level understanding
- Names must follow the following format:
  - First letter must be capital
  - Following letters must be lower case
6.3 Entity Description
The description must describe clearly and in business terms the purpose and meaning of the entity. Any reference to another entity in the architecture must be made with the entity referred to in capitals. If the use of the entity in the description is plural then the entity referred to must be suffixed with a lower case “s” eg PARTYs.

6.4 Associative Entity
An associative entity name must be carefully constructed and follow the following rule in addition to the above Entity name rules:

- Avoid concatenation of the entities associated unless the name would be meaningful in the business community eg Party and Schedule Period can be associated and called Party Schedule

6.5 Attribute Description
The description must describe clearly and in business terms the purpose and meaning of the attribute. Any inclusion of an entity name in the architecture must be made with the entity referred to in capitals.

6.6 Attribute Name
- The attribute name must be singular
- Names will also be subject to the rules found in the preceding sections
- The name must be business based and reflect a business level understanding
- Non-identifying attributes – generally the name should not be prefixed with the entity unless the attribute needs to relate to another attribute that is prefixed with the entity name eg effective and end dates
- Identifying Attributes – as the attribute will propagate to other entities and will not be immediately obvious as to its original context without looking through the relationships, the name should be prefixed with the entity name.
- Names must follow the following format:
  - First letter must be capital
  - Following letters must be lower case
  - Must be singular
Must not contain any special characters other than “_“ (underscore) where shown below.

Must use only letters no numbers unless the attribute refers to a numbered standard

Must not use abbreviations unless necessary if the name would exceed 30 characters. Only the approved abbreviations below are allowed

Maximum attribute name length 30 characters

Where the name is constructed of more than one word then there should be a “_” (underscore) between words.

Identity attribute must use the abbreviation of Id for Identity eg Party_Id

Attributes that use the data type of Simple_Flag must be suffixed with the name _Flag

Externally defined Data Item and Data Type Standards documents (eg ISO, BSI, IETF, ITU etc) may be referred to and used as a source subject to the application of the above rules when using in the Business Data Architecture

Relationship entity - identify parent and child inherited duplicated attributes via suffix in the attribute name

Inherited attributes that need to be qualified in the child entity (eg to prevent keys merging or for context) - refer to Modelling Concept section title Role Names

Definitions:

For Controlled list Types the definition should start “A Controlled list of values …”

For unique Id suffixed attributes, the definition should start “The identifier that uniquely identifies …”

For compound key identifiers the definition should start “The identifier that combined with other identifiers uniquely …”

6.7 Reference Data

Reference data such as Country lists should use the same naming rules as for Entity and attribute above but will have the word “Type” added at the end of the name eg Party Role Type.

6.8 Aggregated Data

Aggregated data should use the same naming rules as for Entity and attribute and Reference data above but will be accompanied by the prefix ‘Agg’ immediately followed by an abbreviation of the data standard name in capitals eg the data standard Agg Leavers Destination would result in the prefix AggLD being applied to the entity and AggLD_ to attributes within the data standard.

6.9 Entity Relationships

Each name must be a verb construct
Whilst Erwin will support descriptions for both directions of a relationship, most relationship 2nd descriptions tend to be the reverse of the main direction relationship and as such do not add value. As a consequence, only the parent to child should be named unless the relationship is a many-to-many as yet unresolved or by adding the reverse relationship the description will clarify the relationship.

For each parent to child relationship direction the name may consist of the name of one of the entities involved in the relationship but not both

Ideally, the Entity used in the phrase should be the parent entity

The use of “is the subject of” or “is related to” should be avoided unless it is the only concise description

Where the same entity is related to another entity more than once eg The Party and Party Relationship entities results in two relationships from Party to Party Relationship, then the primary relationship role will be named “is the parent in” and the secondary relationship role will be named “is the child in”

6.10 Master Data

Master Data is an entity that:

- Contains prime non-transactional sets of data that are used in the enterprise, such as customer, supplier, or product.
- For effective information across the enterprise, this data set needs to have the same values and be in synchronisation across the enterprise.
- Are not reference data sets in that the data in the set can change frequently

For Master Data, a unique id across the enterprise is required. To prevent requiring a compound key that would then propagate across the entities and add no value in other entities, a single key should be used prefixed with “_Id”.

If there are supertypes and subtypes involved with a master data set then the type that identifies the subtypes must be held on the supertype as a mandatory identifying attribute containing values that are the same as the subtype entity names.

7 APPROVED ABBREVIATIONS

The following table contains the list of currently Approved Abbreviations for use with the Business Data Architecture entity or attribute naming.

The version of the BDA Erwin data model that this applies to is Version 9.02

<table>
<thead>
<tr>
<th>Word/Phrase</th>
<th>Abbreviation1</th>
<th>Abbreviation2</th>
<th>Example</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Role) Party Id</td>
<td>(Role) Id</td>
<td></td>
<td>Learning Provider Id</td>
<td>Ensure consistency if parent/child suffixes involved</td>
</tr>
</tbody>
</table>

File: BDA-Data Architecture Standards | Page 34 of 40 | Version: 13.0 | Status: Final | Issue Date: 19/12/2016
<table>
<thead>
<tr>
<th>(Role) Party Schedule Period</th>
<th>(Role) Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregated Levers Destination</td>
<td>AggLd</td>
</tr>
<tr>
<td>Allowed</td>
<td>Allwd</td>
</tr>
<tr>
<td>Amendment</td>
<td>Amdmnt</td>
</tr>
<tr>
<td>Alternative</td>
<td>Alt</td>
</tr>
<tr>
<td>Application</td>
<td>App</td>
</tr>
<tr>
<td>Assessment</td>
<td>Assmnt</td>
</tr>
<tr>
<td>Assessed</td>
<td>Assesd</td>
</tr>
<tr>
<td>Assigned</td>
<td>Assgnd</td>
</tr>
<tr>
<td>Attendance</td>
<td>Attend</td>
</tr>
<tr>
<td>Calendar</td>
<td>Cal</td>
</tr>
<tr>
<td>Capacity</td>
<td>Cap</td>
</tr>
<tr>
<td>Category</td>
<td>Cat</td>
</tr>
<tr>
<td>Certificate</td>
<td>Cert</td>
</tr>
<tr>
<td>Child</td>
<td>Cd</td>
</tr>
<tr>
<td>Classification</td>
<td>Clas</td>
</tr>
<tr>
<td>Component</td>
<td>Comp</td>
</tr>
<tr>
<td>Consideration</td>
<td>Consid</td>
</tr>
<tr>
<td>Constraint</td>
<td>Constr</td>
</tr>
<tr>
<td>Child Services Assessment Event</td>
<td>CSAE</td>
</tr>
<tr>
<td>Date</td>
<td>Dt</td>
</tr>
<tr>
<td>Date Time</td>
<td>DateT</td>
</tr>
<tr>
<td>Declared</td>
<td>Decl</td>
</tr>
<tr>
<td>Delete</td>
<td>Del</td>
</tr>
<tr>
<td>Delivery</td>
<td>Deli</td>
</tr>
<tr>
<td>Department</td>
<td>Dept</td>
</tr>
<tr>
<td>Description</td>
<td>Descr</td>
</tr>
<tr>
<td>Term</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Direct Service</td>
<td>DS</td>
</tr>
<tr>
<td>Effective</td>
<td>Eff</td>
</tr>
<tr>
<td>Effective End Date</td>
<td>End Date</td>
</tr>
<tr>
<td>Employment</td>
<td>Employ</td>
</tr>
<tr>
<td>Enrolment</td>
<td>Enrol</td>
</tr>
<tr>
<td>Estimate/ estimation/</td>
<td>Est</td>
</tr>
<tr>
<td>Framework</td>
<td>Frmwrk</td>
</tr>
<tr>
<td>Fulfilment</td>
<td>Fulfil</td>
</tr>
<tr>
<td>Generic Service</td>
<td>GServ</td>
</tr>
<tr>
<td>Geography</td>
<td>Geog</td>
</tr>
<tr>
<td>Grade Range</td>
<td>GR</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Hier</td>
</tr>
<tr>
<td>Identifier</td>
<td>Id</td>
</tr>
<tr>
<td>Immigration</td>
<td>Imm</td>
</tr>
<tr>
<td>Information</td>
<td>Info</td>
</tr>
<tr>
<td>Information Object</td>
<td>IO</td>
</tr>
<tr>
<td>Learner Year Group</td>
<td>LYG</td>
</tr>
<tr>
<td>Learning Assessment</td>
<td>LAS</td>
</tr>
<tr>
<td>Specification</td>
<td></td>
</tr>
<tr>
<td>Learning Enrolment</td>
<td>LE</td>
</tr>
<tr>
<td>Learning Opportunity</td>
<td>LO</td>
</tr>
<tr>
<td>Learning Opportunity</td>
<td>Learning Opp</td>
</tr>
<tr>
<td>Learning Opportunity</td>
<td>LOD</td>
</tr>
<tr>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td>Learning Opportunity</td>
<td>LOE</td>
</tr>
<tr>
<td>Enrolment</td>
<td></td>
</tr>
<tr>
<td>Learning Place Provider</td>
<td>LPP</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Learning Provider</td>
<td>Learning Prov</td>
</tr>
<tr>
<td>Learning Provision</td>
<td>LP</td>
</tr>
<tr>
<td>Learning Opportunity Provider</td>
<td>LOP</td>
</tr>
<tr>
<td>Locator</td>
<td>Locr</td>
</tr>
<tr>
<td>Number</td>
<td>No</td>
</tr>
<tr>
<td>Operating</td>
<td>Op</td>
</tr>
<tr>
<td>Opportunity</td>
<td>Opp</td>
</tr>
<tr>
<td>Organisation</td>
<td>Org</td>
</tr>
<tr>
<td>Parent</td>
<td>Pt</td>
</tr>
<tr>
<td>Party Time Period Reference</td>
<td>PTPR</td>
</tr>
<tr>
<td>Performance</td>
<td>Perf</td>
</tr>
<tr>
<td>Party Personal Assessment</td>
<td>PPA</td>
</tr>
<tr>
<td>Party Personal Assessment Associate</td>
<td>PPAA</td>
</tr>
<tr>
<td>Payment</td>
<td>Paymt</td>
</tr>
<tr>
<td>Payment To</td>
<td>PyT</td>
</tr>
<tr>
<td>Performance</td>
<td>Perf</td>
</tr>
<tr>
<td>Period</td>
<td>Perd</td>
</tr>
<tr>
<td>Personal</td>
<td>Pers</td>
</tr>
<tr>
<td>Personal Group</td>
<td>PG</td>
</tr>
<tr>
<td>Premises</td>
<td>Prem</td>
</tr>
<tr>
<td>Previous</td>
<td>Prev</td>
</tr>
<tr>
<td>Provision</td>
<td>Prov</td>
</tr>
<tr>
<td>Provision Component</td>
<td>Prov Comp</td>
</tr>
<tr>
<td>Qualification</td>
<td>Qual</td>
</tr>
<tr>
<td>Qualification Element</td>
<td>QE</td>
</tr>
<tr>
<td>------------------------</td>
<td>----</td>
</tr>
<tr>
<td>Reference</td>
<td>Ref</td>
</tr>
<tr>
<td>Regulation</td>
<td>Reguln</td>
</tr>
<tr>
<td>Relationship</td>
<td>Rel</td>
</tr>
<tr>
<td>Relationship Role</td>
<td>RR</td>
</tr>
<tr>
<td>Required</td>
<td>Req</td>
</tr>
<tr>
<td>Responsibility</td>
<td>Resp</td>
</tr>
<tr>
<td>Risks/Issues/Dependencies</td>
<td>RID</td>
</tr>
<tr>
<td>Rule Of Combination</td>
<td>ROC</td>
</tr>
<tr>
<td>Schedule</td>
<td>Sched</td>
</tr>
<tr>
<td>Service Provision Component</td>
<td>SPC</td>
</tr>
<tr>
<td>Party _Role</td>
<td>Role</td>
</tr>
<tr>
<td>Total</td>
<td>Tot</td>
</tr>
<tr>
<td>Type</td>
<td>Tp</td>
</tr>
<tr>
<td>Verification</td>
<td>Vern</td>
</tr>
<tr>
<td>Representative</td>
<td>Rep</td>
</tr>
<tr>
<td>Immigration</td>
<td>Imm</td>
</tr>
</tbody>
</table>

Notes on abbreviations.

1. Avoid abbreviating unless absolutely necessary - use the richness of 30 characters to describe as clearly as possible the purpose of the attribute.
8 MODELLING CONCEPTS

8.0 Objective

Data Architecture projects will generally define a set of standards such as naming etc. The actual principals used in the modelling may vary from project to project and depend upon the specialist developing the model. For the purposes of defining a data architecture that is used for interoperability data standards, the methodology of the data model design becomes very important. If the project data model design does not adhere to a strict methodology, then there could be overlapping areas in the model or inconsistent behaviour when using the data standards.

The concepts used to design the model are based on a separate methodology and use the following principles:

- Role naming of attributes via the relationship where the same attribute propagated via two or more relationships would merge incorrectly
- Role naming of identification attributes from the super to subtype
- Natural Identifiers
- Full lifecycle entities
- Current point in time entity design
- Sub typing to differentiate different properties of the same concept

9 CHANGES FROM PREVIOUS VERSION

1. Amended processes as a result of the upgrade from Erwin 7.3 to 9.64
2. Inclusion of additional subject areas to support automatic ISB Data Standards web site directly from the Erwin model
3. Details on how to include CL values into the Erwin model and the use of the Erwin API CL Spreadsheet

10 REFERENCES

The following references relate to this document and should be read in conjunction with this document:

- ESCS ISB Enterprise Architecture - Business Data Architecture
- ESCS ISB Business Data Architecture Data Types
11 COPYRIGHT NOTICE

© Crown copyright 2016

The Information Standards Board (ISB) is an advisory body to the Department for Education (DfE) and the Department for Business, Innovation and Skills (BIS). The information it produces is subject to Crown copyright, which is administered by the National Archives.

The Crown copyright protected information in this document (other than ISB or Departmental logos) may be reproduced free of charge in any format or medium under the terms of the Open Government Licence, available from the National Archives website.

Any reuse is subject to the material being reproduced accurately and not used in a misleading context. It must be acknowledged as being protected by Crown copyright and the title of the source material must be supplied with the ISB named as the corporate author.

Authorisation to reproduce any information in this document which is identified as being the copyright of a third party must be obtained from the copyright holders concerned.