



## System Development and Life-Cycle Management (SDLCM) Methodology

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

Approval

CISSCO Program Director

### A. PURPOSE

This standard specifies the format and content of data, or entity, models.

### B. APPLICABILITY

This standard applies to all Projects/Tasks subject to the SDLCM Methodology that include software development or integration.

Members of the Development Team, designated by the Technical Project Manager, are responsible for developing and maintaining the Project's data models, key managers and quality assurance personnel for reviewing them, and the Technical and Business Project Managers for approving them. The data models are made available to all members of the Project team preferably in electronic form.

### C. REFERENCE PUBLICATIONS

The following publications contain related information:

- *SDLCM Methodology Handbook*, Component 1
- *SDLCM Methodology Handbook*, Component 3
- *Systems Development CASE Tool Guidelines*, Systems Development and Integration Branch (SDIB) Office of Information Resources Management (OIRM), September 12, 1995
- *Standards and Conventions*, SDIB OIRM, August 28, 1995
- SDLCM Methodology Procedure P-3101, Data Modeling
- SDLCM Methodology Procedure P-3051, System Requirements Specification
- SDLCM Methodology Procedure P-3053, System Operations Concept
- SDLCM Methodology Standard S-3171, Logical Design Document
- SDLCM Methodology Standard S-3172, Physical Design Document

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

## D. STANDARD

Three levels of Data Models may be produced by a Project or update by a Maintenance task following the SDLCM Methodology:

- The **Conceptual Data Model**, also known as the scope-setting version of the logical data model, identifies groupings of data important to the business situation that the Project addresses. This data model is documented in the System Requirements Specification (SRS).
- The **Logical Data Model**, completed version, presents a fully attributed and normalized data model and identifies the relationships among all data entities. This data model is documented in Logical Design Document
- The **Physical Data Model**, the design model, describes how data will be distributed to different processing nodes and structured to meet performance objectives in a specific physical implementation. This data model is documented in the Physical Design Document.

Table 3151-1 identifies the items that make up each of the three data models:

**Table 3151-1. Data Models**

Conceptual Model	Entity List
	Entity Definition
Logical Model	Entity List
	Entity Definition
	Entity Relationship Diagram (ERD)
	Attribute
	Attribute Definition
Physical Model	Information Type (Domain)
	Data Catalog
	Relational Schema
	Relational Table Structure Diagram

Refer to the *Systems Development CASE Tool Guidelines* and to SDLCM Methodology Procedure P-3101, Data Modeling, for descriptions of the processes used to create and refine data models.

### D.1 Definitions

An *entity* describes a person, place, thing, or event about which data is stored. It may refer to a tangible object in the real world, such as CUSTOMER or PRODUCT, or it may refer to an intangible business concept, such as ASSIGNMENT or CLAIM.

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

A *relationship* describes the connection between two entities. For example, “EMPLOYEE **is assigned to** Project” identifies two entities, EMPLOYEE and Project, and describes their relationship, **is assigned to**. The relationship is usually expressed in the form of a verb.

*Cardinality* describes the number of occurrences of a first entity that may exist in a relationship with a second entity. The possible types of cardinality are one-to-one, one-to-many, and many-to-many.

Entity *subtypes* are used to clarify otherwise confusing relationships. For example, BOAT and AUTO may be subtypes of an entity VEHICLE that can be either a BOAT or an AUTO. Each subtype has at least one unique relationship with another entity. For example, only a BOAT **is moored at** a DOCK.

An *attribute* is a data element that is assigned as a characteristic of a specific entity. As an example, the attributes of a data entity EMPLOYEE may include the employee number, name, address, job code, and department.

Two special types of attributes are the primary key and the foreign key. A *primary key* is an attribute or set of attributes that uniquely identifies an entity. In the example above, the employee number is the primary key. A foreign key is an attribute of one entity that is part of a primary key of another entity.

## D.2 Conceptual Data Model

Create the conceptual data model to identify and briefly describe the high-level entities that are within the scope of the Project. Note: Use the conceptual data model as the starting point for a more detailed definition of Project data requirements.

### D.2.1 ENTITY LIST

Identify and list only the high-level data entities important to the Project. Ensure that each listed entity is independent of any physical constraints; do not reflect the constraints of any database or data file.

Note: The entity list, generated as an activity of Component 1 (Define Initial Project Requirements) of the SDLCM Methodology, is based on partial understanding and will be refined as a data modeling activity of Component 3, Design the Solution.

Refer to Section 3.1, Data Objects, of the *Standards and Conventions* document for the conventions used to name data entities.

### D.2.2 ENTITY DEFINITION

Provide as much of the following information as is available to define each entity in the entity list:

- Name
- Identifier (primary key)

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

- List of included subtypes, if any
- Description (several sentences)
- Average volume at conversion
- Annual growth percentage
- Active life
- Estimated size in characters (preliminary)

Refer to Appendix C, Guidelines for Object Definitions, of the *Standards and Conventions* document for guidelines in defining data entities.

### D.3 Logical Data Model

Create the logical data model to identify all of the entities needed by the Project, their attributes, relationships, and the domains, or information types, that specify the format and value sets of the attributes.

#### D.3.1 ENTITY LIST

Refine the entity list from the conceptual data model to include all data entities needed by the Project. Ensure that each listed entity is independent of any physical constraints; do not reflect the constraints of any database or data file.

Refer to Section 3.1, Data Objects, of the *Standards and Conventions* document for the conventions used to name data entities.

#### D.3.2 ENTITY DEFINITION

Include the following information to define each entity in the entity list:

- Name
- Short Name
- Identifier (primary key)
- Description (several sentences)
- List of included attributes
- Average volume at conversion
- Annual growth percentage
- Active life
- Archived life
- Estimated size in characters
- Aliases (use with caution, if at all)

Refer to Appendix C, Guidelines for Object Definitions, of the *Standards and Conventions* document for guidelines in defining data entities.

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

### D.3.3 ENTITY RELATIONSHIP DIAGRAM

Develop one or more entity relationship diagrams (ERDs) that show the Project's entities and their relationships. Ensure that the ERD:

- Shows all entities (although not necessarily on one diagram)
- Resolves all many-to-many relationships into pairs of one-to-many relationships
- Identifies and defines all entity attributes of interest to the business process
- Removes subtypes, if applicable
- Identifies foreign keys, if applicable
- Is in at least third normal form (but may include derived data attributes)
- Identifies attribute domains
- Is independent of physical constraints such as those imposed by the target DBMS
- Is limited to the scope of the Project

Refer to SDLCM Methodology Standard S-xxxx, Entity Relationship Diagrams, and to Section 3.2.12, Entity Relationship Diagram Graphical Conventions, of the *Systems Development CASE Tool Guidelines* for guidance in creating the actual ERD.

### D.3.4 ATTRIBUTE AND ATTRIBUTE DEFINITION

Document the attributes of each entity. Briefly define each attribute by identifying its optionality and uniqueness. Refer to Section 3.1.6, Attribute, of the *Standards and Conventions* document for the conventions used to name and define data attributes.

Refer to SDLCM Methodology Procedure P-3101, Data Modeling, and Section 3.2.5, Attributes, of the *Systems Development CASE Tool Guidelines* for the process involved in identifying and refining the attributes selected for the data entities.

### D.3.5 ATTRIBUTE DOMAIN

Specify the characteristics that portray the structure of the attribute and define how it will be stored physically. Include the physical characteristics, such as length and character type information, about each attribute. Identify the domains of allowable values of the attribute, such as days of the week or months of the year, and the range of values.

Refer to Section 3.1.8, Information Type, of the *Standards and Conventions* document for the conventions used to characterize attributes.

<b>Subject</b> Data Models	<b>Type</b>	Standard
	<b>Identifier</b>	S-3151
	<b>Effective Date</b>	February 2002
	<b>Revision No.</b>	2

## D.4 Physical Data Model

Create the physical data model to depict all aspects of how data will be stored in the computer system.

### D.4.1 DATA CATALOG

Document the results of translating the data entity definitions to a baseline data element model.

Each object (entity type, attribute type, relationship type, information type, and subtype sets) in the logical data model relates to a corresponding object in the physical model.

Refer to Section 4.2.1, Translation to the Data Catalog, of the *Systems Development CASE Tool Guidelines* for further information on the data catalog.

### D.4.2 RELATIONAL SCHEMA

Provide a relational schema diagram to depict the relational tables derived from the ERD and the relationships between the tables.

Refer to Section 4.1.1, Relational Schema, of the *Standards and Conventions* document for the conventions used to document the schema.

### D.4.3 RELATIONAL TABLE STRUCTURE DIAGRAM

Provide a relational table structure diagram for each entity type contained in the data catalog.

Refer to Section 4.3.1.2, Data Structure, of the *Systems Development CASE Tool Guidelines* for information on how the translation is accomplished.

Refer to Section 4.1.2, Relational Table, of the *Standards and Conventions* document for the conventions used to document the relational table structure.