BRITISH STANDARD

BS 7738-1: 1994

Incorporating Amendment No. 1

Specification for

Information systems products using SSADM (Structured Systems Analysis and Design Method) —

Part 1: Implementation of SSADM version 4

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/15, upon which the following bodies were represented:

British Computer Society
CCTA (the Government Centre for Information Systems)
Department of Trade and Industry (IT Standards Unit ITD6A)
Institution of Electrical Engineers
International Computers Limited
Lloyd's Register of Shipping
Ministry of Defence
National Computing Centre Ltd.
Oracle Corporation (UK) Ltd.

The following bodies were also represented in the drafting of the standard, through BSI Panel IST/15/-/200:

Leeds Metropolitan University
The International SSADM Users Group
Information Systems Examination Board
SSADM Design Authority Board
Computer Services Association

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Standards Board and comes into effect on
15 June 1994

The following BSI references relate to the work on this standard:
Committee reference IST/15
Draft for comment 91/64274 DC

ISBN 0 580 22849 5

Amendments issued since publication

Amd. No.	Date	Comments
8939	December 1995	Indicated by a sideline in the margin

Contents

Foreword

This British Standard has been prepared under the direction of the Information Systems Technology Standards Policy Committee.

NOTE To comply with this specification, the user has to comply with all its relevant requirements. He may depart from recommendations, but this would be on his own responsibility and he would be expected to have good reasons for doing so.

The Structured Systems Analysis and Design Method (SSADM) is a systematic approach to the analysis and design of information systems applications.

SSADM Version 4 is a product-oriented specification of what practitioners and managers need to do to undertake the tasks of systems analysis and design. It describes an effective process of system development, which users can adapt to suit project needs.

Relationship between the standard and the method

Presently under development is:

BS 7738-2, *Implementation of SSADM version 4.2.*

Other parts may be added to BS 7738 as and when other versions of the SSADM manuals are published.

The details of the method itself are contained in the current version of the SSADM manuals (SSADM Version 4 NCC/Blackwell, 1990 [1]). This British Standard is designed to be self-contained. It will not depend upon the SSADM manuals in its role as a standard. It will not, however, be independent of the method (and of the manuals in a wider sense). The standard will be the responsibility of BSI; the method will be the responsibility of the Design Authority Board (DAB). The two bodies understand the importance of maintaining a clearly understood relationship with mutual recognition of purpose by each organization.

Control of the method

The method was developed by the CCTA[1)], it is now controlled by the Design Authority Board (DAB). The DAB provides a forum for discussion of the method, its application and evolution. The DAB includes representatives from all major groups with interests in the method: government and private sector, customers and suppliers of SSADM-related products and services and members of relevant professional bodies. The relationship between the standard and the method will be maintained by liaison between the DAB and the BSI Technical Committee; a representative of the Technical Committee being a member of the DAB. The DAB members are solely responsible for the method, its publication in the form of the manuals and supporting documentation, and any changes to the method.

BSI acknowledges that the copyright in SSADM is the property of the Crown. This British Standard is published with the consent of CCTA and the permission of the Controller of HMSO.

Change control

The evolution of the standard is the responsibility of the Technical Panel (IST/15/-/200) of BSI, reporting directly to Technical Committee IST/15 Software Engineering. The standard will be revised and updated to reflect changes in the method as agreed by DAB and as published in the manuals. In addition, the standard itself will be reviewed periodically, and in response to suggestions for revision received from users of the standard.

This edition of this standard makes specific reference to the 1990 version of the manuals for SSADM. Future releases of the standard will similarly make explicit reference to the prevailing edition. If the method changes, as reflected in the manuals, the Technical Panel will consider the changes and amend the standard accordingly. The minimum change would be an alteration to the normative reference.

[1)] Formerly the Central Computer and Telecommunications Agency of HM Treasury, now CCTA — The Government Centre for Information Systems.

Production certification. Users of this British Standard are advised to consider the desirability of third party certification of product conformity with this British Standard based on testing and continuing product surveillance which may be coupled with assessment of a supplier's quality systems against the appropriate Part of BS 5750.

Enquiries as to the availability of third party certification schemes are forwarded by BSI to the Association of Certification Bodies. If a third party certification scheme does not already exist, users should consider approaching an appropriate body from the list of Association members.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i to vi, pages 1 to 136, an inside back cover and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

blank

Introduction

Purpose of the SSADM standard

The Structured Systems Analysis and Design Method (SSADM) is a method widely used in the United Kingdom for the development of information systems. In developing such systems, many organizations require the procurement of products and services from external suppliers. In this context there is a need for a clearly established and uniformly recognized statement of the specification for information systems products using SSADM. This standard provides a clearly identified basis for the interface between customers for, and suppliers of, SSADM application products for information systems development. Application products are those normally associated with the development of a computer system.

Aligning with the modular structure of SSADM, there is recognition of the necessity to be able to locate products within the identified modules and stages of the method, thereby enabling procurement of coherent groups of products.

The establishment of a standard in this fashion is intended to promote consensus and good practice in information systems development.

Specifically this British Standard will:

a) contribute to an open market in SSADM products and services, by providing a recognized specification of products;

b) provide a basis for negotiation between customers for and suppliers of SSADM products and services;

c) facilitate the transfer of products across stages and modules of the method; supporting the flexibility of SSADM;

d) facilitate the procurement of services, training, tool support, and other related support facilities for an SSADM environment.

NOTE Guidance on the management of product design can be found in BS 7000.

Readership

The audience for the SSADM manuals includes practitioners, user managers, trainers, service and product providers, project and technical managers and IT directorate staff, standards makers and co-ordinators, data administrators information managers and academics. This British Standard is aimed at a subset of this group: those involved directly in the clear specification of required SSADM products delivered by suppliers to customers in a contractual context. It is, however, specifically aimed at those non-specialists involved in such contexts, who would not expect to have to be familiar with the detailed knowledge of SSADM contained in the manuals. It will, therefore, not be necessary for users of this standard to have a detailed knowledge of the method itself.

A specification for products

This British Standard has been produced as a specification for products (see BS 0-3). The scope for such a standard is to specify complete or partial products in terms of certain key characteristics. This standard is not intended to be a specification for the method (SSADM) itself.

Structure of SSADM

SSADM is modular in structure. (See Figure 1 for the modules and stages involved.) It consists of five modules, each being composed of one or more stages. Completion of any stage or module is marked by acknowledged completion of a defined and agreed set of products. Initiation of any stage or module is marked by acceptance of inputs; these inputs may be products from preceding stages or modules, or they may be derived from non-SSADM origins.

SSADM activities

SSADM stages are further subdivided into steps, each step consisting of a number of activities or tasks. The details relating to the performance of these tasks are contained in the SSADM manuals. They are not of direct concern to this standard for the specification for products.

Conformity by presentation

The manual for SSADM version 4 suggests notations and conventions for aspects of products that can be represented graphically. In this standard where a product can be represented graphically the product entry in section **3** includes details of presentation. The aim of such detail is to present illustrations of components of the product. It is not the aim of this standard to offer completed examples of the products themselves, but to define the preferred notation for component parts.

This "preferred presentation" is a stronger view of presentation than that defined in the manuals. It permits the standard to offer a baseline for readability and comprehensibility across a range of contexts: between customers, suppliers and practitioners.

Conformity to the standard may, however, be accomplished in terms of individual products without precluding alternative presentation conventions, as long as the products delivered can be shown to fulfil the other criteria specified in the standard.

In most cases the preferred presentation has been that exemplified in the SSADM manuals. In other cases it has been necessary to select one amongst several options for the preferred presentation.

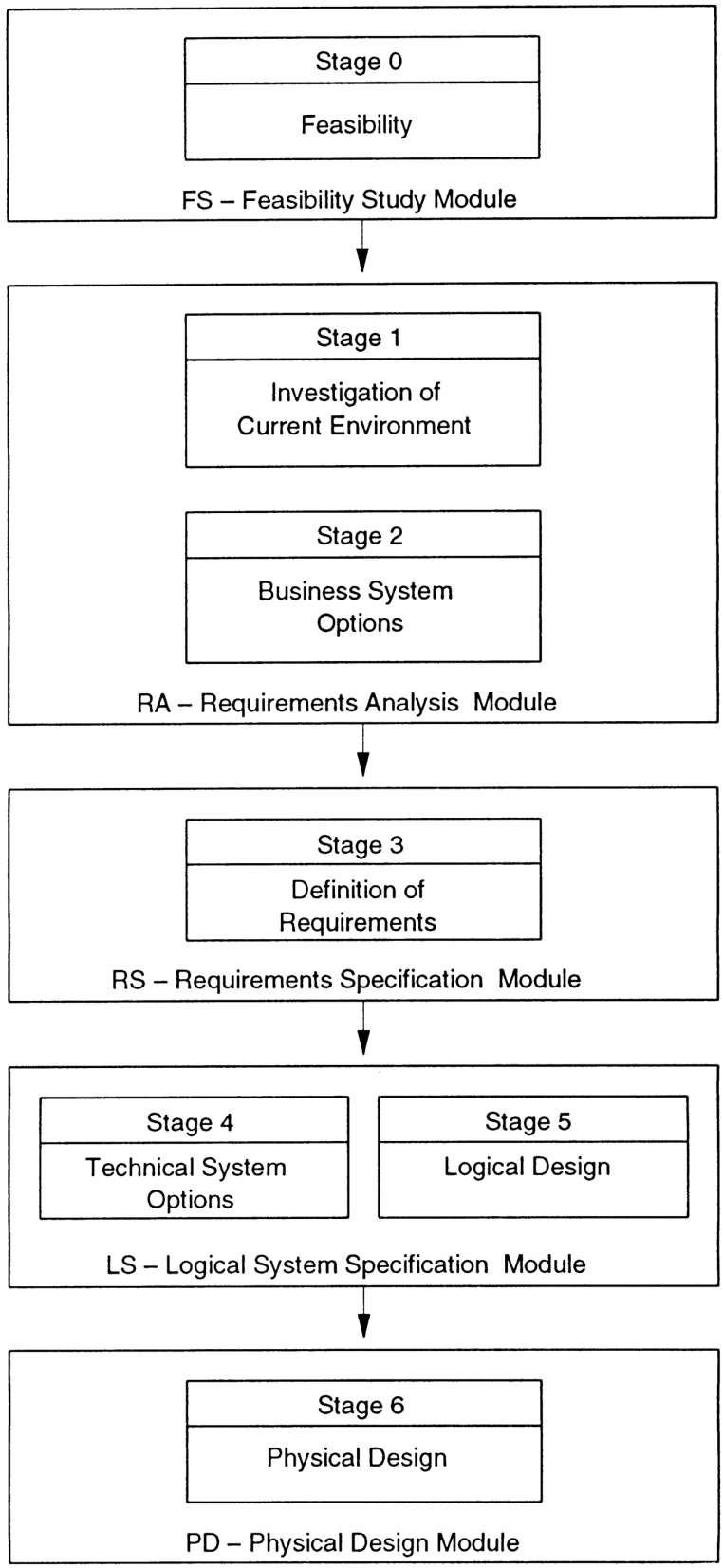


Figure 1 — SSADM Modules and Stages

Section 1. General

1.1 Scope

This British Standard specifies the application products encompassed by the use of SSADM in the development of information systems. The complete list of products covered by this standard is given in Annex A.

This British Standard focuses on the interface between customer and supplier when supply of products and services relating to systems development, using SSADM, is the key concern. The focus of the standard is on products, and the means for assessing delivery of completed products.

The products produced during an SSADM development project can be classified in a number of ways:

a) by Module or Stage;

b) by concept and technique;

c) alphabetically.

Annex B groups products according to the Module/Stage in which they are produced. Section **3** gives more detailed entries for products grouped by concept and technique; this precludes the necessity to repeat common details for products which occur in more than one Stage or Module of the method. Annex A lists products alphabetically, cross-referenced to the relevant section in this standard.

1.2 References

1.2.1 Normative references

This British Standard incorporates, by dated or undated reference, provisions from other publications. These normative references are made at the appropriate places in the text and the cited publications are listed on the inside back cover. For dated references, only the edition cited applies; any subsequent amendments to or revisions of the cited publication apply to this British Standard only when incorporated in the reference by amendment or revision. For undated references, the latest edition of the cited publication applies, together with any amendments.

1.2.2 Informative references

This British Standard refers to other publications that provide information or guidance. Editions of these publications current at the time of issue of this standard are listed on the inside back cover, but reference should be made to the latest editions.

1.3 Definitions

For the purposes of this British Standard, the following definitions apply.

NOTE See **C.1** for an explanation of the conventions used in this clause.

1.3.1
acceptance (testing) criteria

Object

criteria setting out the details of how the final implemented system is required to perform specific tasks so that the users are satisfied that it caters for their needs

NOTE Initial details will be compiled based on SSADM documentation, but will require further development outside of SSADM.

1.3.2
access path (A)

Object

route to be taken through the Logical Data Model from an entry point to the entity, or entities, required for a particular piece of processing

NOTE The Logical Data Model is constructed with a view to implementing update processing, so update processing access paths are not documented separately. Enquiry processing access paths need to be documented explicitly.

1.3.3
activity (N)

General

that which, within SSADM steps, effects transformation in one or more products

1.3.4
activity descriptions (A)

Product

details of what the transformation (in one or more products) is, the inputs, the references, and the participants

1.3.5
activity network

Product

all of the activities placed into logical sequence, thus enabling timescales to be estimated and work to be scheduled

1.3.6
ad-hoc enquiry (A)

Object

enquiry which is not pre-defined but is created by the user as and when it is needed

NOTE Ad-hoc enquiries may be documented for each applicable user role as individual function definitions.

1.3.7
analysis of requirements

Product

module Product formed from the Requirements Analysis Module

NOTE It consists of the Current Services Description, Requirements Catalogue, User Catalogue and the Selected Business System Option.

1.3.8
application development standards

Product

standards defining the standards which apply to the physical design and development activities for this project/application

1.3.9
application naming standards

Product

standards defining the naming conventions for all aspects of the application under development, with particular emphasis being placed on constraints imposed by the (physical) implementation environment

1.3.10
application style guide

Product

set of standards, covering the user interface, to be followed within a particular application development

NOTE 1 This document is based on the Installation Style Guide and tailored to the specific needs of this project.
NOTE 2 This is also known as Style Guide.

1.3.11
attribute (A)

Object

characteristic property of an entity type, that is, any detail that serves to describe, qualify, identify, classify, quantify or express the state of an entity

NOTE 1 If possible the logical attributes for an entity will be translated directly into data items within the physical representation.
NOTE 2 An attribute may be optional for an entity type, meaning that it does not apply to all occurrences of the entity.

1.3.12
attribute/data item description

Product

each description within this set documenting all known details about an attribute (data item) in the logical system

NOTE Details are later translated into the physical data item(s) for the implementation of the system. Also known as Data Item/Attribute Description.

1.3.13
batch

Object

logical grouping of events (or functions) which always happen within the same time-frame

NOTE See also **1.3.102**.

1.3.14
bottom-level process

Object

process on a Data Flow Diagram which is not further decomposed into another Data Flow Diagram

NOTE See also **1.3.61**.

1.3.15
business system option (BSO)

Technique

the means by which users agree desired functionality of the new application with developers

NOTE BSOs are used to define the functionality needs and the boundary for the system, with reference to the business needs.

1.3.16
business system options

Structural Model Element. Stage 2

the Stage aiming to take the Requirements Catalogue, Current Services Description and User Catalogue and use this information as the basis on which to decide the most appropriate way for development to meet the business needs

NOTE This Stage has two Steps:
a) Define Business System Options;
b) Select Business System Options.

1.3.17
business system options

Product

set of Business System Options which is compiled so that a selection can be made

1.3.18
capacity planning

Technique

predicting the (hardware/software) configuration required to satisfy the constraints and requirements of the proposed system

NOTE It is also used to assist in the development of service level agreements (outside of SSADM).

1.3.19
capacity planning input

Product

processing and data information passed outside of SSADM activities to capacity planning techniques, explicitly during development of Technical System Options

1.3.20
command structure

Product

product showing the directions that control can take when a user decides to complete or terminate a particular dialogue

NOTE This allows navigation to be implemented with or without menus.

1.3.21
common process

Object

aspect of processing identified as common to several processes (or functions) within a SSADM system specification

NOTE Functionality such as a calculation of entitlement under a benefits scheme or discount arrangements may occur in more than one process or function. Details for each common process need to be documented in an Elementary Process Description with the relevant cross-references. These may be carried forward as part of the Function Definitions within the Requirements Specification. Common processes are vital components of the Physical Design, documented in the Function Component Implementation Map.

1.3.22
component

Object

object represented in the Function Component Implementation Map

NOTE It is usually classified as one of the following:
- — super function;
- — function;
- — I/O process;
- — database process;
- — common process.

1.3.23
composite data flow

Object

a data flow on a Data Flow Diagram which is decomposed into simpler data flows on a lower-level Data Flow Diagram

1.3.24
context diagram

Product

diagram drawn to illustrate the initial scope of the proposed system

NOTE The diagram concentrates on the major inputs and outputs of the system and shows the external sources and recipients of system data.

1.3.25
core SSADM

General

central, fundamental part of SSADM which refers to the following Modules:

- — Feasibility Study (FS);
- — Requirements Analysis (RA);
- — Requirements Specification (RS);
- — Logical System Specification (LS);
- — Physical Design (PD).

1.3.26
cost/benefit analysis

Product

objective way to judge the merit of one option against another

NOTE This forms the vital financial part of the specification of each option (business or technical).

1.3.27
current environment

General

current services (computer and manual) and the wish list for the future system

NOTE Where no current system exists this will simply reflect the wish list for the new system. This is a generic term.

1.3.28
current environment logical data model

Product

product providing a detailed description of the information used or produced by the current environment

NOTE It is developed during the Requirements Analysis Module and forms part of the Current Services Description. (See also **1.3.117** and **3.6.5**.)

1.3.29
current physical data flow model

Product

model showing how the current services are organized and processing is undertaken

NOTE An overview of current services is provided by documenting only the level 1 Data Flow Diagram. (See also **1.3.35** and **3.2.4**.)

1.3.30
current services

General

services encompassing all existing processing within the business area being addressed by the project, irrespective of whether those functions are manual or computerized

NOTE This is a generic term.

1.3.31
current services description

Product

product providing the details of the logicalized current system which, with the Requirements Catalogue and User Catalogue, is output from Stage 1: Investigation of Current Environment

1.3.32
data catalogue

Product

central repository for all the descriptive information about items of data

NOTE 1 This includes physical details which may be found during data flow modelling activities as well as physical design activities. Logical data modelling will provide information about attributes (the logical equivalent to data items).
NOTE 2 See also **1.3.12**, **1.3.101**, **3.6.1** and **3.6.4**.

1.3.33
data flow

Object

object showing where data is being passed between different elements on a Data Flow Diagram

NOTE The name associated with the data flow should be meaningful to those reviewing the Data Flow Diagram. Data flows will pass into and out of the system and between processes (generally via data stores except on the Current Physical Data Flow Model where process-to-process flows may reflect inadequacies in the current system). When the system boundary is being defined there may even be data flows between external entities. At the lowest level of the Data Flow Diagram these are simple data flows, though they may be combined into composite data flows on higher level diagrams.

1.3.34
data flow diagram

Product

diagram showing how services are organized and processing is undertaken

NOTE It should be a simple diagram that is readily understood, so that it can act as an effective means of communication between analysts and users. (See also **1.3.35** and **3.2.4**.)

1.3.35
data flow model

Product

set of Data Flow Diagrams and their associated documentation

NOTE The diagrams form a hierarchy with the Data Flow Diagram Level 1 showing the scope of the system and the lower level diagrams expanding the detail as appropriate. Additional documentation provides a description of the processes, input/output data flows and external entities.

1.3.36
data flow modelling (DFM)

Technique

technique used to help define the scope of the system and ensure that the analysts have a clear understanding of the users' problems and requirements

NOTE The technique is used to build a model of the information flows and not to define the detail of the processing performed by the system.

1.3.37
data item (A)

Object

element or field of a data store or component of inputs or outputs (which may be held within an entity type on the Logical Data Structure)

NOTE A data item is any detail that serves to describe, qualify, identify, classify, quantify or express the state of the data store. The logical attributes for an entity will, in general, translate directly into data items within the physical representation.

1.3.38
data store (A)

Object

collection of any type of data in any form as represented on a Data Flow Diagram

NOTE This may be a computer file or a box of documents or any other means of storing data. Data stores may be standing, long-term files such as sales and purchase ledgers, or short-term accumulations such as daily batches of documents. Data stores on logical DFDs cannot be manual.

1.3.39
database management system (DBMS)

General

mechanism for managing data held within a computerized system

NOTE Conceptually data is held within one file regardless of how the content is physically organized.

1.3.40
database process

Object

a physical process to be applied to the implemented database

NOTE The valid processes will vary depending on the DBMS used for implementation. They will include read and write operations to varying levels of sophistication.

1.3.41
DBMS data storage classification

Product

product for analysing and recording data storage and retrieval mechanisms of DBMS or file handler

1.3.42
DBMS performance classification

Product

product recording the factors which impact on the performance of a DBMS or file handler

1.3.43
definition of requirements

Structural Model Element. Stage 3

stage aiming to take the Analysis of Requirements and produce the Requirements Specification within the boundary identified in the Selected Business System Option

NOTE This Stage has eight Steps:
— Define Required System Processing;
— Develop Required Data Model;
— Derive System Functions;
— Enhance Required Data Model;
— Develop Specification Prototypes;
— Develop Processing Specification;
— Confirm System Objectives;
— Assemble Requirements Specification.

1.3.44
detail entity (A)

Object

the entity type at the several end where two entity types are connected by a 1 : m relationship, a single instance of one entity type is related to several instances of the other

NOTE See also **1.3.132**.

1.3.45
determinant (A)

Object

values for some of the attributes within an entity occurrence directly dependent on the value of a group of specific attributes (possibly a group of one), which effectively identify this occurrence of the entity

NOTE The identification attributes are known as the determinant within relational data analysis.

1.3.46
dialogue

Object

on-line activity required by a particular user role so that they can action (perform) a particular function

1.3.47
dialogue control table (A)

Product

table used to identify and capture the navigation between the logical groupings of dialogue elements

1.3.48
dialogue design (DD)

Technique

technique used to define the on-line activity of the system

NOTE Dialogues are identified as part of the Requirements Specification and then logically designed explicitly as part of the Logical Design. Physical dialogue design activities are undertaken during the physical design activities to complete the design prior to system implementation.

1.3.49
dialogue element

Object

section of an input or output data flow which may consist of many data items

NOTE Each dialogue element is represented as a box on a Dialogue Structure.

1.3.50
dialogue element description

Product

product used to describe a dialogue element

NOTE A set of these provides the detailed documentation for a Dialogue Structure.

1.3.51
dialogue identification (A)

Technique

technique used to identify the complete set of the requirements for dialogues within the proposed system

NOTE Dialogues which are identified as critical in function definition may be subjected to specification prototyping.

1.3.52
dialogue level help

Product

product used to detail the level of help that the user (user role) requires to progress through this dialogue

1.3.53
dialogue structure (A)

Product

diagrammatic representation of a dialogue

1.3.54
dialogues

Product

product used to package together details of all identified dialogues within the system

1.3.55
document flow diagram

Product

diagram showing how documents pass around the system

NOTE This may be the initial diagram drawn within the data flow modelling technique to assist in defining/identifying the boundary of the system. This diagram will be produced if the current system is predominantly clerical and involves the passing of information using forms or other documents.

1.3.56
domain

Object

pool of values from which the actual values for an attribute can be drawn

1.3.57
effect (A)

Object

change caused to a single entity occurrence because of a single event

NOTE An effect can be one of four kinds: create (birth), modify (update), logical delete (death) or update of state indicators.

The effect has to be qualified where an occurrence of an entity is affected in one of severally mutually exclusive ways by an event.

1.3.58
effect correspondence diagram (ECD)

Product

diagram showing all the effects an event has on data within the system and how those effects impact upon each other

1.3.59
effect correspondence diagramming

Technique

technique used to analyse and document how effects are related to each other for each event

1.3.60
effect qualifier (N)

Object

object given after the event name to qualify the effect where an occurrence of an entity is affected in one of severally mutually exclusive ways by an event

NOTE The qualifier is bracketed after the event name and such brackets should differ in nature to those used to qualify an entity by its role.

1.3.61
elementary process

Object

lowest-level process on a Data Flow Diagram which relates directly to the user (business) environment in which the system (computerized or manual) is to be operated

NOTE 1 It will assist in the identification of functions.
NOTE 2 Whole, or partial, elementary processes may be regarded as common across a number of processes, in which case they will be separately described.

1.3.62
elementary process description (EPD)

Product

description of the business environment in which the process is trying to operate

NOTE A requirement for common processing may also be described within an Elementary Process Description and cross-referenced to the elementary processes or functions which use it.

1.3.63
end-module product (N)

Object

a product which is delivered at the end of a Module, having been created or updated within that Module

1.3.64
end-stage product (N)

Object

a product which is delivered at the end of a Stage, having been created or updated within that Stage

1.3.65
enquiry access path (EAD)

Product

route through the Logical Data Model from an entry point to the entity type(s), required for a particular enquiry

1.3.66
enquiry element

Object

element which requires information to be read from the database but involves no update processing

NOTE 1 Some update functions contain enquiries as well as updates (events); these elements are also called enquiry elements.
NOTE 2 This is also known as enquiry.

1.3.67
enquiry function (A)

Object

function which requires information to be read from the database but involves no associated update processing

1.3.68
enquiry process model (EPM) (A)

Product

model consisting of a structure diagram for an enquiry processing requirement and the associated Operations List

NOTE The structure is based on the Enquiry Access Path and the appropriate I/O Structure.

1.3.69
enquiry trigger

Object

data items that have to be input to the system to initiate an enquiry

1.3.70
entity (A)

Object

something, whether concrete or abstract, which is recognized as being of importance within the environment under investigation and which is capable of an independent existence

NOTE 1 Each occurrence of an entity (type) should be uniquely identifiable.
NOTE 2 Logical data modelling identifies types of entity, not individual occurrences, i.e. tenant and applicant not John Smith. (See **1.3.76** and **1.3.78**.)

1.3.71
entity description

Product

product documenting all of the details concerned with an entity type on the Logical Data Structure, including details of state indicators which are applied during entity life history analysis

NOTE There will be associated Relationship Descriptions for each related entity type on the Logical Data Structure.

1.3.72
entity-event modelling (EEM)

Technique

combination of entity life history analysis and effect correspondence diagramming

1.3.73
entity life histories (ELH) (A)

Product

set of structures combining possible lives of possible occurrences of an entity type

1.3.74
entity life history (A)

Object

object that charts the events that may cause any one occurrence of an entity type to be changed in any way

NOTE It shows the valid structure of events (initially identified through use of data flow modelling and function definition techniques) affecting an entity type on the Logical Data Structure.

1.3.75
entity life history analysis

Technique

technique bringing together and validating the processing and data requirements of the system

NOTE This is done by investigating the Required System Logical Data Model. Events which affect the entity lives are documented showing the sequence of effects which take place.

1.3.76
entity occurrence (N)

Object

specific (uniquely identifiable) instance of an entity type

1.3.77
entity roles

Object

the entity deemed to be assuming different roles if a single event affects more than one occurrence of a particular entity type and the effects are different for each occurrence affected

NOTE Each possible role has to be separately identified on the ELH diagram for the relevant entity, as separate processing will have to be specified for each role. The qualifier for the role is placed in brackets after the event name to provide a complete identification of the entity role and brackets should differ in nature to those used to qualify an event by its effect.

1.3.78
entity type (N)

Object

something, whether concrete or abstract, which is recognized as being of importance within the context of the system being developed, and which is capable of an independent existence

NOTE Each occurrence of an entity type should be uniquely identifiable.

1.3.79
event (A)

Object

whatever triggers a process (on a Data Flow Diagram) to update the values or status of the system

NOTE An event may cause more than one entity occurrence or type to be changed. In the logical system, an event initiates an update process.

1.3.80
event/entity matrix

Product

grid used to identify which entity types are affected by a particular event

NOTE 1 It provides two checks: that each event affects at least one entity type, and that each entity type is affected by at least one event.
NOTE 2 Also known as Entity/Event Matrix.

1.3.81
evolving product (N)

Object

product which is delivered but expected to evolve or be the subject of further refinement in later Stages or Modules

NOTE A distinction is made between end-stage/end-module products which are delivered as complete at the end of the Stage or Module.

1.3.82
exclusive relationship group

Object

group where the participation of an entity occurrence in one relationship precludes its participation in one or more other relationships

1.3.83
expression

General

statement which can be evaluated

NOTE The statement may be forming a comparison between individual items. An expression may itself consist of several sub-expressions linked by “and” or “or”.

1.3.84
external entity

Object

source or recipient (or both) of data which exists outside the boundary of the defined system but which communicates with the system

NOTE An external entity may be another system, an organization, an individual or a group of people. These are documented within the Data Flow Model.

1.3.85
external entity description (EED)

Product

brief explanation of the relevance of an external entity in relation to the existing or proposed system

NOTE The detail will cover responsibilities or functions of the external entity and any constraints on the interface with the proposed system.

1.3.86
feasibility

Structural model element. Stage 0

stage aiming to investigate the requirements laid down within the Project Initiation Document and suggest the way ahead

NOTE This Stage has four Steps:
— Prepare for the Feasibility Study;
— Define the problem;
— Select Feasibility Options;
— Assemble Feasibility Report.

1.3.87
feasibility

Technique

technique used to investigate the possibilities for future work in a particular area of the business

NOTE The scope of the study will be laid down within the Project Initiation Document. Much of the analysis will be done using other techniques including logical data modelling, data flow modelling, requirements definition and dialogue design.

1.3.88
feasibility options

Product

set of Feasibility Options which is compiled so that a selection can be made

NOTE Each option documents the functions to be incorporated and details implementation requirements. Each description is textual with some planning information.

1.3.89
feasibility report

Product

module Product from the Feasibility Study Module, documenting the possible approaches to the system development and assessing the impact of each so that the most appropriate way ahead can be fully investigated

1.3.90
feasibility study module (FS)

Structural Model Element

module whose objective is to produce the Feasibility Report which will suggest the way ahead for the project

NOTE 1 The activities form a short assessment of a proposed information system to determine whether the system would be feasible and appropriate to the business needs of the organization. Feasibility is assessed in terms of the managerial, business, financial, technological and cultural needs of the organization.
NOTE 2 This Module has one Stage: Stage 0: Feasibility.

1.3.91
fragment

Object

defined processing element which has a specific identity, purpose, set of inputs, outputs and correspondences

NOTE It may be at any level. 1 : 1 with component or 1 : 1 with operation, e.g. Read Master. It may be in support of a process, or a data group, screen or error message.

1.3.92
function

Object

set of system processing which the users wish to schedule together, to support their business activity

1.3.93
function component implementation map (FCIM)

Product

classification and specification of all implementation fragments for all function components defined in the Function Definitions to meet the processing requirements

NOTE See also **1.3.22**.

1.3.94
function definition

Product

description of the function providing a cross-reference to other associated SSADM products

1.3.95
function definition (FD)

Technique

identification and documentation functions which are the units of processing carried forward to physical design

1.3.96
function definitions

Product

packaging of all details about functions to be included in the Requirements Specification

NOTE These details are further expanded during physical design activities. Basic information about a function consists of a Function Definition with one or more I/O Structures. Further information may be found in Enquiry Access Paths (for enquiry functions) and (common) Elementary Process Descriptions.

1.3.97
function type

General

category of the function

NOTE There are three ways of categorizing functions:
— processing requirements: update or enquiry;
— access details: on-line or off-line;
— method of initiation: user or system.

1.3.98
functional area (A)

General (in DFM)

view that the user has of the processing system based upon the tasks which they undertake

NOTE Each of these views will represent one or more Functional Areas depending upon how widely the individual user uses the system. Each Functional Area will be directly related to a specific task which the user undertakes.

1.3.99
functional requirement

Object

requirement for a particular facility or feature which describes what the system should do, e.g. include updates, enquiries and reports

1.3.100
grouped domain

Object

pool of values associated with validation rules and formats common to several attributes

1.3.101
grouped domain description

Product

products used to document validation rules and formats common to several attributes

1.3.102
group of events

Object

collection of event occurrences containing more than one event type

NOTE There may be more than one occurrence of each event type. The term is used to describe the batching or grouping of data for input to a function.

1.3.103
help

General

facility provided in the implemented system to guide the user through the options available to them at any point in time

NOTE Dialogue level help may be associated with Dialogue Structures so that the need to consider (and implement) these features can be passed forward into the physical design activities.

1.3.104
I/O descriptions

Product

product used to document data flows which cross the Data Flow Model system boundary

1.3.105
I/O process

Object

process, documented in the physical design, for producing detail to or accepting information from a specific piece of functional processing

1.3.106
I/O structure

Product

product documenting the input to and outputs from a function, or part of a function

NOTE See also **3.5.5** and **3.5.6**.

1.3.107
I/O structures (for all functions)

Product

packaging of all I/O Structures for all identified functions

NOTE This product exists because some SSADM activities do not require the complete set of documentation for all functions.

1.3.108
impact analysis

Product

description of the effects of the option (business or technical) on the user environment covering issues concerned with organization, procedures, and implementation factors

NOTE This product is used to document the ramifications of pursuing a particular course of action.

1.3.109
installation development standards (A)

Product

standards documenting the criteria which should be used during the development of all Information Systems within the organization/installation

1.3.110
installation style guide

Product

set of standards about the nature, approach and style of the human factors aspects of computerized systems

NOTE 1 The standards should be followed by all projects undertaken within an organization.
NOTE 2 Also known as Style Guide.

1.3.111
integrity errors

Object

inconsistencies in a database

NOTE The data in a database have to be consistent for the system to be able to function correctly. Where inconsistency arises the database is not maintaining referential integrity of the database.

1.3.112
interim product (N)

Object

product created during a stage or module, but which is not specifically offered as a delivered product at the end of the stage or module itself

NOTE It is most often an in-progress product created as part of the continuing development process, and used in the derivation of other (end-stage, end-module) products.

1.3.113
investigation of current environment

Structural Model Element. Stage 1

stage aiming to look at the existing services and requirements and produce a logical view of the processing requirements

NOTE This Stage has six Steps:
— Establish Analysis Framework;
— Investigate and Define Requirements;
— Investigate Current Processing;
— Investigate Current Data;
— Derive Logical View of Current Services;
— Assemble Investigation Results.

1.3.114
key (A)

Object

attribute or a combination of attributes which can be used to identify (uniquely) each occurrence of an entity type

1.3.115
list

Object (in DBMS)

representation of relationship between master and detail entities by a list of keys, including the key of the master and the key of each detail entity, with pointers connecting each entry to the next entry

1.3.116
logical data flow model

Product

set of Data Flow Diagrams and their associated documentation produced during logicalization in Stage 1

NOTE See also **1.3.35**.

1.3.117
logical data model

Product

product providing an accurate model of the information requirements of all or part of an organization

NOTE This serves as a basis for file and database design, but is independent of any specific implementation technique or product. The Logical Data Model consists of a Logical Data Structure, Entity Descriptions and Relationship Descriptions. Associated descriptions of attribute/data items and grouped domains are maintained in the Data Catalogue.

1.3.118
logical data modelling (LDM)

Technique

technique used to produce a conceptual model of the information requirements of all or part of an organization

1.3.119
logical data store/entity cross-reference (A)

Product

matrix showing the correspondence between logical data stores in the Data Flow Model and the entity types on the Logical (current or required) Data Model

NOTE 1 This is used to ensure that a main data store corresponds to an entity type or group of types. Also each entity type on the Logical Data Model has to be held completely within one and only one main data store. There may be two variants of this product: Logical, Requirements Specification.
NOTE 2 Also known as Entity/Logical Data Store Cross-reference.

1.3.120
logical data structure (LDS) (A)

Product

diagrammatic representation of the information needs of an organization in the form of entity types and the important relationships between them

1.3.121
logical database process design (LDPD)

Technique

technique used to translate the information gathered during the Requirements Specification Module into a logical specification which can then be translated into a physical design for the system in any implementation

NOTE This also provides a logical definition of the system which assists in maintenance of the implemented system. This technique covers the definition of the processing for data input to, and output from, the database.

1.3.122
logical design

Product

stage Product from Stage 5: Logical Design, packaging the logical view of processing with the Required System Logical Data Model and the Requirements Catalogue

1.3.123
logical design

Structural Model Element. Stage 5

stage aiming to take the Requirements Specification and further develop the processing requirements in such a way as to be implementation independent prior to physical design activities

NOTE This Stage has four Steps:
— Define User Dialogues;
— Define Update Processes;
— Define Enquiry Processes;
— Assemble Logical Design.

1.3.124
logical grouping of dialogue elements (LGDE)

Object

dialogue elements within a particular Dialogue Structure grouped together where input/output requirements suggest that there are benefits to be gained by doing so, such as simpler processing

1.3.125
logicalization

Technique

sub-technique of data flow modelling, used in Stage 1, which converts the Current Physical Data Flow Model into a logical model with no physical implications

1.3.126
logical key (A)

Object

form of key meaningful to the users of the system and which is usually allocated by them

NOTE Each instance of such a key uniquely identifies one record. It does not tell the system where the data is physically stored. (See also **1.3.162**.)

1.3.127
logical/physical data store cross-reference (A)

product

matrix that is used within the data flow modelling technique (logicalization) to cross-reference the logical view of data organization from the current (physical) services

1.3.128
logical process model

Product

product used to package all processing details within the Logical Design

1.3.129
logical system specification

Product

module product from the Logical System Specification Module consisting of the Selected Technical System Option, the Technical Environment Description and the Logical Design

1.3.130
logical system specification module (LS)

Structural Model Element

module with the objective to produce the Logical System Specification

NOTE The Selected Technical System Option and the Technical Environment Description define the scope of the physical implementation. This detail has to be consistent with the Logical Design. The Application Style Guide is also developed during this Module. This Module has two Stages:
— Stage 4. Technical System Options;
— Stage 5. Logical Design.

1.3.131
main data store

Object

representing elements on Data Flow Diagrams; the data that is held centrally in a system so that it can be used by a number of different processes

NOTE Entities on the Logical Data Model have to reside in one and only one main data store. Each main data store has to have one or more entities associated with it.

1.3.132
master entity (A)

Object

the entity at the single end where two entity types are connected by a 1 : m relationship and a single instance of one entity is related to several instances of the other

NOTE During Stage 3, all relationships on the LDM have to be represented as 1 : m relationships. This provides a clear hierarchical structure to provide the necessary input to the following techniques: entity life history analysis, logical database process design and physical design. (See also **1.3.44**.)

1.3.133
menu

Object

hierarchical structure used to provide a user (role) with access to available and applicable functions

1.3.134
menu structure

Product

product providing a diagrammatic representation of the menus to be used within the system

1.3.135
message pair

Object

exchange between the user and the system comprising an input message and an output message

1.3.136
module

Structural Model Element

element comprising a defined set of products and activities, a finite lifespan and an organizational structure

NOTE The SSADM framework requires a project to be subdivided into a number of Modules each of which consists of one or more Stages. Each Module forms a distinct unit for management purposes. The production of the defined end products, to agreed quality standards, signals the completion of the Module.

1.3.137
non-functional requirement

Object

requirement describing how, how well or to what level of quality a facility of the system should be provided

NOTE Examples include service level requirements, access restrictions, security, monitoring, audit and control, and constraints.

1.3.138
non-procedural specification

Object

specification of a process (or entire function), generally used with a program generator, which may contain no more than the specification of the inputs and outputs, and the correspondences between them

1.3.139
normal form

General

result of applying relational data analysis techniques

NOTE 1 There are several stages of normalization; relations are translated into:

— first normal form (1NF);
— second normal form (2NF);
— third normal form (3NF);
— fourth normal form (4NF);
— fifth normal form (5NF).

NOTE 2 SSADM recognizes the existence of fourth and fifth normal form but does not tend to put them to practical use.

1.3.140
normalization

Technique

relational data analysis, effectively using rules to analyse the way items of data depend upon one another for their meaning

NOTE In a normalized Logical Data Model all entity types, considered as relations, have to be in third (or higher) normal form.

1.3.141
normalized relation (A)

Object

result of normalization

NOTE It is a group of data (items) which are a convenient package for access and manipulation purposes. The information held within a normalized relation will be concerned with a particular concept.

1.3.142
object type (N)

Object

object having a defined range of behaviours

NOTE Typically, a function component (e.g. I/O process) or an element type supported in the physical environment. Similar to an entity type.

1.3.143
off-line function

Object

function where all the data are input and the whole of the database processing for the function is completed without further interaction with the user

1.3.144
on-line function

Object

function where the system and the user communicate through input and output messages, i.e. message pairs

NOTE The system responds in time to influence the next input message. On-line functions may include off-line elements such as printing an off-line report.

1.3.145
operation (A)

Object

discrete pieces of processing which may combine to constitute effects; initially identified during entity life history analysis and expanded during process modelling

1.3.146
operations list

Product

list of all the operations which are detailed on a processing structure which may be for an enquiry or an update

NOTE See also **3.7.1** and **3.7.3**.

1.3.147
outline current environment description (A)

Product

description of the current services and any existing problems; produced during the Feasibility Study

NOTE The analysis is similar to that carried out in Stage 1, but is not completed to the same level of detail.

1.3.148
outline development plan (A)

Product

product providing management with information on the development strategy for the remainder of the project for the specific Technical System Option so that provisional timescales and resource requirements, and therefore development costs, can be estimated

1.3.149
outline required environment description (A)

Product

description of the requirements to be included within the proposed system; produced during the Feasibility Study

NOTE The analysis reflects that carried out in Stage 1 but is not completed to the same level of detail.

1.3.150
overview logical data structure (A)

Product

product describing the major entity types and their relationships within the area of the investigation

NOTE See also **1.3.117**.

1.3.151
parallel structure (A)

Object

structure appearing on Entity Life Histories, used to show where certain events may happen within the lifetime of an entity, but not in any prescribed order

1.3.152
physical data design

Product

definition for the physical database which is to be implemented

NOTE The design is developed in two Steps; the first produces a first-cut design based on applying rules about the DBMS to the Required System Logical Data Model; the second is a design optimized for performance reasons.

1.3.153
physical data design (PDD)

Technique

technique taking the Required System Logical Data Model and translating it into a product specific database design within the chosen technical environment with consideration having been given to performance and space constraints

1.3.154
physical design

Product

module product from the Physical Design Module which defines the data and processing elements of the implementable system

1.3.155
physical design

Structural Model Element. Stage 6

stage whose objective is to specify the physical data and physical processes using the language and features of the chosen implementation environment and incorporating installation standards

NOTE This Stage has seven Steps:
— Prepare for Physical Design;
— Create Physical Data Design;
— Create Function Component Implementation Map;
— Optimize Physical Data Design;
— Complete Function Specifications;
— Consolidate Process Data Interface;
— Assemble Physical Design.

1.3.156
physical design

Technique

technique taking the Logical Design and developing it into a specification for the implementation which is wholly dependent on the technical environment as documented in the Physical Environment Specification

NOTE This technique has two aspects: physical data design, and physical process specification.

1.3.157
physical design module (PD)

Structural Model Element

module whose objective is to produce the Physical Design for the system based on the Logical System Specification and the Physical Environment Specification, i.e. it will be implementation dependent

NOTE This Module has one Stage: Stage 6, Physical Design.

1.3.158
physical design strategy

Product

product documenting all aspects relating to designing the physical implementation of the application including all planning documentation

1.3.159
physical environment

Object

implementation environment including, where appropriate, the development environment and the migration path between them

1.3.160
physical environment classification

Product

classification of the environment in which the application is to be implemented which also describes the development environment and migration path where necessary

NOTE See also **1.3.41**, **1.3.42** and **1.3.172**.

1.3.161
physical environment specification (PES)

Product

specification of the hardware and software products and services to be supplied, commissioned and made available for implementation

NOTE Generally this will be provided by the vendor.

1.3.162
physical key (A)

Object

object which tells the system the address on the magnetic disc where a specified record is to be found

NOTE This is often called a pointer.

1.3.163
physical process specification

Product

packaging of all of the specifications for processing which are required in the proposed system

1.3.164
physical process specification

Technique

technique taking the Logical Design and tailoring it to the physical environment to produce the specification for all necessary processing (programs)

1.3.165
physical system specification

Product

specification comprising the Physical Design, Application Development Standards (both from SSADM) and the Physical Environment Specification (which is likely to be the response to a procurement exercise)

1.3.166
problem definition statement

Product

statement of the user requirement for the system produced during the Feasibility Study

NOTE This may include charts and diagrams to supplement the detail.

1.3.167
procedural specification (A)

Object

specification of a process, derived from the input and output data relevant to a procedure, including operations and conditions allocated to it

NOTE It may take the form of a diagrammatic structure, based on SSADM structure diagram.

1.3.168
process

Object

object transforming or manipulating information (data) in a system

1.3.169
process data interface

Product

product documenting how the Logical Data Model can be mapped onto the Physical Data Design, showing how it interfaces with the Physical Processing Specification

NOTE It allows the designer to implement the logical update and enquiry processes as physical programs, independently of the physical database structure.

1.3.170
process/entity matrix

Product

cross check that each entity type is used as the basic information for at least one process and identifying logical groupings of bottom-level processes during logicalization of the Data Flow Model

NOTE Also known as Entity/Process Matrix.

1.3.171
processing specification (PS)

Product

development of this product may identify errors in other products and force previous Steps to be revisited

NOTE Product of Stage 3 (Develop Processing Specification). This product is not shown explicitly on the SSADM structural model.

1.3.172
processing system classification

Product

classification of the details of the processing environment which is to be used for implementation

NOTE Where appropriate, it also defines the development environment.

1.3.173
product (A)

Object

item of software, hardware, or documentation which may itself be a collection of other products

NOTE Each product is predefined and its production and delivery is planned within the project.
(See also **1.3.63**, **1.3.64**, **1.3.81**, **1.3.112**.)

1.3.174
product specification (N)

General

description of the purpose, form, components and completion criteria of a product

1.3.175
program (A)

Object

stored set of computer instructions assembled to meet the requirements of one or more functions

1.3.176
project

General

undertaking having the following characteristics:

— a defined and unique set of technical products to meet the business needs;

— a corresponding set of activities to construct those products;

— a certain amount of resources;

— a finite lifespan;

— an organizational structure with defined responsibilities.

1.3.177
project initiation document (PID)

Product

definition of the terms of reference and objectives for the project, approved by the project board at project initiation

NOTE It is used to identify business requirements, as well as organizational and general information needs, security aspects and an initial Project Plan.

1.3.178
prototype

Object

animated view provided for the user of how the system being developed will work

NOTE It enhances user understanding, allowing better identification of discrepancies and deficiencies in the user requirement.

1.3.179
prototype demonstration objective document

Product

list of the points of discussion to be addressed between the user and the analyst during the prototype demonstration completed prior to any prototype demonstration for each Prototype Pathway

NOTE Assumptions and queries for each menu, screen and/or report are listed under their respective component number.

1.3.180
prototype pathway

Product

combination of screen and report components, identified within the prototyping activities, with the existing menus

NOTE The pathways are a script for the prototyping session and hence use a limited and simple serial diagrammatic representation to convey the structure.

1.3.181
prototype result log

Product

document used to record the results of the prototype demonstration

NOTE This document is used in a similar capacity to minutes of a meeting. Each request made by the user is documented on the log, with a change grade, and the log is updated later to show what changes are required.

1.3.182
prototyping

Technique

variety of techniques which may be used to produce prototypes

NOTE See also **1.3.210**.

1.3.183
prototyping report

Product

report prepared by the technical manager of the prototyping exercise to report back to the project board setting out whether the objectives for the exercise were achieved, or alternatively, the reason why they were not

NOTE It includes estimates of the value of the work done and where necessary suggests whether more work would (or would not) be beneficial.

1.3.184
prototyping scope

Product

product used by the project board to define the boundaries and objectives for the prototyping activities

1.3.185
quality

General

the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs

NOTE This definition is identical with that given in BS 4778-1.

1.3.186
quit and resume (A)

Object

object appearing on an Entity Life History to show how the prescribed sequence of events for an entity may be altered in particular circumstances

1.3.187
random event

Object

event which may occur at any point either during the life of an entity or during a particular stage of an entity life

1.3.188
RDA working paper

Product

documentation of the progress through relational data analysis, taking relations which are un-normalized through to third normal form

1.3.189
relation

Object

group of data items (or attributes)

NOTE When the Logical Data Model is validated using relational data analysis, a relation equates to an entity.

1.3.190
relational data analysis (RDA)

Technique

method of deriving data structures which have the least redundant data and the most flexibility

NOTE The flexibility is achieved by breaking down the data groups into smaller groups without losing any of the original information. It is the objective of this technique to transform all relations into at least third normal form.

1.3.191
relationship (A)

Object

association between two entity types, or one entity type and itself (recursion/involution), to which all instances (occurrences) of the relationship conform

1.3.192
relationship degree (A)

Object

indication where two entity types are directly related, of the number of instances of each such entity type that may participate in one instance of the relationship

1.3.193
relationship description (A)

Product

documentation of the details of a relationship between two entity types on the Logical Data Structure

1.3.194
report format

Product

product showing the layout of a printed report as desired by the user

1.3.195
required system data flow model

Product

model showing how the proposed services are to be organized and what processing is to be undertaken

NOTE See also **1.3.35**.

1.3.196
required system logical data model

Product

product providing the detail of the proposed system information requirements which is developed during the Requirements Specification and Logical System Specification Modules

NOTE See also **1.3.117**.

1.3.197
requirement (A)

Object

description of required feature of the proposed system

NOTE Requirements may be functional (describing what the system should do) and non-functional (describing how a facility should be provided, or how well, or to what level of quality). (See also **1.3.99** and **1.3.137**.)

1.3.198
requirements analysis module (RA)

Structural Model Element

element where objective is to produce the Analysis of Requirements, within which the Selected Business System Option will define the scope of further investigation

NOTE This Module has two Stages:
— Stage 1. Investigation of Current Environment;
— Stage 2. Business System Options.

1.3.199
requirements catalogue

Product

central repository for information covering all identified requirements, both functional and non-functional, in which each entry is textual and describes a required facility or feature of the proposed system

1.3.200
requirements definition

Technique

technique focusing on the future, required system, used to identify and describe requirements for the proposed system which meet the needs of users, and of the business as a whole

NOTE As the project progresses, requirements are resolved using more formal techniques such as function definition.

1.3.201
requirements specification

Product

module product from the Requirements Specification Module, packaging all of the details which are required in order to decide upon the technical direction of the project

1.3.202
requirements specification module (RS)

Structural Model Element

module whose objective is to produce the Requirements Specification

NOTE This Module has one Stage: Stage 3. Definition of Requirements.

1.3.203
resource flow diagram

Product

diagram documenting how resources move within an organization

NOTE This may be the initial diagram drawn within the data flow modelling technique to define the boundary of the system. This diagram will be produced if the current system is predominantly concerned with movement of physical objects, e.g. goods.

1.3.204
screen format

Product

product showing the layout that the user requires on the visual display unit screen

1.3.205
selected business system option

Product

description of a chosen system development direction documenting the system boundary, inputs, outputs and the transformation taking place within the boundary

NOTE Essentially the description is textual with supporting (annotated) elements from the Current Services Description.

1.3.206
selected technical system option (TSO) (A)

Product

description of the chosen technical system option

NOTE The technical details are placed in the Technical Environment Description.

1.3.207
service level requirement

Object

non-functional requirement which states the required quality of service the user expects from a functional aspect of the system

NOTE The service level requirement may be associated with Function Definitions.

1.3.208
space estimation

Product

assessment of the storage requirements of the data design using a particular implementation environment

NOTE The relevant information will vary widely depending upon the technical environment and has to be designed and compiled for each application.

1.3.209
specific function model

Object

model of the components of a particular SSADM function

1.3.210
specification prototyping (SP) (A)

Technique

technique used to identify and trap errors in the specification of the user requirement, and so to enhance requirements definitions prior to detailed logical design activities being undertaken

1.3.211
SSADM structure diagram

Product

diagrammatic representation of structures in terms of sequences, selections and iterations which may be extended to include parallel constructs and operations lists

1.3.212
stage

Structural Model Element

unit of activity with a single goal and rationale

NOTE The SSADM framework requires a project to be subdivided into a number of Modules each of which consists of one or more Stages.

1.3.213
state indicator

Object

attribute used as a re-expression of the structure of an Entity Life History, in a format which will be used during the Logical Design Stage to enforce the defined sequence of events

NOTE A state indicator should be thought of as an additional attribute within each entity.

1.3.214
step

Structural Model Element

subdivision of a Stage with a defined set of products and activities

NOTE The SSADM framework requires a project to be subdivided into a number of Modules each of which consists of one or more Stages, each of which consists of one or more Steps.

1.3.215
success unit

Object

set of processing which has to succeed or fail as a whole within the system

NOTE 1 When a failure occurs the system is restored to the state it was in immediately before the success unit began.
NOTE 2 Also known as logical success unit.

1.3.216
super function

Object

function comprising several lower-level functions

1.3.217
system

General

complete technical output of the project including all technical products (i.e. all hardware, software, documentation, etc.)

NOTE The system will live beyond the life of the project and, in the case of an enhancement project, will have existed before the project.

1.3.218
system description

Product

product showing how the Requirements Specification is met by the Technical Environment Description for a particular Technical System Option

NOTE In many cases the major decisions in this area will have already been taken in choosing a Business System Option.

1.3.219
table

Object (in DBMS)

tabular representation of the relationship between master and detail entities represented by the physical contiguity of the key of the master entity and the keys of each of its detail entities

1.3.220
take-on requirements description

Product

details of the data conversion requirements which have to be implemented before a fully working system can be available compiled during technical system option activities

1.3.221
technical environment description (TED)

Product

details of the specification of the technical environment which is produced once the Technical System Option has been selected

NOTE This detail is then passed on to physical design activities.

1.3.222
technical system option (TSO)

Technique

means by which users agree the new applications implementation strategy incorporating the desired functionality, as defined in the Requirements Specification

NOTE Several Technical System Options are developed and one is selected (or combined elements from several). This gives the technical direction for future development.

1.3.223
technical system options (TSO)

Product

set of Technical System Options which has been developed so that the system development direction can be chosen

NOTE Each option documents the functions to be incorporated and details implementation requirements. Each description is textual with some planning information. Functional elements are taken directly from the Requirements Specification.

1.3.224
technical system options (TSO)

Structural Model Element. Stage 4

stage aimimg to take the Requirements Specification and decide on the most appropriate way for development to meet the technical needs

NOTE This Stage has two Steps:
— Define Technical System Options;
— Select Technical System Options.

1.3.225
testing outline

Product

product compiled during technical system option activities which documents the basic requirements for testing the implementable system so that it will meet user requirements

1.3.226
testing timing factors definition

Product

product used to identify the DBMS read and write operations that will be used to implement the types of logical reads and writes used in the logical update and enquiry processes

NOTE Disc and processor performance information is required for all relevant operations.

1.3.227
timing estimation

Product

product used to assess the timing requirements using the data design within a particular implementation environment

NOTE The relevant information will vary wildly depending upon the technical environment and has to be compiled for each application.

1.3.228
training requirements description

Product

product compiled during technical system option activities which documents the amount of training required for the staff who will be using/working on the new system, so that they will be fully effective at the appropriate time (i.e. when the system is available for use)

1.3.229
transient data store

Object

element where transient data is held for a short time before being used by a process and then deleted

NOTE Data held in transient data stores may not be structured in the same way as the data in a main data store. These are elements of a Data Flow Diagram but are not reflected in the Logical Data Model. They are likely to be used to overcome a constraint within the physical system implementation.

1.3.230
universal function model (UFM)

Object

standard model used to identify the general components of an SSADM function

1.3.231
update function

Object

function containing some update processing of the database

NOTE This may also include some enquiry functions.

1.3.232
update process model (UPM)

Product

structure diagram for the update (event) processing and the associated operations list

NOTE This is based on the Effect Correspondence Diagrams which provide a data-oriented view of the system, and the associated Entity Life Histories which provide an event-oriented or process-oriented view of the system.

1.3.233
user catalogue

Product

description of the on-line users of the proposed system, including details of job titles and the tasks undertaken by each of the identified users

1.3.234
user manual requirements description

Product

product compiled during technical system option activities, which documents the basic information which can be supplied from the SSADM products and other available sources

NOTE It relates to the system and its smooth running from the user point of view.

1.3.235
user role

Object

collection of job holders who share a large proportion of common tasks

1.3.236
user role/function matrix

Product

the mapping of correspondences between functions and user roles, thus identifying the dialogues as being the cross-reference between user roles and on-line functions (either enquiry or update)

NOTE 1 Reading down from the user roles axis also provides the (initial) Menu Structure for the system.
NOTE 2 Also known as Function/User Role Matrix.

1.3.237
user roles

Product

documentation of the details for each user role identified as having a direct interest in the required system

1.3.238
variant (N)

General

product developed to meet specific requirements which differ from other types of the same product, as well as meeting requirements generally applicable to all types of the same product

NOTE The specification for each variant has to have a unique name. A product may have several variants.

1.3.239
version (A)

General

subdivision of a product, or variant, relating to the continuing development of that product

NOTE A product, or variant, may have a number of versions.

1.4 Abbreviations

For the purposes of this British Standard the following abbreviations apply.

1NF	first normal form
2NF	second normal form
3NF	third normal form
BSO	business system option (the technique)
	Business System Options (the product)
	Selected Business System Option (the product)
CBA	Cost/Benefit Analysis
CCTA	CCTA — The Government Centre for Information Systems
DBA	Database Administrator
DBMS	Database Management System
DD	dialogue design
DFD	Document Flow Diagram
DFM	data flow modelling
EAP	Enquiry Access Path
ECD	Effect Correspondence Diagram
EED	External Entity Description
EEM	entity event modelling
ELH	Entity Life Histories (the product)
	entity life history analysis (the technique)
EPD	Elementary Process Description
EPM	Enquiry Process Model
FCIM	Function Component Implementation Map
FD	function definition
FS	Feasibility Study
I/O	Input/Output
LDM	Logical Data Model (the product)
	logical data modelling (the technique)
LDPD	logical database process design
LDS	Logical Data Structure
LGDE	logical grouping of dialogue elements
LS	Logical System Specification
PD	physical design
PDD	physical data design (the technique)
PDI	Process Data Interface
PED	Physical Environment Description
PES	Physical Environment Specification
PID	Project Initiation Document
PPS	Physical Process Specification
PS	Processing Specification
RA	Requirements Analysis
RD	Relationship Degree
RDA	relational data analysis
RS	Requirements Specification
SP	specification prototyping
TED	Technical Environment Description
TSO	technical system option (the technique)
	Selected Technical System Option (the product)
	Technical System Options (the product)
UFM	universal function model
UPM	Update Process Model

Section 2. Conformity

2.1 Individual products

To claim compliance with this British Standard, each individual product (as specified in section **3**) supplied shall conform to the relevant clauses only, i.e. those relating to the specific product. Upon request, the supplier shall supply a copy of the charts (given in Annex D) appropriately completed and annotated for the agreed range of products.

For any product, conformity shall be stated in one of the following ways.

a) *Conformity by product*. Conformity in respect of delivery of the product specified (at the overall level of the product itself) reference being made to the relevant purpose, composition, completion criteria, status and presentation conventions clauses.

b) *Conformity by specified component*. Conformity in respect of delivery of the product specified, but with specific reference to the composition clause of the product.

NOTE This is to permit agreed delivery of a product which does not incorporate all aspects of the stated composition, but which is deemed to serve its purpose in the specific development context.

c) *Conformity by presentation*. In addition, a product or component can conform in terms of the presentation conventions given in the specification for that particular product.

2.2 Marking

Each individual product supplied shall be labelled or supplied with documentation stating the following:

a) the number and date of this British Standard, i.e. BS 7738-1:1995[2)];

b) means of identifying the supplier.

[2)] Marking BS 7738-1:1995 on or in relation to a product represents a manufacturer's declaration of conformity, i.e. a claim by or on behalf of the manufacturer that the product meets the requirements of the standard. The accuracy of the claim is solely the clamaint's responsibility. Such a declaration is not to be confused with third party certification of conformity, which may also be desirable.

Section 3. Detailed product specifications

NOTE See **C.2** for an explanation of the conventions used in this section.

3.1 Business System Options products

COMMENTARY AND RECOMMENDATIONS ON **3.1**. *A Business System Option (BSO) is a description of a system, its boundary, inputs, outputs and the transformations taking place within it. It is not concerned with how the transformations are effected, except where the mechanisms for carrying them out impact the transformations themselves.*

Following the establishment of the Requirements Catalogue, and preceding the Requirements Specification Module, BSO products are presented to users to enable them to decide upon the functionality of the required system.

Each specific option comprises sets of features designed to satisfy Requirements Catalogue entries, and to assist the evaluation of the impacts and benefits of the possible new system upon the organizational context.

3.1.1 Business System Options

3.1.1.1 *Purpose*

A set of Business System Options shall encompass a number of possible solutions, each of which shall satisfy the user requirements to a greater or lesser extent. Each option shall present a high-level system design, and be developed and evaluated from the business aspects.

Each BSO shall contain descriptions of the functional areas to be included in the system (i.e. encompassed by the system boundary), the requirements to be addressed and the potential impact on the organization.

NOTE 1 There may be only one BSO.

NOTE 2 Presentation and consideration of BSOs provides users with the opportunity to assess alternative scenarios for meeting selections among required functions. It also allows users to decide to abandon the project, having decided that requirements cannot be adequately satisfied within the pre-defined constraints.

3.1.1.2 *Composition*

Business System Options shall be textual documents, which may be supported by Data Flow Diagrams and a Logical Data Structure, as follows:

a) heading:

— Option name and/or identifier;

b) detailed documentation describing:

— functionality (system) boundary;

— levels of functionality (of the entire application and its components);

— other technical considerations such as operating constraints;

— Cost/Benefit Analysis;

— Impact Analysis;

— training needs.

3.1.1.3 *Completion criteria*

The following shall be considered for any example:

a) for each:

1) Is the scope of the system described consistent with the constraints defined in the project initiation document, the Feasibility Report, or both?

2) Are the minimum requirements met by this option?

3) Is the proposed system (option) feasible?

4) Will the proposed option meet the installation standards for data, processing, communications, etc.?

5) Is the format of the option in keeping with installation standards?

b) for the set:

1) Are all options documented here?

2) Does the set of options cover all business requirements?

3.1.1.4 *Status*

The Business System Option shall be an interim product (completed); an interim product produced during Stage 2 (RA), but options shall be presented to users, and the selected BSO will be an end-stage product (RA).

COMMENTARY AND RECOMMENDATIONS ON **3.1.1**. *For any proposed system a number, or set, of Business System Options can be compiled and presented to users in order that a specific option be selected. Alternatively it may be decided by the users that no option embodies a feasible basis for development and that the project be abandoned.*

Derivation:

Requirements Analysis:

— *Current Services Description;*

— *Feasibility Report (if it exists);*

— *project initiation document;*

— *Requirements Catalogue;*

— *User Catalogue.*

Where used:

— *Business System Options: RA.*

Presentation conventions are not applicable.

3.1.2 Selected Business System Option

3.1.2.1 *Purpose*

The Selected Business System Option shall document the description of the system functionality of the chosen option, so supporting the further development of the Requirements Specification.

3.1.2.2 *Composition*

The Selected Business System Option shall be essentially a textual document, which may be supported by Data Flow Diagrams and a Logical Data Structure.

Detailed documentation shall describe the following:

— functionality (system) boundary;

— levels of functionality (of system and components);

— technical considerations (e.g. operating constraints);

— Cost/Benefit Analysis;

— Impact Analysis;

— selection reasoning (documenting reasons for choosing this option, and for not selecting other available options).

3.1.2.3 *Completion criteria*

The following shall be considered for any example.

a) Is the scope of the system described consistent with the requirements defined in the Project Initiation Document?

b) Has one and only one option been selected? (This may incorporate aspects from several of the original options.)

c) Are the minimum requirements met by this option?

d) Is the proposed system (option) feasible and possible?

e) Is the proposed system (option) economically sound and within budget?

f) Will the proposed option meet the installation standards?

g) Is the format of the option in keeping with installation standards?

h) Does the Selected Business Option look as if it will meet the project objectives?

i) Will the Selected Business Option contribute towards enabling the organization's objectives?

3.1.2.4 *Status*

The Selected Business System Option shall be delivered complete at the end of Stage 2.

COMMENTARY AND RECOMMENDATIONS ON **3.1.2**. *The Selected Business System Option provides the basis for further development of the Requirements Specification.*

Derivation:

Requirements Specification:

— *Business System Options;*

— *Current Services Description;*

— *Feasibility Report;*

— *project initiation document;*

— *Requirements Catalogue;*

— *User Catalogue.*

Where used:

— *Business System Options: RA;*

— *Data Flow Modelling: RS;*

— *Dialogue Design: RS;*

— *Logical Data Modelling: RS;*

— *Requirements Definition: RS;*

— *Technical Systems Options: LS.*

Presentation conventions are not applicable.

3.1.3 Cost/Benefit Analysis

3.1.3.1 *Purpose*

The CBA shall present costs of a proposed option for completion of the project which are balanced against benefits, therefore assisting in the decision making process.

3.1.3.2 *Composition*

The composition shall be as follows:

a) variant identification (one of):

— Feasibility Option;

— Business System Option;

— Technical System Option;

b) header information:

— option name and/or identifier;

c) details:

— development costs;

— operating costs;

— replaced costs;

— benefits;

— tangible benefits (visible/quantifiable);

— projected cost containment (how proposed system is cheaper than current);

— intangible benefits (meeting critical requirements).

3.1.3.3 *Completion criteria*

The following shall be considered for any example.

a) Is the option type identified as one of feasibility, business system, or technical system?

b) Is the option name/identifier correct?

c) Are the stated direct tangible benefits valued correctly? Do they indeed exist?

d) Are the projected cost containment figures reasonable?

e) Are the intangible benefits believable?

f) Has the source for all estimates been clearly identified?

g) Are the development, operating and replaced costs believable and reasonable?

h) Are relevant benefits metrics available?

3.1.3.4 *Status*

NOTE CBAs may be produced for several options.

For a specific selected option the CBA shall be delivered as a complete product at the end of the relevant module as follows:

— Feasibility CBA: FM;

— BSO CBA: RA;

— TSO CBA: LS.

For non-selected options, the CBA shall be an interim product.

COMMENTARY AND RECOMMENDATIONS ON **3.1.3**.

Cost Benefit Analysis offers users a focus for discriminating between different systems options, by providing details of the potential benefits and costs of an option. Such an analysis may be produced for Feasibility Options (Feasibility Module), Business System Options (Requirements Analysis), and/or Technical System Options (Logical System Specification).

Derivation:

Feasibility Module (Feasibility Options):

— *Feasibility Option;*

— *Impact Analysis.*

Requirements Analysis (Business System Options):

— *(Selected) Business System Option;*

— *Impact Analysis.*

Logical System Specification (Technical System Options):

— *Impact Analysis;*

— *Outline Development Plan;*

— *Technical Environment Description;*

— *(Selected) Technical System Option.*

Where used:

— *Feasibility CBA: FS;*

— *BSO CBA: RA;*

— *TSO CBA: LS.*

Presentation conventions are not applicable.

3.1.4 Feasibility Options

3.1.4.1 *Purpose*

The set of Feasibility Options shall encompass all the options to be considered, during a feasibility study, in selecting the way ahead. Each alternative shall present a high-level system design, and be evaluated from the following viewpoints:

a) business (support for business requirements and objectives);

b) organizational (impact on people and tasks);

c) technical (IS requirements, viability of development and implementation route);

d) financial (cost, benefits and risks).

3.1.4.2 *Composition*

Feasibility Options shall be essentially textual documents, which may be augmented by components of a Logical Data Model and a Data Flow Model, as follows:

a) heading:

— option name and/or identifier;

b) detailed documentation including:

1) textual description of the scope of the information system (clerical and IT aspects) and the requirements to be addressed, possibly supported by models of the major processes, entities and external interfaces of the proposed information system;

2) an overview of the software and hardware configuration required to run the information system option and the technical resources required to develop it;

3) Impact Analysis: an outline assessment of the impact of the option on the organization;

4) an approximate investment/appraisal (Cost/Benefit Analysis) detailing a broad order of cost and financial, non-financial and indirect benefits;

5) resource requirements;

6) outline time scale/plan for each implementation;

7) risks in terms of business, technical, financial and cultural factors;

8) advantages and disadvantages of the options and a conclusion as to whether it is achievable and desirable.

3.1.4.3 *Completion criteria*

The following shall be considered for any example.

a) For each: Is the option feasible from all of the following viewpoints:

— business;

— organizational;

— technical;

— financial?

b) For the set: Is the set of options complete?

3.1.4.4 *Status*

The Feasibility Options shall be end-module products; delivered complete at the end of the Feasibility Module.

COMMENTARY AND RECOMMENDATIONS ON **3.1.4**. *The set of Feasibility Options is compiled so that a selection can be made. Each option documents the functions to be incorporated and details implementation requirements. Each description is textual with some planning information.*

Derivation:

Feasibility Module:

Outline Current Environment Description;

Outline Required Environment Description;

Problem Definition Statement;

Requirements Catalogue;

User Catalogue.

Where used:

Feasibility: FS;

Business System Options: FS;

Technical System Options: FS;

Logical Data Modelling: FS;

Data Flow Modelling: FS;

Requirements Definition: FS.

Presentation conventions are not applicable.

3.1.5 Feasibility Report

3.1.5.1 *Purpose*

The feasibility report shall serve several purposes as follows.

a) It records management decisions about the options for further development, including whether a proposed information system should be cancelled, re-scoped, split or merged with another.

b) It forms the basis for any decision to commit resources for a more detailed study.

c) It provides information for a more detailed study as a record of decisions, assumptions, estimates, user requirements and outline options.

d) It provides an outline Project Plan for managing a more detailed study.

e) It records the findings of the study team according to the terms of reference agreed at the start of the study.

3.1.5.2 *Composition*

The composition shall be as follows:

a) introduction:

— reasons for study;

— terms of reference;

— study objectives;

— scope of study;

— constraints;

— completion date;

— consultation;

— management of study;

b) executive summary:

— recommended solution;

— options considered but rejected;

— plans for full study;

— preferred procurement path;

— implementation plan;

c) study approach:

— details of costs and conduct of study;

d) existing business and IS support to the organization, documenting the current situation in the area of study;

— business objectives;

— functions and processes currently undertaken;

— organization of the business area, with various roles and responsibilities;

— current and potential areas of strength and weakness;

— any relationships with other business areas and organizations;

— any existing IS support detailing functions supported or not, and weaknesses, technological opportunities and constraints;

e) future IS support required:

— description of the system's place in the IS strategy;

— overview of scope of required system and functionality;

— details expressed in measurable terms for requirements;

— implications of any geographic distribution of IS support;

— details of service performance required of the proposed system;

f) proposed system, describing how the requirements can be met:

— narrative overview of the logical system, based on selected business option(s);

— outline of the alternative technological options, with summary of the necessary technical framework;

— advantages and disadvantages of the proposals;

g) options considered but rejected:

— described in fashion similar to that for the proposed system, but with less detail;

h) financial assessment:

— summary of costs comparing proposed system with options;

— summary of costs against benefits;

i) project plan:

— resource requirements;

— expected implementation timescale;

— proposed management structure for development and implementation;

j) conclusion and recommendations;

k) annexes:

— SSADM support documentation.

3.1.5.3 *Completion criteria*

The following shall be considered for any example.

a) Does the report conform to installation standards?

b) Is the proposed solution within the identified budget constraints?

c) Is the project within the scope set? Can it be continued on this basis?

d) Do the users agree that their requirements have been addressed?

e) Is one and only one option selected as the way ahead?

f) Are all the requirements mutually consistent? If not, have priorities been identified?

g) Is it an accurate statement of requirements, constraints, and possible future enhancements to the system?

3.1.5.4 *Status*

The Feasibility Report shall be a completed product, delivered at the end of the Feasibility Module.

COMMENTARY AND RECOMMENDATIONS ON **3.1.5**. *The Feasibility Report provides the basis upon which decisions can be made to continue or redefine a proposed IS project.*

Derivation:

— *FS.*

Where used:

— *Feasibility: FS;*

— *Requirements Definition: FS, RA;*

— *Data Flow Modelling: FS, RA;*

— *Logical Data Modelling: FS, RA;*

— *BSO: FS, RS;*

— *TSO: FM;*

— *Dialogue Design: RA.*

Presentation conventions are not applicable.

3.1.6 Impact Analysis

3.1.6.1 *Purpose*

Impact Analysis shall explain the effects of the option under consideration on the user environment. This document shall be used to supplement Feasibility Options, Business System Options and Technical System Options.

3.1.6.2 *Composition*

The composition shall be as follows:

a) version identification (one of):

— Feasibility Option;

— Business System Option;

— Technical System Option (unselected);

— Technical Environment Description (selected);

b) heading information:

— option name and/or identifier;

c) details:

— Take-On Requirements Description;

— testing outline;

— Training Requirements Description;

— User Manual Requirements Description;

— additional points to be covered including organization and staffing; changes in user operating procedures;

— implementation considerations including: training requirements, conversion and staff learning curves;

— savings, in terms of replaced equipment, maintenance, supplies, etc.;

— comparative impact (advantages and disadvantages) with the other Technical System Options.

NOTE For Feasibility Options and Business System Options there will be less detail available and some of these items may have no associated section.

3.1.6.3 *Completion criteria*

The following shall be considered for any example.

a) Is the option type identified correctly (feasibility, business system, technical system)?

b) Is the option name/identifier correct?

c) Are the following aspects covered:

— improved turnover/work rates;

— ease and speed of implementation;

— scale of development costs;

— reliability;

— projected cost containment;

— resourcing consequences;

— staff and equipment;

— performance and service;

— limitations;

— difficulties in implementation and/or increased time scale?

3.1.6.4 *Status*

Impact Analysis shall be delivered complete for a specified option and type of option; dependent on Stage/Module by option type.

COMMENTARY AND RECOMMENDATIONS ON **3.1.6**. *Impact Analysis describes the effects of a specified option (feasibility, business or technical) on the user environment and will cover issues concerned with organization, procedures, and implementation factors. This product is used to document the ramifications of selecting a particular option.*

Derivation:

a) *Feasibility Module (Feasibility Options):*

— *Feasibility Option;*

— *Outline Current Environment Description;*

— *Outline Required Environment Description;*

b) *Requirements Analysis (Business System Options):*

— *(Selected) Business System Option;*

— *Current Services Description;*

— *Requirements Catalogue;*

— *User Catalogue.*

c) *Logical System Specification (Technical System Options):*

— *Requirements Specification;*

— *(Selected) Technical System Option;*

— *Technical Environment Description.*

Where used:

— *Business System Options: RA;*

— *Technical System Options: LS;*

— *Feasibility: FS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.2 Data Flow Modelling products

COMMENTARY AND RECOMMENDATIONS ON **3.2**. *Data Flow Modelling is used to develop a hierarchical decomposition of a specific system into identified data sources and receivers, functions, data stores and data flows between them. The techniques employed have the following objectives:*

a) *define the system context in terms of its boundary, scope and environment;*

b) *identify the external entities that supply data to the system and/or receive data output by the system;*

c) *identify data flows across the system boundary;*

d) *identify data stores within the system boundary;*

e) *establish the main functions, or groups of functions, to be performed by the system.*

The technique plays a crucial part in the development of the Requirements Definition of the system under development. The Data Flow Model, being hierarchical, moves from a top-level user-oriented view to a more detailed view at the lower levels. These lower levels, consisting of descriptions of elementary processes, provide the bases for products completed in the Logical Systems Specification Module.

Derivation:

For many of these products there might be three variants: Current Physical, Logical and Required System. These will be produced in the order listed. Consequently the derivation of later variants will depend on the production of earlier ones if produced.

3.2.1 Context Diagram

3.2.1.1 *Purpose*

The Context Diagram shall provide an overall view of the external data flows to and from the system and also the external entities that supply data to and/or receive data from the system within the boundary. The diagram shall be drawn by the analyst using data flow modelling notation to represent the context in which the system operates.

3.2.1.2 *Composition*

The composition shall be as follows:

— the system (a single process);

— multiple external entities;

— links (data flows) between external entities and processes.

3.2.1.3 ***Completion criteria***

The following shall be considered for any example.

a) Is the diagram syntactically correct and semantically adequate?

b) Is the scope of the system clearly shown?

c) Are all known external entities represented?

d) Is there one and only one "process" box, representing the system?

3.2.1.4 ***Status***

The Context Diagram shall be an evolving product if produced during Feasibility. It shall be a completed, end-stage product delivered at the end of Stage 1 (RA).

3.2.1.5 ***Presentation conventions***

The components of Context Diagrams shall be in accordance with Figure 2.

COMMENTARY AND RECOMMENDATIONS ON **3.2.1**. *The Context Diagram is drawn to illustrate the initial scope of the proposed system. The diagram concentrates on the major inputs and outputs of the system and shows the external sources and recipients of system data.*

Derivation:

a) *Feasibility:*

— *discussion with users;*

— *existing system documentation;*

b) *Requirements Analysis:*

— *discussion with users;*

— *existing system documentation;*

— *Context Diagram (if produced during Feasibility Study).*

Where used

— *Feasibility;*

— *Data Flow Modelling: RA.*

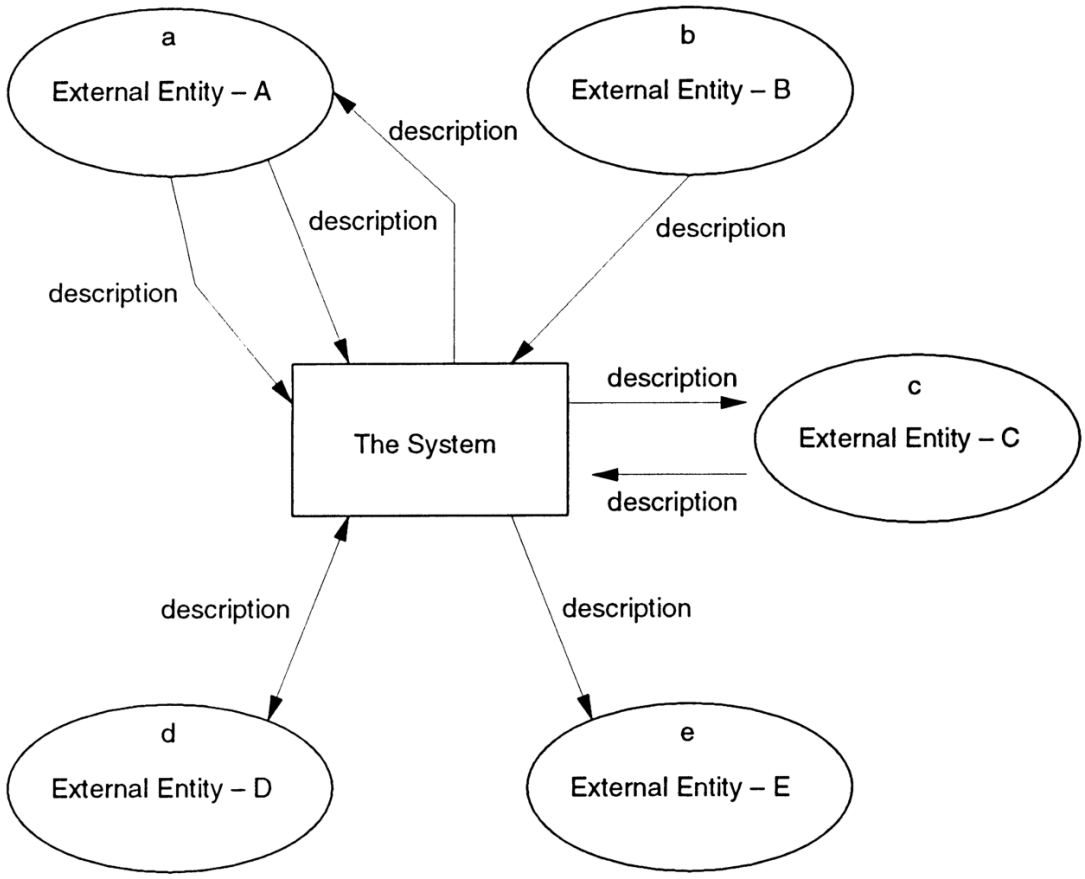


Figure 2 — Presentation conventions for the Context Diagram

3.2.2 Data Flow Diagram: Level 1

3.2.2.1 *Purpose*

The Level 1 Data Flow Diagram shall serve to contain the overall information flow model of the system.

NOTE It will be revised during the lifetime of the system to reflect current, logical or required functional areas.

There shall be three possible variants of the Data Flow Diagram: Level 1: Current System, Logical and Required System.

The Level 1 Data Flow Diagram of the Current System shall be drawn by the analyst to describe the scope of the system under investigation.

The Level 1 Diagram of the Logical Data Flow Model shall be built from the bottom up, by grouping bottom-level processes into more abstract units and drawing the data flows that represent information flowing at the lower level.

The Level 1 Diagram of the Required System Data Flow Model shall be reached when processes map on to a user's view of the functions that the system is intended to support.

3.2.2.2 *Composition*

The composition shall be as follows:

a) variant identifier (one of):
- — Current Physical;
- — Logical;
- — Required System;

b) diagram elements representing:
- — processes;
- — data flows;
- — data stores;
- — external entities;
- — resource flows;
- — resource stores.

3.2.2.3 *Completion criteria*

The following shall be considered for all variants.

a) Is the variant identifier properly completed?

b) Are agreed notational conventions correctly applied?

c) Is the boundary of the system clear?

d) Are all user-perceived functional areas represented?

e) Are meaningful names used for processes and data stores?

f) Are all identifiers unique?

g) Do external entity names accurately reflect the environment outside the system?

h) Does the diagram accurately reflect the physical, logical or required system in terms of external entities, data stores and data flows?

i) Does the diagram avoid giving an inappropriate level of detail, such as sequencing or detailed processing logic?

The following shall be considered for the Logical Data Flow Diagram.

1) Are all physical aspects of the current system removed, unless they are constraints on the requirement?

2) Are any enquiries remaining after logicalization major ones?

3) Are all external entities identified as potential users of the system?

It shall be considered for the Required System Data Flow Diagram if all the facilities defined by the Selected Business System Option, and nothing else, are modelled in the Required System Data Flow Diagrams.

3.2.2.4 *Status*

There shall be three possible variants of this product: Current Physical, Logical and Required System.

Where the Current Physical variant is started during Feasibility, it shall then be delivered as an evolving end-module product. It shall then be updated in Stage 1 and delivered complete as an interim product during Stage 1.

The Logical variant shall be produced during Stage 1, and shall be delivered as an end-stage product on completion of Stage 1.

The Required System variant shall be developed during Stage 3 as an interim product.

3.2.2.5 *Presentation conventions*

The components of the Level 1 Data Flow Diagram shall be in accordance with Figure 3 to Figure 8.

COMMENTARY AND RECOMMENDATIONS ON **3.2.2**. *The Level 1 diagram illustrates the main functional areas contained within the scope of the system. It should be a simple diagram readily understood by users, so that it can act as an effective means of communication between analysts and users. (See also **3.2.4**.)*

Derivation:

a) *Feasibility (for Current Physical Data Flow Diagram Level 1):*
- — *existing system documentation;*
- — *project initiation document;*

b) *Requirements Analysis (for Current Physical Data Flow Diagram Level 1):*
- — *Context Diagram;*
- — *Current Physical Data Flow Diagram Level 1;*
- — *Document Flow Diagram;*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Resource Flow Diagram.*

c) *Requirements Analysis (for Logical Data Flow Diagram Level 1):*

— *Current Physical Data Flow Diagram Level 1.*

d) *Requirements Specification (for Required System Data Flow Diagram Level 1):*

— *Selected Business System Option;*

— *Logical Data Flow Diagram Level 1.*

Where used:

— *Data Flow Modelling (all three variants);*

— *Feasibility: Current Physical;*

— *Logical Data Modelling (all three variants);*

— *Business System Options: Logical, Required variants;*

— *Function Definition: Required;*

— *Entity-event Modelling: Required.*

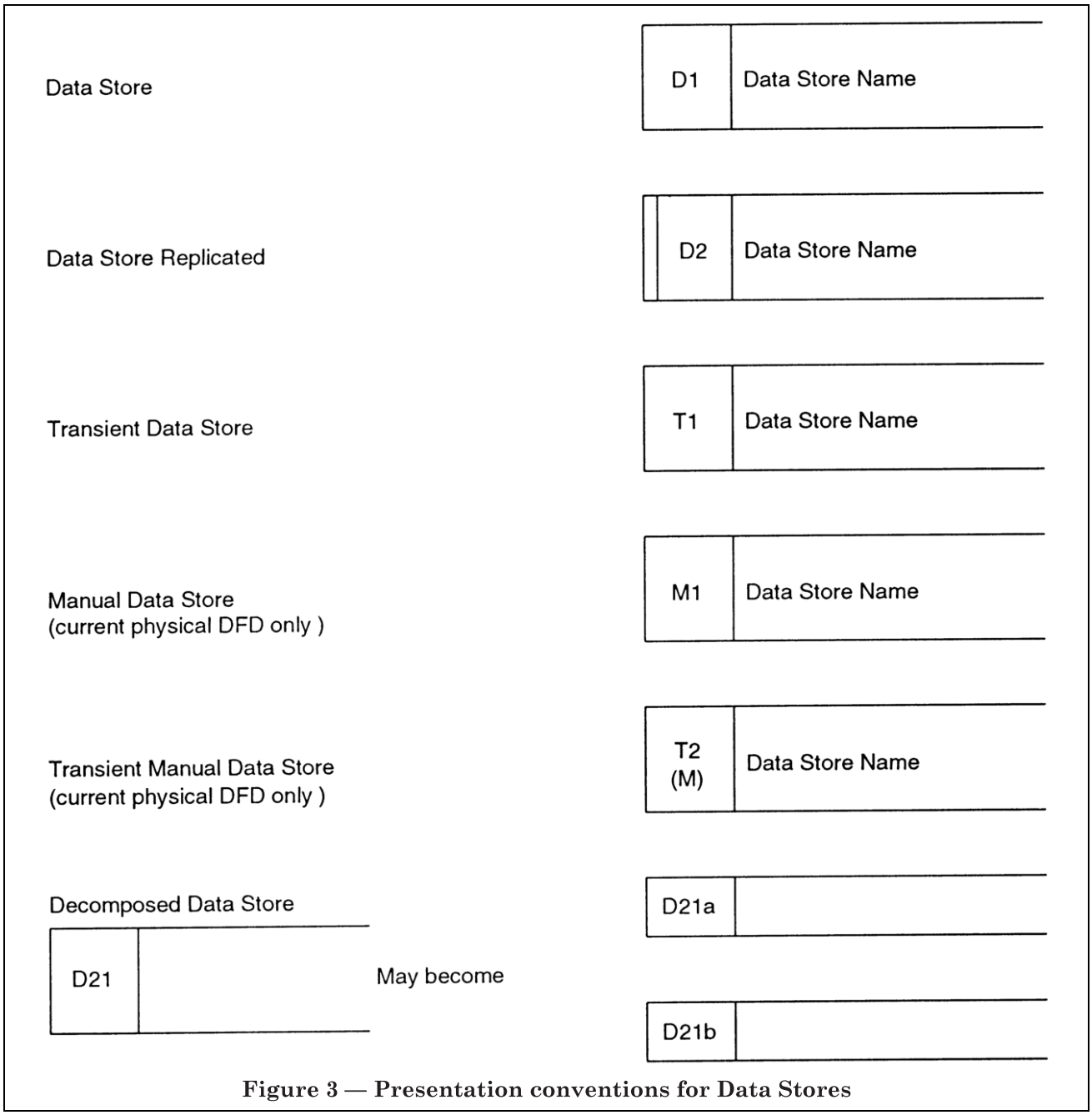


Figure 3 — Presentation conventions for Data Stores

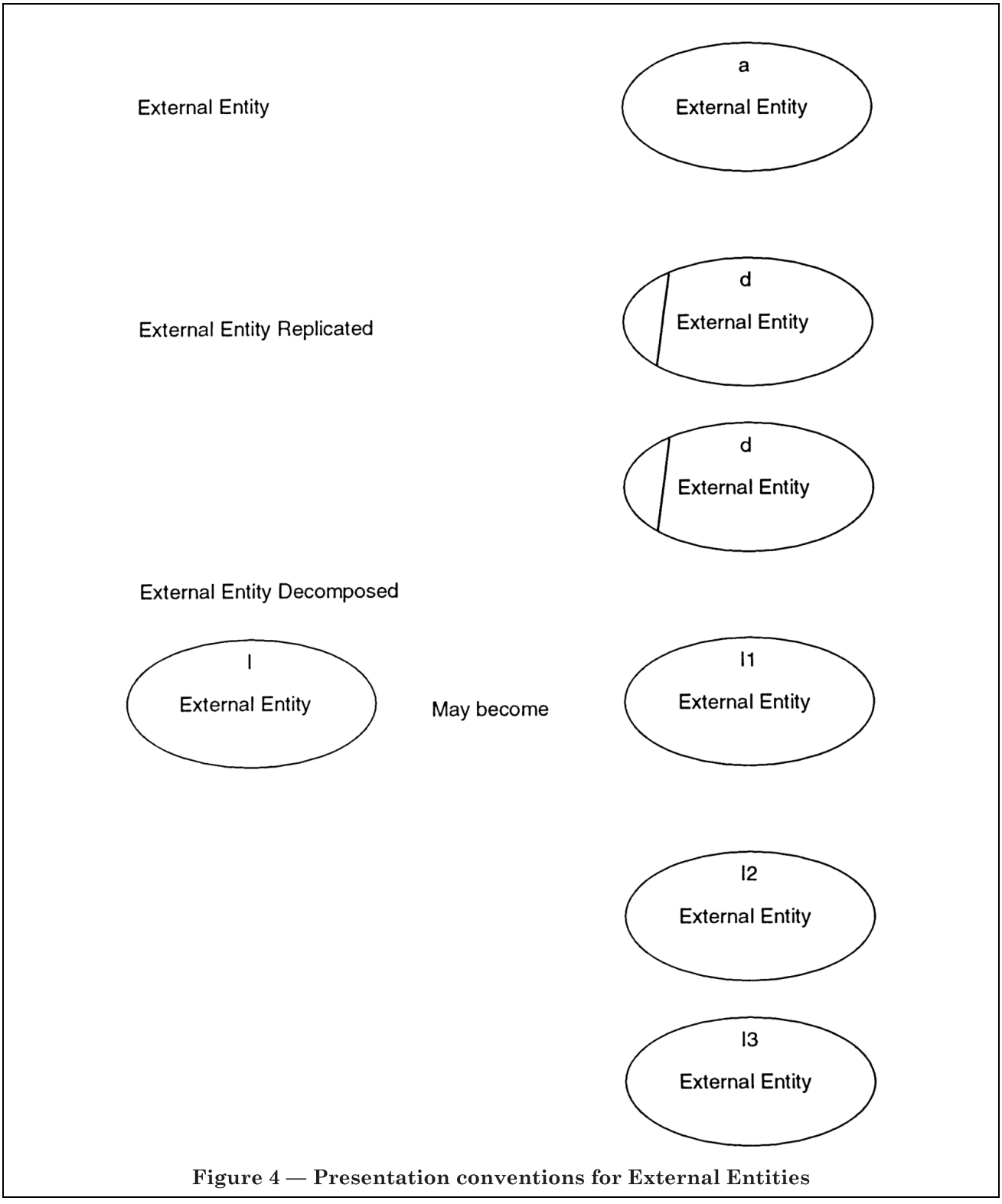


Figure 4 — Presentation conventions for External Entities

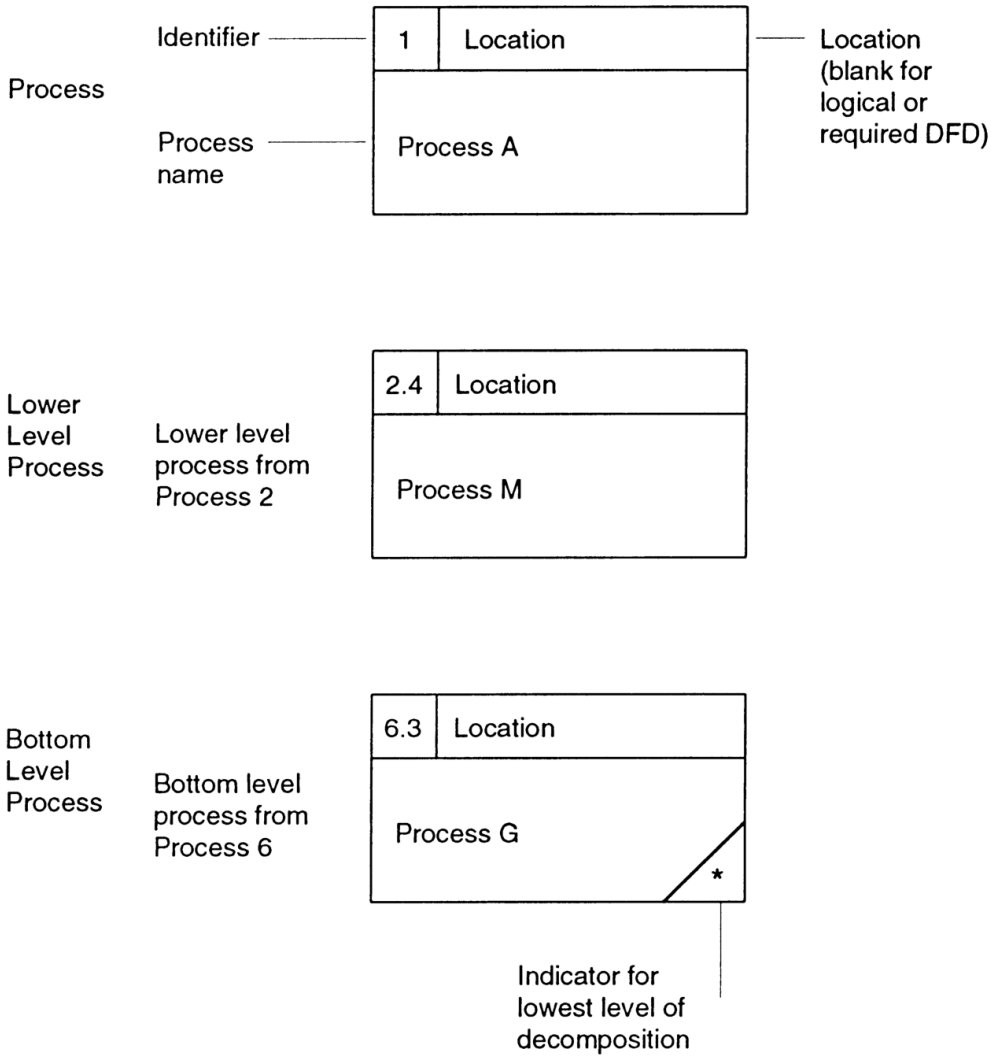


Figure 5 — Presentation conventions for Processes

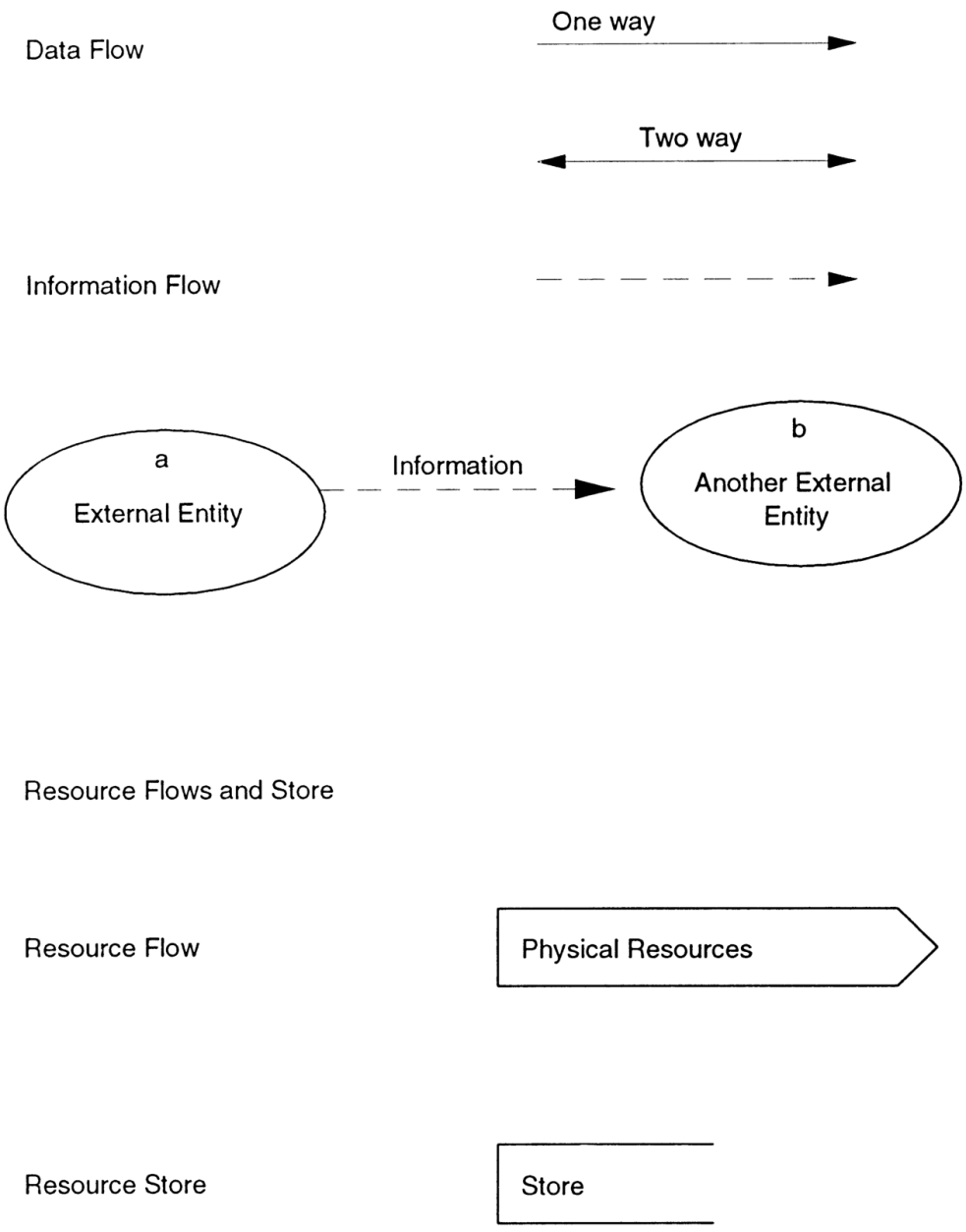


Figure 6 — Presentation conventions for Data Flows

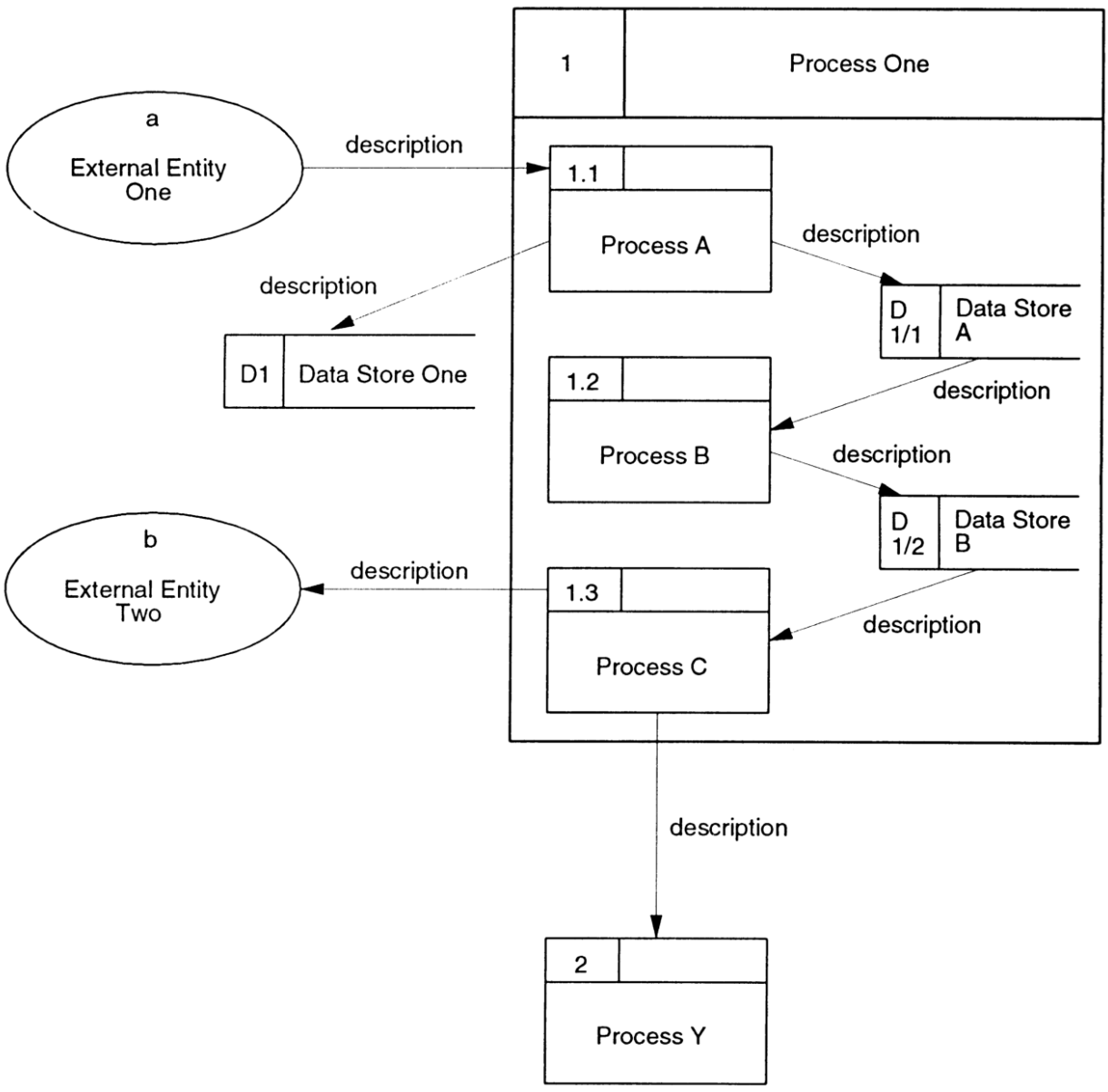


Figure 7 — Presentation conventions for Process Decomposition

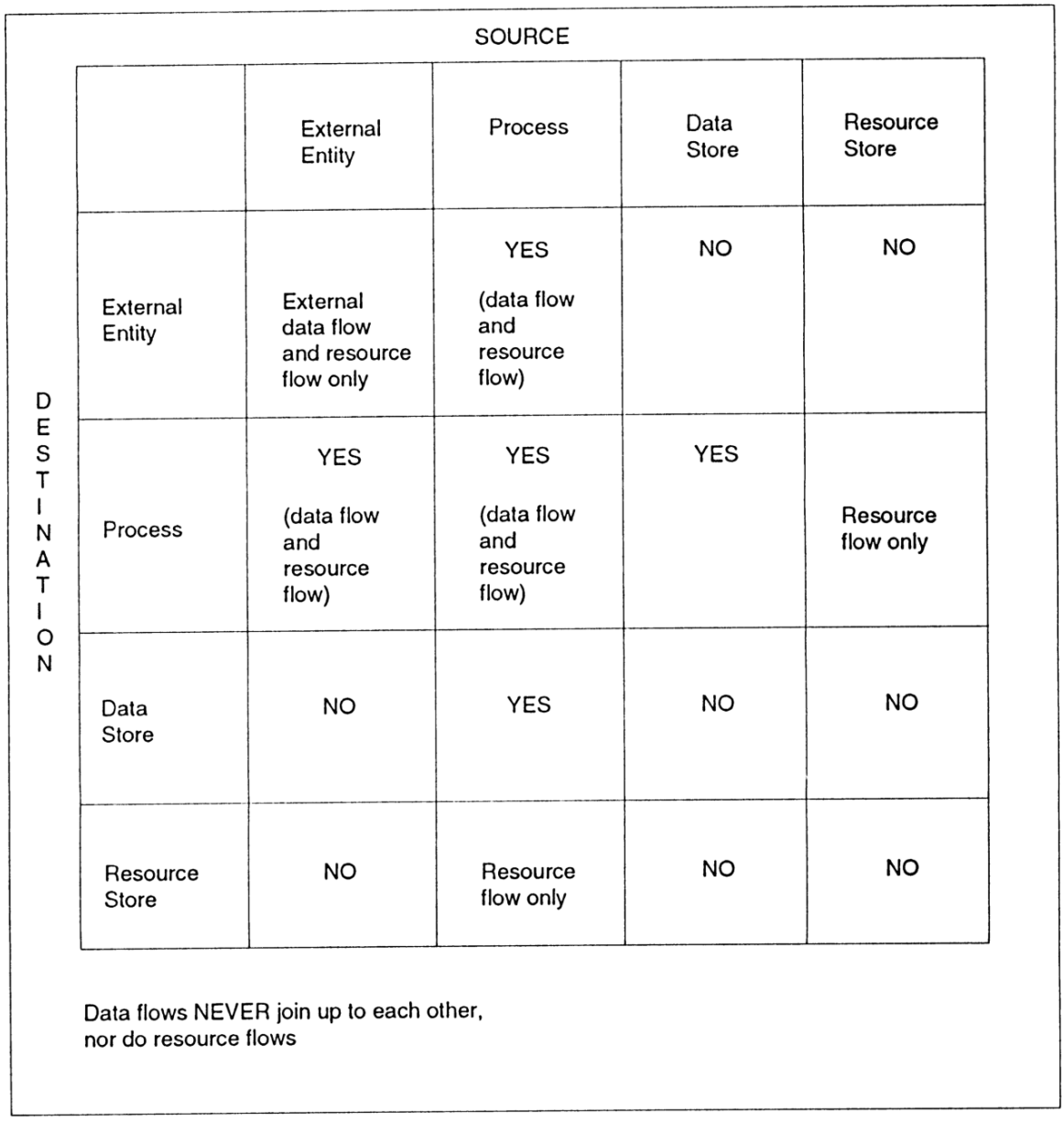


Figure 8 — Legal sources and destinations of data flows

3.2.3 Data Flow Diagrams: Lower Level

3.2.3.1 *Purpose*

This level of Data Flow Diagram shall describe processes shown on a higher level Data Flow Diagram in more detail.

NOTE 1 The higher level may be the Level 1 Data Flow Diagram, or may be itself a Lower Level Data Flow Diagram.

The process box on the lower level diagram shall be drawn much larger to accommodate the further processes and data stores subsumed within the higher level process.

The external entities, data stores and other processes with which the decomposed process communicates shall be re-represented in this diagram outside the process boundary and the data flows that represent this communication shall be drawn going to the boundary of the process and, when the diagram is complete, crossing it.

NOTE 2 External entities and data stores outside the process boundary may be decomposed from higher level objects.

NOTE 3 Lower level Data Flow Diagram forms are used for the decomposition of Data Flow Diagrams or Resource Flow Diagrams, as required.

3.2.3.2 ***Composition***

The composition shall be a set of Data Flow Diagrams: Lower Level, each consisting of:

a) variant identifier (one of):

— Current Physical;

— Logical;

Required System;

b) details from higher level:

— higher level process number;

— higher level process name;

— external entities;

— data stores from higher level;

— processes from higher level.

c) additional details for this level (appearing within the outer process box):

— data stores;

— processes;

— resource flows;

— resource stores.

3.2.3.3 ***Completion criteria***

The following shall be considered for each.

a) Is the variant correctly identified?

b) Are agreed notational conventions correctly applied?

c) Is the boundary of the process clear?

d) Are meaningful names used for processes and data stores?

e) Do external entity names accurately reflect the environment outside the system?

f) Does the diagram avoid giving an inappropriate level of detail, such as sequencing or detailed processing logic?

The following shall be considered for the Logical Data Flow Diagram.

1) Are all physical aspects of the current system removed, unless constraints on the requirement?

2) Are any enquiries remaining after logicalization major ones?

It shall be considered for the Required System Data Flow Diagram if all and only the facilities defined by the Selected Business System Option are modelled in the Required System Data Flow Diagrams.

It shall be considered for Lowest-level Diagrams if all data flows are one-way.

The following shall be considered for the set of diagrams.

i) Are all identifiers unique?

ii) Is the set of diagrams complete?

iii) Is the integrity of the hierarchy of diagrams maintained?

3.2.3.4 ***Status***

There shall be three possible variants of the Lower Level Data Flow Diagram: Current Physical, Logical and Required System.

Where the Current Physical variant is started during Feasibility, it shall then be delivered as an end-module product. It shall then be updated in Stage 1, and shall be delivered complete as an interim product during Stage 1.

The Logical variant shall be produced during Stage 1, and shall be delivered as an end-stage product on completion of Stage 1.

The Required System variant shall be developed during Stage 3 as an interim product.

3.2.3.5 ***Presentation conventions***

The components of the Lower Level Data Flow Diagrams shall be in accordance with Figure 3 to Figure 8.

COMMENTARY AND RECOMMENDATIONS ON **3.2.3**.

These diagrams enhance higher level Data Flow Diagrams. They show further details relating to system processes. Each diagram should be clear and simple so that it can be readily understood by relevant users: so that it can act as an effective means of communication between analysts and users. (See also ***3.2.4****.)*

Derivation:

a) *Requirements Analysis:*

1) *for Current Physical Data Flow Diagrams: Lower Level:*

— *Document Flow Diagram;*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Current Physical Data Flow Diagram Level 1;*

— *Resource Flow Diagram;*

2) *for Logical Data Flow Diagrams: Lower Level:*

— *Current Physical Data Flow Model (if a current system exists);*

b) *Requirements Specification:*

for Required System Data Flow Diagrams: Lower Level:

— *Selected Business System Options;*

— *Logical Data Flow Model.*

Where used:

— *Data Flow Modelling (all three variants);*

— *Logical Data Modelling (all three variants);*

— *Business System Options: Logical, Required variants;*

— *Function Definition: Required;*

— *Entity-Event Modelling: Required.*

3.2.4 Data Flow Model

3.2.4.1 *Purpose*

The Data Flow Model shall amalgamate the overall information flow model of the system.

NOTE This will be revised during the lifetime of the system to reflect Current Physical, Logical or Required functional areas.

3.2.4.2 *Composition*

The composition shall be as follows:

— Data Flow Diagram: Level 1 (see **3.2.2**);

— Data Flow Diagrams: Lower Level(s) (see **3.2.3**);

— Elementary Process Descriptions (see **3.2.6**);

— External Entity Descriptions (see **3.2.7**);

— I/O Descriptions (see **3.2.8**);

NOTE See Figure B.9 for the product breakdown structure diagram.

3.2.4.3 *Completion criteria*

The following shall be considered for each model.

a) Is the model variant identified and consistent for all components of the model?

b) Does the model accurately reflect the current physical, logical or required system in terms of processes, external entities, data stores and data flows?

c) Is the model consistent with the previous version?

d) Are external entities, data stores and data flows represented consistently between levels?

e) Is there a consistent level of complexity between diagrams?

f) Are some processes decomposed into too many lower level processes, suggesting the need for a further level of hierarchy?

g) Is this a complete set of documentation to describe fully the Data Flow Diagrams currently available (applies to Current Physical, Logical, and required system Data Flow Models)?

h) Have lower-level diagrams been created to represent detail of complex processes?

i) Is the set of diagrams complete?

j) Is there an adequate Elementary Process Description for all the bottom-level processes on the Data Flow Diagrams, including those that appear on the Level 1 diagram?

k) Has the description of any common process been placed in Elementary Process Descriptions?

l) Are the common and other Elementary Process Descriptions cross-referenced consistently?

m) Are process identifiers and names consistent between the Data Flow Diagrams and the Elementary Process Descriptions?

n) Is there an adequate description of all the external entities identified within the Data Flow Model, including those that are decomposed and appear only on lower-level diagrams?

o) Are identifiers and names consistent within the model?

p) Are all bottom-level flows across the system boundary described?

q) Are only inputs and outputs crossing the system boundary described in I/O Descriptions?

r) Do the I/O Descriptions document all identified data flows which cross the system boundary?

The following shall be considered for the Logical Data Flow Model.

1) Are all physical aspects of the current system removed, unless constraints on the requirement?

2) Are only major enquiries remaining after logicalization?

3.2.4.4 *Status*

There shall be one data flow model, with three possible variants: Current Physical, Logical and Required System.

Where the Current Physical variant is started during Feasibility, it shall be delivered as an end-module product. It shall then be updated in Stage 1, and shall be delivered complete as an interim product during Stage 1.

The Logical variant shall be produced during Stage 1, and shall be delivered as an end-stage product on completion of Stage 1.

The Required System variant shall be developed during Stage 3 as an interim product.

COMMENTARY AND RECOMMENDATIONS ON **3.2.4**.
The model consists of a set of Data Flow Diagrams and associated documentation. The diagrams form a hierarchy with the Data Flow Diagram Level 1 showing the scope of the system and the lower level diagrams expanding the detail as appropriate. Additional documentation provides a description of the processes, input/output data flows and external entities.

Derivation:

a) *Feasibility (for Current Physical Data Flow Diagram Level 1):*

— *existing system documentation;*

— *project initiation document;*

b) *Requirements Analysis:*

1) *for Current Physical Data Flow Diagram Level 1:*

— *Context Diagram;*

— *Current Physical Data Flow Diagram Level 1;*

— *Document Flow Diagram(s) (if developed);*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Resource Flow Diagram (if developed);*

2) *for Logical Data Flow Model:*

— *Current Physical Data Flow Model;*

c) *Requirements Specification (for Required System Data Flow Model):*

— *Selected Business System Option;*

— *Logical Data Flow Model.*

Where used:

— *Data Flow Modelling (all three variants);*

— *Feasibility: Current Physical;*

— *Logical Data Modelling (all three variants);*

— *Business System Options: Logical, Required variants;*

— *Function Definition: Required;*

— *Entity-event Modelling: Required;*

Presentation conventions are not applicable.

3.2.5 Document Flow Diagram

3.2.5.1 *Purpose*

The Document Flow Diagram shall model, as a graphical representation, the flow of documentation in a current system between external entities (sources and recipients).

NOTE 1 Several individual document flows may be drawn and then combined.

NOTE 2 The Document Flow Diagram assists in data flow modelling, either to start or verify the analysis of data flows on the Current Physical Data Flow Model.

3.2.5.2 *Composition*

Diagram elements shall represent:

— document flows;

— external entities;

— data flows.

3.2.5.3 *Completion criteria*

The following shall be considered for the model.

a) Are agreed notational conventions adhered to?

b) Is the scope of the system clearly shown?

3.2.5.4 *Status*

The Document Flow Diagram shall be an interim product used within Stage 1.

3.2.5.5 *Presentation conventions*

Components shall be marked, as appropriate, in accordance with Figure 6, relating to DFDs.

COMMENTARY AND RECOMMENDATIONS ON **3.2.5**. *The Document Flow Diagram illustrates how documents pass around the system. This may be the initial diagram drawn within the data flow modelling technique to assist in defining and identifying the boundary and major components of the system. This diagram will be produced if the current system is predominantly clerical and involves the passing of information using forms or other documents.*

Derivation:

Requirements Analysis:

— *discussion with users;*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *by examination of data flows in the current information system.*

Where used:

— *Data Flow Modelling: RA.*

3.2.6 Elementary Process Description(s)

3.2.6.1 *Purpose*

Elementary Process Descriptions shall serve to record brief descriptions of the functionality of the following:

a) Data Flow Model processes which are not the subject of decomposition to lower level data flow diagrams;

b) discrete elements of processing which are common to several bottom-level Data Flow Model processes (elementary processes).

Each instance of either a) or b) shall be recorded as a separate Elementary Process Description, and these in their turn shall be packaged as a complete set. These descriptions shall be used to support understanding of the Data Flow Model for use in later techniques.

3.2.6.2 *Composition*

The composition shall be as follows:

a) variant identifier (one of):

— Current Physical;

— Logical;

— Required System;

— Function Definition;

b) process identification (indicating if common or elementary process; common processing details are subsequently associated directly with the set of Function Definitions):

— process identifier;

— process name;

— common processing cross-references (as appropriate);

— description of process.

3.2.6.3 *Completion criteria*

The following shall be considered for each description.

a) Is the variant identifier completed and valid?

b) Is the description of processing suitably detailed and appropriate to the data flow modelling technique?

c) Is the common processing cross-reference (if used) valid?

d) Is the description consistent with any previous version?

Within Function Definitions (Stage 3) it shall be considered if all the necessary descriptions of common processing are linked with the Function Definitions.

It shall be considered for the set of descriptions if all elementary processes are described.

3.2.6.4 *Status*

An Elementary Process Description shall be either:

a) an evolving product delivered at completion of Stage 1; or

b) a completed product delivered at end of Stage 3.

COMMENTARY AND RECOMMENDATIONS ON **3.2.6**.
Each description outlines the functionality of processes included in a specified data flow model variant. A requirement for common processing may also be described within an Elementary Process Description and cross-referenced to the elementary processes or functions which use it.

Derivation:

a) *Requirements Specification:*

1) *for Current Physical Data Flow Model:*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Current Physical Data Flow Diagram, Level 1;*

2) *for Logical Data Flow Model:*

— *Current Physical Data Flow Model;*

b) *Requirements Analysis:*

1) *for Required System Data Flow Model:*

— *Selected Business System Option;*

— *Logical Data Flow Model;*

2) *for Function Definitions:*

— *Required System Data Flow Model.*

Where used:

— *Data Flow Modelling;*

— *Function Definition;*

— *Logical Database Process Design;*

— *Physical Design.*

Presentation conventions are not applicable.

3.2.7 External Entity Description(s)

3.2.7.1 *Purpose*

All External Entity Descriptions associated with a Data Flow Model shall be packaged together to ensure that a complete set is compiled.

Each External Entity Description shall describe a real-world external entity (whether another system, an organization, an individual or group of people) that interfaces to the system. It shall record any relevant detail about the responsibilities or functions of the external entity, and possible constraints on how it interfaces, or is required to interface to the system. The descriptions also serve to determine the match between user roles and the external entities on the Required System Data Flow Model.

3.2.7.2 *Composition*

The composition shall be as follows:

a) variant identifier, one of:

— Current Physical;

— Logical;

— Required System;

b) Multiple External Entity Descriptions each composed of:

— external entity identifier;

— external entity name;

— external entity description.

3.2.7.3 *Completion criteria*

The following shall be considered for any example.

a) Is the variant identifier completed and valid?

b) Is the external entity and its interaction with the system described sufficiently?

The following shall be considered for the set.

1) Are all external entities described?

2) Is the set consistent with previous versions (modified as appropriate by Business System Options)?

3.2.7.4 *Status*

The status shall be as follows:

a) Current Physical: interim product used within Stage 1;

b) Logical: completed end-stage product for Stage 1;

c) Required System: completed end-module product, RS;

d) Function Definition: completed end-module product RS.

COMMENTARY AND RECOMMENDATIONS ON **3.2.7**.
These descriptions are used to explain the relationship between an external entity and the existing or proposed system with which it operates. The detail covers responsibilities or functions of the external entity and any constraints on the interface with the proposed system.

Derivation:

a) *Requirements Analysis:*

1) *for Current Physical Data Flow Model:*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *User Catalogue;*

2) *for Logical Data Flow Model:*

— *Current Physical Data Flow Model;*

b) *Logical System Specification (for Required System Data Flow Model):*

— *Logical Data Flow Model;*

— *Selected Business System Option;*

— *User Roles;*

Where used:

— *Data Flow Modelling;*

— *Business System Options;*

— *Dialogue Design;*

— *Function Definition.*

Presentation conventions are not applicable.

3.2.8 I/O Descriptions

3.2.8.1 *Purpose*

I/O Descriptions shall consolidate all of the I/O Descriptions associated with a Data Flow Model to ensure that a complete set is compiled.

Each I/O Description entry shall record the data contained in one data flow that crosses the system boundary of a Data Flow Diagram (a data flow between an external entity and a process, in either direction).

The data items on the flows shall be listed and bottom-level flows shall be documented.

NOTE The I/O Structure is not shown, neither repeating groups, nor optional items, nor choice of items. This lower level of detail will be rigorously defined during function definition. However, the practitioner may wish to record such detail in a comments column as analysis proceeds.

3.2.8.2 *Composition*

The composition shall be as follows:

a) variant identifier (one of):

— Current Physical;

— Logical;

— Required System;

b) I/O Descriptions, each consisting of:

— source object identifier (from);

— destination object identifier (to);

— data flow name;

— data content, including data item identifiers where appropriate;

— comments.

3.2.8.3 *Completion criteria*

The following shall be considered for any example:

a) for each:

1) Is the variant identifier completed correctly?

2) Is the list of data content complete from information available?

3) Does the data flow include one or more data items?

b) for the set:

1) Are all input and output flows across the system boundary described?

2) Are only data flows which cross the system boundary described?

3) Is this set consistent with the previous version?

3.2.8.4 *Status*

The status shall be as follows:

a) Current Physical: interim product, Stage 1;

b) Logical: end-stage product delivered complete, Stage 1;

c) Required System: interim product, Stage 3.

COMMENTARY AND RECOMMENDATIONS ON **3.2.8**.
The I/O *Structures are used to document all data flows which cross the Data Flow Model system boundary.*

Derivation:

a) *Requirements Analysis:*

1) *for Current Physical Data Flow Model:*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Current Physical Data Flow Diagram Level 1;*

2) *for Logical Data Flow Model:*

— *Current Physical Data Flow Model;*

b) *Requirements Specification (for Required System Data Flow Model):*

— *Selected Business System Option;*

— *Logical Data Flow Model;*

Where used:

— *Data Flow Modelling;*

— *Function Definition.*

Presentation conventions are not applicable.

3.2.9 Logical Data Store/Entity Cross-reference

3.2.9.1 *Purpose*

The Logical Data Store Entity Cross-reference shall document the correspondence between logically-related groupings of entities from the Logical Data Model and main logical data stores derived during logicalization of the Data Flow Diagrams. The product shall later be used to identify the entity types that are updated by events (found by inspection of the Required System Data Flow Model).

NOTE It should accurately reflect changes made in the transition from Logical Data Flow Model to Required System Logical Data Flow Model.

3.2.9.2 *Composition*

Each cross-reference shall detail the following:

a) Logical Data Store identity and name;

b) entity type names (possibly hierarchy diagram).

3.2.9.3 *Completion criteria*

The following shall be considered for any example.

a) Are all main logical data stores, derived during logicalization of the Data Flow Diagrams or elsewhere, defined in terms of entity types?

b) Is an entity type in one and only one main logical data store?

c) Are all entity types from the Logical Data Model mapped?

d) Does any logical data store appear more than once within the documentation? If so, why?

e) Where logically related groupings of entity types are shown, is the structure consistent with the Logical Data Structure?

3.2.9.4 *Status*

There shall be two possible variants of this product: Logical and Requirements Specification.

The Logical variant shall be delivered as a complete product at the end of Stage 1. The Requirements Specification variant is an interim product developed and used in RS.

COMMENTARY AND RECOMMENDATIONS ON **3.2.9**.

This product is in the form of a matrix showing the correspondence between logical data stores in the Data Flow Model and the entity types on the Logical Data Model. It is used to ensure that each main data store corresponds to an entity type or group of entity types; also that each entity type on the Logical Data Model is held completely within one and only one main data store.

Derivation:

a) *Requirements Analysis:*

— *Current Logical Data Flow Model;*

— *Current Environment Logical Data Model;*

b) *Requirements Specification:*

— *Logical Data Flow Model;*

— *Required System Logical Data Model.*

Where used:

— *Stage 1 and Stage 3.*

Presentation conventions are not applicable.

3.2.10 Logical/Physical Data Store Cross-reference

3.2.10.1 *Purpose*

The Logical/Physical Data Store Cross-reference shall serve to cross-reference main logical data stores to the physical data stores from the Current Physical Data Flow Model. The product shall be created to aid the complete and consistent logicalization of the Current Physical Data Flow Model, by listing logical data stores as they are derived. It shall help to ensure that no data held in the current system has been ignored, and shall serve as a way of communicating these details to users.

NOTE 1 This product is an aid to logicalization and is not maintained after the Logical Data Flow Model has been derived.

NOTE 2 A data store from the current system may be referenced to more than one main logical data store, and one main logical data store may be referenced to more than one physical data store.

3.2.10.2 *Composition*

The composition shall be as follows:

a) Logical Data Store details:

— identifier;

— name.

b) Physical Data Store details (may be for more than one):

— identifier;

— name.

3.2.10.3 *Completion criteria*

The following shall be considered for any example:

a) Have all physical data stores been mapped to a logical data store? That is, for each physical data store is there a logical data store containing the same entities?

b) Can physical data stores not appearing in the cross-reference be accounted for (for example, deleted as redundant)?

3.2.10.4 *Status*

The Logical/Physical Data Store Cross-reference shall be an interim product developed in Stage 1.

COMMENTARY AND RECOMMENDATIONS ON **3.2.10**.
This matrix is used in the construction of the logical view of the data organization when derived from the current (physical) services.

Derivation:

Requirements Analysis:

— *Logical Data Flow Model;*

— *Current Physical Data Flow Model.*

Where used:

— *Data Flow Modelling Stage 1.*

3.2.11 Process/Entity Matrix

3.2.11.1 *Purpose*

The Process/Entity matrix shall support the identification of processes that access the same data, when deriving high-level groupings of processes at logicalization. It shall be created with reference to the Logical Data Store/Entity Cross-reference, but not maintained after logicalization.

This matrix shall be a guide-line to help when logical groupings are not obvious. The intersections of the matrix shall show whether a given process updates a given entity type (via a logical data store) or simply reads it, or does both or neither.

It shall be used to obtain groupings of bottom-level processes during the logicalization of the Current Physical Data Flow Model.

3.2.11.2 *Composition*

The composition shall be as follows:

— column headings: entity type names;

— row headings: bottom-level process names;

— cells completed as appropriate.

3.2.11.3 *Completion criteria*

The following shall be considered for any example.

a) Are all of the entity types from the Current Environment Logical Data Model used as column headings?

b) Are all of the bottom-level processes that exist after logicalization used as row headings?

c) Are all data flows which access a main logical data store matched by a correct entry in the matrix?

d) Are all necessary cell values completed correctly [i.e. “u” (for “update”) when a process updates an entity occurrence; “r” (for “read”) when there is a flow from the main logical data store to the process; null (an empty cell) denoting that the process does not access that entity type]?

3.2.11.4 *Status*

The Process/Entity Matrix shall be an interim product produced within Stage 1.

COMMENTARY AND RECOMMENDATIONS ON **3.2.11**.
This product is used to check that all entity types are manipulated in at least one process and to identify logical groupings of bottom-level processes during logicalization of the Data Flow Model.

Derivation:

Requirements Analysis:

— *Current Environment Logical Data Model;*

— *Current Physical Data Flow Model;*

— *Logical Data Store/Entity Cross-reference.*

Where used:

— *Requirements Analysis.*

Presentation conventions are not applicable.

3.2.12 Resource Flow Diagram

3.2.12.1 *Purpose*

The Resource Flow Diagram shall illustrate the movement of actual resources in the current services, rather than the information flows that document them. Each Resource Flow Diagram shall illustrate the movement of one resource. A set of such diagrams shall be produced to cover all resources.

3.2.12.2 *Composition*

Diagrammatic representation shall consist of the following:

— process box(es);

— one or more external entities;

— links (broad arrows) between external entities and processes;

— resource flows;

— resource stores.

3.2.12.3 *Completion criteria*

The following shall be considered for any example.

a) Are agreed notational conventions adhered to?

b) Is the scope of the system clearly shown?

It shall be considered for the set if all resources are documented.

3.2.12.4 *Status*

The Resource Flow Diagram shall be an interim product developed during Stage 1.

3.2.12.5 *Presentation conventions*

The components shall be marked, as appropriate, in accordance with Figure 3 to Figure 8.

COMMENTARY AND RECOMMENDATIONS ON **3.2.12**. *This product documents how resources move within the section of the organization of concern to the system under development. This may be the initial diagram drawn within the data flow modelling technique to define the boundary of the system. The diagram may be produced if the current system is predominantly concerned with movement of physical objects, e.g. goods.*

Derivation:

Requirements Analysis:

— *discussion with users;*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Current Physical Data Flow Diagrams (examining* I/O *flows).*

Where used:

— *Stage 1.*

3.3 Dialogue Design products

COMMENTARY AND RECOMMENDATIONS ON **3.3**. *Dialogue Design products are developed to deal with those aspects of the system concerned with system-user interaction.*

3.3.1 Command Structure(s)

3.3.1.1 *Purpose*

The Command Structure shall serve to provide navigation around the menus and dialogues when a dialogue terminates.

NOTE In many instances it will be used with a Menu Structure.

3.3.1.2 *Composition*

The composition shall be as follows:

a) heading:

— dialogue name;

— User Role;

b) repeating group which shall consist of:

— option of what the next action could be;

— dialogue or menu (whether that action is contained in another dialogue or a menu);

— Dialogue/Menu name: the name of the dialogue or menu to be used.

3.3.1.3 *Completion criteria*

The following shall be considered.

a) Is the dialogue correctly identified?

b) Is the user role correctly identified?

c) Does the user agree with the command structures?

3.3.1.4 *Status*

The Command Structure(s) shall be either:

a) an evolving end-module product delivered at the end of RS; or

b) delivered complete as end-stage product, Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.3.1**. *The Command Structure shows the directions that control can take when a user decides to complete or terminate a particular dialogue. This allows navigation to be implemented with or without menus.*

Derivation:

a) *Requirements Specification:*

— I/O *Structures;*

— *Menu Structures;*

b) *Logical System Specification:*

— *Dialogue Structures;*

— *Menu Structures.*

Where used:

— *Dialogue Design: LS;*

— *Specification Prototyping: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.3.2 Dialogue Control Table

3.3.2.1 *Purpose*

The Dialogue Control Table shall be used to identify the navigation between logical grouping of dialogue elements (LGDEs), both default and alternative.

3.3.2.2 *Composition*

The composition shall be as follows:

a) heading:

— Dialogue Name;

b) footing (one per dialogue not per form):

— Percentage Path Usage;

c) details of logical grouping of dialogue elements:

— identity of the logical grouping of dialogue elements;

— the number of occurrences of this grouping within the dialogue;

— the default pathway through these dialogue elements;

— list of all alternative pathways that may be taken for this dialogue.

3.3.2.3 *Completion criteria*

The following shall be considered.

a) Is the dialogue identified correctly?

b) Are the number of occurrences noted correctly?

c) Has the default pathway been correctly identified?

d) Have all the necessary alternative pathways been identified correctly?

e) Does the total of the percentage path usage (which is calculated for the default path and all alternative pathways within the dialogue) equal 100?

3.3.2.4 *Status*

The Dialogue Control Table shall be an end-stage product delivered complete: Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.3.2**.
This is used to identify and capture the navigation between the "logical groupings of dialogue elements" under "normal conditions". This table also details the different order in which particular aspects of the dialogue may be undertaken.

Derivation:

Logical System Specification:

— *dialogue documentation;*

— *discussion with users.*

Where used:

— *Dialogue Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.3.3 Dialogue Element Description(s)

3.3.3.1 *Purpose*

Dialogue Element Descriptions shall be used to provide support documentation for the graphical Dialogue Structures. These descriptions shall be used within several activities of dialogue design, with more information added by each task.

NOTE The set of completed descriptions comprises most of the supporting documentation for the graphical Dialogue Structures.

Two end products of dialogue design shall be documented within this product as follows:

— Logical Grouping of Dialogue Elements;

— Dialogue Element Descriptions.

A complete set of these descriptions shall be packaged together to describe all the dialogues in the system.

3.3.3.2 *Composition*

The composition shall be as follows:

a) heading:

— Dialogue name;

— User role identification (possibly more than one);

— Function name;

b) Dialogue Element Description:

— element name within the dialogue;

— the name, or names, of the associated data items;

c) Logical Grouping of Dialogue Elements (LGDE):

— LGDE identifier;

— LGDE status indicator (mandatory/optional).

3.3.3.3 *Completion criteria*

The following shall be considered:

a) for each description:

1) Is the dialogue identified correctly?

2) Is the user role identified correctly?

NOTE Exceptionally, there may be more than one, in which case the major user role should be identified first.

3) Is the function identified correctly?

4) Are all the dialogue elements in this dialogue included on the form (which may include continuation sheets)?

5) Are all data elements and LGDE details filled in for every instance of dialogue element?

6) Does every dialogue element include one or more data items?

b) for the set: Is this a complete set for all identified dialogues?

3.3.3.4 *Status*

The Dialogue Element Description shall be an end-stage product, delivered complete, Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.3.3**.
Each description is used to describe a dialogue element. A set of these provides the detailed documentation for a Dialogue Structure.

Derivation:

Logical System Specification:

— I/O *Structures;*

— *results from prototyping;*

— *User Role/Function Matrix.*

Where used:

— *Dialogue Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.3.4 Dialogue Level Help

3.3.4.1 *Purpose*

Dialogue Level Help shall record any dialogue level help that is identified after discussions with the users in Stage 5.

NOTE The type of information recorded is that of help for the dialogue as a whole and not for screen help that will be designed as part of Stage 6: Physical Design.

3.3.4.2 *Composition*

The composition shall be a textual document that records all dialogue level help that is required by the users.

3.3.4.3 *Completion criteria*

The following shall be considered.

a) Is the dialogue identified correctly?

b) Is the user role identified correctly? (Exceptionally there may be more than one, in which case the major user role should be identified first.)

c) Has all the necessary help been identified?

3.3.4.4 *Status*

Dialogue Level Help shall be an end-stage product, delivered complete, Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.3.4**.
This product is used to detail the level of help that the user (user role) requires to progress through a specific dialogue.

Derivation

Logical System Specification:

— *Discussion with users;*

— *Dialogue Structures.*

Where used:

— *Dialogue Design;*

— *Physical Design.*

Presentation conventions are not applicable.

3.3.5 Dialogues

3.3.5.1 *Purpose*

Dialogues shall consolidate all of the dialogue details to ensure that there is a complete set which is passed between steps in the method. Each Dialogue shall be used to describe the components of a dialogue and how they fit together.

3.3.5.2 *Composition*

Multiple Dialogues shall each consist of the following:

— Dialogue Structure;

— Dialogue Element Descriptions;

— Dialogue Control Table;

— Dialogue Level Help.

NOTE See Figure B.14 for the product breakdown structure diagram for this product.

3.3.5.3 *Completion criteria*

The following shall be considered.

a) For each Dialogue:

1) Are all four components completed for this dialogue? If help is unnecessary, are reasons given?

2) Are all names within boxes on any Dialogue Structure cross-referenced to the corresponding Dialogue Element Descriptions?

3) Is the logical grouping of dialogue elements (LGDE) within a Dialogue Structure composed of the same elements as the corresponding Dialogue Element Descriptions?

4) Are all dialogue elements shown on a Dialogue Structure?

5) Do all the dialogue elements in any dialogue appear in a Dialogue Element Description?

6) Is each completed Dialogue Control Table consistent with its associated Dialogue Structure and Dialogue Element Descriptions?

b) for the set: Is this a complete set of documentation for all identified Dialogues?

3.3.5.4 *Status*

Dialogues shall be end-stage products, delivered complete, Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.3.5**.
This set of entries is used to consolidate details of all identified dialogues within the system.

Derivation:

Logical System Specification:

— *I/O Structures;*

— *discussion with users;*

— *(Installation) Style Guide.*

Where used:

— *Dialogue Design: LS;*

— *Physical Design: PD.*

NOTE Presentation conventions are not applicable.

3.3.6 Dialogue Structure

3.3.6.1 *Purpose*

The dialogue structure shall be used to show, in a graphical fashion, all the components of a dialogue and how they fit together.

3.3.6.2 *Composition*

The composition shall be graphical representation using the concepts outlined for the SSADM Structure Diagram. (See **3.16.6**.)

NOTE Logical grouping of dialogue elements (LGDE) may also be shown.

3.3.6.3 ***Completion criteria***

The following shall be considered.

a) Is the dialogue correctly identified (in the top box)?

b) Are diagramming conventions correctly applied?

c) Have all dialogue elements been identified and assigned unique names?

3.3.6.4 ***Status***

Dialogue Structures shall be end-stage products delivered complete, Stage 5.

3.3.6.5 ***Presentation conventions***

The presentation conventions shall be in accordance with Figure 9.

COMMENTARY AND RECOMMENDATIONS ON **3.3.6**.

This is a diagrammatic representation of a dialogue. Each box on the Dialogue Structure equates to a dialogue element. Input/output operations are allocated to dialogue elements.

Derivation:

Logical System Specification:

— I/O *Structures.*

Where used:

— *Dialogue Design: LS;*

— *Physical Design: PD.*

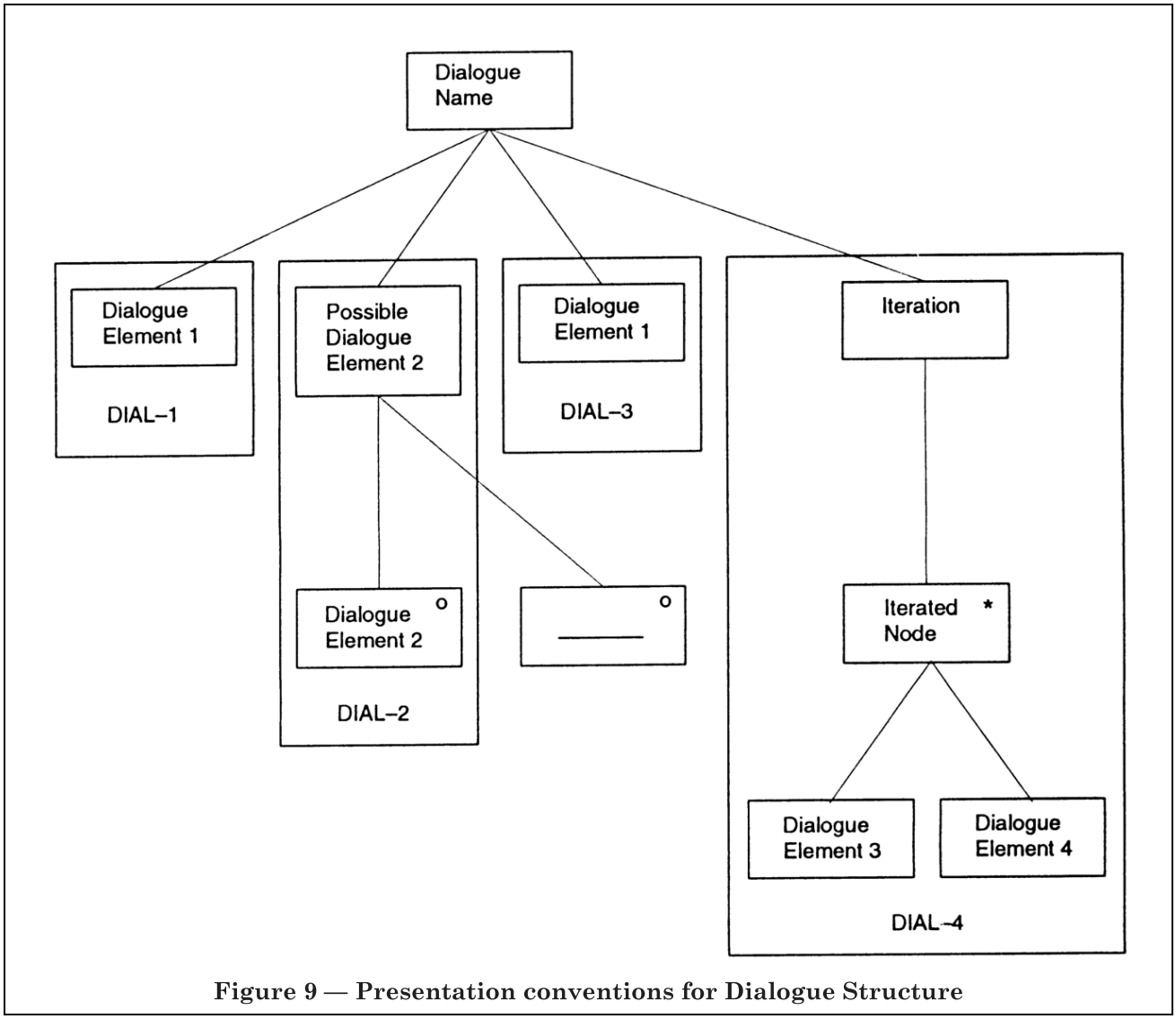


Figure 9 — Presentation conventions for Dialogue Structure

3.3.7 Menu Structure

3.3.7.1 *Purpose*

The Menu Structure shall be used to represent the menu structure for a user role.

3.3.7.2 *Composition*

The Menu Structure shall be a tree structure showing how menus and dialogues interrelate.

3.3.7.3 *Completion criteria*

The following shall be considered.

a) Is the user role identified correctly (named in the top box)?

b) Are the boxes drawn and labelled correctly?

c) Do the boxes accurately reflect a menu or dialogue?

d) Is the "tree" drawn correctly?

Within Stage 3, user(s) belonging to the user role concerned shall have the opportunity to approve the Menu Structures once implemented using the support tool.

In Stage 5 users shall be involved in approving the Menu Structures, even if they are not implemented.

3.3.7.4 *Status*

The Menu Structure shall be an evolving end-module product: RS; completed end-stage product, Stage 5.

3.3.7.5 *Presentation conventions*

The presentation conventions shall be in accordance with Figure 10.

COMMENTARY AND RECOMMENDATIONS ON **3.3.7**.

This product provides a diagrammatic representation of the menus to be used within the system.

Derivation:

a) *Requirements Specification:*

— *discussion with users;*

— *User Role/Function Matrix;*

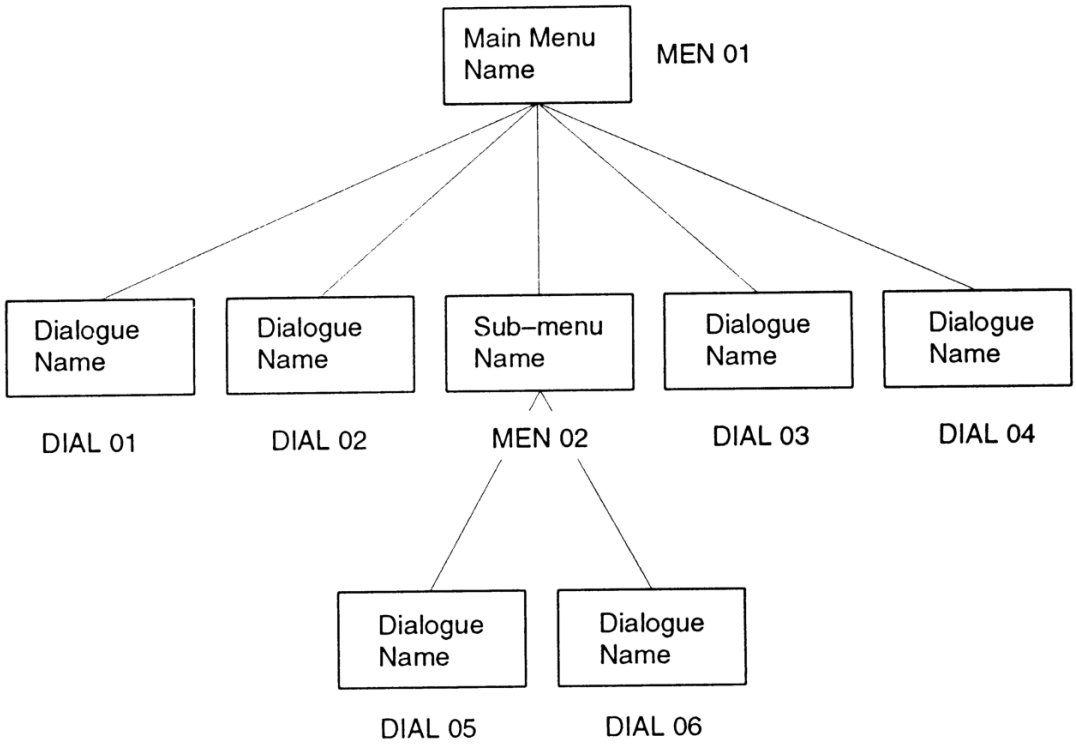


Figure 10 — Presentation conventions for Menu Structures

b) *Logical System Specification:*

— *Dialogues;*

— *User Role/Function Matrix.*

Where used:

— *Dialogue Design: LS;*

— *Specification Prototyping: RS;*

— *Physical Design: PD.*

3.3.8 Report Format

3.3.8.1 *Purpose*

The Report Format shall show the output format required from the system.

NOTE Critical reports may be prototyped (in Stage 3: Develop Specification Prototypes) to ensure user requirements are fully documented.

3.3.8.2 *Composition*

The composition shall be as follows:

a) header details: title of report;

b) report facsimile size shall depend on page requirements: generally maximum size per page of: 160-col by 66-line grid with intended page layout superimposed.

3.3.8.3 *Completion criteria*

The following shall be considered.

a) Does the Report Format conform to existing standards for reports?

b) Do the users agree with the Report Formats?

c) Does the Report Format contain all of the relevant detail?

3.3.8.4 *Status*

The Report Format shall be an end-module product, delivered complete, RS.

COMMENTARY AND RECOMMENDATIONS ON **3.3.8**.
This format shows the layout of a printed report as desired by users.

Derivation:

a) *Requirements Specification:*

— *discussion with users;*

— *existing system documentation;*

— *Installation Style Guide;*

— *Requirements Catalogue;*

b) *Physical Design PD: in physical design, the most effective way of implementing the report to meet the specification has to be considered.*

Where used:

— *Specification Prototyping: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.3.9 Screen Format

3.3.9.1 *Purpose*

The Screen Format shall show how items will be presented on the visual display unit to the user.

NOTE During Stage 3, Develop Specification Prototypes, these may be drawn on a form or be dumped from the prototyped screen on the demonstration system.

3.3.9.2 *Composition*

The composition shall be as follows:

a) header details: title or name of Screen Format;

b) screen facsimile: intended Screen Format superimposed on physical screen layout;

c) footer details:

— cross-references;

— notes.

3.3.9.3 *Completion criteria*

The following shall be considered:

a) Do the Screen Formats conform to any existing standards?

b) Do the users agree with the Screen Formats?

c) Is the Screen Format in the order of entry of data?

3.3.9.4 *Status*

The Screen Format shall be an end-module product, delivered complete, RS.

COMMENTARY AND RECOMMENDATIONS ON **3.3.9**.
This product shows the layout that users require on the visual display unit screen.

Derivation:

a) *Requirements Specification:*

— *discussion with users;*

— *existing system documentation;*

— *Installation Style Guide;*

— *Requirements Catalogue;*

b) *Physical Design PD.*

In physical dialogue design, the most effective way to implement the report to meet the specification has to be considered.

Where used:

— *Specification Prototyping: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.3.10 User Role/Function Matrix

3.3.10.1 *Purpose*

The User Role matrix shall enable the identification of all dialogues to be provided by the system.

3.3.10.2 *Composition*

Column headings shall be function names. Row headings shall be user role identifiers. Cell values shall be as appropriate.

3.3.10.3 *Completion criteria*

The following shall be considered.

a) Have all on-line functions been included in the matrix (as column headings)?

b) Have all user roles been included in the matrix (as row headings)?

c) Have all the necessary intersection points been identified? [i.e. "x" marks intersections between the two (i.e. dialogues) and blank entry denotes that this user role does not make use of this function.]

For critical functions it shall be considered whether all critical functions have been identified with a "c" in the appropriate cell?

3.3.10.4 *Status*

The User Role/Function Matrix shall be an end-module product delivered complete, RS.

COMMENTARY AND RECOMMENDATIONS ON **3.3.10**.
Dialogues are identified as being the cross-reference between user roles and on-line functions (either enquiry or update). This matrix maps correspondences between functions and user roles, thus identifying these dialogues.

Reading down from the user roles axis also provides the (initial) Menu Structure for the system.

Derivation:

Requirements Specification:

— *Function Definitions;*

— *User Roles.*

Where used:

— *Dialogue Design: RS;*

— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.3.11 User roles

3.3.11.1 *Purpose*

User roles shall be used to consolidate all user roles in the system into a complete set.

NOTE By identifying where jobs are described by more than one job title, these can be amalgamated into a single user role. Also, jobs which are very similar, but not identical, can also be amalgamated into a single user role.

3.3.11.2 *Composition*

Each user role shall have the following details:

a) user role name/identifier;

b) job details, repeating group consisting of:

— job title;

— multiple user role activities.

3.3.11.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Does the user role subsume job titles which should, to avoid compromising security, be kept separate?

2) Are more than three job titles covered by one user role? If so this shall be investigated.

3) Is one job title split between more than three user roles? (If so this shall be flagged as a work organization problem, outside the scope of SSADM.)

4) Are activities correctly identified for the associated job title?

b) for the set: is this a complete set of documentation for all identified user roles in the proposed system?

3.3.11.4 *Status*

User roles shall be end-module products, delivered complete, RS.

COMMENTARY AND RECOMMENDATIONS ON **3.3.11**.
These are used to document the details for each user role identified as having a direct interest in the required system.

Derivation:

Requirements Specification:

— *Selected Business System Option;*

— *User Catalogue.*

Where used:

— *Dialogue Design: RS;*

— *Data Flow Modelling: RS;*

— *Function Definition: RS;*

— *Technical Systems Options: LS.*

Presentation conventions are not applicable.

3.4 Entity-Event Modelling products

COMMENTARY AND RECOMMENDATIONS ON **3.4**.
Entity-Event Modelling products are derived from the application of two techniques: entity life history analysis and effect correspondence diagramming.

Entity life history analysis validates the high-level processing and data views of the system, and identifies further detailed processing and data requirements.

Effect correspondence diagramming produces an event-oriented view of the system requirements and the results are used to specify the Update Process models in logical design.

Business rules modelled in the Logical Data Model are handled during entity life history analysis and progressed through the method on the processing side to physical design where any overlap with the Logical Data Model is resolved.

Entity Life Histories (ELHs) describe the business rules and Effect Correspondence Diagrams (ECDs) express the systems organization. The two need to be used in communication with users who have a detailed knowledge of the order in which events need to take place. Such users should also be capable of describing unusual situations which need to be catered for by the system, and of situations which should be rejected as errors.

During Stage 5, ELHs and ECDs are input to the logical database process design technique for designing the database update processing.

3.4.1 Effect Correspondence Diagram

3.4.1.1 *Purpose*

The Effect Correspondence Diagrams shall be used to show the different effects caused by events and how they interrelate. Each event shall be described in a single diagram and a set of these shall be consolidated for use in the later development stages.

3.4.1.2 *Composition*

The header shall be the event name.

Several small SSADM Structure Diagrams shall be drawn illustrating the entity types that are affected by this event; also including details such as effect qualifiers, entity roles, read-onlys, correspondence arrows and knock-on effects on other entity types as appropriate.

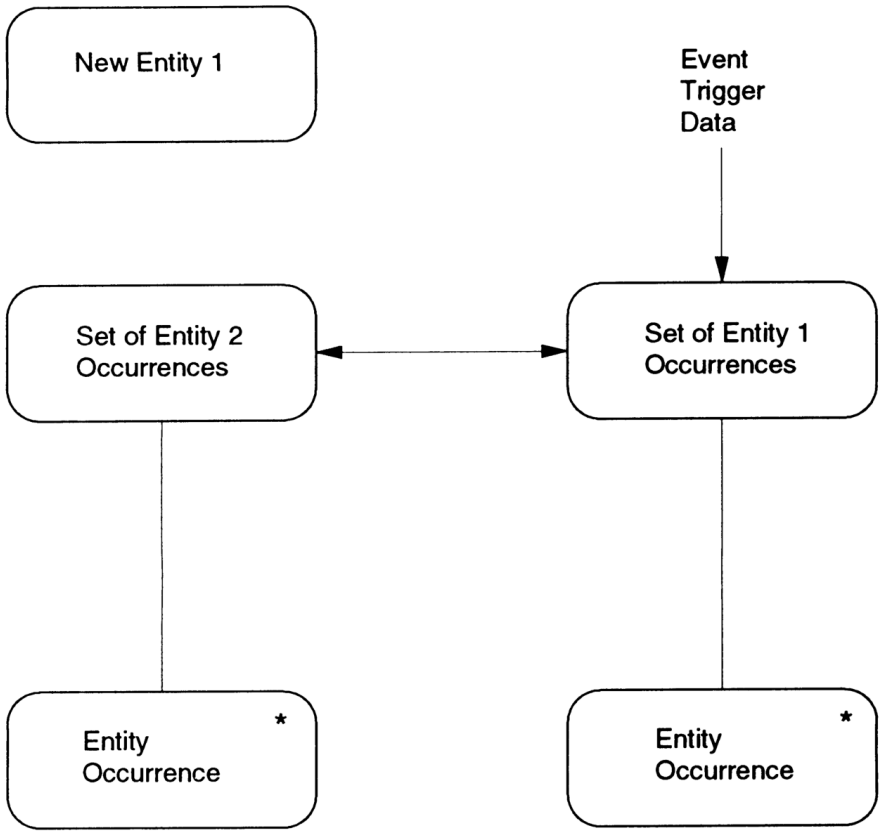


Figure 11 — Presentation conventions for Effect Correspondence Diagrams

3.4.1.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Has the event name been completed correctly for this diagram?

2) Does the Effect Correspondence Diagram show all the effect correspondences between entity types and occurrences?

3) Have all of the entity types which are affected and accessed by a specific event been documented on the same Effect Correspondence Diagram?

4) Does the Effect Correspondence Diagram accurately reflect requirements?

5) Are all agreed notational conventions adhered to?

b) for the set:

1) Is the set of Effect Correspondence Diagrams complete?

2) Have any duplicate ECDs for a single event been resolved?

3.4.1.4 *Status*

The Effect Correspondence Diagrams shall be an end-module product comprising a completed set of diagrams produced at conclusion of RS.

3.4.1.5 *Presentation conventions*

Effect Correspondence Diagrams shall be presented in accordance with Figure 11 and Figure 12.

COMMENTARY AND RECOMMENDATIONS ON **3.4.1**.

These diagrams show all the effects an event has on data within the system and how those effects impact upon each other.

Each diagram shows the effects caused by an event on entity types and occurrences within the system. Effect Correspondence Diagrams provide the access path details for update functions which are used in logical design activities.

Derivation:

Requirements Specification:

— *Entity Life Histories;*

— *Required System Logical Data Model;*

— *Requirements Catalogue.*

Where used:

— *Entity-Event Modelling: RS;*

— *Function Definition: RS;*

— *Logical Database Process Design: LS.*

3.4.2 Entity Life History(s)

3.4.2.1 *Purpose*

Each diagram shall show the sequence and constraints of the required system processing for an entity type.

Each entity type shall be described by a single Entity Life History and a set of these shall be packaged to provide this information for all entities as part of the Requirements Specification (Processing Specification).

3.4.2.2 *Composition*

Each Entity Life History shall comprise a SSADM Structure Diagram with representations for entity type:

— events;

— effects;

— operations;

— state indicators.

NOTE An event may need to be further qualified by an entity role or an effect qualifier so that Effect Correspondence Diagrams can be correctly drawn. Highlight the difference in the type of qualifier by using a different style of brackets to identify an effect as opposed to an entity role.

3.4.2.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Does the entity type name appear in the top box of the diagram?

2) Have all create events which may apply to this entity type been documented on the Entity Life History?

3) Have all modify events which may apply to this entity type been documented on the Entity Life History?

4) Have all delete events which may apply to this entity type been documented on the Entity Life History?

5) Have all events which affect more than one entity occurrence for an entity type been qualified by an entity role?

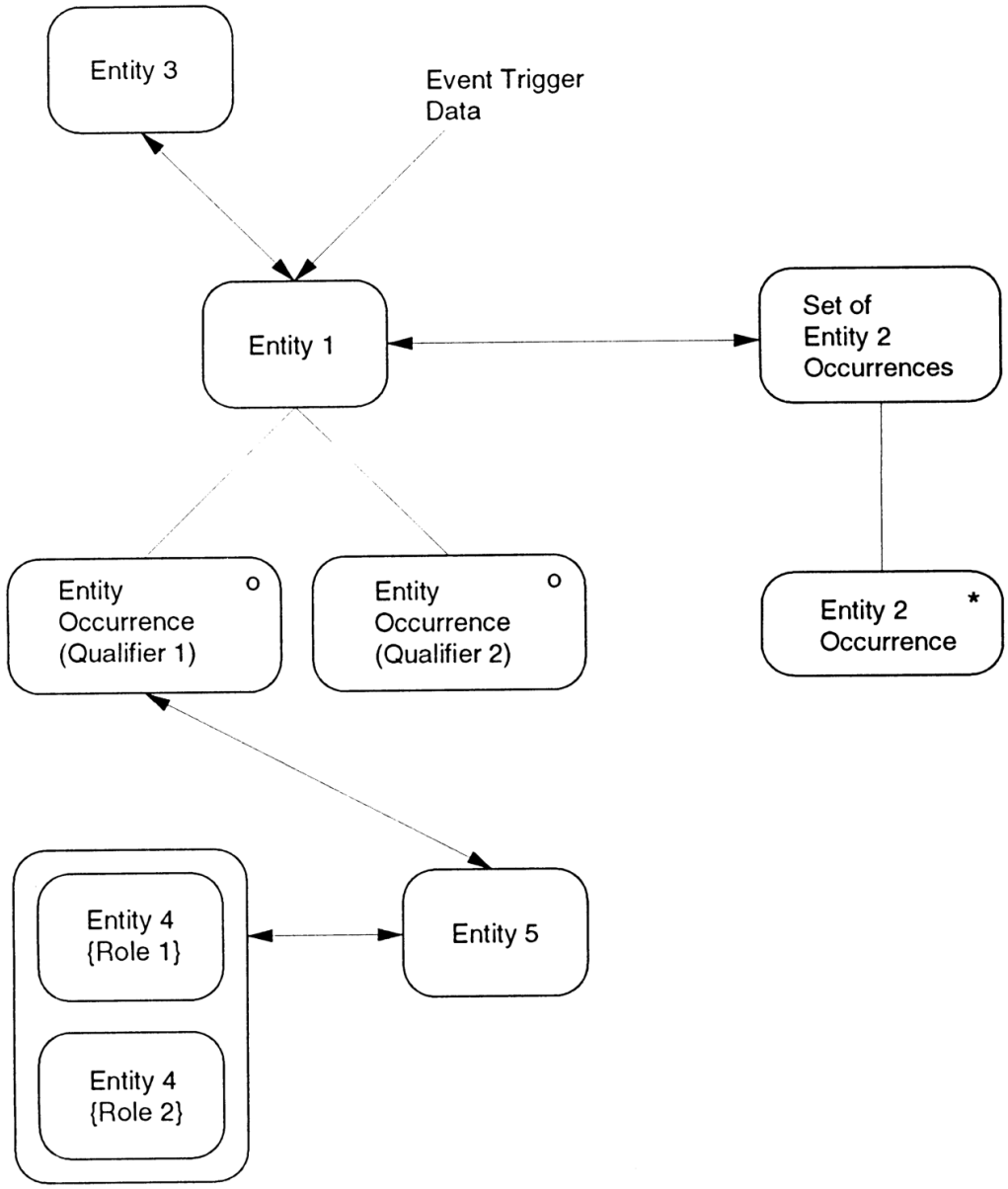
6) Have all events which affect an entity type in mutually exclusive ways been qualified by an effect qualifier?

7) Have all of the major processing operations been identified for each effect on the Entity Life History and correctly allocated?

8) Are all agreed notational conventions adhered to?

9) Does the Entity Life History support the sequence of events required by the business?

10) Has the operations list been provided for this Entity Life History?



NOTE Different brackets for Entity Roles and Effect Qualifiers.

Figure 12 — Presentation conventions for Effect Correspondence Diagrams: Entity Roles and Effect Qualifiers

11) Do the operations conform to the format required by the technique?

b) in Stage 5: Have state indicators been included as appropriate?

c) for the set:

1) Does the set of all Entity Life Histories meet all appropriate business requirements?

2) Does the set of Entity Life Histories accurately illustrate processing constraints?

3.4.2.4 *Status*

The Entity Life History shall be an end-module product; delivered complete, apart from state indicators, RS. State indicators shall be added in the Logical System Specification module.

3.4.2.5 *Presentation conventions*

The Entity Life History shall be presented in accordance with Figure 13 to Figure 19.

COMMENTARY AND RECOMMENDATIONS ON **3.4.2**. *An Entity Life History is a diagrammatic structure combining all possible lives of every possible occurrence of the entity type.*

Derivation:

a) *Requirements Specification:*

— *Data Catalogue;*

— *Function Definition;*

— *Logical Data Store/Entity Cross-reference;*

— *Required System Data Flow Model;*

— *Required System Logical Data Model;*

— *Requirements Catalogue;*

b) Logical System Specification:

— *Entity Life Histories (addition of State Indicators).*

Where used:

— *Entity-Event Modelling: RS;*

— *Logical Database Process Design: LS.*

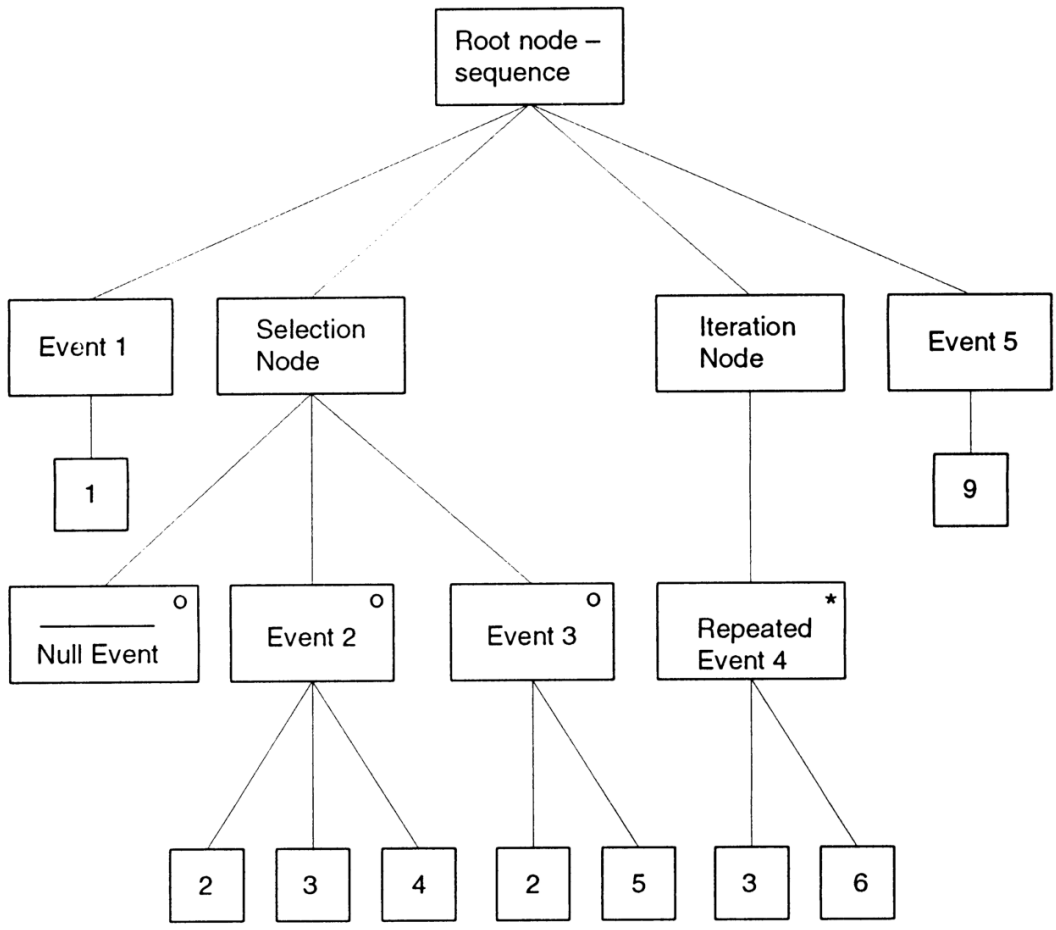


Figure 13 — Presentation conventions for Entity Life Histories

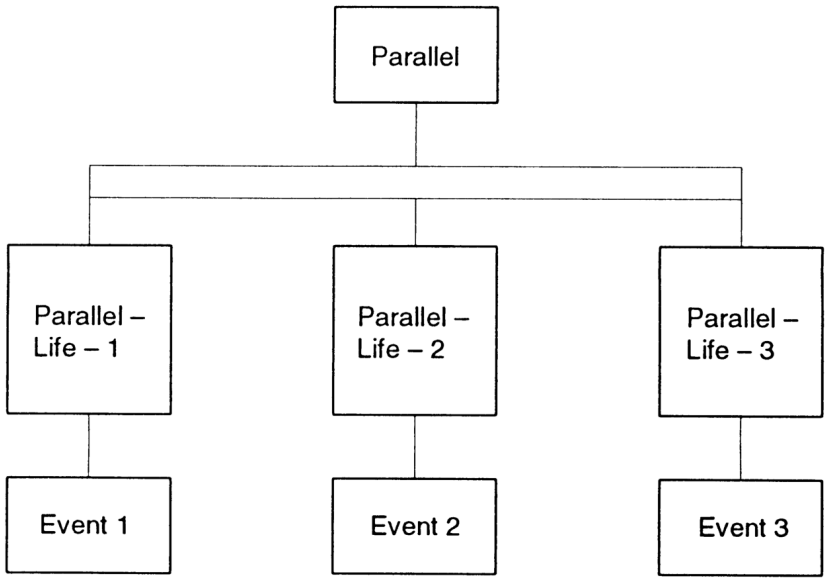


Figure 14 — Presentation conventions for Entity Life Histories: parallel lives

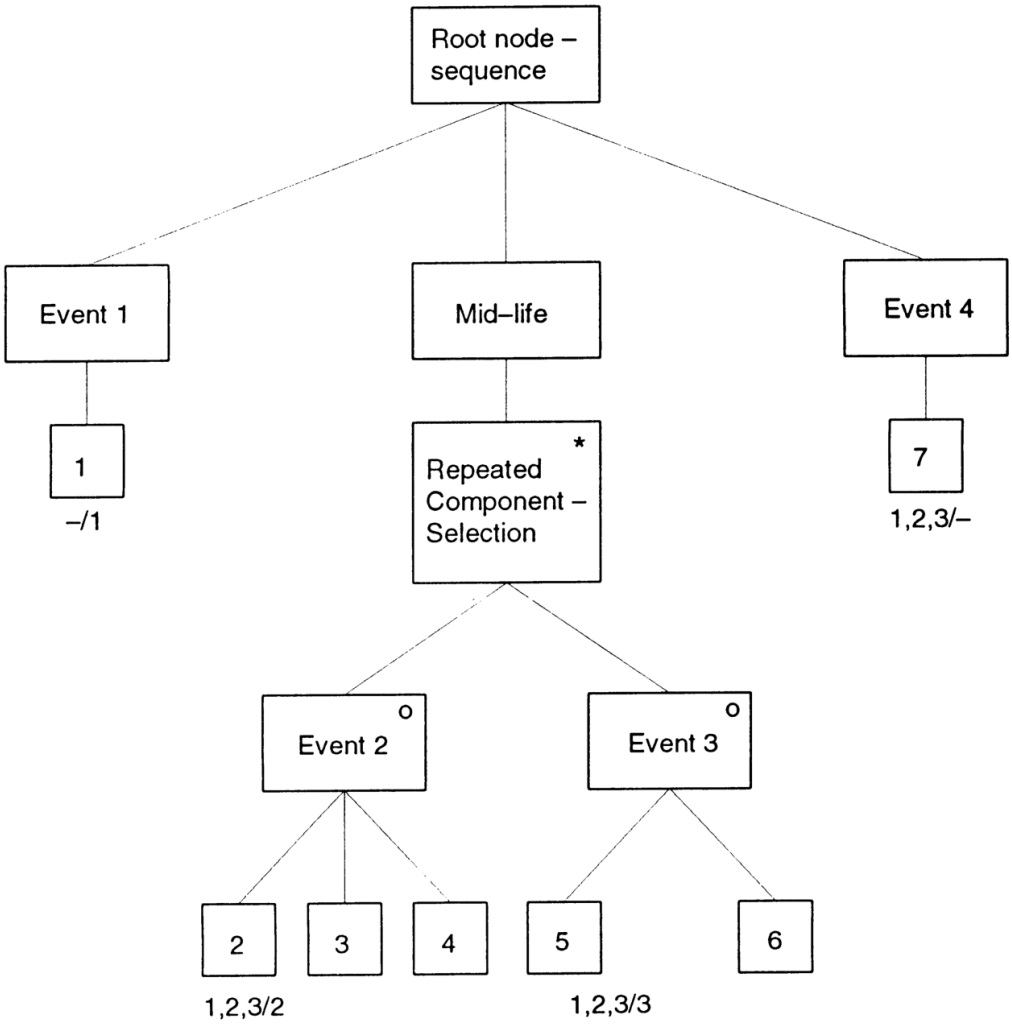


Figure 15 — Presentation conventions for Entity Life Histories: State Indicators

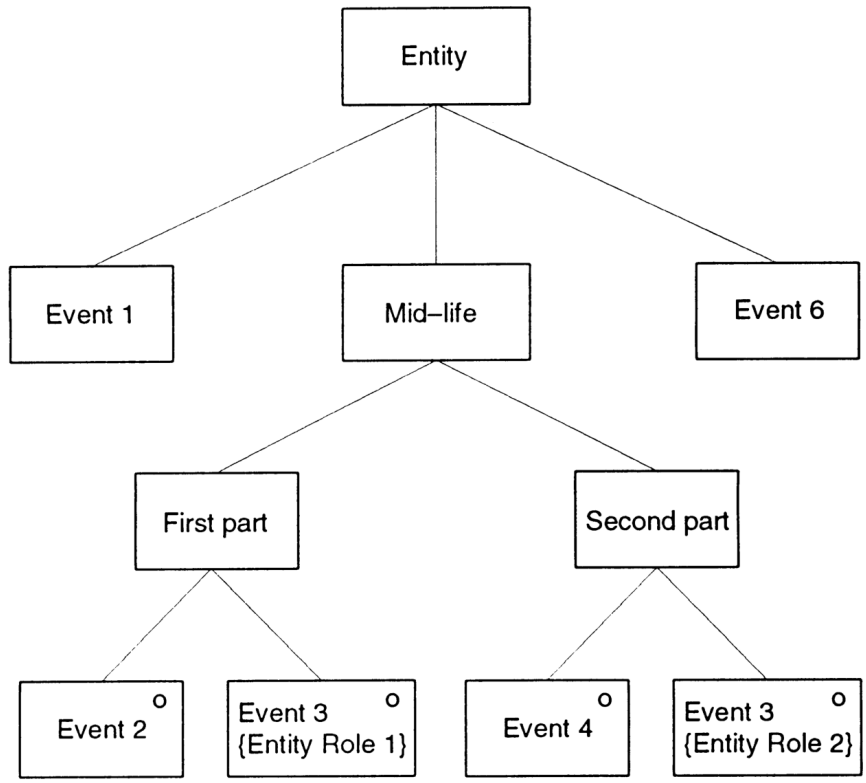
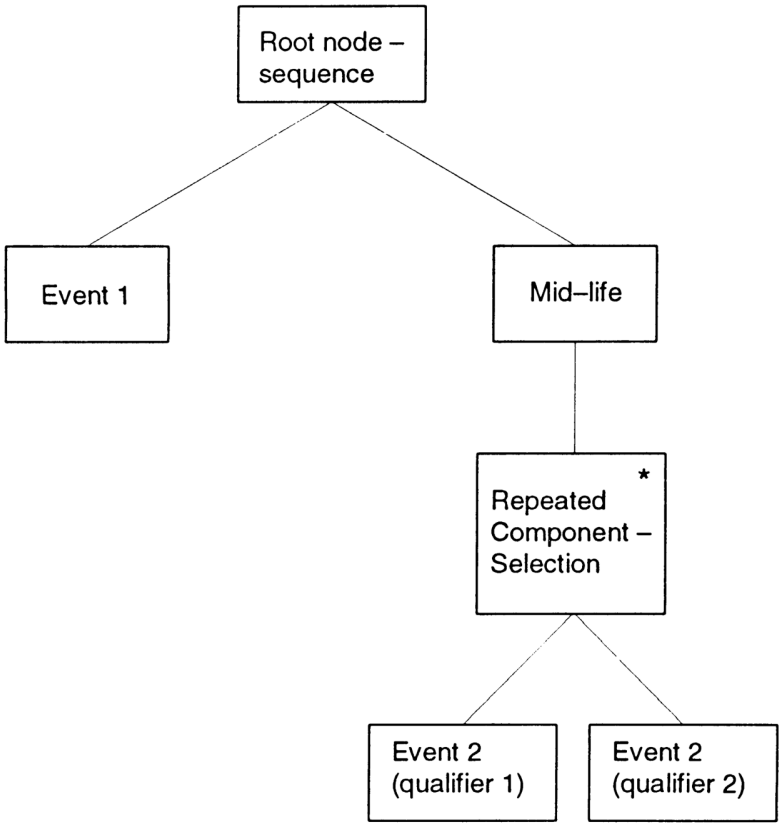


Figure 16 — Presentation conventions for Entity Life Histories: Entity Roles



NOTE Brackets different from those used for Entity Roles.

Figure 17 — Presentation conventions for Entity Life Histories: Effect Qualifiers

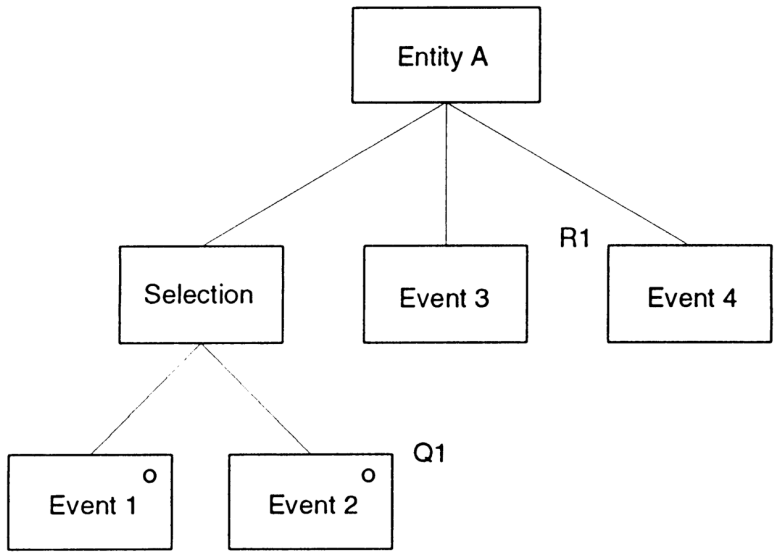


Figure 18 — Presentation conventions for Entity Life Histories: Quit and Resume-1

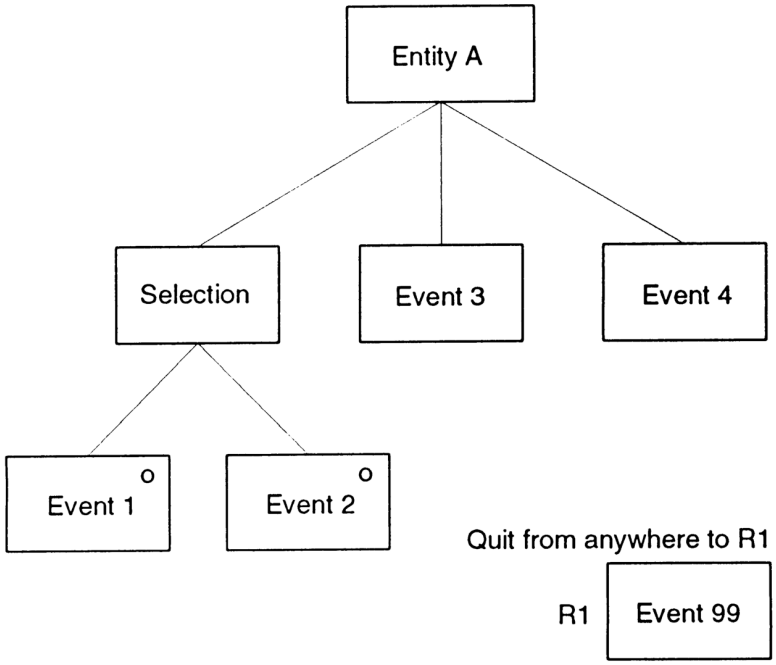


Figure 19 — Presentation conventions for Entity Life Histories: Quit and Resume-2

3.4.3 Event/Entity Matrix

3.4.3.1 *Purpose*

The Event Entity Matrix shall help to ensure all events affecting a Logical Data Model entity type are documented.

NOTE 1 All Logical Data Structure entity types should be affected by one or more events, otherwise they are not a necessary part of the required system.

NOTE 2 Practitioners may use this as a start-up document for entity life history analysis; it is unlikely to be maintained afterwards.

3.4.3.2 *Composition*

Column headings shall be entity names. Row headings shall be event names. Cell values shall be as appropriate.

3.4.3.3 *Completion criteria*

The following shall be considered.

a) Are all events identified during function definition activities listed down the matrix side?

b) Are all Logical Data Model entity types listed across the matrix top?

c) Are all the cell entries C(reate), M(odify), D(logical Delete) or null?

d) Is there any event with no corresponding entity type to act on?

e) Is there a create and delete event for each entity type?

3.4.3.4 *Status*

The Event/Entity Matrix shall be an interim product; completed during Stage 3, Requirements Specification Module.

COMMENTARY AND RECOMMENDATIONS ON **3.4.3**.
This grid is used to identify which entity types are affected by a particular event. It provides two checks: that each event affects at least one entity type, and that each entity type is affected by at least one event.

Derivation:

Requirements Specification

— *Function Definitions;*

— *Required System Logical Data Model;*

— *Logical Data Store/Entity Cross-reference.*

Where used:

— *Entity-event Modelling: RS.*

Presentation conventions are not applicable.

3.5 Function Definition products

COMMENTARY AND RECOMMENDATIONS ON **3.5**.
Function Definition comprises a procedure with the objective of identifying units of processing specification, or functions, from which the physical design of the system may be derived.

Function Definition has several purposes as follows.

a) *It identifies and defines the units of processing specification to be carried forward to physical design.*

b) *It consolidates the products of analysis and design which together specify a function.*

c) *It supports the optimal organization of the system processing to support user tasks.*

d) *It develops and confirms a common understanding between practitioners and users of how the system processing is to be organized.*

e) *It reconciles the two views of the system processing developed during requirements definition as embodied in the Required System Data Flow Diagrams and the events that emerge from entity-event modelling.*

f) *It provides a basis for sizing and deriving design objectives.*

3.5.1 Enquiry Access Path

3.5.1.1 *Purpose*

The Enquiry Access Path shall be used to document the data access paths implied by enquiry components.

3.5.1.2 *Composition*

The Enquiry Access Path shall be a function identifier whose graphical representation shall be based on concepts described in the SSADM Structure Diagram Product Description, with arrows showing navigation.

3.5.1.3 *Completion criteria*

The following shall be considered.

a) Is the function correctly identified?

b) Is the data structure visible on the Logical Data Model?

c) Does the description of access rights in the entity type and attribute/data item descriptions support the accesses defined in the access path?

d) Is the access path valid for this function?

3.5.1.4 *Status*

The Enquiry Access Path shall be an end-module product, delivered complete, RS.

3.5.1.5 *Presentation conventions*

The Enquiry Access Path shall be presented in accordance with Figure 20.

COMMENTARY AND RECOMMENDATIONS ON **3.5.1**.
This product illustrates the route through the Logical Data Model from an entry point to the entity type or types, required to satisfy a particular enquiry function.

Derivation:

Requirements Specification

— *Data Catalogue;*

— *Function Definitions (enquiry functions only);*

— I/O *Structures;*

— *Required System Logical Data Model.*

Where used:

— *Logical Data Modelling: RS;*

— *Logical Database Process Design: LS.*

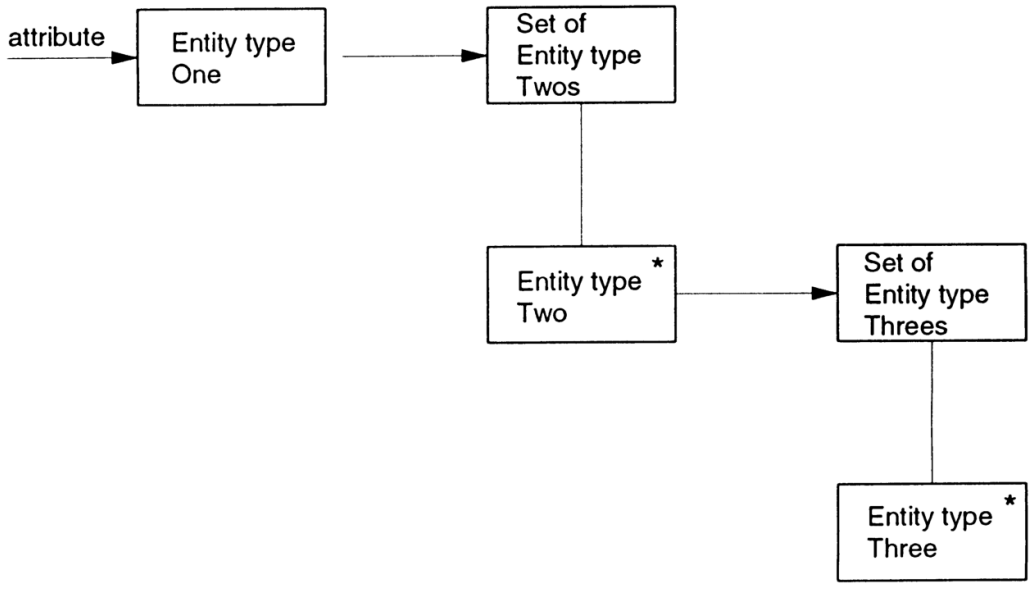


Figure 20 — Presentation conventions for Enquiry Access Paths

3.5.2 Function Definition

3.5.2.1 *Purpose*

Each Function Definition shall serve to define a function which is to be provided by the required system in order to draw together the SSADM documentation which describes the components of a function and to provide a user view of the system processing in preparation for further process design.

3.5.2.2 *Composition*

The composition shall be as follows:

a) heading details:
- — Function name;
- — Function identifier;

b) Function categorization:
- — Function Type;
- — User Roles;

c) Function details:
- — Function description;
- — error handling description(s);

d) references to:
- — DFD Processes;
- — Event details, repeating group consisting of event name/identifier and event frequency;
- — I/O Descriptions;
- — I/O Structures;
- — Requirements Catalogue Reference;
- — volumes;
- — related functions;
- — enquiry details, repeating group consisting of Enquiries and Enquiry Frequency;
- — common processing reference;
- — Dialogue names;
- — Service Level Requirements, repeating group consisting of Service Level Description, Service Level Target Value, Service Level Range and Service Level Comments.

3.5.2.3 *Completion criteria*

The following shall be considered.

a) Is the function definition entry complete, or as complete as is feasible at this time?

b) Is the function identifier unique?

c) If this is an enquiry function, is it specified without a list of events?

d) If this is an update function, does it include one or more events?

e) Is the function classified according to all three types:
- — update or enquiry;
- — on-line or off-line;
- — user or system initiated?

It shall be considered during physical design if the description reflects the appropriate physical restrictions, based on the implementation vehicle.

3.5.2.4 *Status*

The Function Definition shall be an evolving product, delivered at conclusion of RS module or delivered complete as end-module product PD module.

COMMENTARY AND RECOMMENDATIONS ON **3.5.2**.
This definition provides the description of a function included within the required system. In addition it encompasses reference to other SSADM products associated with the function.

Derivation:

a) *Requirements Specification*
- — *Data Catalogue;*
- — *Effect Correspondence Diagrams;*
- — *Required System Data Flow Model;*
- — *Required System Logical Data Model;*
- — *Requirements Catalogue;*
- — *User Roles;*

b) *Physical Design:*
- — *Application Development Standards;*
- — *Function Component Implementation Map;*
- — *Function Definitions;*
- — *Logical Design;*
- — *Physical Data Design (first-cut);*
- — *Process Data Interface.*

Where used:
- — *Function Definition: RS;*
- — *Entity-Event Modelling: RS;*
- — *Logical Database Process Design: LS;*
- — *Physical Design: PD.*

Presentation conventions are not applicable.

3.5.3 Function Definitions (Set)

3.5.3.1 *Purpose*

This set of items shall consolidate all of the documentation relating to functions to ensure that a complete set is passed between stages in development.

3.5.3.2 *Composition*

The composition shall be as follows:

a) Multiple function details each consisting of:
- — Function Definition;
- — one or more I/O Structures;

b) Enquiry Access Path (for enquiry functions only):

— Multiple (Common) Elementary Process Descriptions.

NOTE Some of the I/O Structures are subsumed within the Process Models produced in Stage 5; others pass directly into Physical Design. (See Figure B.12 for the product breakdown structure diagram.)

3.5.3.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Is this a complete set of documentation for all identified functions?

2) Are the cross-references to/from common processing descriptions and function definitions complete and correct?

3) Are all referenced common Elementary Process Descriptions included in the set?

4) Does each Function Definition have the appropriate I/O Structures, i.e. do the I/O Structures fully document input/output details for each function?

5) Is each I/O Structure associated with the appropriate Function Definition?

b) during Stage 3: Is each enquiry function associated with an Enquiry Access Path?

c) during Stage 5: Are I/O Structures included only where necessary (i.e., only those which have not been translated into process models)?

3.5.3.4 *Status*

The Function Definitions (set) shall be an evolving product delivered at completion of RS module or an end-module product completed at conclusion of PD module.

COMMENTARY AND RECOMMENDATIONS ON **3.5.3**.

This product consolidates the details for all functions to be included in the Requirements Specification. These details are further expanded during physical design activities.

Basic information about a function consists of a Function Definition with one or more I/O *Structures. Further information may be found in Enquiry Access Paths (for enquiry functions) and (common) Elementary Process Descriptions.*

Derivation:

a) *Requirements Specification:*

— *Data Catalogue;*

— *Required System Data Flow Model;*

— *Required System Logical Data Model;*

— *Requirements Catalogue;*

— *User Roles;*

b) *(for Enquiry Access Paths only):*

— *Data Catalogue;*

— *Function Definitions (enquiry functions only);*

— I/O *Structures;*

— *Required System Logical Data Model;*

c) *Physical Design:*

— *Application Development Standards;*

— *Function Component Implementation Map;*

— *Function Definitions;*

— *Logical Design;*

— *Physical Data Design (first-cut);*

— *Process Data Interface.*

Where used:

— *Function Definition: RS;*

— *Logical Data Modelling: RS;*

— *Entity-Event Modelling: RS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.5.4 I/O Structure

3.5.4.1 *Purpose*

I/O Structures shall specify the input and output data items for a function, defining the sequencing within them. In particular they provide the detail from which the on-line dialogues are developed.

3.5.4.2 *Composition*

The composition shall be as follows:

— I/O Structure Diagram (see **3.5.6**);

— I/O Structure Description (see **3.5.5**).

3.5.4.3 *Completion criteria*

The following shall be considered.

a) Are all I/O Structure elements on the I/O Structure Diagram documented on the I/O Structure Description?

b) Does the I/O Structure Description describe only elements on the I/O Structure Diagram?

3.5.4.4 *Status*

The I/O Structure shall be an end-module product; delivered complete at conclusion of RS module.

COMMENTARY AND RECOMMENDATIONS ON **3.5.4**.

This product encompasses the inputs to and outputs from a function, or part of a function. It comprises an I/O *Structure Diagram and* I/O *Structure Description.*

Derivation:

Requirements Specification:

— *Data Catalogue;*

— *Function Definition;*

— I/O *Descriptions;*

— *Requirements Catalogue.*

Where used:

— *Function Definition: RS;*

— *Relational Data Analysis: RS;*

— *Specification Prototyping: RS;*

— *Entity-event Modelling: RS;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.5.5 I/O Structure Description

3.5.5.1 *Purpose*

The I/O Structure Description shall document the I/O Structure down to data item level. It also documents additional information about the I/O Structure elements, for example the number of occurrences of a repeating group.

3.5.5.2 *Composition*

The composition shall be as follows:

a) Heading consisting of:

— I/O Structure Description identification;

— Data flows represented;

b) Structure element details, a repeating group of:

— I/O Structure element name;

— Data items associated with this element;

c) Comments.

3.5.5.3 *Completion criteria*

The following shall be considered:

a) Is the I/O Structure Description complete?

b) Does every I/O Structure element include at least one data item?

c) If the I/O Structure has been developed from one or more I/O Descriptions, does the I/O Structure Description contain all the relevant data items on those I/O Descriptions?

3.5.5.4 *Status*

The I/O Structure Description shall be an end-module product, delivered complete at conclusion of RS module.

COMMENTARY AND RECOMMENDATIONS ON **3.5.5**.

This product details the inputs to and outputs from a function, or part of a function.

Derivation:

Requirements Specification:

— *Data Catalogue;*

— *Function Definition;*

— I/O *Descriptions;*

— *Requirements Catalogue.*

Where used:

— *Function Definition: RS;*

— *Relational Data Analysis: RS;*

— *Specification Prototyping: RS;*

— *Entity-event Modelling: RS;*

— *Dialogue Design: RS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.5.6 I/O Structure Diagram

3.5.6.1 *Purpose*

The I/O Structure Diagram shall be used to show graphically the sequencing of data items or groups of data items within the data flows into and out of a function.

3.5.6.2 *Composition*

The Identifier shall be the function name. The I/O Structure Diagram shall be a diagrammatic representation, based on the concepts in the SSADM Structure Diagram Product Description, of the inputs and outputs of the function.

3.5.6.3 *Completion criteria*

The following shall be considered.

a) Is the function name completed and correct?

b) Does the structure conform to the diagramming rules?

c) Are the I/O Structure elements labelled as input or output?

3.5.6.4 *Status*

The I/O Structure Diagram shall be an end-module product, delivered complete at conclusion of RS module.

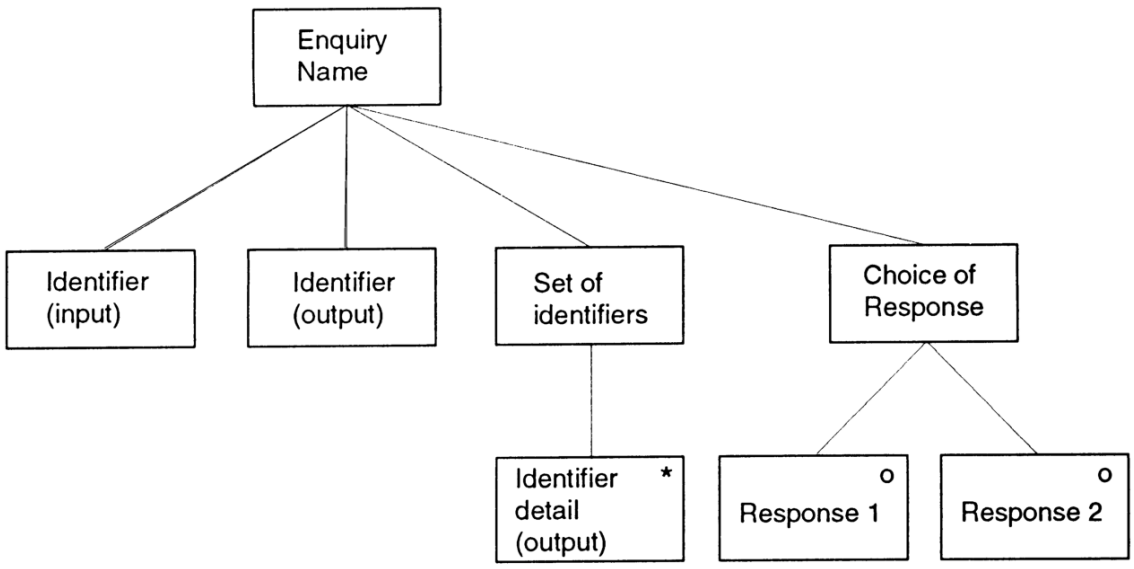


Figure 21 — Presentation conventions I/O Structure Diagram

3.5.6.5 ***Presentation conventions***

The I/O Structure Diagram shall be presented in accordance with Figure 21.

COMMENTARY AND RECOMMENDATIONS ON **3.5.6**.
This product illustrates diagrammatically the inputs to and outputs from a function, or part of a function.

Derivation:

Requirements Specification:

— *Data Catalogue;*
— *Function Definition;*
— I/O *Descriptions;*
— *Requirements Catalogue.*

Where used:

— *Function Definition: RS;*
— *Relational Data Analysis: RS;*
— *Specification Prototyping: RS;*
— *Entity-event Modelling: RS;*
— *Dialogue Design: RS;*
— *Logical Database Process Design: LS;*
— *Physical Design: PD.*

3.5.7 I/O Structures (for all functions)

3.5.7.1 ***Purpose***

I/O Structures shall consolidate all of the input/output details of data flows used within all of the functions.

NOTE On the Structural Model the I/O Structures may be passed to Steps in the Method without the Function Definitions (never the other way round); this ensures that it is a complete set which is passed.

3.5.7.2 ***Composition***

The composition shall be a set of I/O Structures for all defined functions.

NOTE See Figure B.20 for the product breakdown structure diagram.

3.5.7.3 ***Completion criteria***

It shall be considered if this is a complete set of documentation for all identified data flows for the complete set of functions.

3.5.7.4 ***Status***

The I/O Structures (for all functions) shall be end-module products, delivered complete at the conclusion of RS Module.

COMMENTARY AND RECOMMENDATIONS ON **3.5.7**.
This product consolidates all I/O *Structures for all identified functions.*

Derivation:

Requirements Specification:

— *Data Catalogue;*
— *Function Definition;*
— I/O *Descriptions;*
— *Requirements Catalogue.*

Where used:

— *Function Definition: RS;*
— *Relational Data Analysis: RS;*
— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.6 Logical Data Modelling products

COMMENTARY AND RECOMMENDATIONS ON **3.6**.
Logical Data Modelling comprises a group of techniques aimed at the development of an accurate model of the information structure of the context within which the system is to be developed.

As the modelling progresses, the organizational and users' view of the information model is transformed into a system processing view.

The overall Logical Data Model is developed to satisfy the following objectives:

a) *help the analyst(s) understand the application area;*

b) *provide a clear, precise and simple diagrammatic representation of principal features of the system, so that the model can serve as a basis for effective communication with users;*

c) *serve as a basis for common understanding between development personnel early in the development process;*

d) *offer the basis for file and database design, but remaining independent of any specific implementation, technique or product;*

e) *act as a definition of relevant terminology for use in user guides.*

3.6.1 Attribute/Data Item Description(s)

3.6.1.1 *Purpose*

The Attribute/Data Item Descriptions shall consolidate all Attributes/Data Item Descriptions together (being a component of the Data Catalogue).

Each description shall document all of the details relating to a particular attribute or data item, regardless of which technique has been used in obtaining the information.

Only one central description of an attribute or data item shall be maintained and accessed whenever necessary.

NOTE In the logical system, attributes are documented (though these may be thought of as logical data items); these generally translate into physical data items for the implemented system.

3.6.1.2 *Composition*

The composition shall be as follows:

a) attribute/data item name;

b) attribute/data item identifier;

c) cross-reference details; repeating group consisting of cross-reference, name/identifier, cross-reference type:

— synonyms;

— description;

— validation/derivation details;

— mandatory indicator;

— default value;

— optional indicator;

— value if null;

d) logical details:

— logical format;

— unit of measure;

— logical length;

— length description;

e) user roles details, repeating group of: user role name, access rights;

f) owner;

g) standard messages;

h) notes.

3.6.1.3 *Completion criteria*

The following shall be considered:

a) for each description:

1) Has this been identified as an attribute or a data item?

2) Is the attribute assigned to one and only one entity type?

b) in Stage 3:

1) Are the descriptions of entity type and data item user role owners consistent?

2) Is the documentation complete (except where this details a state indicator)?

c) for the set:

1) Is the set complete for a specific variant?

2) Is the version number of the data catalogue consistent throughout the set?

3) Is the set consistent with or correctly derived from the previous version?

3.6.1.4 *Status*

The Attribute/Data Item Description shall be an evolving product, updated throughout the method and finally delivered as a complete, end-module product at the end of Stage 6.

NOTE Specific variants of this product may be delivered as components of delivered variants of the Data Catalogue.

COMMENTARY AND RECOMMENDATIONS ON **3.6.1**.
Each description documents all known details about an attribute (data item) in the logical system. Details are later translated into the physical data item(s) for the implementation of the system.

Derivation:

Updated at all points within the method (except: Stage 2, Business System Options, and Stage 4, Technical System Options) based on the information available at that time.

Where used:

— *Logical Data Modelling: RA, RS;*

— *Data Flow Modelling: RA, RS;*

— *Function Definition: RS;*

— *Relational Database Analysis: RS;*

— *Specification Prototyping: RS;*

— *Entity-event Modelling: RS;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD;*

Presentation conventions are not applicable.

3.6.2 Data Catalogue

3.6.2.1 *Purpose*

The catalogue shall be developed to encompass all of the details relating to attributes and/or data items, regardless of which technique has been used in obtaining the information. The aim shall be to ensure that attribute/data item details are maintained centrally and thus form a complete and consistent, and readily available, set for use in all activities.

3.6.2.2 *Composition*

The composition shall be:

— Multiple Attribute/Data Item Descriptions;

— Multiple Grouped Domain Descriptions.

NOTE See Figure B.10 for the product breakdown structure diagram.

3.6.2.3 *Completion criteria*

The following shall be considered.

a) Are all identified attributes and data items fully documented?

b) Are all identified grouped domains fully documented?

c) Do all attributes mentioned in the grouped domains exist? Are they consistent with details?

d) When data items and attributes describe the same or similar details, are they correctly cross-referenced?

e) Are version numbers consistent?

3.6.2.4 *Status*

Overall, the data catalogue shall be an evolving product, only finally completed after Stage 6.

NOTE Distinct versions of the data catalogue, consistent with end-stage data models (Current Environment LDM, Required System LDM, Physical Data Design), may be frozen and retained as separate variants.

COMMENTARY AND RECOMMENDATIONS ON **3.6.2**.
This is the central repository for all the descriptive information about attributes and items of data. This may include physical details found during data flow modelling and physical design activities.
*See also **3.6.1** and **3.6.4**.*

Derivation:

Updated at all points within the method based on the information available at that time.

Where used:

— *Logical Data Modelling: RA, RS;*

— *Data Flow Modelling: RA, RS;*

— *Function Design: RS;*

— *Relational Data Analysis: RS;*

— *Specification Prototyping: RS;*

— *Entity-event Modelling: RS;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.6.3 Entity Description(s)

3.6.3.1 *Purpose*

The descriptions shall be developed to provide comprehensive documentation for all entity types.

Each entity shall be described under the headings indicated in **3.6.3.2**; all the entries shall then be consolidated as part of the Logical Data Model.

3.6.3.2 *Composition*

Each Entity Description shall consist of the following:

a) Variant identifier, one of:

— Outline;

— Current Environment;

— Required System;

b) Entity Type:

— Entity type name;

— Entity type identifier;

c) location;

d) number of occurrences;

— average;

— maximum;

e) description;

f) synonym(s);

g) attribute details, repeating group of:

— attribute name/identifier;

— primary key;

— foreign key;

h) relationship details, repeating group of:
— relationship identifier;
— "must be/may be" indicator;
— "either/or" (for exclusive relationships);
— relationship link phrase;
— "one and only one/one" or more phrase;
— object entity name;

i) User Role details, repeating group of:
— User Role name;
— access rights;

j) owner;

k) growth per period;

l) additional relationships;

m) archive and destruction details;

n) security measures;

o) state indicator values;

p) notes.

3.6.3.3 *Completion criteria*

The following shall be considered:

a) for each Entity Description:

1) Is the variant identifier completed and valid?

2) Is this entity type really an entity type, i.e. is it meaningful to users in the sense that it represents a group of objects about which information needs to be held in the context of the system under development?

3) Is the entity type name singular and meaningful?

4) Does the entity type have a primary key?

5) Is this entity type fully defined?

6) Can unique instances of the entity type be visualized?

7) Has volumetric information been included? (where known)

8) Are all attribute names singular and meaningful?

9) Have all attributes been identified for this entity type?

10) Are the user role, access, and owner details consistent with the entity type and attribute descriptions?

11) Have all entity type synonyms been identified?

12) Are relationships correctly represented as foreign keys?

b) for the set: Is the set of entity descriptions complete for this version?

3.6.3.4 *Status*

An Entity Description shall be an evolving product, moving through named variants and connected with a specific logical data model finally completed in Stage 6.

COMMENTARY AND RECOMMENDATIONS ON **3.6.3**.

These descriptions document all the details concerned with all entity types on the Logical Data Structure, including details of state indicators which are applied during entity life history analysis (Logical System Specification).

There will be associated Relationship Descriptions for each related entity type on the Logical Data Structure.

Derivation:

a) *Feasibility Module (for Outline Logical Data Model):*

— *High-level Logical Data Model;*

b) *Requirements Analysis (for Current Environment Logical Data Model):*

— *discussion with users (interview notes);*

— *Overview Logical Data Structure;*

c) *Requirements Specification:*

1) *for Required System Logical Data Model:*

— *Current Environment Logical Data Model;*

— *discussion with users;*

— *Elementary Process Descriptions;*

— *Overview Logical Data Structure;*

— *Requirements Catalogue;*

— *Selected Business System Option;*

2) *for Required System Logical Data Model (normalized):*

— I/O *Structures;*

— *Required System Logical Data Model;*

3) *for Required System Logical Data Model (extended):*

— *Entity Life Histories;*

— *Required System Logical Data Model;*

d) *Logical System Specification:*

for Required System Logical Data Model (in logical design):

Entity Life Histories;

Required System Logical Data Model.

Where used:

Use can be distinguished in terms of each of the three LDMs;

— *Logical Database Modelling: RA, RS;*

— *Feasibility: FS;*

— *Relational Data Analysis: RS;*

— *Entity-event Modelling: RS, LS;*

— *Logical Database Process Design: RS;*

— *Physical Design: PD;*

Presentation conventions are not applicable.

3.6.4 Grouped Domain Description

3.6.4.1 *Purpose*

Grouped Domain Descriptions shall document the details of a grouped domain. The concept of a grouped domain shall be used as part of logical data modelling to represent validation and formatting rules, permitted classes and ranges of values which are common to more than one attribute, for example, all "Date" attributes would be based on the same grouped domain, "Dates".

NOTE A Grouped Domain Description may be used to record common attribute descriptions if this saves effort.

3.6.4.2 *Composition*

The composition shall be as follows:

a) grouped domain identification:

— grouped domain name;

— grouped domain Identifier;

— synonym(s);

— description;

— validation/derivation;

— default value;

— value for null;

b) logical details:

— logical format;

— logical length;

— unit of measure;

— length description;

c) User Role details, repeating group of:

— User role;

— access rights;

d) owner;

e) notes.

3.6.4.3 *Completion criteria*

The following shall be considered:

a) Does the Grouped Domain Description apply to all the corresponding attributes?

b) Where formatting and validation rules are further refined in individual Attribute/Data Item Descriptions, are these refinements consistent with the general rules in the Grouped Domain Description?

c) Does the grouped domain include more than one attribute?

d) Are the details completed for each entry?

3.6.4.4 *Status*

The Grouped Domain Description shall be an evolving product, finally delivered complete after Stage 6.

COMMENTARY AND RECOMMENDATIONS ON **3.6.4**.
These descriptions are used to document validation rules and formats common to several attributes.

Derivation:

Updated at all points within the method based on the information available.

Where used:

— *Logical Data Modelling: RA, RS;*

— *Data Flow Modelling: RA, RS;*

— *Function Definition: RS;*

— *Relational Data Analysis: RS;*

— *Specification Prototyping: RS;*

— *Entity-event Modelling: RS;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

3.6.5 Logical Data Model

3.6.5.1 *Purpose*

The Logical Data Model shall provide a detailed logical description of the data and its structure.

3.6.5.2 *Composition*

The composition shall be as follows:

— Logical Data Structure (LDS);

— Entity Descriptions;

— Relationship Descriptions.

NOTE 1 Logical data modelling generates information about attributes. The Attribute/Data Item Descriptions and Grouped Domain Descriptions are maintained within the Data Catalogue.
NOTE 2 The Overview LDM consists only of the Overview LDS.
NOTE 3 See Figure B.11 for the product breakdown structure diagram.

3.6.5.3 *Completion criteria*

The following shall be considered:

a) Is the model variant (e.g. current environment or required system) correctly and consistently assigned to all components of the model?

b) Is every entity type depicted in the Logical Data Structure also described in the Entity Descriptions?

c) Is every relationship depicted in the Logical Data Structure also described in the Relationship Descriptions?

d) Are only entity types depicted in the Logical Data Structure also described in the Entity Descriptions?

e) Are only relationships depicted in the Logical Data Structure also described in the Relationship Descriptions?

f) Is the model consistent with previous versions?

3.6.5.4 *Status*

There shall be one Logical Data Model. There shall be six possible variants of this model: Overview, Current Environment, Required System, Required System (normalized), Required System (expanded) and Required System (logical design).

NOTE 1 The Overview variant may be produced in Feasibility and completed in RA.

NOTE 2 The Current Environment variant may be produced in RA.

NOTE 3 The forms of the Required System LDM may be produced during RS.

COMMENTARY AND RECOMMENDATIONS ON **3.6.5**.

This model provides a detailed description of the information requirements for the system under development. It serves as a basis for file and database design, but is independent of any specific implementation technique or product.

The Logical Data Model consists of a Logical Data Structure, Entity Descriptions and Relationship Descriptions. Associated descriptions of attribute/data items and grouped domains are maintained in the Data Catalogue.

Derivation:

a) *Feasibility Module [for Overview Logical Data Model (structure)]:*

— *Project Initiation Document;*

b) *Requirements Analysis:*

1) *for Overview Logical Data Model (structure):*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *project initiation document;*

2) *for Current Environment Logical Data Model:*

— *discussion with users (interview notes);*

— *Overview Logical Data Model (structure);*

c) *Requirements Specification:*

1) *for Required System Logical Data Model:*

— *Current Environment Logical Data Model;*

— *discussion with users;*

— *Elementary Process Descriptions;*

— *Requirements Catalogue;*

— *Selected Business System Option;*

2) *for Required System Logical Data Model (normalized):*

— I/O *Structures;*

— *Required System Logical Data Model;*

3) *for Required System Logical Data Model (expanded):*

— *Entity Life Histories;*

— *Required System Logical Data Model (normalized).*

d) *Logical System Specification:*

for Required System Logical Data Model (logical design):

— *Entity Life Histories;*

— *Required System Logical Data Model expanded).*

Where used:

— *Logical Data Modelling: FS, RA, RS;*

— *Relational Data Analysis: RS;*

— *Data Flow Modelling: RA, RS;*

— *Feasibility: FS;*

— *Business System Options: RA;*

— *Function Definition: RS;*

— *Entity-event Modelling: RS*

— *Technical System Options: RS;*

— *Logical Database Process Design: LS;*

— *Physical Process Design: PD.*

Presentation conventions are not applicable.

3.6.6 Logical Data Structure

3.6.6.1 *Purpose*

The Logical Data Structure shall provide a logical structure of the non-transient system data.

3.6.6.2 *Composition*

The Logical Data Structure shall be a model variant identifier comprising one of the following:

— Overview;

— Current Environment;

— Required System.

The Logical Data Structure shall be a graphical representation using entity-relationship modelling.

3.6.6.3 *Completion criteria*

The following shall be considered.

a) Is the variant identifier completed correctly?

b) Is each entity type really an entity?

c) Are all entity type names singular and meaningful?

d) Does each entity type have a unique identifier?

e) Is each relationship really a relationship, i.e. a significant association between entity types?

f) Is each relationship end named and capable of being understood accurately and sensibly?

g) Does each relationship have an entity type at each end?

h) Do all relationship ends in an exclusive arc have the same optionality?

i) Have all 1 : 1 relationships been resolved?

j) Have all m : n relationships been resolved?

k) If mandatory relationships, will there always be an occurrence of the entity type at the other end?

l) Are any relationships redundant?

m) Is the structure consistent with the previous version of the LDS?

3.6.6.4 ***Status***

There shall be five possible variants of this product: Overview, Current Environment, Required System, Required System (expanded), and Required System (normalized).

NOTE The Overview variant may be delivered as an evolving product at the end of Feasibility.

The Current Environment variant shall be a completed product at the end of Stage 1. The other three variants shall be produced as interim products during RS.

3.6.6.5 ***Presentation conventions***

The Logical Data Structure shall be presented in accordance with Figure 22 to Figure 25.

COMMENTARY AND RECOMMENDATIONS ON **3.6.6**.

This product is a diagrammatic representation of the enduring information required by the system in the form of entity types and the relationships between them.

Derivation:

a) *Feasibility Module (for Overview Logical Data Model):*

— *project initiation document;*

b) *Requirements Analysis:*

1) *for Overview Logical Data Structure:*

— *existing system documentation;*

— *Feasibility Report (if it exists);*

— *Project Initiation Document;*

2) *for Current Environment Logical Data Model:*

— *discussion with users (interview notes);*

— *Overview Logical Data Structure;*

c) *Requirements Specification:*

1) *for Required System Logical Data Model:*

— *Current Environment Logical Data Model;*

— *discussions with users;*

— *Elementary Process Descriptions;*

— *Requirements Catalogue;*

— *Selected Business System Option;*

2) *for Required System Logical Data Model (normalized):*

— I/O *Structures;*

— *Required System Logical Data Model;*

3) *for Required System Logical Data Model (expanded):*

— *Required System Logical Data Model (normalized).*

Where used:

— *Logical Data Modelling: FS, RA, RS;*

— *Relational Data Analysis: RS;*

— *Data Flow Modelling: RA, RS;*

— *Feasibility: FS;*

— *Business System Options: RA;*

— *Function Definition: RS;*

— *Entity-event Modelling: RS;*

— *Technical System Options: RS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

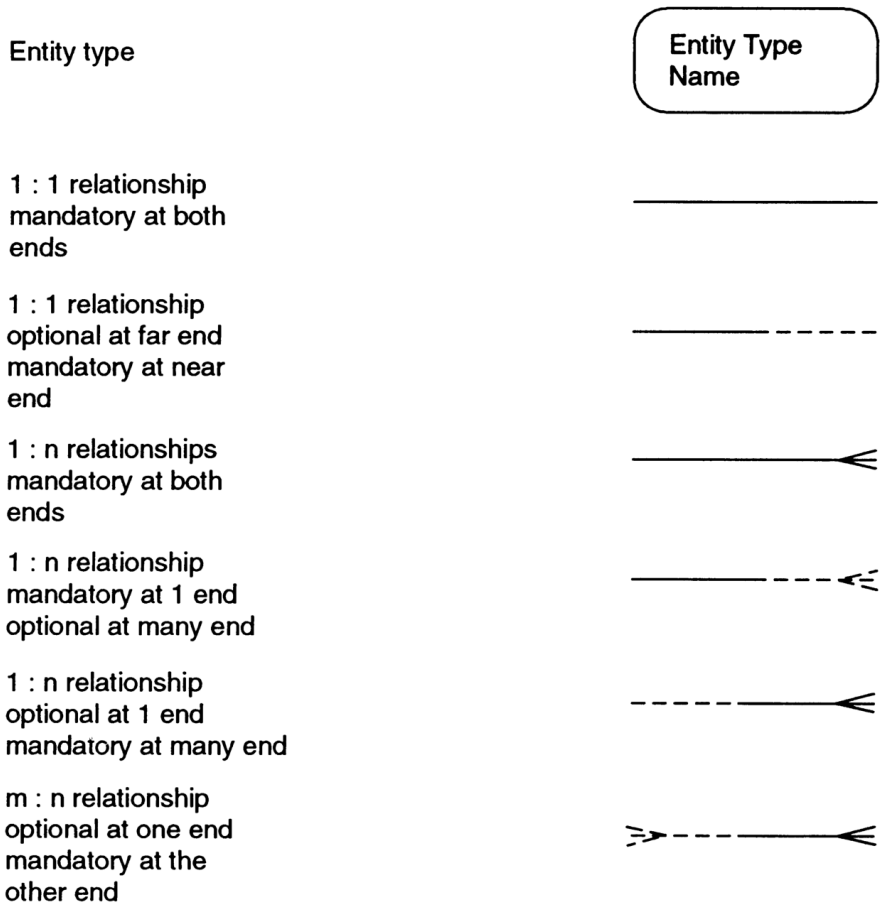
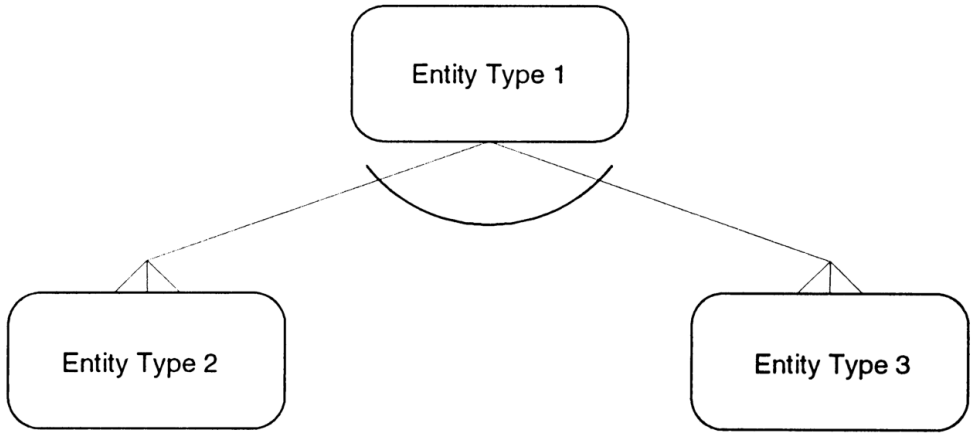
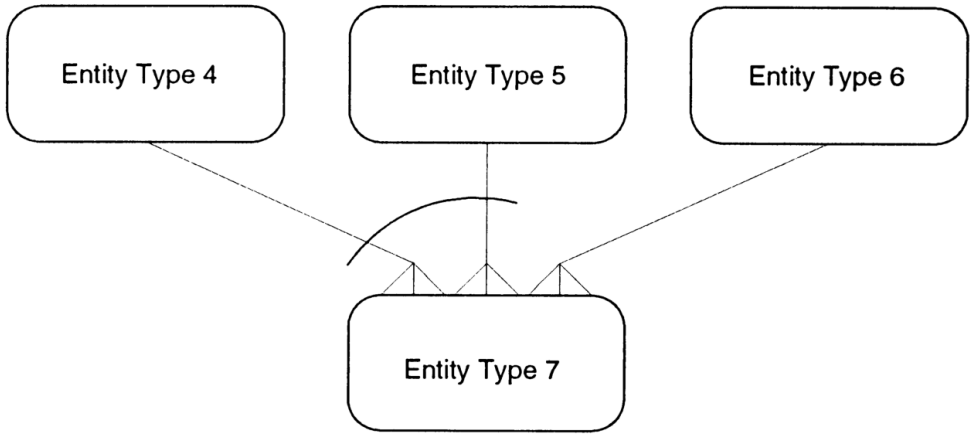


Figure 22 — Presentation conventions for Logical Data Structures

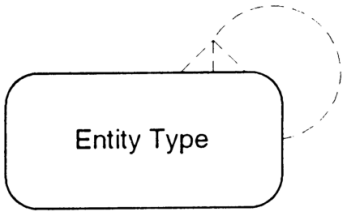


Either one or more occurrences of entity type 2 associated with an occurrence of entity type 1 : Or one or more occurrences of entity type 3 associated with an occurrence of entity type 1.



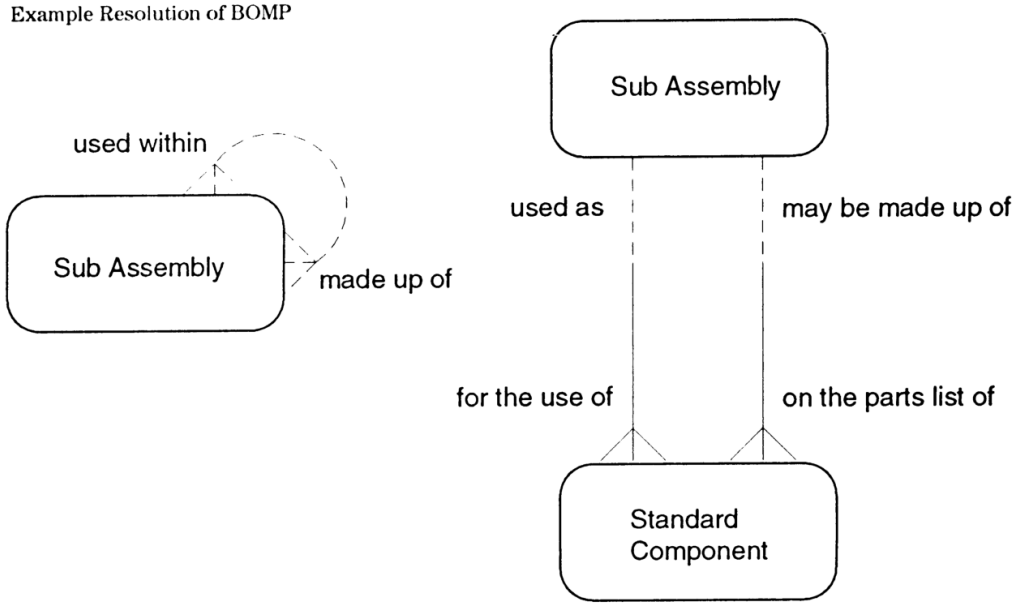
Either one and only one occurrence of entity type 4 associated with an occurrence of entity type 7 : Or one and only one occurrence of entity type 5 associated with an occurrence of entity type 7.

Figure 23 — Presentation conventions for Logical Data Structures: exclusive arcs



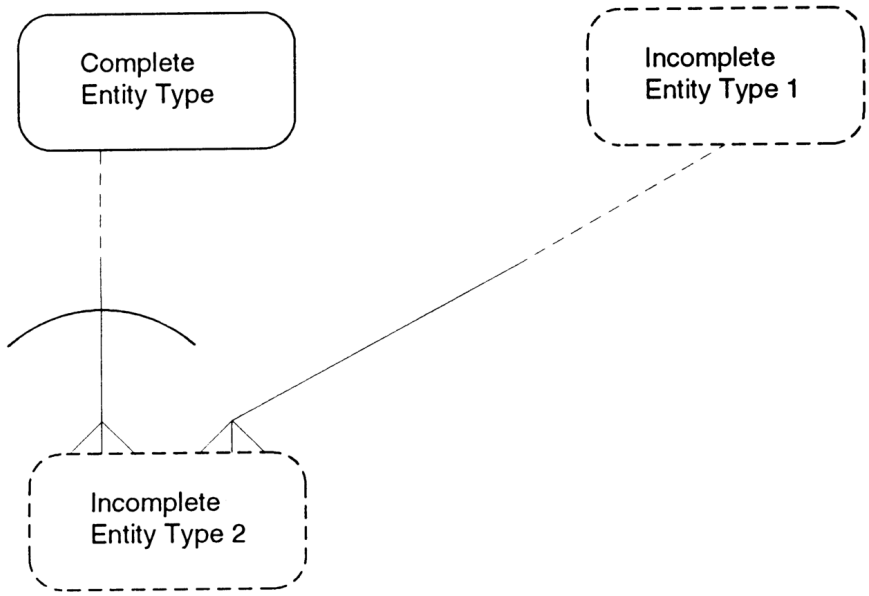
Hierarchy – the diagram indicates that one occurrence of the entity type may be associated with one or more other occurrences of the same entity type.

Example Resolution of BOMP



Network – the diagram on the left indicates that one or more occurrences of the entity type may be associated with one or more other occurrences of the same entity type. This is the situation found in Bill of Materials environments. The notation may be resolved in the form shown in the diagram on the right.

Figure 24 — Presentation conventions for Logical Data Structures: Recursive relationships



NOTE For a large LDS it may be reasonable to present only partial views of the overall structure. This is indicated in the example. The entity type on the left hand side is part of an incomplete exclusive relationship group, with other relationships in the group omitted. The entity type on the right hand side has relationships and other connected entity types which are not shown in the diagram.

Figure 25 — Presentation conventions for Logical Data Structures: partitions

3.6.7 Relationship Descriptions

3.6.7.1 *Purpose*

The set of Relationship Descriptions shall provide detailed documentation for all relationships on the Logical Data Structure.

A full set of entries for each entity type shall comprise a part of the Logical Data Model.

3.6.7.2 *Composition*

The composition shall be as follows:

a) Logical Data Model variant identifier (one of):
- — Current Environment;
- — Required System;

b) Entity type:
- — Entity type name;
- — Entity type identifier;

c) Relationship Description:
- — mandatory/optional indicator;
- — percentage optional;
- — link phrase;

d) description;

e) synonym(s);

f) object entity details:
- — object entity name;
- — object entity identifier;

g) one/many;

h) number of occurrences: minimum, maximum, average;

i) cardinality description;

j) growth per period;

k) additional properties;

l) User Role details, repeating group of:
- — User Role name;
- — access rights;

m) owner;

n) notes.

3.6.7.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Is the variant identifier completed correctly?

2) Is this relationship really a relationship, i.e. a significant association between entity types?

3) Is each relationship end named and capable of being read accurately and sensibly?

4) Does the relationship start and end at an entity type (not in mid-air)?

5) Does each relationship end have the correct degree and optionality?

6) If mandatory relationships, will there always be an occurrence of the entity type at the other end?

7) Are all entries complete?

8) Has retention of historical data been catered for correctly?

b) for the set: Is the set of Relationship Descriptions complete?

3.6.7.4 *Status*

The Current Environment variant shall be delivered complete at the end of Stage 1. The Required System shall be delivered as a completed product at the end of RS.

COMMENTARY AND RECOMMENDATIONS ON **3.6.7**. *Each entry documents the details of a relationship between two entity types on the Logical Data Structure.*

Derivation:

a) *Requirements Analysis (for Current Environment Logical Data Model):*

— *discussion with users (interview notes);*

— *Overview Logical Data Structure;*

b) *Requirements Specification:*

1) *for Required System Logical Data Model:*

— *Current Environment Logical Data Model;*

— *discussion with users;*

— *Elementary Process Descriptions;*

— *Overview Logical Data Structure;*

— *Requirements Catalogue;*

— *Selected Business System Option;*

2) *for Required System Logical Data Model (normalized):*

— I/O *Structures;*

— *Sub-models of RDA Working Paper;*

— *Required System Logical Data Model;*

3) *for Required System Logical Data Model (expanded):*

— *Required System Logical Data Model (normalized).*

Where used:

— *Logical Data Modelling: RA, RS;*

— *Relational Data Analysis: RS;*

— *Entity-event Modelling: RS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD;*

Presentation conventions are not applicable.

3.7 Logical Database Process Design products

COMMENTARY AND RECOMMENDATIONS ON **3.7**. *These products are designed to satisfy two objectives:*

a) *translate the information gathered during Requirements Specification into a detailed logical specification which can then be transformed into a physical design of the system;*

b) *provide a logical definition of the system which will assist maintenance of the implemented system.*

LDPD products encompass both those concerned with updates and enquiries on the relevant system. The term "logical database" is used to invoke the concept of a database, but remaining independent of any specific physical database implementation. The issues of user interfaces to the database, and the batching of data are not addressed by these products; they are dealt with respectively by dialogue design and physical design products.

3.7.1 Enquiry Process Model

3.7.1.1 *Purpose*

Each Enquiry Process Model shall represent the logical process structure of an enquiry component, including the operations. The model shall be constructed by considering the input (Access Path) and output (I/O Structure) and merging them. Any clashes in the input and output data structures shall be identified, but resolution shall be left to physical design.

The completed set of models shall be collected together and passed to the later stages of the method.

3.7.1.2 *Composition*

The composition shall be as follows:

a) Identification: enquiry name;

b) Enquiry Process Structure: a diagrammatic representation using the concepts described in the SSADM Structure Diagram Product Description;

c) Operations List consisting of multiple operations entries as follows:

— operation number/reference (unique within this model only);

— operation description;

d) Conditions for selection and iteration.

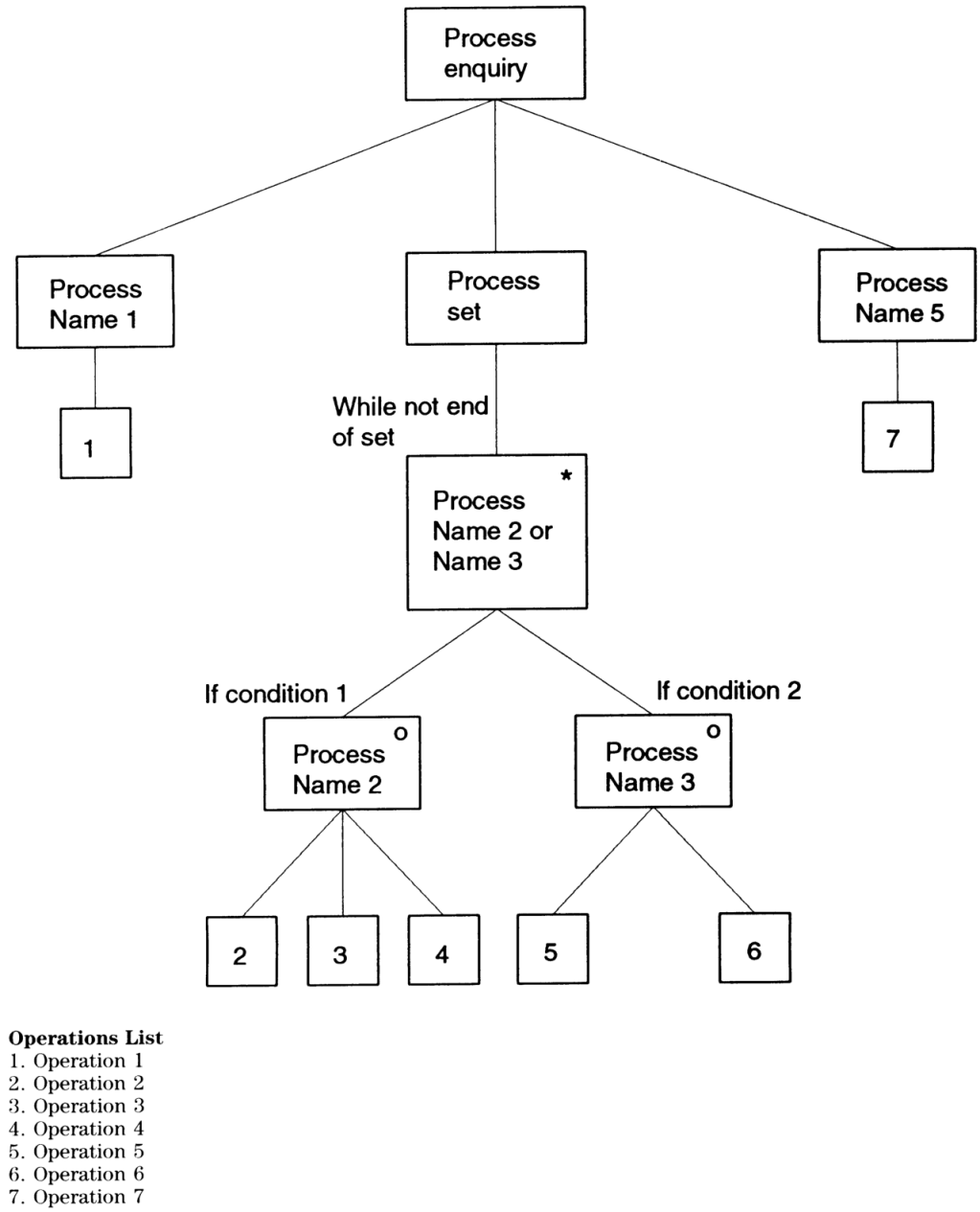


Figure 26 — Presentation conventions for Enquiry Process Models

3.7.1.3 ***Completion criteria***

The following shall be considered:

a) for each:

1) Is the enquiry name completed and valid?

2) Is the notation correctly used within the structure diagram?

3) Are all the operations used at least once?

4) Are all the aspects of the data structures incorporated into the Enquiry Process Structure, or are the clashes highlighted?

b) for the set: Is this a complete set of documentation for all identified enquiry processes?

3.7.1.4 ***Status***

The Enquiry Process Model shall be an end-module completed product, produced for the end of the RS Module.

3.7.1.5 *Presentation conventions*

The Enquiry Process Model shall be presented in accordance with Figure 26 and based on the generic SSADM diagram.

NOTE Ideally the structure and operations list should appear side by side, but physical constraints on paper size may make this impossible.

COMMENTARY AND RECOMMENDATIONS ON **3.7.1**.
This model consists of a structure diagram for an enquiry processing requirement and the associated Operations List. The structure is based on the Enquiry Access Path.

Derivation:

Requirements Specification:

— *Enquiry Access Path;*

— *Entity Life Histories;*

— *Function Definition (enquiry functions only);*

— *Installation Style Guide;*

— I/O *Structures (enquiry functions only);*

— *Required System Logical Data Model.*

Where used:

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

3.7.2 Logical Process Model

3.7.2.1 *Purpose*

The Logical Process Model shall serve to consolidate all of the models of processing (dialogues, enquiry and update) to ensure that a complete set is passed into physical design activities.

3.7.2.2 *Composition*

The composition shall be as follows:

— Dialogues;

— Effect Correspondence Diagrams;

— Enquiry Process Models;

— Function Definitions;

— Update Process Models.

NOTE See Figure B.15 for the product breakdown structure diagram.

3.7.2.3 *Completion criteria*

The following shall be considered.

a) Is this a complete set of documentation for all identified processing requirements? (That is, does each Function Definition have all the necessary instances of the following associated with it:

— Dialogues;

— Enquiry Access Paths;

— Enquiry Process Models;

— Update Process Models;

— I/O Structures)?

b) Are the Function Definitions consistent with the other elements of documentation including the Elementary Process Descriptions?

c) Do the Function Definitions include only those I/O Structures which have not been superseded by the Enquiry or Update Process Models?

d) Are all the Effect Correspondence Diagrams present?

3.7.2.4 *Status*

The Logical Process Model shall be an end-stage product produced complete at the end of Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.7.2**.
This model is used to consolidate all processing details within the Logical Design.

Derivation:

Logical System Specification:

— *Installation Style Guide;*

— *Requirements Specification.*

Where used:

— *Stage 4;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.7.3 Update Process Model

3.7.3.1 *Purpose*

Each Update Process Model shall represent the logical processing structure of an update function, including the operations. A processing model for an event shall be constructed by converting the Effect Correspondence Diagram for that event into a processing structure diagram with ELH operations and some additional operations.

A complete set of models shall be established and passed to later stages in the method.

3.7.3.2 *Composition*

The composition shall be as follows:

a) identifier, event name;

b) Update Process Structure, a diagrammatic representation using concepts described in the SSADM Structure Diagram Product Description;

c) Operations List consisting of multiple operations entries as follows:

— operation number/reference (unique within this model only);

— operation description;

d) conditions for selection and iteration.

3.7.3.3 *Completion criteria*

a) for each:

1) Is the event name completed and correct?

2) Is the notation correctly used within the structure?

3) Are all the operations from the Operations List used at least once?

4) Are all the aspects of the data structures incorporated into the enquiry processing structure, or clashes highlighted?

b) for the set: Is this a complete set of documentation for all identified update processes?

3.7.3.4 *Status*

The Update Process Model shall be an end-stage product produced complete at the end of Stage 5.

3.7.3.5 *Presentation conventions*

The Update Process Model shall be presented in accordance with Figure 27 (based on generic SSADM general diagram).

NOTE Ideally the Structure and Operations List appear side by side, but they will not if the paper is too small.

COMMENTARY AND RECOMMENDATIONS ON **3.7.3**. *This model is a structure diagram for the update (event) processing and the associated Operations List. This is based on the Entity Life Histories, which provide a data-oriented view of the system, and the associated Effect Correspondence Diagrams, which provide an event-oriented or process-oriented view of the system.*

Derivation:

Logical System Specification:

- *Installation Style Guide;*
- *Effect Correspondence Diagrams;*
- *Entity Life Histories;*
- *Function Definitions (update functions only);*
- *Required System Logical Data Model.*

Where used:

- *Logical Database Process Design: LS;*
- *Physical Design: PD.*

3.8 Processing Specification products

COMMENTARY AND RECOMMENDATIONS ON **3.8**. *The products within the category define processing requirements for the overall system at different stages of development. In all cases, the product is a hierarchy of other products. These are combined to form a specific end-stage or end-module product.*

3.8.1 Processing Specification

3.8.1.1 *Purpose*

This product shall be used to specify the processing requirements in order that possible solutions can be identified.

NOTE The components of the product are closely linked, and development of any one may lead to reworking of others.

3.8.1.2 *Composition*

The composition shall be as follows:

- Entity Life Histories;
- Effect Correspondence Diagrams;
- Function Definitions;
- Required System Logical Data Model;
- User Roles/Function Matrix.

NOTE See Figure B.13 for the product breakdown structure diagram.

3.8.1.3 *Completion criteria*

The following shall be considered.

a) Is there an ELH for every entity type on the Required System Logical Data model?

b) Are the ELHs consistent with the Required System Logical Data Model?

c) Do all the data items described in operations for an ELH appear as attributes for that entity type on the Data Catalogue?

d) Are the ELHs consistent with the update functions on the Function Definition?

e) Does every event in the ELHs appear in the appropriate ECD?

f) Are the ELHs consistent with the ECD?

g) Have all the events identified as a result of ELH analysis been allocated to at least one function?

h) Does the entity type for which each ELH exists appear on the Required System Logical Data Model?

i) Has the Enquiry Access Path been defined for each enquiry function?

j) Are the descriptions of average cardinality in the entity descriptions consistent with the access path descriptions?

k) Do the Entity Descriptions allow for state indicators, where such will be required in the ELHs?

l) Are the I/O Structures consistent with the Logical Data Model?

m) Does the event data for an ECD exist as input data on all the functions for the event?

n) Are all input data items represented as event data on the ECD for the corresponding event?

o) Are the Function Definitions consistent with the other products?

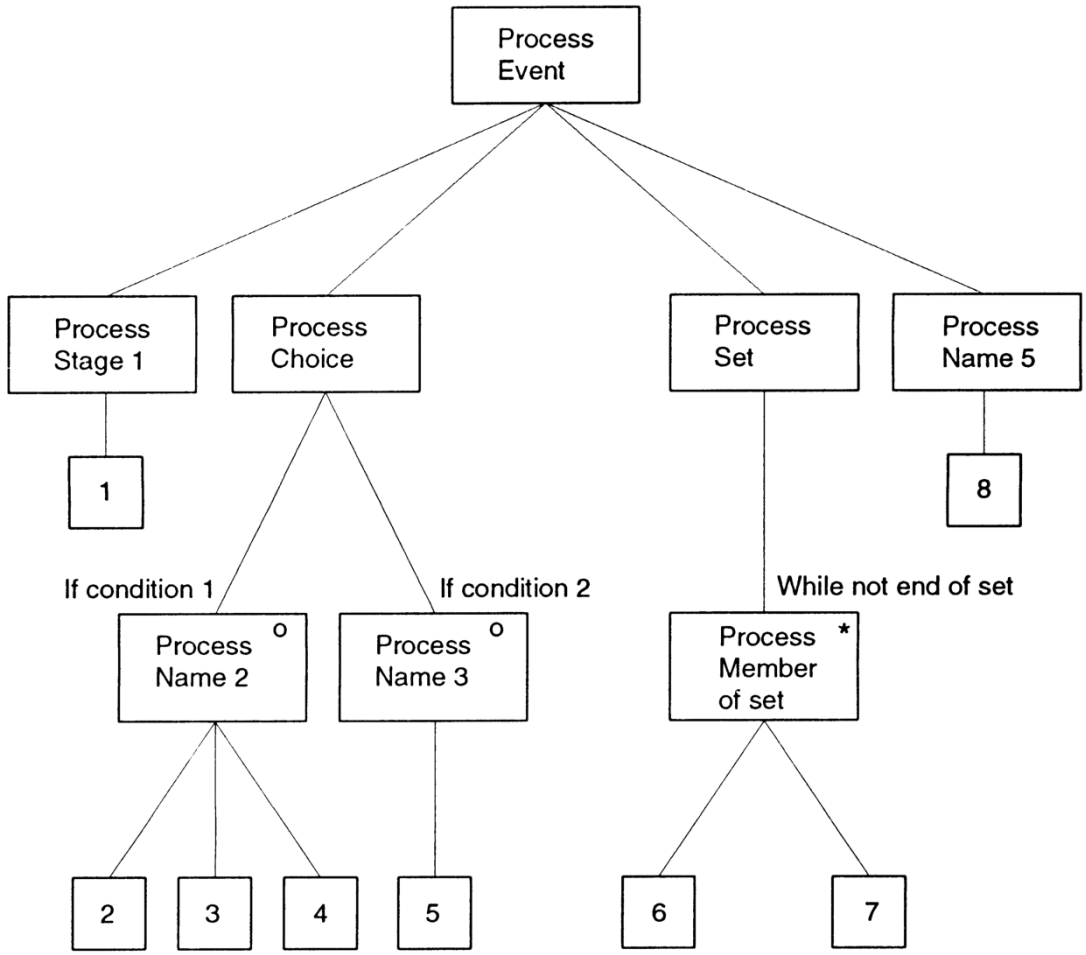


Figure 27 — Presentation conventions for Update Process Models

3.8.1.4 ***Status***

The Processing Specification shall be an end-module product, delivered complete (apart from state indicators on ELHs): RS.

COMMENTARY AND RECOMMENDATIONS ON **3.8.1**.

This product serves to consolidate a number of other products contributing to the overall specification of the system under development.

Derivation:

Requirements Specification:

— *Function Definitions;*

— *Required System Data Flow Model;*

— *Required System Logical Data Model;*

— *Requirements Catalogue;*

— *User Roles/Function Matrix.*

Where used:

— *Entity-event Modelling: RS.*

Presentation conventions are not applicable.

3.8.2 Logical Design

3.8.2.1 *Purpose*

This product shall provide a detailed logical structure of the processing requirements of the proposed system. Processing shall be combined with data details to form the end-stage product from the Logical Design Stage.

3.8.2.2 *Composition*

The composition shall be as follows:

— Command Structures;

— Data Catalogue;

— Logical Process Model;

— Menu Structures;

— Required System Logical Data Model;

— Requirements Catalogue.

3.8.2.3 *Completion criteria*

The following shall be considered.

a) Does the Logical Process Model meet all the functional requirements (in the Requirements Catalogue)?

b) Does the Required System Logical Data Model support the processing functions?

c) Do the Menu and Command Structures fully support the Dialogues?

d) Is the Required System Logical Data Model consistent with the Data Catalogue?

3.8.2.4 *Status*

The Logical Design shall be an end-stage product, delivered complete on completion of Stage 5.

COMMENTARY AND RECOMMENDATIONS ON **3.8.2**.
This consolidates six main products from the Logical System Specification stage. It is then itself combined with other products as part of the overall stage product for Stage 5.

Derivation:

Logical Systems Specification:

— *Installation Style Guide;*

— *Requirements Specification.*

Where used:

— *Physical Design: PD;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS.*

Presentation conventions are not applicable.

3.8.3 Logical System Specification

3.8.3.1 *Purpose*

This product shall provide a detailed set of information relating to the chosen optimum technical solution and the expanded details of the processing requirements, including those aspects concerned with interactive processing (dialogues).

3.8.3.2 *Composition*

The composition shall be as follows:

— Logical Design;

— Selected TSO;

— Technical Environment Description.

NOTE See Figure B.21 for the product breakdown structure diagram.

3.8.3.3 *Completion Criteria*

The following shall be considered.

a) Does the Logical System Specification conform to installation standards?

b) Is the project within the scope of the project terms of reference?

c) Do the users agree that their requirements have been addressed?

d) Is the Logical Design for the processing consistent with the Technical Environment Description?

e) Has only one Technical System Description, possibly a hybrid of several proposed options, been chosen as the basis to proceed?

f) Does this option satisfy the minimum requirements in the Requirements Catalogue?

g) Is the Technical Environment Description consistent with the Selected Technical System Option?

3.8.3.4 *Status*

The Logical System Specification shall be an end-module product, delivered complete on completion of the Logical System Specification module.

COMMENTARY AND RECOMMENDATIONS ON **3.8.3**.
This consolidates products from Stage 4 with products from Stage 5. It is the end-module product for the Logical System Specification module.

Derivation:

Logical System Specification:

— *Project Initiation Document;*

— *Requirements Specification;*

— *Selected BSO.*

Where used:

— *Technical System Options: RS;*

— *Dialogue Design: LS;*

— *Logical Database Process Design: LS;*

— *Physical Design: PD.*

3.8.4 Physical System Specification

3.8.4.1 *Purpose*

This product shall define an operationally viable system which meets the system requirements and is consistent with the Logical System Specification.

3.8.4.2 *Composition*

The composition shall be as follows:

— Application Development Standards;

— Physical Design;

— Physical Environment Specification.

NOTE See Figure B.22 for the product breakdown structure diagram.

3.8.4.3 *Completion criteria*

The following shall be considered.

a) Does the design conform to installation standards?

b) Are the project budget and resources within the scope set in the Project Initiation Document?

c) Do the users agree that this solution addresses their requirements?

d) Can the project continue to the construction and testing phase?

e) Is the design viable and implementable?

f) Does the Physical Environment Specification agree with the Technical Environment Description on which it is based?

g) Does the design take account of hardware/software limitations for development and implementation? Is there consistency between the Physical Design and the Physical Environment Description?

h) Are appropriate migration paths identified if the development and implementation paths differ?

i) Are maintenance issues addressed satisfactorily?

j) Does the design achieve a satisfactory balance between performance and functionality?

k) Do all aspects of the user interface relate consistently to existing systems, and have future modifications been identified?

l) Do the detailed processes of the DBMS:

— provide all required functionality?

— meet system performance requirements?

— make acceptable demands on human or material resources?

— have acceptable hardware requirements?

— conform to operations/management rules?

m) Have acceptance criteria been defined?

n) Has an impact analysis of version changes in the Physical Environment Specification been identified?

3.8.4.4 *Status*

The Physical System Specification shall be an end-module product, delivered complete on completion of Physical Design module.

COMMENTARY AND RECOMMENDATIONS ON **3.8.4**.

This consolidates products from Stage 6. It is a key end-module product for the Physical Design module.

Derivation:

— *Logical System Specification.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.9 Relational Data Analysis products

COMMENTARY AND RECOMMENDATIONS ON **3.9**.

Relational Data Analysis is a technique for complementing and checking the Logical Data Model.

The technique is used to derive data structures which have a minimum of redundant data combined with maximum flexibility in terms of modifiability and extensions to the structure.

The purpose of the technique is to:

a) *capture users' detailed knowledge of the meaning and significance of the data;*

b) *validate the Logical Data Model;*

c) *ensure the data is logically maintainable and extendible;*

d) *produce data in optimum groups to provide a basis for sharing data across many applications.*

3.9.1 Relational Data Analysis (RDA) Working Paper

3.9.1.1 *Purpose*

The RDA Working Paper shall help to validate the Logical Data Model, which will have been developed top-down, against relations developed bottom-up.

3.9.1.2 *Composition*

The composition shall be as follows:

a) heading:

— Source Name;

b) entries consisting of:

1) Un-normalized form;

2) attribute;

3) level:

— First Normal Form (1NF);

— Second Normal Form (2NF);

— Third Normal Form (3NF);

c) Result:

— relation;

— attributes.

3.9.1.3 *Completion criteria*

The following shall be considered.

a) Are the normalization rules applied correctly at each stage?

b) Are any keys redundant?

c) Is each entry complete?

3.9.1.4 *Status*

The RDA Working Paper shall be an interim product produced during the RS Module.

COMMENTARY AND RECOMMENDATIONS ON **3.9.1**.
This product is used to document the progress through relational data analysis, taking relations which are Un-normalized through to Third Normal Form.

Derivation:

Requirements Specification:

— I/O *Structures;*

— *Required System Logical Data Model.*

Where used:

— *Relational Data Analysis: RS;*

— *Logical Data Modelling: RS.*

Presentation conventions are not applicable.

3.10 Requirements Definition products

COMMENTARY AND RECOMMENDATIONS ON **3.10**.
Requirements definition involves the identification and articulation of organizational and systems objectives, and provides the focus for identifying users and their requirements. The aim is to define quantifiable and measurable objectives for each requirement.

Requirements definition products are developed to meet the following objectives:

a) *identify requirements for the proposed system, meeting the needs of users and of the organization as a whole;*

b) *describe the requirements in quantifiable terms;*

c) *provide a basis for decisions concerning the development of the system, in the light of constraints within which development occurs;*

d) *develop the Requirements Specification, providing a complete and accurate specification of requirements as a platform for system design and implementation;*

e) *ensure that the analysis of the system concentrates on user requirements.*

Specific requirements can be classified as:

1) *Functional Requirements;*

2) *Non-Functional Requirements, including Service Level Agreements.*

3.10.1 Requirements Catalogue

3.10.1.1 *Purpose*

The Requirements Catalogue shall be produced to encompass, in an identifiable and usable format, details of all requirements identified at a particular time within the project.

Each entry in the Requirements Catalogue shall provide the description of a requirement for the proposed new system. Requirements shall be recorded in the Requirements Catalogue and updated throughout analysis and design activities to ensure a full, quantified expression of requirements. Where requirements are further expanded using other techniques, such as function definition or logical data modelling, which model requirements with greater rigour, a reference shall be given to specification products which resolve the requirement.

3.10.1.2 *Composition*

The composition shall be as follows:

a) each Requirements Catalogue Entry consisting of:

— requirement identification details;

— source of requirement;

— priority of requirement;

— owner of requirement;

— requirement identifier;

b) Functional Requirement description;

c) non-functional requirement(s) details, repeating group consisting of:

— description;

— target value;

— acceptable range;

— comments;

d) benefits;

e) comments/suggested solution;

f) related documents;

g) related requirements;

h) resolution.

3.10.1.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Is the description of each functional requirement as complete as possible?

2) Are non-functional requirements relating to functional requirements as fully documented as possible?

3) Have source, owner, priority and benefit been identified?

4) If this requirement was previously defined, is the new version consistent with the old? If not, why not?

b) for the set:

1) Does the Requirements Catalogue describe all identified requirements of the new system (with any appropriate cross-references to other SSADM products)?

2) Are requirements consistent with the project objectives?

3) Have all necessary previous requirements been carried forward?

3.10.1.4 *Status*

The Requirements Catalogue shall be an evolving product. It shall be continually updated through Stages and Modules, and shall only be regarded as completed prior to the Physical Design Module.

NOTE Certain levels of detail will only become apparent at later stages of the development process, but, for any specific context, contractual parties may wish to indicate the levels of detail which have to have been completed at specified points.

COMMENTARY AND RECOMMENDATIONS ON **3.10.1**.

The Requirements Catalogue is the central repository for information covering all identified requirements, both functional and non-functional. Each entry is textual and describes a required facility or feature of the proposed system.

Derivation:

a) *Feasibility:*

— *discussion with users;*

— *existing system documentation;*

— *project initiation document;*

— *Requirements Catalogue;*

b) *Requirements Analysis:*

— *Context Diagram;*

— *Current Environment Logical Data Model;*

— *Current Physical Data Flow Model;*

— *Current Physical Data Flow Diagram, Level 1;*

— *Data Catalogue;*

— *discussion with users;*

— *existing system documentation;*

— *Overview Logical Data Structure;*

— *Project Initiation Document;*

— *Requirements Catalogue (if produced in Feasibility);*

— *User Catalogue;*

c) *Requirements Specification:*

— *Current Environment Logical Data Model;*

— *Data Catalogue;*

— *Function Definitions;*

— *Input/Output Structures;*

— *Installation Style Guide;*

— *Logical Data Flow Model;*

— *Prototyping Scope;*

— *Requirements Catalogue;*

— *Required System Data Flow Model;*

— *Selected Business System Option;*

— *User Catalogue;*

— *User Roles;*

— *User Role/Function Matrix;*

d) *Logical System Specification:*

— *Command Structures;*

— *Data Catalogue;*

— *Function Definitions;*

— I/O *Structures;*

— *Installation Style Guide;*

— *Menu Structures;*

— *Requirements Catalogue;*

— *User Role/Function Matrix;*

e) *Physical Design.*

In Physical Design the Requirements Catalogue will be annotated to document how particular aspects are implemented.

Where used:

Essentially the Requirements Catalogue is used throughout the method; either as an input for possible updating (e.g. throughout Stage 1 Investigation of Current Environment), or as the key reference point for development (e.g. Stage 2, Business Systems Options).

Presentation conventions are not applicable.

3.10.2 Take-on Requirements Description

3.10.2.1 *Purpose*

The Take-on Requirements Description shall be used to define the take-on needs for the project under way. Also it shall provide information to decide on the strategy for take-on.

3.10.2.2 *Composition*

The Take-on Requirements Description shall be a textual document covering the following topics:

— estimates covering times, resources, costs;

— data model: source and target;

— file structures: source and target;

— input and output descriptions: target;

— sizing information: source and target;

— sourcing information: where, when and how;

— validation tests and procedures;

— outline functions to perform take-on;

— possible conversion method;

— outline resourcing schedule;

— outline Plan for the take-on exercise.

3.10.2.3 *Completion criteria*

The following shall be considered:

a) Is the data to be taken-on:

— clearly identified as a source (within the existing system) in logical and physical terms?

— targeted to the appropriate home, in logical and physical terms?

b) Is volumetric information clearly stated?

c) Is validation information clearly stated?

d) Is the hardware and software configuration for both take-on activity and target system clearly indicated (stated)?

e) Are Take-on Plans compatible with implementation/application plans?

f) Are estimates within tolerable and agreed levels of accuracy?

g) Is there an identified organization to implement take-on activity?

3.10.2.4 *Status*

The Take-on Requirements Description shall be an end-stage product, delivered complete at the end of Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.10.2**.

This product is compiled during technical system option activities and details the data conversion requirements which have to be implemented before a fully working system can be available. It embodies recognition of the essential activities and procedures required to establish the initial data for the target system.

Derivation:

Logical System Specification:

— *existing system documentation;*

— *Outline Project Plan (not mentioned in the standard);*

— *Requirements Specification;*

— *Selected Business System Option;*

— *Technical Environment Description.*

Where used:

— *Technical System Option: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.10.3 Training Requirements Description

3.10.3.1 *Purpose*

The Training Requirements Description shall be used to describe the training needs implied by the Selected Technical Environment Description. It shall support development of a training strategy.

3.10.3.2 *Composition*

The composition shall be as follows:

a) training subjects:

— for users;

— for technicians (operators);

— for implementors;

— for management;

b) training topics:

— for particular skill needs and personnel profiles;

— for particular aspects of hardware and software;

— configuration;

— for particular functional/procedural aspects;

c) resourcing for a) and b);

d) costs of providing a) and b);

e) timing of training for a) and b).

3.10.3.3 *Completion criteria*

The following shall be considered.

a) Is each user role covered by training needs? Or explicitly excluded?

b) Are all the required skills covered by a training profile?

c) Are hardware items covered? Or explicitly excluded?

d) Are software items covered? Or explicitly excluded?

e) Are procedural aspects of the system covered?

f) Are times and costs clearly identified?

g) Are volumes (of all types) clearly identified?

h) Are the times related to the milestones identified in the Project Development Plan?

3.10.3.4 *Status*

The Training Requirements Description shall be an end-stage product delivered completed at the end of Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.10.3**.
This product is compiled during Technical System Option activities, and documents the nature, levels and degree of training required for the staff who will be using/working on the new system, so that they will be fully effective at the appropriate time in the development of the system.

Derivation:

Logical System Specification:

— *Requirements Specification;*

— *Selected Technical System Option;*

— *Technical Environment Description.*

Where used:

— *Technical Systems Options: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.10.4 User Catalogue

3.10.4.1 *Purpose*

The User Catalogue shall provide a list of all on-line users of the required system and the tasks associated with them. This shall be used as input to the formation of user roles.

3.10.4.2 *Composition*

Each entry shall consist of:

— job title;

— multiple job activity descriptions.

3.10.4.3 *Completion criteria*

The following shall be considered.

a) Are all user tasks identified for each job title?

b) Have all the necessary job titles been investigated?

3.10.4.4 *Status*

The User Catalogue shall be an evolving product, if initially produced during Feasibility, or an end-stage product, completed at the end of Stage 1.

COMMENTARY AND RECOMMENDATIONS ON **3.10.4**.
This product provides a description of the on-line users of the proposed system. It includes details of job titles and the tasks undertaken by each of the identified users.

Derivation:

a) *Feasibility:*

— *Context Diagram;*

— *Current Physical DFD Level 1;*

— *discussions with users;*

— *Overview Logical Data Model;*

— *Project Initiation Document;*

— *Requirements Catalogue;*

b) *Requirements Analysis:*

— *Context Diagram;*

— *Current Environment Logical Data Model;*

— *Current Physical Data Flow Model;*

— *discussion with users;*

— *Project Initiation Document;*

— *Requirements Catalogue;*

— *User Catalogue (if produced in Feasibility).*

Where used:

— *Stage 1.*

Presentation conventions are not applicable.

3.10.5 User Manual Requirements Description

3.10.5.1 *Purpose*

The User Manual Requirements Descriptions shall be produced to consolidate user issues relating to the system and its smooth running. The descriptions shall provide outline information for inclusion in the final user manual.

3.10.5.2 *Composition*

The composition shall be as follows:

a) description of activity relating to:

— normal running and operation;

— interfaces to manual procedures;

— abnormal condition recognition and control/correction procedures;

— help facilities;

b) description of general user facilities;

c) operations timetables;

d) computer requirements.

3.10.5.3 *Completion criteria*

The following shall be considered:

a) Is each user role covered in terms of:

— normal running and operation;

— interfaces to manual procedures;

— abnormal condition recognition and control/correction procedures;

— help facilities?

b) Do the I/O Descriptions match with descriptions of system activity and user activity?

c) Similarly, do the related functions match their user activity descriptions?

d) Are abnormal conditions adequately covered?

e) Are exception procedures provided for the user?

f) Is there a clear fallback position, e.g. to a help desk?

g) Has responsibility to implement a user manual been agreed?

3.10.5.4 *Status*

The User Manual Requirements shall be delivered complete as an end-stage product in Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.10.5**. *This is compiled during technical system option activities, and documents the basic information, derived from both SSADM and non-SSADM sources, which will be required by user for systems operation. It relates to the system and its smooth running from the user point of view.*

Derivation:

Logical System Specification:

— *Non-functional description (i.e. non-SSADM);*

— *Requirements Specification;*

— *Technical Environment Description.*

Where used:

— *Technical Systems Options: LS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.10.6 Current Services Description

3.10.6.1 *Purpose*

The Current Services Description shall detail the investigations identifying the constraints of, and facilities offered by the current system.

3.10.6.2 *Composition*

The composition shall be as follows.

— Data Catalogue;

— Current Environment Description;

— Context Diagram;

— Logical Data Flow Model;

— Logical Data Store/Entity cross-reference.

3.10.6.3 *Completion criteria*

The following shall be considered:

a) Is the scope of the system described consistent with the constraints defined in the Project Initiation Document?

b) Are synonyms used for data item names, and are they identified as such?

c) Have all appropriate people been consulted?

d) Are the Context Diagram and the Logical Data Flow Model mutually consistent?

e) Is the Current Environment Logical Data Model consistent with the Logical Data Flow model (based on the Logical Data Store/Entity cross-reference)?

f) Is the Current Environment Logical Data Model consistent with the Data Catalogue?

3.10.6.4 *Status*

The Current Services Description shall be produced as an end-stage product in Stage 1.

COMMENTARY AND RECOMMENDATIONS ON **3.10.6**. *The investigations required to produce this report are carried out within specified user departments, using techniques such as interviewing and questionnaire completion.*

Derivation:

— *Requirements Analysis.*

Where used:

— *Stage 1, RS.*

Presentation conventions are not applicable.

3.10.7 Requirements Specification

3.10.7.1 *Purpose*

The Requirements Specification shall serve to specify user requirements, including constraints and any future enhancements, in order that possible solutions can be identified and developed.

3.10.7.2 *Composition*

The composition shall be as follows:

— Data Catalogue;

— Processing Specification;

— Requirements Catalogue.

NOTE See Figure B.18.

3.10.7.3 *Completion criteria*

The following shall be considered:

a) Does the Requirements Specification conform to relevant installation standards?

b) Is the project within the scope set? Can it be continued on this basis?

c) Do the users agree that their requirements have been addressed?

d) Is the Requirements Specification compatible with the installation procurement procedures?

e) Does the Requirements Specification form the basis for a range of implementation approaches?

f) Is it a complete and accurate statement of requirements, constraints, and possible future enhancements of the system?

g) Are all requirements mutually consistent? If not, have priorities been identified?

h) Is the Data Catalogue consistent with the Required System Logical Data Model?

i) Does the description of access rights in the entity and attribute descriptions support the accesses defined in the access path descriptions?

j) Are the descriptions of entity volumes in the entity and attribute descriptions consistent with the access path descriptions?

k) Is the Requirements Catalogue correctly cross-referenced to the Function Definitions?

l) Do the functions fully support the duties of each user role?

m) Do all the enquiry functions in the Requirements Catalogue have the appropriate function documentation?

n) Are the functional requirements in the Requirements Catalogue fully supported in the Processing Specification?

3.10.7.4 *Status*

The Requirements Specification shall be a product delivered complete at the end of RS.

COMMENTARY AND RECOMMENDATIONS ON **3.10.7**.
This product forms the overall module product from the Requirements Specification Module. It incorporates all the products and groups of products developed in that Module.

Derivation:

Requirements Specification:

- *— Analysis of Requirements;*
- *— Installation Style Guide;*
- *— Prototyping Scope.*

Where used:

- *— RD: RA, RS, LS;*
- *— Feasibility: FS;*
- *— Data Flow Modelling: RA, RS;*
- *— Dialogue Design: RS;*
- *— Business System Options: RA;*
- *— Function Definition: RS;*
- *— Logical Data Modelling: RA, RS;*
- *— Specification Prototyping: RS;*
- *— Technical System Options: RS;*
- *— Physical Design: PD.*

Presentation conventions are not applicable.

3.11 Specification Prototyping products

COMMENTARY AND RECOMMENDATIONS ON **3.11**.
Specification Prototyping is used to demonstrate aspects of the planned system to users. It is not used to provide a final version of all or part of the system, but to further system specification by allowing users to see an animated working component. The prototypes produced have the overall objectives of facilitating the highlighting of errors and discrepancies, in addition to increasing the fine detail of the system requirements. Specification prototypes should enhance user understanding of the developing system, permitting better and clearer identification of system functioning and integrity constraints.

Specification Prototyping draws on dialogue design products in the derivation of prototypes. The results of prototype reviews may impact upon a wide range of other SSADM products. Contractual parties will need to ensure that the necessary steps are taken to progress the results of prototyping activities to the relevant aspects of the system, and that the prototypes are re-delivered for further assessment if necessary.

The objectives behind Specification Prototyping may be detailed in a prototyping scope document. This may detail the resource allocations and any constraints (equipment and timescales) which apply. Procedures for reporting back to the project board may also be included.

3.11.1 Prototype Demonstration Objective Document

3.11.1.1 *Purpose*

The Prototyping Demonstration Objective Document shall be used to assist in the prototype demonstration with the user. It shall ensure that the prototyping demonstration with the user is organized, and that the maximum benefit is gained from the time spent with the user.

3.11.1.2 *Composition*

The composition shall be as follows:

a) heading consisting of:

- — document number;
- — Prototype Pathway number;
- — function name;
- — User Role;

b) agenda: the objectives to be achieved in the demonstration of the prototype model;

NOTE This will be in a similar form to an agenda for a meeting.

c) component details, repeating group consisting of:

— component number as shown on the Prototype Pathway;

— component queries; a summary of points that require clarification during the demonstration for this component of the Prototype Pathway.

3.11.1.3 *Completion criteria*

The following shall be considered.

a) Is the heading section correctly completed with:

— unique document number;

— Prototype Pathway number;

— function name;

— User Role?

b) Does the agenda cover all queries which the analyst needs to discuss with the user?

c) Are there any components for which there are no queries? If so, is there really nothing that requires checking/clarification for that component?

d) Identification of relevant user(s) for prototype demonstration.

3.11.1.4 *Status*

The Prototype Demonstration Objective Document shall be an interim product, produced as part of SP activities; RS Module. Specification Prototyping products are for the most part interim, but shall be delivered completed to user representatives in order that the prototype results can be adequately reviewed.

COMMENTARY AND RECOMMENDATIONS ON **3.11.1**.

One of these documents is completed prior to any prototype demonstration for each Prototype Pathway. Assumptions and queries for each menu, screen and/or report are listed under their respective component number. This document lists the points of discussion to be addressed between the user and the analyst during the prototype demonstration.

Derivation:

Requirements Specification:

— *Prototype Pathway.*

Where used:

— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.11.2 Prototype Pathway

3.11.2.1 *Purpose*

The Prototype Pathway shall be used to structure the prototype that is to be implemented on the support tool. One Prototype Pathway shall be produced for each dialogue or report to be prototyped, for each user role. It shall be a serial representation, intended to show at a high level how all the components (menus, screen and reports) of the prototype are linked together.

3.11.2.2 *Composition*

The heading shall include the following:

— function name;

— User Role;

— Prototype Pathway number.

The diagram shall be composed of boxes and vertical linking lines. Each box shall represent one menu, screen or report component.

3.11.2.3 *Completion criteria*

The following shall be considered.

a) Is the heading correctly completed?

b) Does the Prototype Pathway represent one dialogue or report for one user role?

c) Do the components form a logical sequence; or are any components missing or in the wrong place on the diagram?

d) Does the implemented version reflect the diagram?

e) Once implemented on the support tool, does the Prototype Pathway meet the requirements of the user(s)? If not, this shall be reflected in the Prototype Result Log.

f) Have users belonging to the user role for which the Prototype Pathway was created been identified for inclusion in reviews?

3.11.2.4 *Status*

The Prototype Pathway shall be an interim product, produced as part of SP activities; RS Module. Specification Prototyping products are for the most part interim, but shall be delivered completed to user representatives in order that the prototype results can be adequately reviewed.

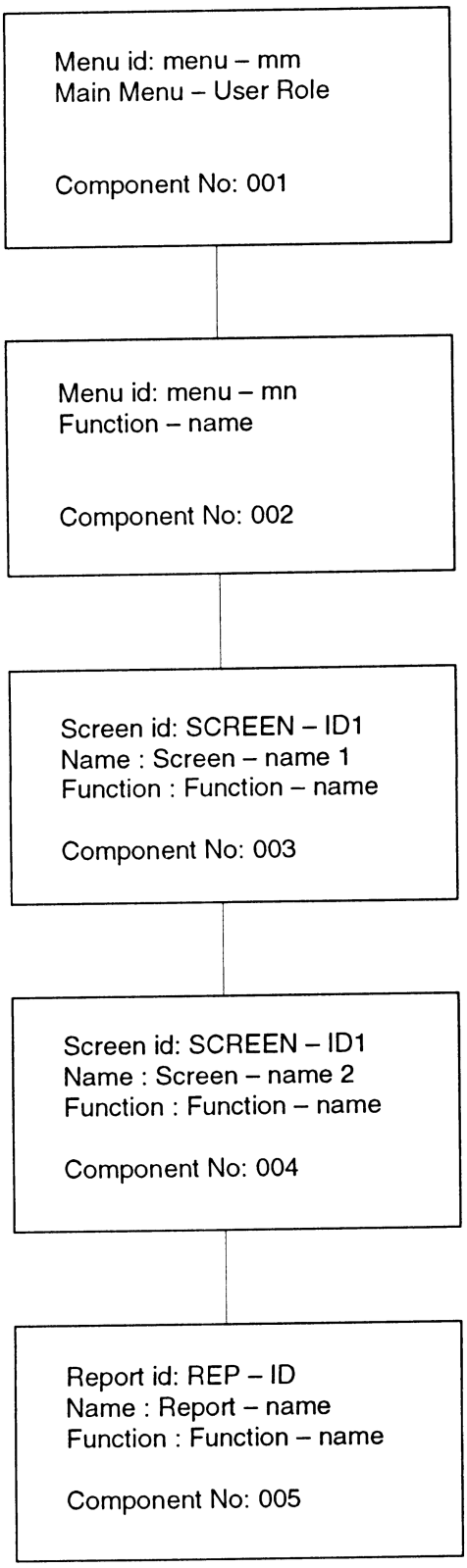


Figure 28 — Presentation conventions for the Prototype Pathway

3.11.2.5 *Presentation conventions*

The Prototype Pathway shall be presented in accordance with Figure 28.

COMMENTARY AND RECOMMENDATIONS ON **3.11.2**.

Once screen and report components have been identified within the prototyping activities, they can be combined with the existing menus to form Prototype Pathways. The pathways are a script for the prototyping session and hence use a limited and simple serial diagrammatic representation to convey the structure.

Derivation:

Requirements Specification:

— I/O *Structures;*

— *Prototyping Scope;*

— *User Role/Function Matrix.*

Where used:

— *Specification Prototyping: RS.*

3.11.3 Prototype Result Log

3.11.3.1 *Purpose*

The Prototype Result Log shall provide a record of the prototyping session. A separate log shall be maintained for each prototyping session for a particular Prototype Pathway. During the session, each request made by the user shall be documented on the log concerned. After the session, the log shall be updated to show the type of change required to satisfy each result.

3.11.3.2 *Composition*

The composition shall be as follows:

a) heading consisting of:

— Prototype Result Log number;

— Prototype Pathway number;

— function name;

— User Role;

b) repeating group consisting of:

— Prototype Pathway component number;

— result number;

— result description;

— change grade; the change grade for each result shall be added after the prototype demonstration.

3.11.3.3 *Completion criteria*

The following shall be considered.

a) Is the heading completed correctly?

b) Have all requests made by the user been documented as results?

c) Do the result descriptions make sense, i.e. will the analyst be able to understand what was meant after the demonstration?

d) Has a change grade been allocated to each result?

e) Are change grades correct?

Each shall be one of the following:

— N: no change required;

— C: cosmetic, affecting presentation only;

— D: affecting the dialogue or report only;

— P: affecting only the Prototype Pathway being evaluated;

— S: affecting standards;

— A: indicating that there may be something wrong with the analysis to date; may need to consult project management;

— G: global, affecting something outside the application, or affecting working practices.

3.11.3.4 *Status*

The Prototype Result Log shall be an interim product, produced as part of SP activities, RS module. Specification Prototyping products are for the most part interim, but shall be delivered completed to user representatives in order that the prototype results can be adequately reviewed.

COMMENTARY AND RECOMMENDATIONS ON **3.11.3**.
This log is used to record the results of the prototype demonstration. This document is used in a similar capacity to minutes of a meeting. Each request made by the user is documented on the log, with a change grade, and the log is updated later to show what changes are required.

Derivation:

Requirements Specification:

— *Installation Style Guide;*

— *Prototype Demonstration Objective Document;*

— *Prototype Pathways.*

Where used:

— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.11.4 Prototyping Report

3.11.4.1 *Purpose*

The Prototyping Report shall be used to summarize the result of the prototyping activity, with particular attention being paid to the original objectives laid down in the Prototyping Scope.

3.11.4.2 *Composition*

The Prototyping Report shall be a textual document with no predefined structure.

3.11.4.3 *Completion criteria*

The following shall be considered.

a) Does the report cover all the objectives set out in the Prototyping Scope?

b) Have all major changes to SSADM documentation arising from the demonstrations been summarized and recorded?

c) Are any recommendations for future prototyping work feasible, sensible and soundly based?

3.11.4.4 *Status*

The Prototyping Report shall be a completed end-module product: RS.

COMMENTARY AND RECOMMENDATIONS ON **3.11.4**.
This report is prepared to establish whether or not the objectives for the prototyping exercise were achieved, or alternatively, the reasons why they were not. It includes estimates of the value of the work done and where necessary suggests whether more work would (or would not) be beneficial.

Derivation:

Requirements Specification:

— *Prototype Result Log(s);*

— *Prototyping Scope.*

Where used:

— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.12 Technical System Options products

COMMENTARY AND RECOMMENDATIONS ON **3.12**.
Technical System Options (TSO) products provide detailed implementation features for the selected Business System Option (BSO). TSO products are designed to address the following systems aspects:

a) *specification of the technical environment, e.g. provision and configuration of hardware devices, software, operating regime;*

b) *confirmation of the functions to be covered by the system, and the manner in which they are to be achieved;*

c) *impact of the system and associated changes on the organizational context and working practices;*

d) *impact on the organization during the remainder of the development project.*

If no feasibility study has been completed, and if the selected BSO has not addressed all the issues in full, TSO products may be used to draw attention to strategy and policy issues. These may include the following:

1) *generic technical environment (hardware, communications, etc.);*

2) *development strategy; development phasing, procurement, implementation methods and procedures;*

3) *organizational impact: new business areas and opportunities, staffing and resourcing, implementation targets.*

TSO products are produced to focus user concerns on the detailed nature of the project, and the interaction between the project and the organizational environment.

TSO products can be used in the following manner:

i) *to identify the major constraints;*

ii) *to create outlines of possible solutions;*

iii) *to expand outlines with sufficient detail for users to comprehend the alternatives involved;*

iv) *to present the options to users, explaining the alternatives sufficiently that a decision can be made;*

v) *to record the details of decisions reached;*

vi) *to adjust the selected option in order that it reflects the decisions reached.*

The selected BSO description (including the rationale behind its selection) enhanced by the Requirements Specification, provides the basis for TSO products.

3.12.1 Capacity Planning Input

3.12.1.1 *Purpose*

The Capacity Planning Input shall provide capacity planners with information about the required system. This information shall then be evaluated and returned to SSADM activities in the most appropriate manner. It shall be used explicitly during Stage 4, Technical System Options.

3.12.1.2 *Composition*

The Capacity Planning Input shall be a description of the application, including relevant volumetric information.

3.12.1.3 *Completion criteria*

The following shall be considered.

a) Is the volumetric information for the Required System Logical Data Model components of adequate quality, including access path details?

b) Is the volumetric information for the function/processing components of adequate quality?

c) Is the volumetric information for the appropriate entries in the Requirements Catalogue of adequate quality?

d) Are the service level requirements included at the appropriate level of detail?

e) Have required tools and skills to undertake capacity planning activities been identified?

f) Is information about existing installation configurations and capacity available?

3.12.1.4 *Status*

The Capacity Planning Input shall be delivered during Stage 4; then re-input in Stage 4 having been evaluated by capacity planners.

COMMENTARY AND RECOMMENDATIONS ON **3.12.1**.

This document is used to summarize the processing and data information, passed outside of SSADM activities, to capacity planning techniques, explicitly during development of Technical System Options.

Derivation:

Logical System Specification:

— *Requirements Specification;*

— *(Selected) Technical System Option.*

Where used:

— *Technical System Options: RS.*

Presentation conventions are not applicable.

3.12.2 System Description

3.12.2.1 *Purpose*

The System Description shall serve to illustrate how the Requirements Specification is met within the specified Technical Environment Description.

The System Description shall emphasize the significance of an option by including a list of functions/facilities not being provided to support the full user requirements.

NOTE This may be necessary to trade off facilities against development times and costs.

3.12.2.2 *Composition*

The System Description shall be a textual document supported by the relevant detail from the Requirements Specification, including a list of functions/facilities not provided.

3.12.2.3 *Completion criteria*

It shall be considered if the system description is consistent with the user requirements.

3.12.2.4 *Status*

The System Description shall be an end-stage product, delivered complete in Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.12.2**.

This document shows how the Requirements Specification is met by the Technical Environment Description for a particular Technical System Option. In many cases the major decisions in this area will have already been taken in choosing a Business System Option.

Derivation:

Logical System Specification:

— *Requirements Specification.*

Where used:

— *Technical System Options: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.12.3 Technical Environment Description

3.12.3.1 *Purpose*

The Technical Environment Description shall serve to provide, before selection, sufficient information for users to understand how the system will work, for significant design factors to be explained, and for detailed cost estimates to be made. It shall also provide, after selecting a technical option, a detailed definition of the functional and physical aspects of the system.

3.12.3.2 *Composition*

The Technical Environment Description shall consist of a textual description of the basic details, including the following:

— hardware;

— software;

— system sizing;

— fallback and recovery arrangements;

— access rights;

— access and security methods;

— hardware/software maintenance.

These details shall be supplemented by Impact Analysis and System Description.

3.12.3.3 *Completion criteria*

The following shall be considered.

a) Is the description technically feasible and viable?

b) Does this clearly reflect the management decision?

c) Does this clearly reflect the hardware and software configuration issues?

d) Are any objectives implied by selection? If so are these clearly identified?

e) Is the management decision on the selected option understood by practitioners and user representatives?

3.12.3.4 *Status*

The TED shall be an end-stage product delivered complete in Stage 4. A fully detailed version of this product shall be delivered for the Selected TSO.

COMMENTARY AND RECOMMENDATIONS ON **3.12.3**.

This product provides the specification of the technical environment which is produced once the Technical System Option has been selected. This detail is then passed on to physical design activities.

Derivation:

Logical System Specification:

— *Requirements Specification;*

— *(Selected) Technical System Option.*

Where used:

— *Technical System Options: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.12.4 Technical System Options

3.12.4.1 *Purpose*

Technical System Options shall contain details of a number of possible solutions, each of which would satisfy the user requirements. Each alternative Technical System Option shall present a high-level system design, and shall be evaluated from the technical aspects. The documentation shall provide information on the project's way ahead, shape, timing, costs, implications and timescales (relating to the potential functionality of the system).

3.12.4.2 *Composition*

Technical System Options shall be textual documents which provide details of the proposed solution as follows:

a) heading information:

— Option name and/or identifier;

b) detailed documentation:

NOTE This may be textual and/or compiled from the following documents as appropriate:

— Cost/Benefit Analysis;
— Impact Analysis;
— Outline Development Plan;
— System Description;
— (Outline) Technical Environment Description.

3.12.4.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Is there clear identification of system functionality vis-a-vis non-functional aspects?

2) Does the hardware and software configuration agree with the user roles/function location information?

3) Can the project meet its objectives as agreed?

4) Is the option technically feasible and economically sound?

b) for the set:

1) Are all technical options documented here?

2) Have relevant users and an independent, experienced analyst been identified to join the review team, and to comment on the acceptability of the recommendation?

3) Has a mechanism been identified to implement the capacity planning activity?

4) Is there information on any existing installation configurations and capacity?

3.12.4.4 *Status*

NOTE There may be several TSOs produced during Stage 4. Some may only be produced in Outline form, others may be further developed amongst more than one short-listed form. One will finally be selected (see **3.12.5**).

All non-selected forms shall be interim products produced during Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.12.4**.
This product is the set of Technical System Options which has been developed so that the system development direction can be chosen.

Each option documents the functions to be incorporated and details implementation requirements. Each description is textual with some planning information. Functional elements are taken directly from the Requirements Specification.

Derivation:

Logical System Specification:

— *project initiation document;*

— *Requirements Specification;*

— *Selected Business System Option.*

Capacity planning techniques will be applied to information produced within the Technical System Options Stage (Capacity Planning Input) and the results used to update the appropriate option prior to selection.

Where used:

— *Technical System Options: LS.*

Presentation conventions are not applicable.

3.12.5 Selected Technical System Option

3.12.5.1 *Purpose*

The Selected Technical System Option shall provide management with information about the future development of the project: the direction, timing, costs, implications and timescales (all related to the proposed system functionality).

3.12.5.2 *Composition*

The Selected Technical System Option shall be a textual document providing details of the selection process for the proposed solution.

NOTE Details may include the following:
— Cost Benefit Analysis;
— Outline Development Plan;
— summary of option;
— rationale behind selection.

3.12.5.3 *Completion criteria*

The following shall be considered.

a) Is there clear identification of system functionality with reference to non-functional aspects?

b) Does the hardware and software configuration accord with user roles and function location information?

c) Can the project meet its objectives as agreed?

d) Is the recommendation technically feasible and economically sound?

3.12.5.4 *Status*

The Selected Technical System Option shall be an end-stage product, produced at the end of Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.12.5**.
Several TSOs may be evaluated. Only one will be selected, it may itself be based on aspects of existing TSOs.

Derivation:

Logical System Specification:

— *project initiation document;*

— *Requirements Specification;*

— *Selected BSO;*

— *TSOs.*

Where used:

— *Technical System Options: LS;*

— *Physical Design: PD.*

3.13 Physical Data Design products

COMMENTARY AND RECOMMENDATIONS ON **3.13**.
Physical Data Design is used to cover the features of physical data placement, and DBMS optimization. The technique focuses on identifying required groupings of data elements in physical blocks and then optimizing access paths. An initial physical design can be produced using guidelines given in the method. Product-specific rules can then be applied for a more detailed result. Timing and sizing estimates can then be carried out on the design, and any necessary changes made to the design to meet performance and space criteria.

The products produced under this heading are designed to further the following objectives:

a) *implement the required system's data requirements as specified in the Required System Logical Data Model;*

b) *support the system's processing requirements;*

c) *satisfy any performance and space constraints.*

3.13.1 Physical Data Design

3.13.1.1 *Purpose*

The design itself shall be used to produce an implementation specific data design that ensures that the system meets performance objectives expressed in terms of machine space and levels of service.

The performance of the product specific Physical Data Design shall be evaluated and the design modified to ensure that it meets the objectives defined by the users.

NOTE Several Physical Data Designs may be needed to implement a distributed Required System Logical Data Model.

3.13.1.2 *Composition*

The Physical Data Design is dependent on the implementation vehicle and so shall show how the Required System LDM is to be implemented in the physical system.

NOTE 1 The documentation may include any or all of the following:

— Physical Data Design (first-cut);
— Physical Data Design (optimized);
— space estimation form(s) (to be designed for the characteristics of a particular DBMS);
— timing estimating form(s) (to be designed for the characteristics of a particular DBMS).

NOTE 2 For the space estimation, the following sample data may be collected and manipulated in a spreadsheet:

— block/page type;
— block/page size (kbytes);
— block/page working capacity (e.g. kbytes at 60 %);
— record type (entity id for row);
— primary owner;
— number of record instances per hierarchy;
— record data size in bytes;
— record storage overhead;
— relationship overheads;
— pointer to each master;
— pointer to each detail;
— total space per record (bytes);
— cumulative space for hierarchy;
— cumulative space in block;
— data total;
— block header;
— block total.

NOTE 3 For the timing estimation, the following sample data may be collected and manipulated in a spreadsheet. Disk timing factors should be considered as a matrix with the following entries:

a) rows:
— read direct via index;
— read direct via pointer chain;
— read next in database;
— update in place;
— update in place/modify pointer chain;
— create;
— delete;

b) columns:
— average read time (including rotational delay and latency);
— average write time (including rotational delay, latency and any read after write check);
— load time for blocks into memory;
— average time for DBMS software to interpret and execute call;
— average time for TPMS to queue/de-queue physical I/O calls;
— average time for operating system to schedule operations.

3.13.1.3 *Completion criteria*

The following shall be considered:

a) for each:

1) Is the data design implementable in the chosen Physical Environment Specification?

2) Does the design meet performance objectives?

3) Will the resultant DBMS operate efficiently in the available disk and memory space?

4) Are all Logical Data Model entity types and relationships implemented in the physical DBMS?

5) Are batching elements (transient data stores) incorporated?

6) Can the DBMS support all the physical processing identified in the Physical Process Specification to the required level of service?

7) Is the data design fully documented?

8) Do the columns on the space planning form match the Physical Environment Classification?

9) Do the operations and factors on the timing planning form match the Physical Environment Classification and other sources?

b) in Stage 6:

1) Is the optimized Physical Data Design consistent with the product specific data design?

2) Have all deviations from the logical design been agreed and documented?

c) generally:

1) Is suitable product information to establish the validity (or not) of the design available?

2) Has the chosen product been assessed in terms of its appropriateness to priority requirements?

3) Is required optimization expertise available?

4) Is accurate timing factor information available?

3.13.1.4 *Status*

First-cut PDD shall be an interim product, developed during PD. The optimized variant shall be an end-module product, PD.

COMMENTARY AND RECOMMENDATIONS ON **3.13.1**.
This product is the definition of the physical database which is to be implemented. The design is developed in two steps: the first produces a first-cut design based on applying rules about the DBMS to the Required System Logical Data Model; the second is a design optimized for performance reasons.

Derivation:

Physical Design:

— *expertise of external consultants;*

— *Function Definitions (service level requirements);*

— *Function Component Implementation Map;*

— *monitoring records of performance of existing software;*

— *optimization rules/strategy (for the particular implementation);*

— *Physical Data Design (product specific);*

— *Physical Environment Specification;*

— *Required System Logical Data Model.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.13.2 Physical Design Strategy

3.13.2.1 *Purpose*

The Physical Design Strategy shall be used to record the planned approach for all aspects of physical implementation including cross-checking between data and process classification schemes and defining which features are to be used, and how they are to be documented.

3.13.2.2 *Composition*

The composition shall be as follows:

a) planning information:
- — Activity Descriptions;
- — Activity Network;
- — Product Breakdown Structure;
- — Product Description;

b) program specification format defining:
- — purpose;
- — composition;
- — derivation;
- — inputs;
- — outputs;
- — used by;
- — used for;
- — test criteria;
- — external dependencies;
- — documentation references;

c) product specific data design rules.

NOTE For each type of function component, it is necessary to specify the following factors to be considered in accordance with non-functional requirements for flexibility, performance, security and usability:
- — complexity;
- — suitability of non-procedural features;
- — criticality (need for formal notation);
- — review procedures (need for structured walkthrough).

3.13.2.3 *Completion criteria*

The following shall be considered.

a) Are all necessary standards set?

b) Are standards set for all the implementation object types identified for the physical environment?

c) Are there rules for each class of function component?

d) Are there rules for each option supportable in the physical environment?

e) Does the design team understand the application of the standard criteria for procedural/non-procedural specifications?

f) Are there Activity Descriptions to produce required Stage 6 products to relevant standards?

g) Are all relevant vendor features incorporated?

h) Are roles identified with skill and training requirements?

i) Are techniques and documentation standards identified?

j) Have any relevant existing installation standards been considered in the light of the nature of physical environment?

k) Have the highest priority function characteristics been considered?

l) Does the design team have the necessary skill levels?

m) Have the quality and availability of product specific guides been assessed?

3.13.2.4 *Status*

The Physical Design Strategy shall be an end-module product, delivered complete, PD.

COMMENTARY AND RECOMMENDATIONS ON **3.13.2**.
This product documents all aspects relating to designing the physical implementation of the application. This includes all planning documentation.

Derivation:

— *Physical Design.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.13.3 Physical Design

3.13.3.1 *Purpose*

The Physical Design shall serve to define the data and processing elements of the system which is to be implemented.

NOTE It should exploit the benefits and circumvent the constraints of the chosen implementation environment, without compromising the portability of the Logical Design.

It shall be consistent with the Logical System Specification.

3.13.3.2 *Composition*

The composition shall be as follows:

— Physical Data Design;

— Physical Process Specification;

— Process Data Interface.

NOTE See Figure B.19 for the product breakdown structure diagram.

3.13.3.3 *Completion criteria*

The following shall be considered.

a) Does the Physical Design conform to relevant installation standards?

b) Is the project within the scope set in the project initiation document? Can it continue to the construction and testing phase? Is it operable?

c) Can the users confirm that this solution addresses their requirements? Do they have the means for testing and acceptance?

d) Is the design viable? Does it meet all critical requirements within the constraints of the Physical Environment Specification?

e) Does it satisfy/is it consistent with the Logical System Specification?

f) Does the design take full account of hardware/software limitations?

g) Are the file handling mechanisms selected appropriate to space and performance requirements of the Physical Data Design?

h) Are the file structures appropriate?

i) Do the data structures incorporate existing files and records as required?

j) Are access methods appropriate?

k) Are all aspects of the Physical Design consistent?

l) Is there evidence that the design will meet service level requirements and provide all functions?

m) Has the implementation of all Logical Design components been documented with the proper cross-references traced back to Function Definitions and Requirements Catalogue?

n) Are files and data compatible between different sub-systems?

o) Have unnecessary duplications and redundancy in processing and data been eliminated?

p) Do the detailed processes: provide all functionality; have appropriate controls and restarts; have appropriate security arrangements; meet system performance requirements; conform to operations/management rules?

3.13.3.4 *Status*

The Physical Design shall be an end-module product produced on completion of PD.

COMMENTARY AND RECOMMENDATIONS ON **3.13.3**.
This product is the module product from the Physical Design Module. It forms the description of the implementation of the application.

Derivation:

Physical Design:

— Logical System Specification;

— Physical Environment Specification.

Where used:

— Physical Design: PD.

Presentation conventions are not applicable.

3.14 Physical Process Specification products

COMMENTARY AND RECOMMENDATIONS ON **3.14**.
These products develop the processing requirements of the application in line with the characteristics of the planned physical environment.

3.14.1 Function Component Implementation Map

3.14.1.1 *Purpose*

The FCIM shall serve to classify and specify the implementation components needed to meet the processing requirements for all function components of the Function Definitions. The classification shall be in terms of the object types supported by the physical environment.

The document shall have the following four sub-goals:

a) to eliminate duplicate components;

b) to reuse common components;

c) to specify the implementation route to be followed;

d) to package components into success units.

The FCIM documentation shall be implemented to local standards.

3.14.1.2 ***Composition***

The composition shall be as follows:

a) cross-reference of function components showing uses of and use by relationships (including showing how common components are reused);

b) cross-reference of function components showing form of implementation (tool facility used, procedural and non-procedural specification and languages);

c) detailed processing specifications each including:

1) the purpose of the procedure;

2) its relationship with other procedures;

3) inputs and outputs;

4) summary of operation which may be classified using one or more of the following:

— non-procedural code;

— SSADM generic structure diagrams and supporting documentation;

— other forms of procedural specification including constrained natural language or other notations;

d) for some programs or run units it will be necessary to add all or some of the following:

— structure of the job in terms of constituent programs;

— hardware requirements;

— files (media, sort orders, names, volumes and volatility);

— file and record layouts not previously defined;

— references to defined layouts;

— operations required to drive output from input;

— controls and restarts;

— special requirements (e.g. library routines and security requirements);

e) manual procedures, including detail covering the purpose of the procedure, position in sub-system and relationship to other procedures and programs, inputs and outputs, human and material resource requirements, implications of operations and management policies, timing and development estimate.

3.14.1.3 ***Completion criteria***

The following shall be considered:

a) for each component:

1) Is the implementation technique appropriate?

2) Is the nature of each component or fragment properly described in terms of purpose, inputs, outputs, action (and variations) and usage?

3) Are the non-procedurally encoded specifications validated using the tools of the physical environment (i.e. do they compile)?

4) Has each specification (especially the procedural ones) been subjected to a structured walk through?

5) Have all walk-through results been documented?

6) Can run schedules be implemented feasibly?

7) Will run units meet required service levels?

b) for the set:

1) Is every function definition component implementation shown on the map?

2) Is every function component cross-referenced to a processing fragment?

3) Are the cross-references fully documented (preferably using the facilities of the physical environment)?

4) Do the object types defined in the Function Component Implementation Map match the types available in the Physical Environment Specification?

5) Are there facilities for documentation and cross-referencing in the Physical Environment Specification?

6) Is there flexibility in the Physical Environment Specification facilities?

3.14.1.4 ***Status***

The FCIM shall be an end-module product, delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.14.1**.

This product provides a classification and specification for implementation of all function components defined in the Function Definitions to meet the processing requirements.

Derivation:

Physical Design:

— *Application Development Standards;*

— *Logical Design;*

— *Physical Data Design (first-cut);*

— *Physical Environment Classification;*

— *Physical Environment Specification.*

Where used:

— *Physical Design: PD;*

Presentation conventions are not applicable.

3.14.2 Process Data Interface

3.14.2.1 *Purpose*

The Process Data Interface shall serve to document the interface between the implemented Physical Data Design and all the components of the Physical Processing Specification.

NOTE This interface should make the database implemented using the Physical Environment Specification appear as the Required System Logical Data Model to the processing components. The aim of this is to ease understandability and maintenance. It will also minimize the impact of change.

It shall allow the designer to implement the logical update and enquiry processes as physical programs, independently of the physical database structure.

3.14.2.2 *Composition*

The composition shall be as follows.

a) fragment specification for DBMS accesses (from Function Component Implementation Map);

b) fragment specification for declarative syntax and control utilities (from Function Component Implementation Map).

3.14.2.3 *Completion criteria*

The following shall be considered.

a) Are all necessary requirements documented?

b) Are all update data views supported?

c) Are all enquiry data views supported?

d) Are all physical DBMS navigation paths explicitly cross-referenced to the Required System LDM?

e) Are all processing interfaces defined?

f) Are installation standards followed with regard to data administration facilities?

g) Do the DBMS access components meet performance requirements?

h) Is there a sufficient level of knowledge of the characteristics and functionality of the implementation language?

i) Is there recognition of the extent to which the Physical Data Design is a one-to-one mapping with the Required System Logical Data Model?

3.14.2.4 *Status*

The Process Data Interface shall be an end-module product delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.14.2**.
This product documents how the Logical Data Model can be mapped onto the Physical Data Design, showing how it interfaces with the Physical Processing Specification. It allows the designer to implement the logical update and enquiry processes as physical programs, independently of the physical database structure.

Derivation:

Physical Design:

— *Function Component Implementation Map;*

— *Physical Data Design (optimized);*

— *Process data interface requirements.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.14.3 Physical Process Specification

3.14.3.1 *Purpose*

Each Physical Process Specification shall provide a documentation base on which future maintenance, enhancement, end user support (such as information centre services), error handling and help desk activities can draw. The level of detail required and the format of the specifications shall be dependent on the target environment and installation standards.

All the specifications for the system processing shall be consolidated to enable construction, testing and commissioning of the proposed system to be completed.

3.14.3.2 *Composition*

The composition shall be as follows:

— Data Catalogue (probably implemented in a data dictionary);

— Function Component Implementation Map;

— Function Definition (with additional data optimization and PDI requirements);

— Requirements Catalogue (noting any design trade-offs or compromises made in the course of Physical Design);

— Required System Logical Data Model.

NOTE See Figure B.16 for the product breakdown structure diagram.

3.14.3.3 *Completion criteria*

The following shall be considered.

a) Are all components complete and consistent?

b) Does all the documentation conform to the constraints of the Physical Environment Classification, the Requirements Catalogue and the Technical Environment Description?

c) Is there recognition of the existence of relevant installation standards for producing these facilities?

d) Is there availability of required design and language expertise?

e) Have all deviations from the logical design been agreed and documented?

3.14.3.4 *Status*

The Physical Process Specification shall be an end-module product delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.14.3**.
This product consolidates all of the specifications for processing which are required in the proposed system.

Derivation:

Physical Design:

— *Logical Design;*

— *Physical Environment Classification;*

— *Application, Development Standards.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.15 Physical Environment Description products

COMMENTARY AND RECOMMENDATIONS ON **3.15**.
These products are all classifications containing the information provided by the specific suppliers to be incorporated into further SSADM development.

3.15.1 DBMS Data Storage Classification

3.15.1.1 *Purpose*

The DBMS Data Storage Classification shall be used to analyse and record data storage and retrieval mechanisms of the DBMS or file handler which are being considered for implementing the proposed system.

NOTE 1 There may be more than one DBMS within the proposed solution or development environment.
NOTE 2 The required detail may have been provided in a usable format by a supplier in the interface guide.

3.15.1.2 *Composition*

The composition shall be as follows:

a) DBMS/file handler;

b) relationship representation:

— table;

— list;

c) amalgamation of entity type and relationship data:

— none;

— relationship and master;

— relationship and detail;

— relationship with master and detail;

— relationship and relationship;

d) key representation in relationship (logical or physical):

— master to detail/detail to next detail;

— detail to master;

e) retrieval by logical key:

— search;

— indexing;

— hashing;

f) significant restrictions.

3.15.1.3 *Completion criteria*

The following shall be considered:

a) Is each entry complete?

b) Does the information for each entry accord with the supplier's information?

c) Are the appropriate standards documented?

d) Are all data management systems fully documented?

e) Is a product specific guide, or expert, or the supplier available to provide the required details?

3.15.1.4 *Status*

The DBMS Data Storage Classification shall be an end-module product, delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.15.1**.
This classification is used for analysing and recording data storage and retrieval mechanisms of the DBMS or file handler.

Derivation:

Physical Design:

— *Physical Environment Specification (from supplier).*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.15.2 DBMS Performance Classification

3.15.2.1 *Purpose*

The DBMS Performance Classification shall record various factors impacting upon the performance of a DBMS or file handler being considered for implementing the proposed system.

NOTE 1 There may be more than one DBMS within the proposed solution or development environment.
NOTE 2 The required detail may have been provided in a usable format by a supplier in the interface guide.

3.15.2.2 *Composition*

The composition shall be as follows:

— DBMS/file handler (including relevant software and hardware);

— performance overheads;

— standard timing factors.

3.15.2.3 *Completion criteria*

The following shall be considered:

a) Are all entries complete?

b) Do the entries agree with the supplier's product documentation?

c) Are the appropriate standards documented?

d) Are all data management systems fully documented?

e) Does the product interface guide exist; or is a relevant expert/supplier available to provide the details?

3.15.2.4 *Status*

The DBM Performance Classification shall be an end-module product, delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.15.2**.
This product records the factors which impact on the performance of a DBMS or file handler.

Derivation:

Physical Design:

— *Physical Environment Specification (from supplier).*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.15.3 Physical Environment Classification

3.15.3.1 *Purpose*

The Physical Environment Classification shall classify the environment in which the application is to be implemented. If necessary, the development environment shall also be described and the migration path between the two documented.

NOTE Implementation and development are likely to use the same file handling facilities but may run on differing hardware and so changes in this area are likely to be most significant.

3.15.3.2 *Composition*

The composition shall be as follows:

— DBMS Data Storage Classification;

— DBMS Performance Classification;

— Processing System Classification.

NOTE See Figure B.17 for the product breakdown structure diagram.

3.15.3.3 *Completion criteria*

The following shall be considered.

a) Are the appropriate standards documented?

b) Are all entries valid, relevant and correct?

c) Are all significant facilities classified?

d) Is the classification consistent with vendor information?

e) Is relevant documentation on the development and implementation environments available?

3.15.3.4 *Status*

The Physical Environment Classification shall be an end-module product, delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.15.3**.
This product classifies the environment in which the application is to be implemented. It also describes the development environment and migration path where necessary.
(See also **3.15.1**, **3.15.2** *and* **3.15.4**.*)*

Derivation:

Physical Design:

— *Logical System Specification;*

— *Physical Environment Specification.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.15.4 Processing System Classification

3.15.4.1 *Purpose*

The Processing System Classification shall classify details of the processing environment within which the application is to be implemented (and developed).

NOTE 1 More than one aspect of the tool may be used to provide the full range of processing facilities, including data dictionary, data manipulation and definition and procedural languages. A separate form should be completed for each.
NOTE 2 The required detail may have been provided in a usable format by a supplier in the interface guide.

3.15.4.2 *Composition*

The composition shall be as follows:

— classes of tool feature;

— procedural/non-procedural;

— on-line/off-line;

— success units;

— error handling;

— process components;

— database processing;

— update;

— enquiry;

— I/O processing;

— dialogue processing;

— dialogue navigation;

— process data interface;

— distributed systems.

3.15.4.3 *Completion criteria*

The following shall be considered.

a) Are all relevant standards set?

b) Do the entries agree with the supplier's product documentation?

c) Are the appropriate standards documented?

d) Are all data management systems fully documented?

e) Is the relevant product and environment information available?

3.15.4.4 *Status*

The Processing System Classification shall be an end-module product, delivered complete in PD.

COMMENTARY AND RECOMMENDATIONS ON **3.15.4**. *This product is used to classify the details of the processing environment which is to be used for implementation. Where appropriate, it also defines the development environment.*

Derivation:

Physical Design:

— *Logical System Specification;*

— *Physical Environment Specification.*

Where used:

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.16 Miscellaneous products

COMMENTARY AND RECOMMENDATIONS ON **3.16**. *This group contains products which do not fit into any of the other categories. Some of the products are not project specific, but relate to the overall context in which the system is to be developed and to operate.*

Some of these are products which are not generated specifically for an individual SSADM project but which may be in place prior to SSADM tasks being undertaken. They are specified since, if they exist, they will guide and constrain the expectations of contractual partners.

3.16.1 Application Development Standards

3.16.1.1 *Purpose*

Application Development Standards shall serve to define the appropriate standards to be used throughout the application design, construction and testing stages.

3.16.1.2 *Composition*

The composition shall be as follows:

— Application Naming Standards;

— Application Style Guide;

— Physical Design Strategy;

— Physical Environment Classification.

NOTE See Figure B.8.

3.16.1.3 *Acceptance criteria*

The following shall be considered:

a) Are all necessary standards set?

b) Are the standards readily available to all relevant project members?

c) Do relevant installation standards documents exist?

d) Is the information concerning the (physical) implementation and development environments available?

3.16.1.4 *Status*

Application Development Standards shall be non-SSADM Composite Product; composed of Application Naming Standards (**3.16.2**), Application Style Guide (**3.16.3**), Physical Design Strategy (**3.13.2**), Physical Environment Classification (**3.15.3**) and delivered completed prior to the start of Stage 6.

COMMENTARY AND RECOMMENDATIONS ON **3.16.1**. *This product defines the standards which apply to the physical design and development activities for the project.*

Derivation:

a) *Logical System Specification (Application Style Guide only):*

— *Installation Style Guide (if it exists);*

— *Requirements Catalogue (for user requirements elicited during the specification prototyping activities).*

b) *Physical Design (all other components):*

— *Installation Development Standards;*

— *Physical Environment Specification.*

Where used:

— *Dialogue Design: LS;*

— *Physical Design: PD;*

Presentation conventions are not applicable.

3.16.2 Application Naming Standards

3.16.2.1 *Purpose*

Application Naming Standards shall be used to define standards for naming elements of a system, incorporating any constraints imposed by the physical environment.

3.16.2.2 *Composition*

The composition shall vary according to the installation.

NOTE A suggested example is as follows:
— identifier (unique);
— class of fragment;
— purpose/action;
— inputs/subject/precondition;
— outputs/object/post condition.

3.16.2.3 *Acceptance criteria*

The following shall be considered.

a) Are all necessary standards set?

b) Are conventions understood?

c) Do they cover all aspects of the application?

d) Do relevant installation standards documents exist?

e) Is information concerning the (physical) implementation and development environments available?

3.16.2.4 *Status*

Application Naming Standards shall be non-SSADM products delivered completed prior to the start of PD.

COMMENTARY AND RECOMMENDATIONS ON **3.16.2**.
This product defines the naming conventions for all aspects of the application under development, with particular emphasis being placed on constraints imposed by the (physical) implementation environment.

Derivation:

Physical Design:

— *Installation Development Standards;*

— *Physical Environment Specification.*

Where used:

— *Dialogue Design: RS;*

— *Physical Design: PD.*

Presentation conventions are not applicable.

3.16.3 Application Style Guide

3.16.3.1 *Purpose*

The Application Style Guide shall set the standards for the user environment within a particular project (application). This shall include ergonomic details, such as siting of equipment, and system based requirements such as the style that is to be used for dialogues and reports.

NOTE Here style relates to format (layout) in terms of size, positioning of particular items (in particular headings and help text).

3.16.3.2 *Composition*

The Application Style Guide shall be a textual document that establishes all the standards for the user environment of computer-based systems.

3.16.3.3 *Acceptance criteria*

The following shall be considered.

a) Are all the necessary standards set?

b) Does the Installation Style Guide exist?

3.16.3.4 *Status*

The Application Style Guide shall be a non-SSADM Product delivered completed prior to the start of Stage 4.

COMMENTARY AND RECOMMENDATIONS ON **3.16.3**.
This guide establishes a set of standards relevant to the design of the user interface and associated user aspects relevant to a particular application. It will be derived from the Installation Style Guide (if one exists) and the specific requirements of the project.

Derivation:

Logical System Specification:

— *Installation Style Guide (if it exists);*

— *Requirements Catalogue (for user requirements elicited during the specification prototyping activities).*

Where used:

— *Dialogue Design: RS;*

— *Physical Design: PD.*

3.16.4 Installation Style Guide

3.16.4.1 *Purpose*

The Installation Style Guide shall set the standards for the user environment. This shall include ergonomic details, such as siting of equipment, and system based requirements, such as the style that is to be used for dialogues and reports.

NOTE 1 Style in the latter context relates to format (layout) in terms of size, positioning of particular items (in particular headings and help text).

NOTE 2 The Installation Style Guide sets the general standards that should be adhered to for any systems development.

3.16.4.2 *Composition*

The Installation Style Guide shall be a textual document that will set all the standards for the user environment of computer-based systems.

3.16.4.3 *Acceptance criteria*

The following shall be considered.

a) Are all the necessary standards set?

b) Do standards conform to relevant, existing installation standards?

3.16.4.4 *Status*

The Installation Style Guide shall be a non-SSADM product delivered complete prior to project initiation.

COMMENTARY AND RECOMMENDATIONS ON **3.16.4**.
This is a set of standards concerning the nature, approach and style of the human factors aspects of computerized systems. The standards should be followed by all projects undertaken within an organization.

Derivation:

The product is defined outside of the scope of SSADM.

Where used:

— *Dialogue Design: LS;*

— *Specification Prototyping: RS.*

Presentation conventions are not applicable.

3.16.5 Project Initiation Document

3.16.5.1 *Purpose*

The Project Initiation Document shall amalgamate all the information needed to start a project. It shall be used to convey the information to the project team.

3.16.5.2 *Composition*

The composition shall be as follows:

a) Business Case;

b) Project Brief:

— project objectives;

— major products;

— constraints;

c) Project Boundary;

d) Project Management.

3.16.5.3 *Acceptance criteria*

a) Are all components present?

b) Are names shown against required roles?

c) Have people concerned agreed to act and commit time as scheduled?

d) Have dates been set for control meetings and reports?

e) Have measures been proposed for all risks identified?

f) Have project, plans been reviewed and agreed?

3.16.5.4 *Status*

The Project Initiation Document shall be a non-SSADM product, delivered complete prior to initial stage of development, usually Feasibility or Requirements Analysis.

COMMENTARY AND RECOMMENDATIONS ON **3.16.5**.

This product identifies the boundary, rationale, key objectives and constraints of the system to be developed.

Derivation:

A product of whatever IS strategy and planning procedures and products are in use by the sponsoring organization.

Where used:

— *Feasibility: FS;*

— *Business System Options: RA;*

— *Technical System Options: LS.*

Presentation conventions are not applicable.

3.16.6 SSADM Structure Diagram

3.16.6.1 *Purpose*

The SSADM Structure Diagram shall be used to represent hierarchical structures for a number of different product representations.

3.16.6.2 *Composition*

The SSADM Structure Diagram shall comprise notations for sequence, selection, iteration, and elementary components; also additional notations for parallelism and operations.

3.16.6.3 *Completion criteria*

The following shall be considered.

a) Is there exactly one box with no parent: the root node?

b) Is this box a sequence, selection or iteration? It shall not be a parallel component.

c) Do all boxes below the root node have exactly one parent?

d) Are all children of a single parent of the same type?

e) Is each iteration box the only child of its parent?

f) Do all selections consist of at least two choices (even if one choice is an empty box, indicating a null)?

g) Do all boxes contain either the choice symbol, the iteration symbol, or no symbol?

h) Is every box, other than the root, connected to its parent box by a line?

i) Is every non-elementary box connected to its children by a line?

j) Are there no lines other than these?

k) Is the diagram drawn with no crossing lines?

l) If the diagram has been split, are the links clear and easy to follow?

For parallel constructs the following shall be considered.

1) If a parallel construct has been used, is the diagram an ELH?

2) Is every parallel component part of a sequence?

3) Are there two or more boxes below every parallel bar?

4) Does every box immediately below a parallel bar contain no symbol?

3.16.6.4 *Status*

The SSADM Structure Diagram is not a distinct product, but shall be used in a variety of products, as appropriate, as follows:

a) EEM: entity life history, effect correspondence diagram;

b) FD: I/O structure diagram;

c) LDPD: logical update process model, logical enquiry process model;

d) LDM: enquiry access path;

e) DD: dialogue structure.

3.16.6.5 ***Presentation conventions***

The SSADM Structure Diagram shall be presented in accordance with Figure 29 and Figure 30.

COMMENTARY AND RECOMMENDATIONS ON **3.16.6**.

Where used:

— *Entity-event Modelling: RS*

— *Logical Database Process Design: LS;*

— *Function Definition: RS;*

— *Logical Data Modelling: RA, RS;*

— *Dialogue Design: LS;*

— *Physical Design: PD.*

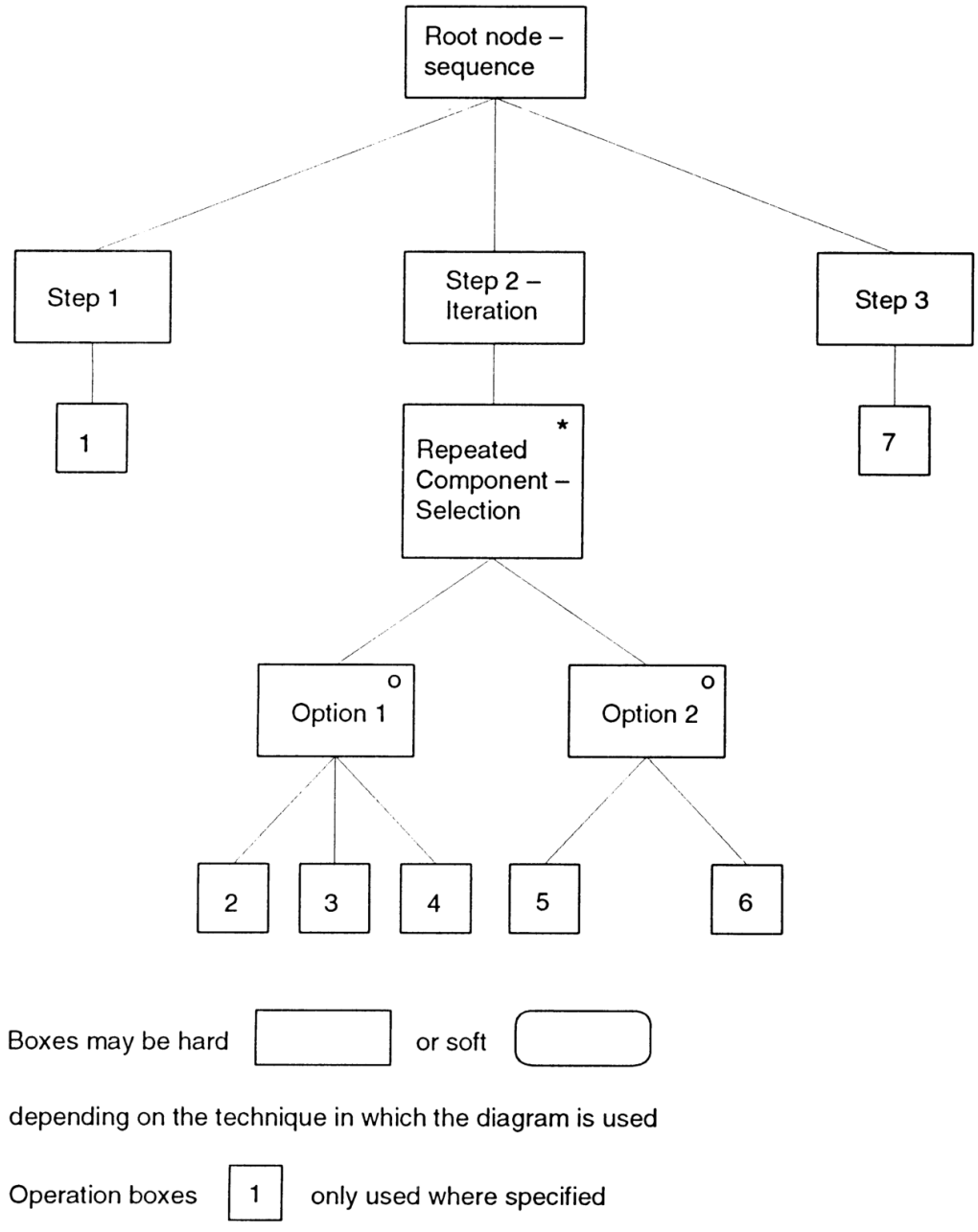
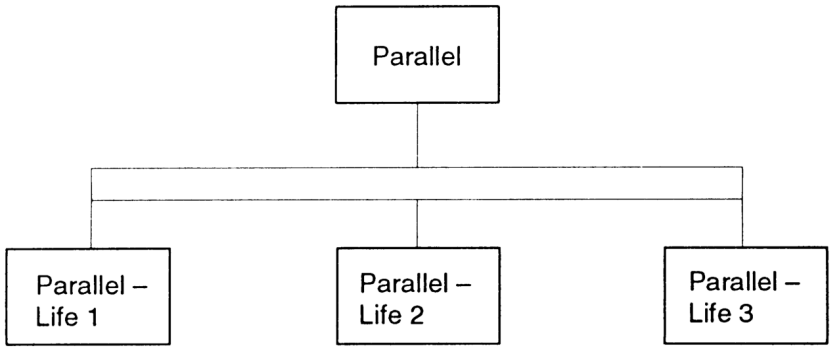


Figure 29 — Presentation conventions for the SSADM Structure Diagram



NOTE Used only in ELHs.

Figure 30 — Presentation conventions for the SSADM Structure Diagram: parallel structure

3.16.7 Physical Environment Specification

3.16.7.1 *Purpose*

The Physical Environment Specification shall be used to specify the hardware and software products and services to be supplied, and the configuration in which they are to be supplied, commissioned and made available for implementation.

3.16.7.2 *Composition*

The composition shall be to agreed standards specified in the contract and the service provider shall document the following:

a) all hardware products and versions to be supplied;

b) all software products and versions to be supplied;

c) hardware configuration;

d) software configuration;

e) operating documentation;

f) reference manuals;

g) product specific guides for use of software products with SSADM.

3.16.7.3 *Completion criteria*

It shall be considered if the document meets all criteria stipulated in the contract.

3.16.7.4 *Status*

The Physical Environment Specification shall be an end-module product delivered complete at the end of PD.

COMMENTARY AND RECOMMENDATIONS ON **3.16.7**.

The document may include a range of different products or configurations. It may also include products to operate in a distributed environment.

Derivation:

A contractual document derived from the Requirements Specification and the Technical Environment Description.

Where used:

— *Physical Design.*

Presentation conventions are not applicable.

Annex A (normative)
Products specified

The products specified in this British Standard are listed in Table A.1. Entries which do not have a section reference specified refer to those products which appear as named products in Annex B, but which do not have a specific entry in section **3**. In these cases the details for those products can be found by referring to the product named in brackets.

Table A.1 — Products specified

Product	Conceptual grouping	Clause number	Product	Conceptual grouping	Clause number
Application Development Standards	MISC	**3.16.1**	I/O Structure Diagram	FD	**3.5.6**
Application Naming Standards	MISC	**3.16.2**	I/O Structures for all functions	FD	**3.5.7**
Application Style Guide	MISC	**3.16.3**	Impact Analysis	BSO	**3.1.6**
Attribute/Data Item Description	LDM	**3.6.1**	Installation Style Guide	MISC	**3.16.4**
Business System Options	BSO	**3.1.1**	Logical Data Flow Model (see DFM)	DFM	—
Capacity Planning Input	TSO	**3.12.1**	Logical Data Model	LDM	**3.6.52**
Command Structure	DD	**3.3.1**	Logical Data Store/Entity Cross-reference	DFM	**3.2.9**
Context Diagram	DFM	**3.2.1**	Logical Data Structure	LDM	**3.6.6**
Cost/Benefit Analysis	BSO	**3.1.3**	Logical Design	PS	**3.8.2**
Current Environment LDM (see LDM)	LDM	—	Logical/Physical Data Store Cross-reference	DFM	**3.2.10**
Current Physical DFM (see DFM)	DFM	—	Logical Process Model	LDPD	**3.7.2**
Current Services Description	RD	**3.10.6**	Logical System Specification	PS	**3.8.3**
Data Catalogue	LDM	**3.6.2**	Menu Structure	DD	**3.3.7**
Data Flow Diagram: Level 1	DFM	**3.2.2**	Non-selected BSOs (see BSOs)	BSO	—
Data Flow Diagram: Lower Level	DFM	**3.2.3**	Non-selected Feasibility Options (see Feasibility Options)	BSO	—
Data Flow Model	DFM	**3.2.4**	Non-selected TSOs (see TSOs)	TSO	—
DBMS Performance Classification	PED	**3.15.2**	Physical Data Design	PD	**3.13.1**
DBMS Data Storage Classification	PED	**3.15.1**	Physical Design	PD	**3.13.3**
Dialogue Control Table	DD	**3.3.2**	Physical Design Strategy	PD	**3.13.2**
Dialogue Element Description	DD	**3.3.3**	Physical Environment Classification	PED	**3.15.3**
Dialogue Level Help	DD	**3.3.4**	Physical Environment Specification	MISC	**3.16.7**
Dialogue Structure	DD	**3.3.6**	Physical Process Specification	PPS	**3.14.3**
Dialogues	DD	**3.3.5**	Physical System Specification	PS	**3.8.4**
Document Flow Diagram	DFM	**3.2.5**	Process Data Interface	PPS	**3.14.2**
Effect Correspondence Diagram	EEM	**3.4.1**	Process/Entity Matrix	DFM	**3.2.11**
Elementary Process Description	DFM	**3.2.6**	Processing Specification	PS	**3.8.1**
Enquiry Access Path	FD	**3.5.1**	Processing System Classification	PED	**3.15.4**
Enquiry Process Model	LDPD	**3.7.1**	Project Initiation Document	MISC	**3.16.5**
Entity Description	LDM	**3.6.3**	Prototype Demonstration Objective Document	SP	**3.11.1**
Entity Life History	EEM	**3.4.2**	Prototype Pathway	SP	**3.11.2**
Event/Entity Matrix	EEM	**3.4.3**	Prototype Result Log	SP	**3.11.3**
External Entity Description	DFM	**3.2.7**	Prototyping Report	SP	**3.11.4**
Feasibility Options	BSO	**3.1.4**	RDA Working Paper	RDA	**3.9.1**
Feasibility Report	BSO	**3.1.5**	Relationship Description	LDM	**3.6.7**
Function Component Implementation Map	PPS	**3.14.1**	Report Format	DD	**3.3.8**

Table A.1 — Products specified

Product	Conceptual grouping	Clause number	Product	Conceptual grouping	Clause number
Function Definition	FD	**3.5.2**	Required System Data Flow Model (see DFM)	DFM	—
Function Definitions (Set)	FD	**3.5.3**	Required System LDM (see LDM)	LDM	—
Grouped Domain Description	LDM	**3.6.4**	Requirements Catalogue	RD	**3.10.1**
I/O Description	DFM	**3.2.8**	Requirements Specification	RD	**3.10.7**
I/O Structure Description	FD	**3.5.5**	Resource Flow Diagram	DFM	**3.2.12**
I/O Structure	FD	**3.5.4**	Screen Format	DD	**3.3.9**
Selected Technical System Option	TSO	**3.12.5**	Training Requirement Descriptions	RD	**3.10.3**
SSADM Structure Diagram	MISC	**3.16.6**	Update Process Model	LDPD	**3.7.3**
System Description	TSO	**3.12.2**	User Catalogue	RD	**3.10.4**
Take-on Requirements Description	RD	**3.10.2**	User Manual Requirement Description	RD	**3.10.5**
Technical Environment Description	TSO	**3.12.3**	User Role/Function Matrix	DD	**3.3.10**
Technical System Options	TSO	**3.12.4**	User Roles	DD	**3.3.11**

Annex B (informative) Product groupings

Figure B.1 to Figure B.7 illustrate the groupings of products for each of the six Stages of SSADM. These are supplemented with product breakdown diagrams for some of the products included in the product grouping diagrams.

For each product grouping diagram the main products associated with the Stage are presented. Products are included in all the Modules in which they may be created, or updated. In some cases the product is itself composed of further products, and details of these are given in referenced product breakdown diagrams. For product grouping diagrams, each product is contained within a box, and also includes reference to the section where the full specification can be found. Boxes drawn with continuous lines refer to products which are expected to be delivered upon completion of the specific Stage; boxes drawn with dashed lines indicate those which may be regarded as interim products, developed during the course of the Stage and then used for the derivation of further products.

The diagrams do not distinguish between those products which evolve after delivery at the end of a Stage, and those which are delivered as complete.

The product grouping for the Feasibility Study Module is slightly different from the others; the main product of this Module is the Feasibility Report. The other products illustrated may not be distinguishable as separate products, but incorporated as sections within the report itself. All the delivered products, apart from the Feasibility Report itself, may be considered as evolving products should the development proceed to later Stages.

The major purpose of the diagrams contained in this annex is to indicate, to both customers and suppliers, the range and association of products for individual Stages. The products depicted in the Stage/Module groupings are those specified in section **3** of this standard. They have been grouped in this fashion in order to facilitate discussion between customers and suppliers. In this sense they are a component of the standard, but there may be information derived from these structures which is not identical with that stated in the manuals, e.g. some items included as “components” of products in their respective entries in the manual are omitted from this standard, and hence from the diagrams. This results from the extent of the descriptions in the manual, which in some cases includes details outside the range of specific SSADM products.

The product breakdown diagrams (Figure B.1 to Figure B.22) are identical with those in the manual.

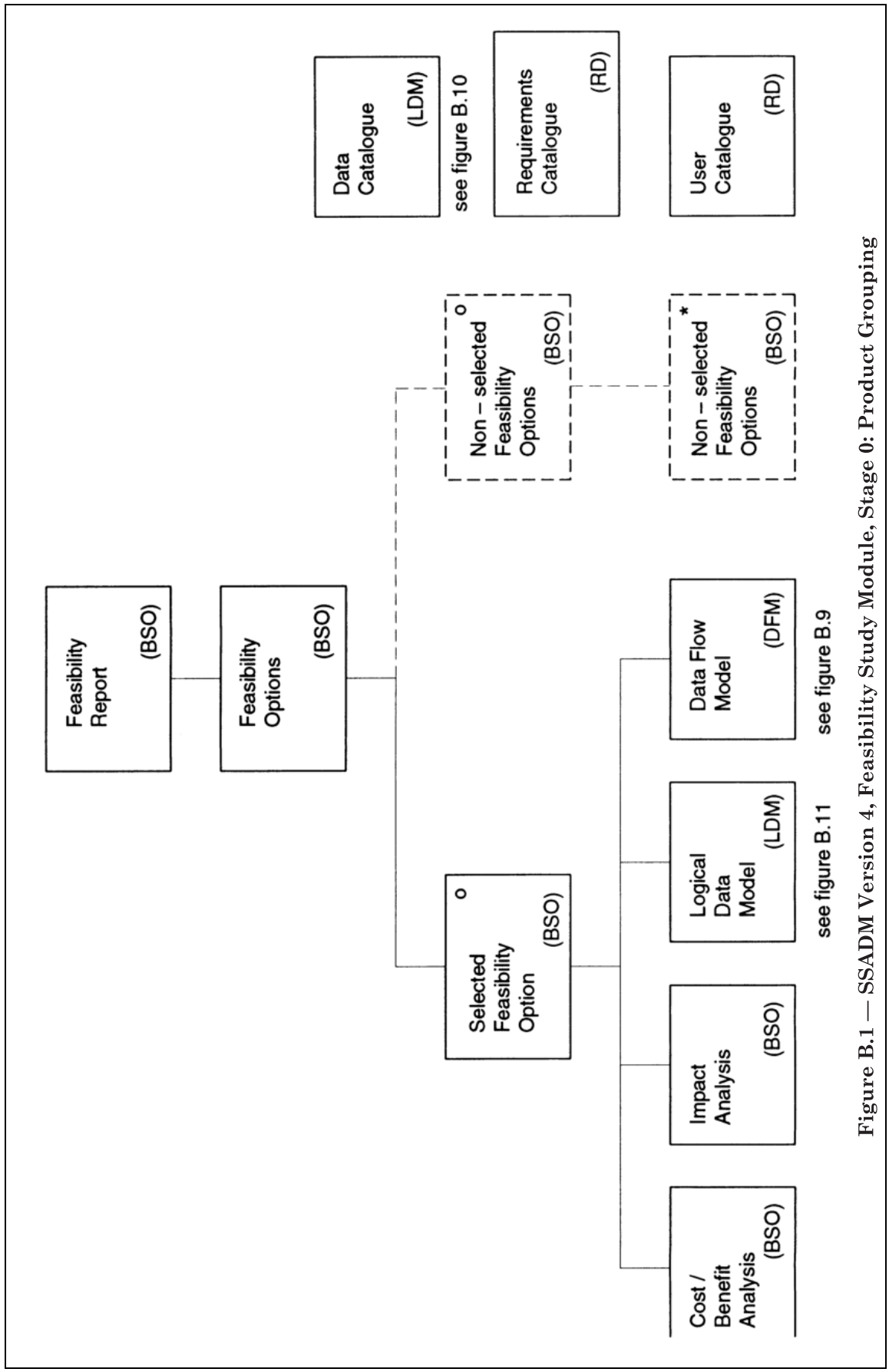


Figure B.1 — SSADM Version 4, Feasibility Study Module, Stage 0: Product Grouping

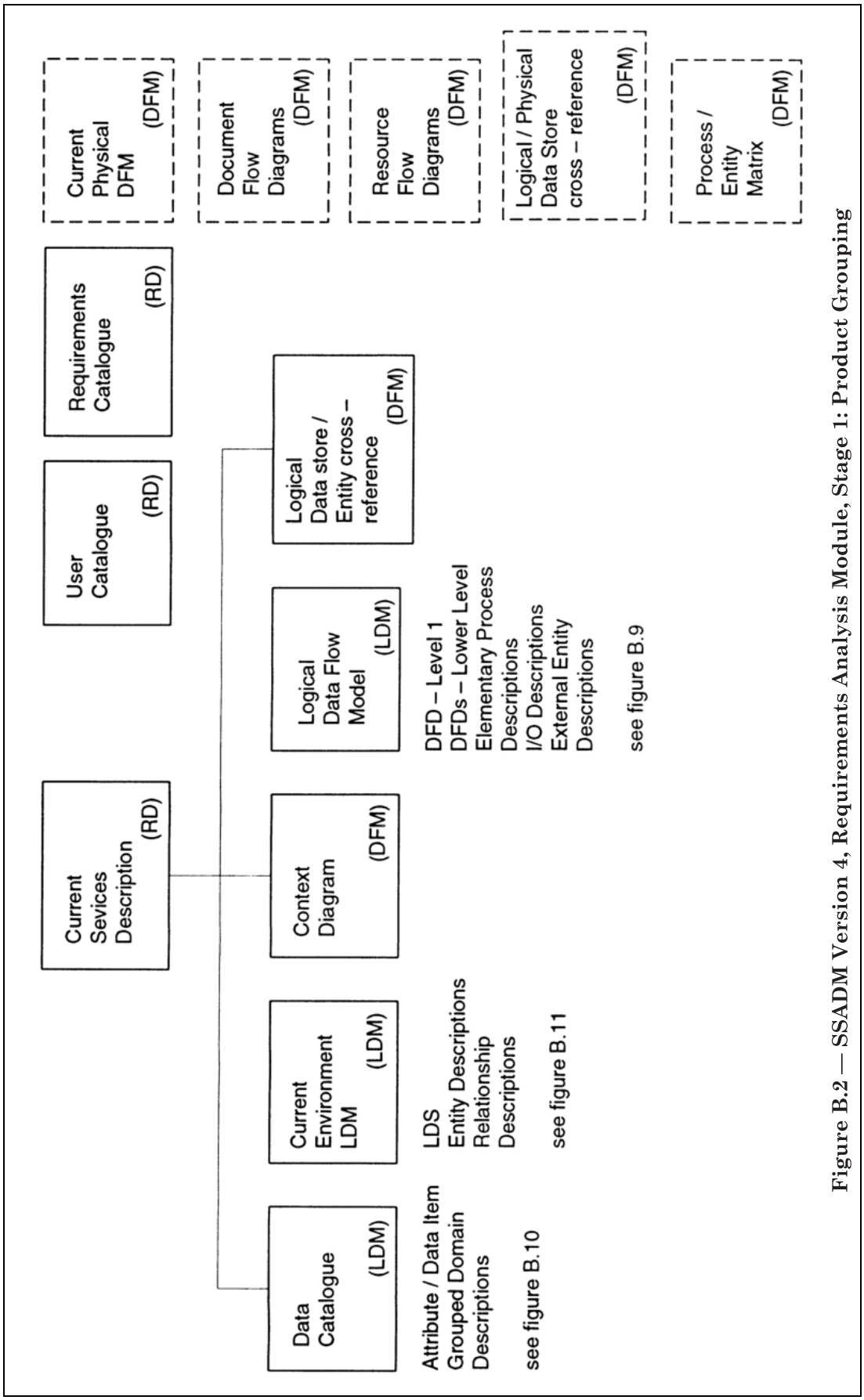


Figure B.2 — SSADM Version 4, Requirements Analysis Module, Stage 1: Product Grouping

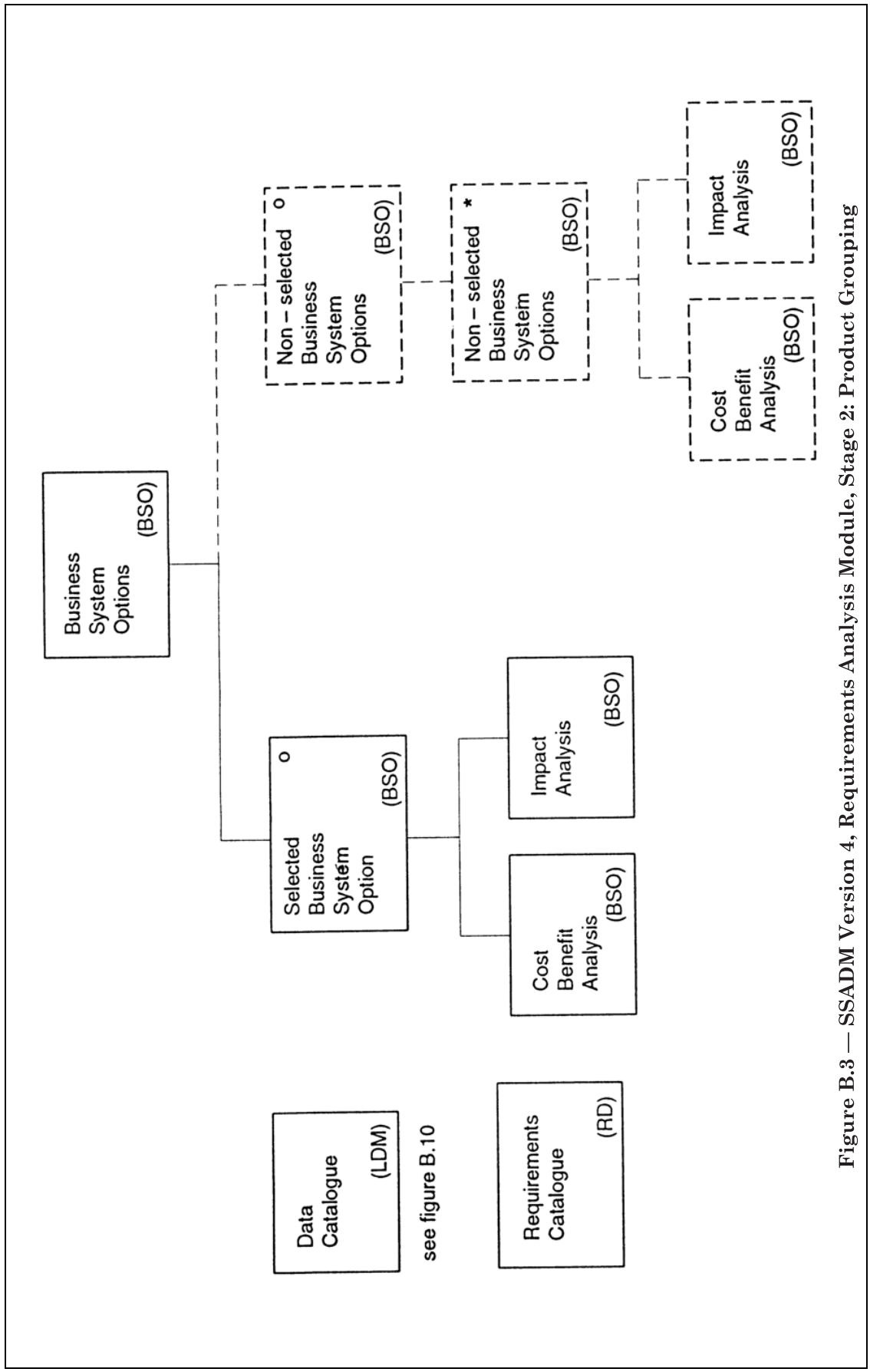


Figure B.3 — SSADM Version 4, Requirements Analysis Module, Stage 2: Product Grouping

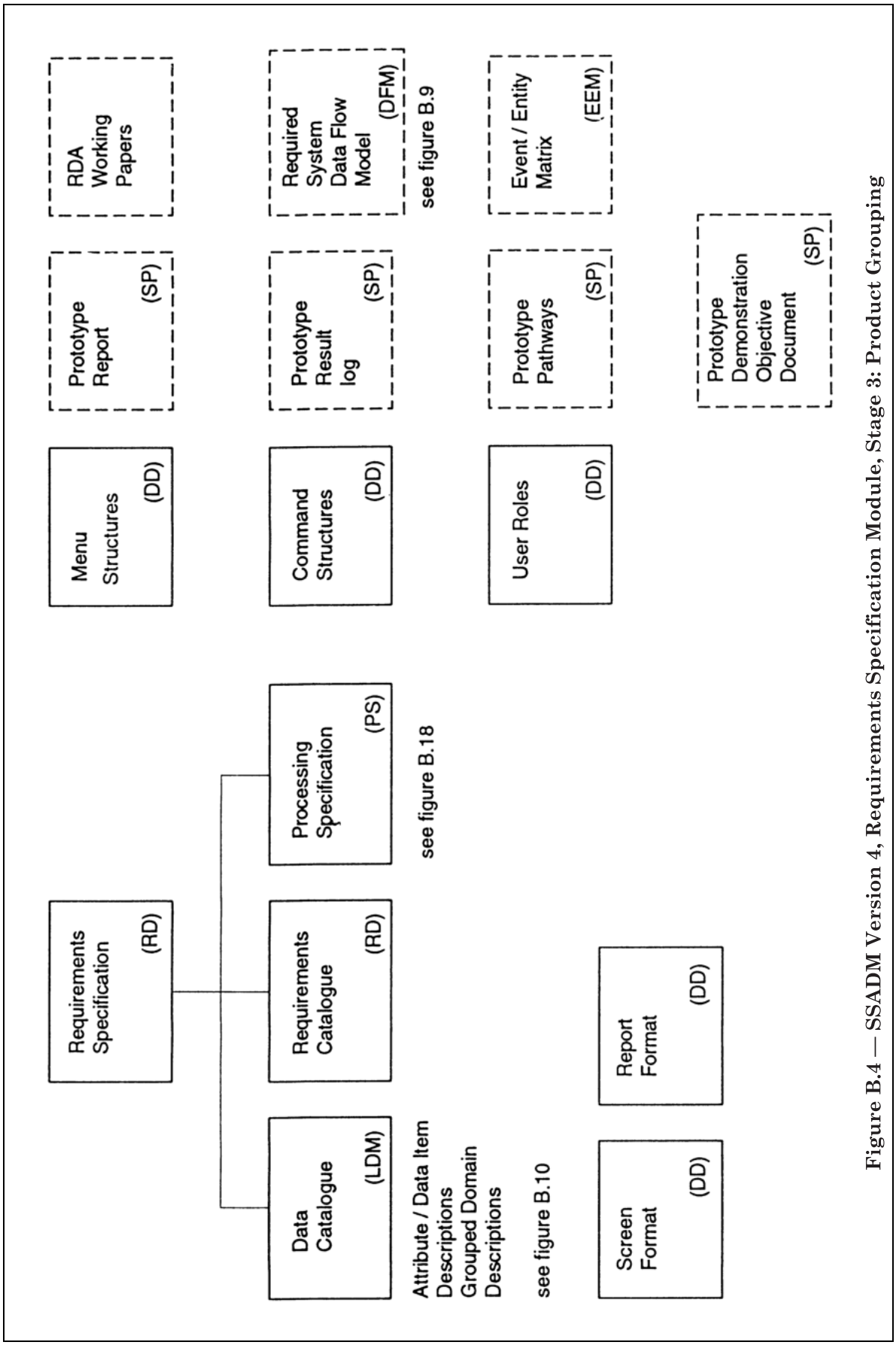


Figure B.4 — SSADM Version 4, Requirements Specification Module, Stage 3: Product Grouping

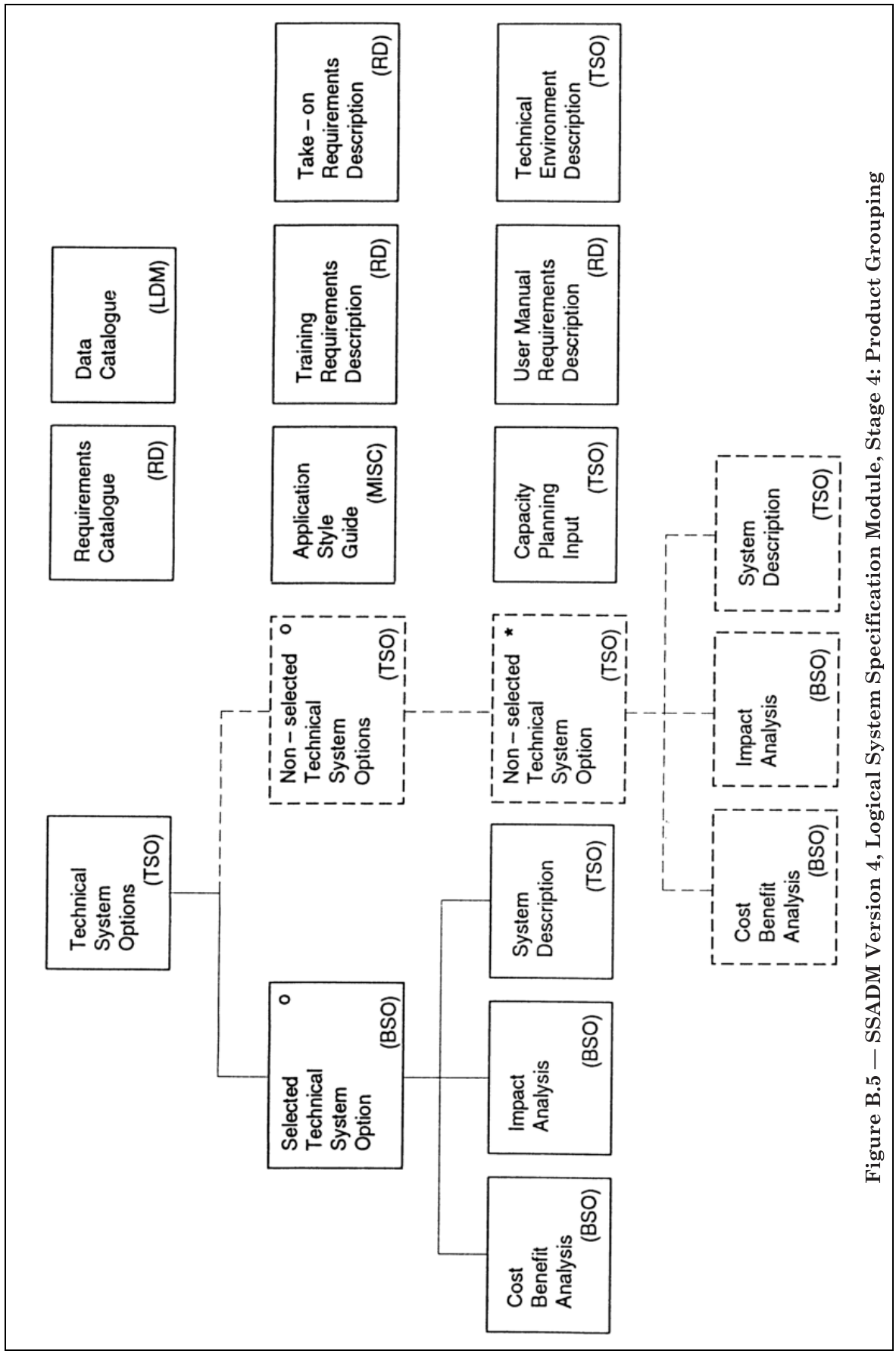


Figure B.5 — SSADM Version 4, Logical System Specification Module, Stage 4: Product Grouping

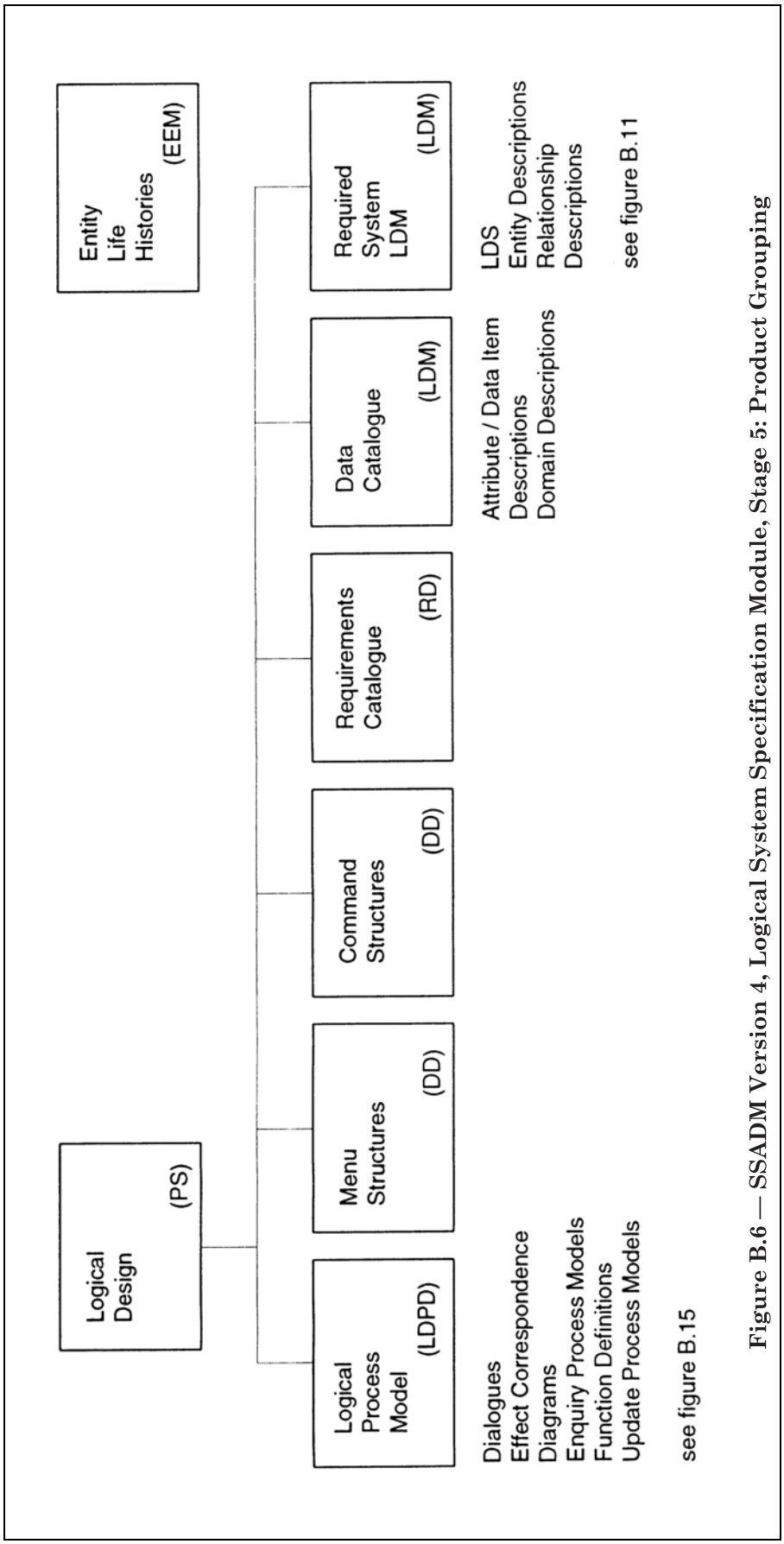


Figure B.6 — SSADM Version 4, Logical System Specification Module, Stage 5: Product Grouping

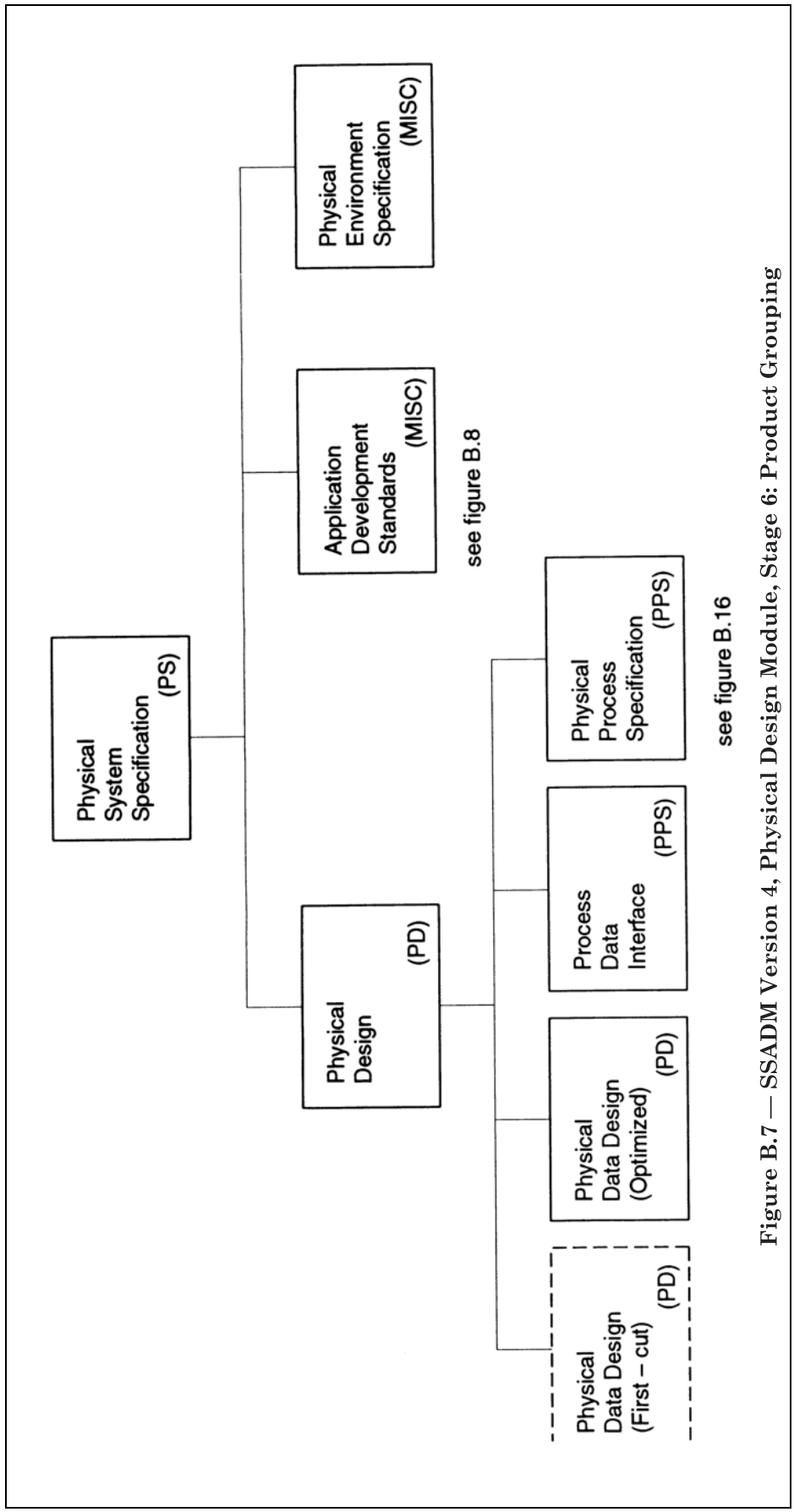


Figure B.7 — SSADM Version 4, Physical Design Module, Stage 6: Product Grouping

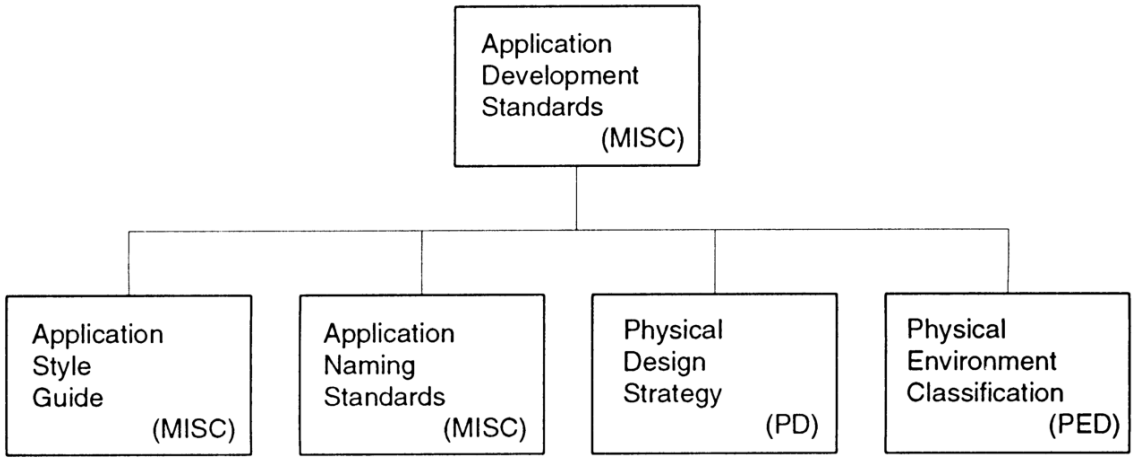


Figure B.8 — Application Development Standards

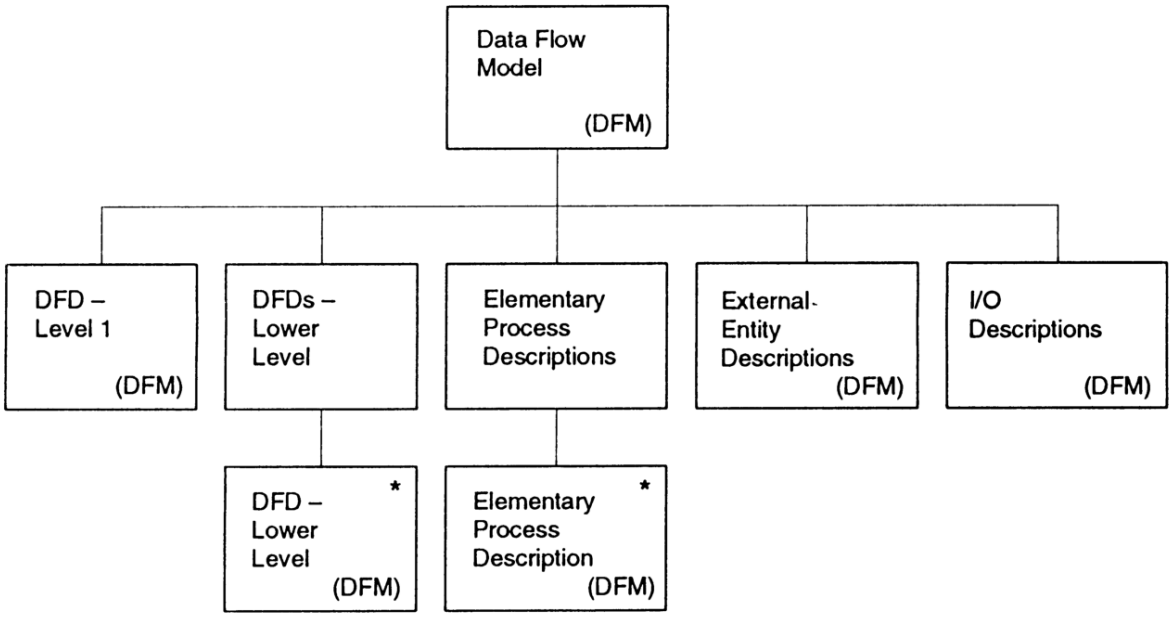


Figure B.9 — Data Flow Model

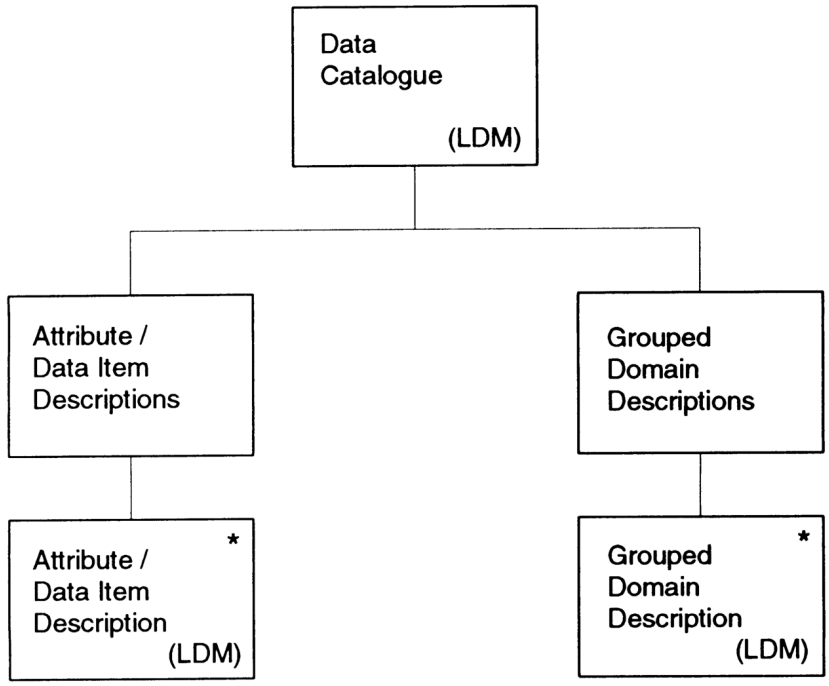


Figure B.10 — Data Catalogue

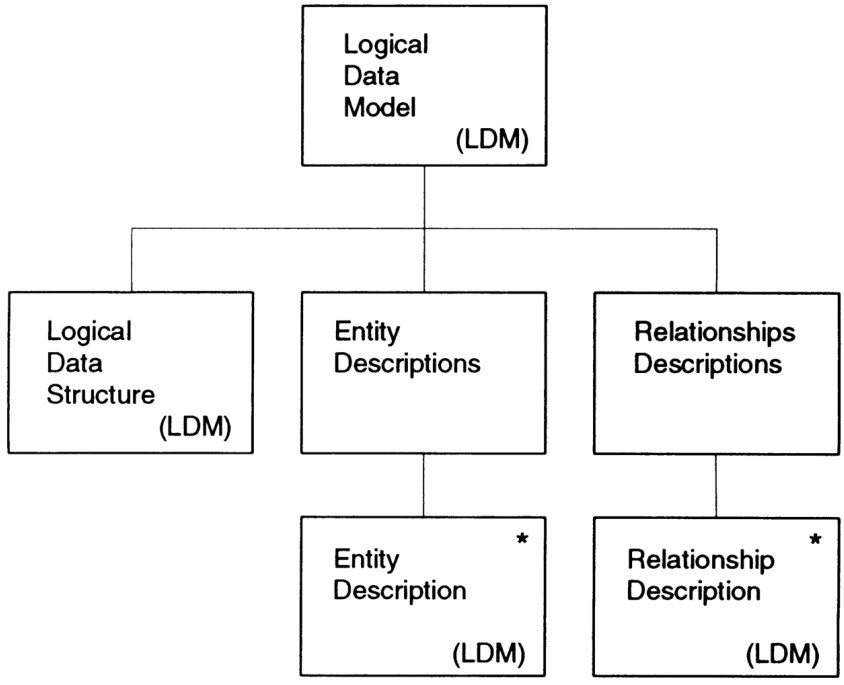


Figure B.11 — Logical Data Model

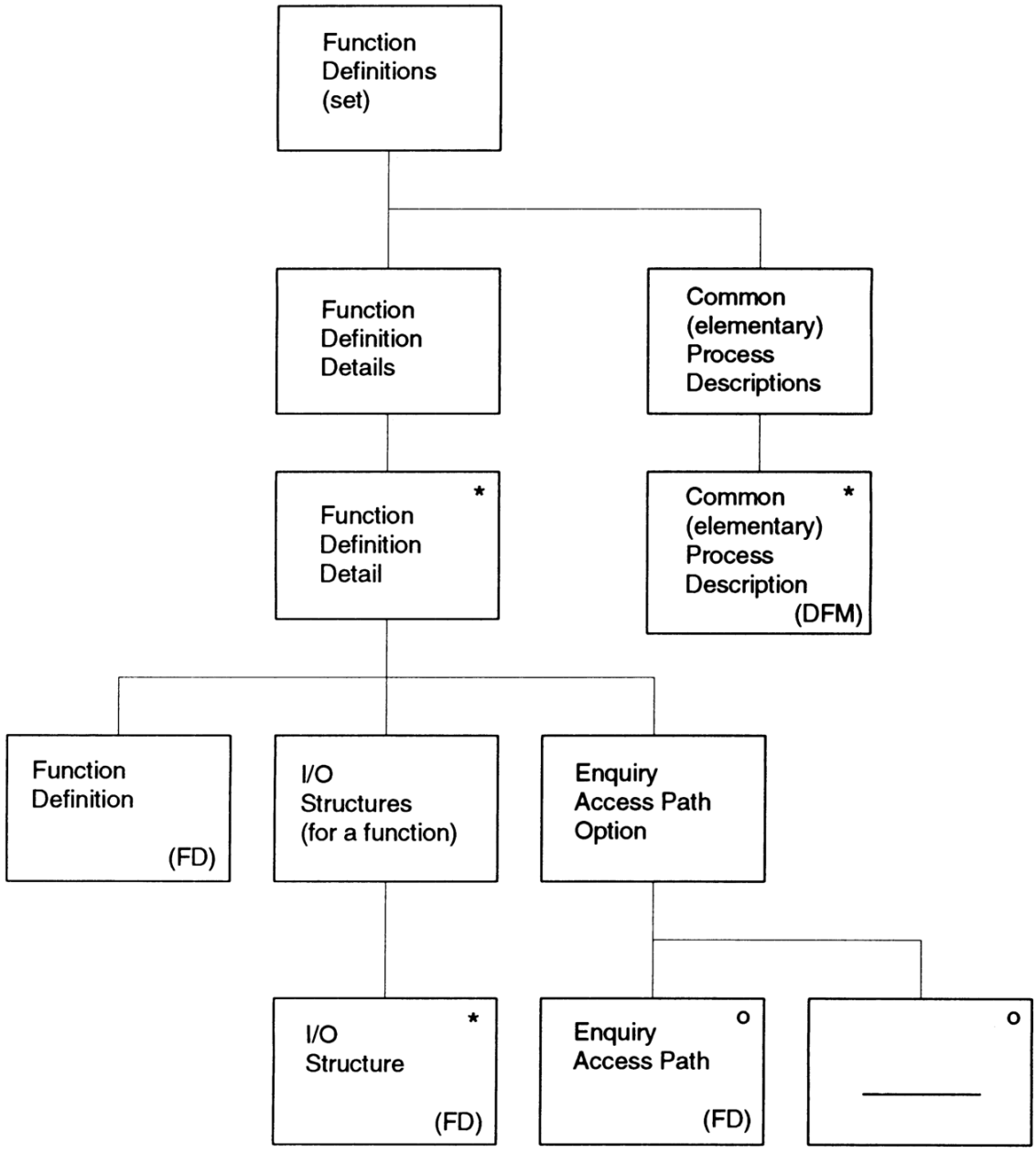


Figure B.12 — Function Definitions

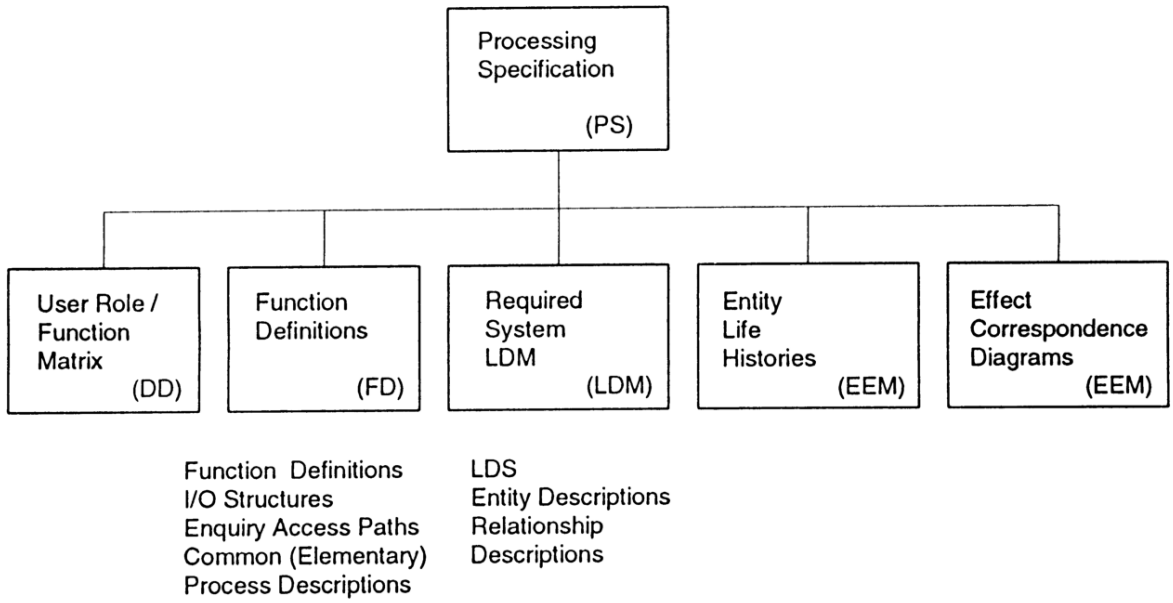


Figure B.13 — Processing Specification

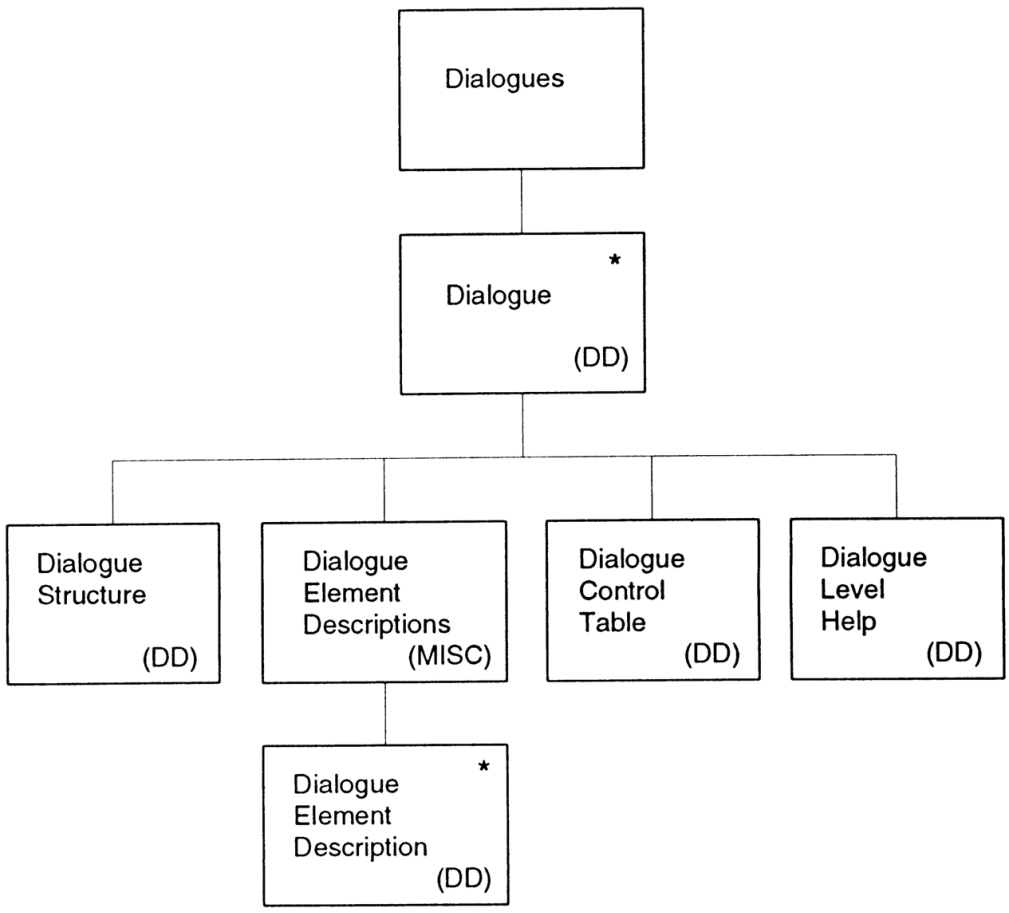


Figure B.14 — Dialogues

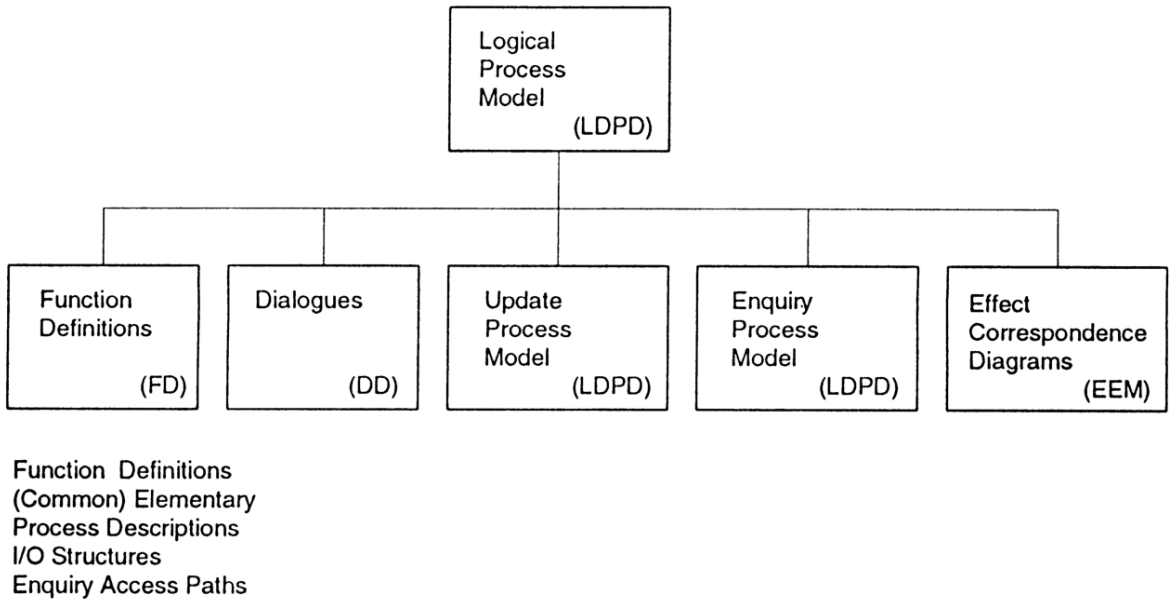


Figure B.15 — Logical Process Model

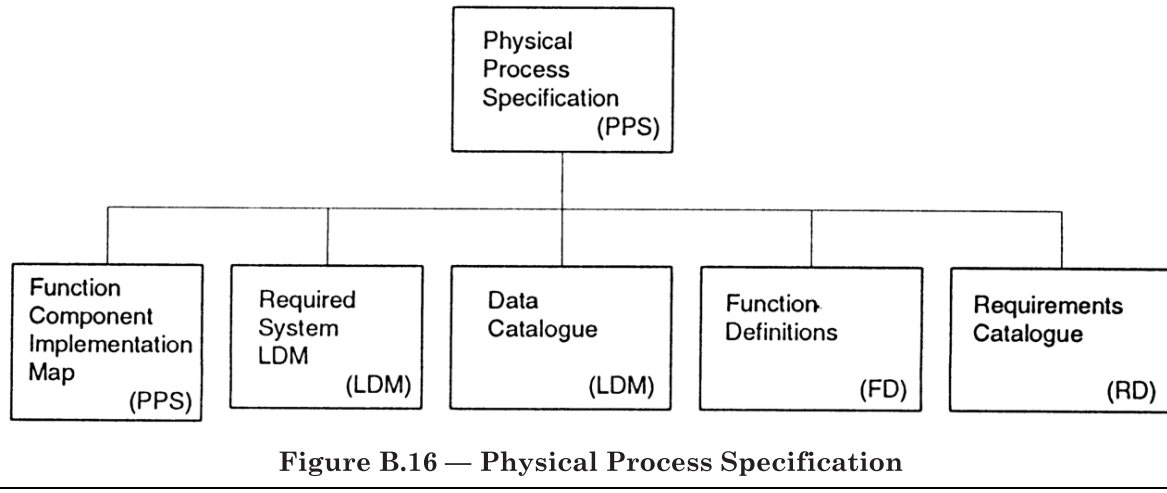


Figure B.16 — Physical Process Specification

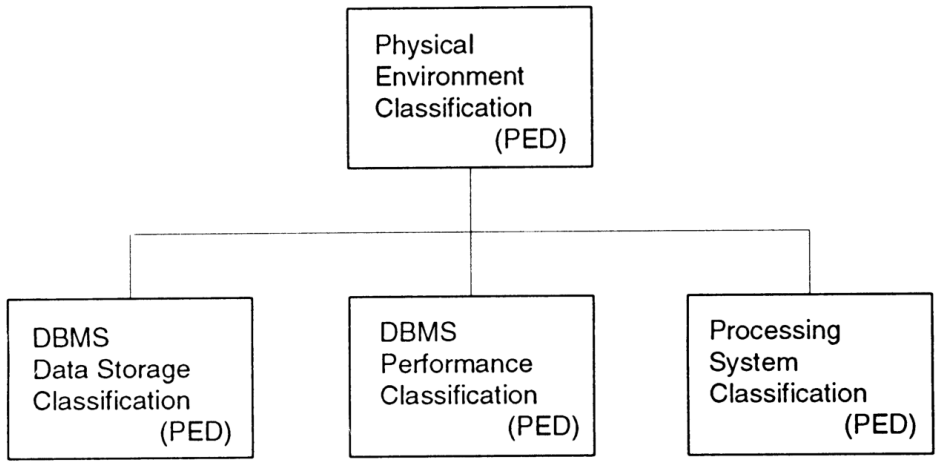


Figure B.17 — Physical Environment Classification

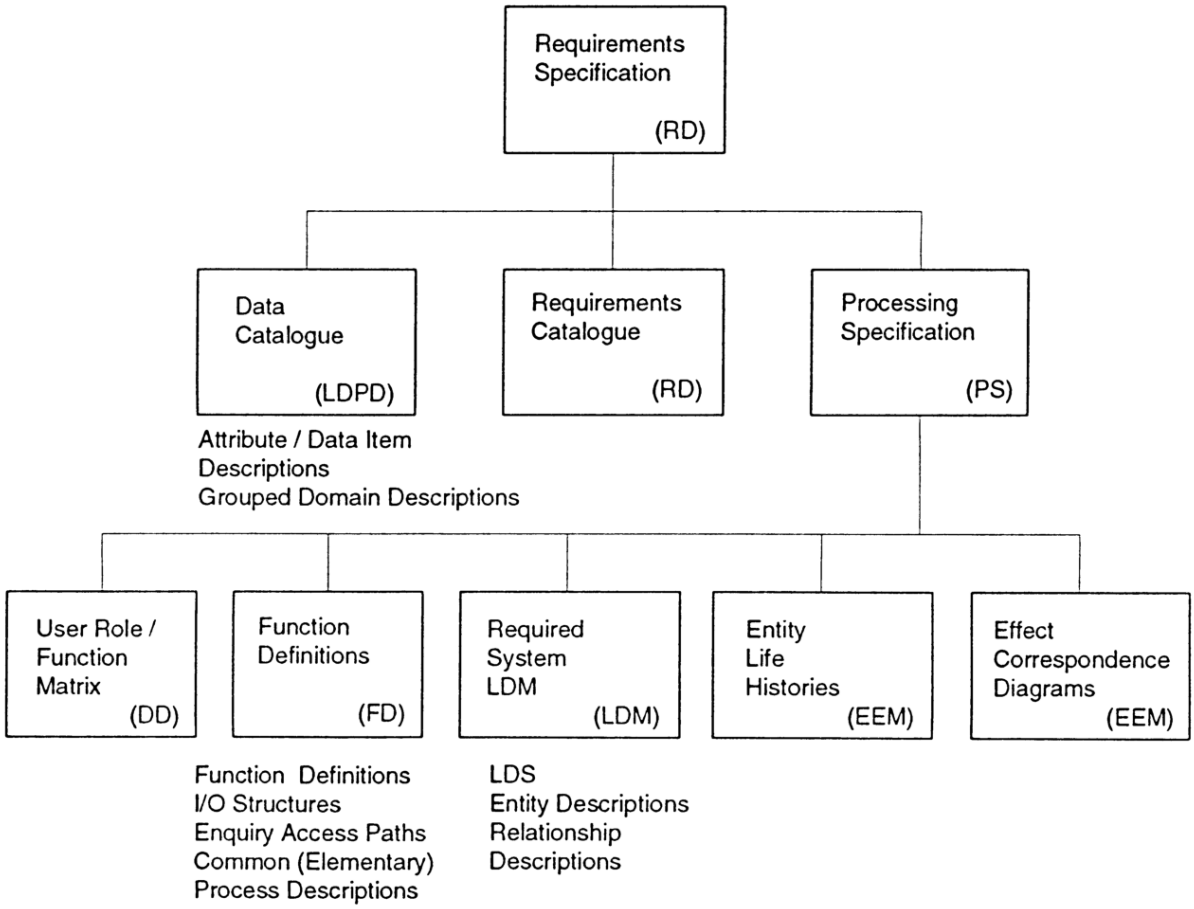


Figure B.18 — Requirements Specification

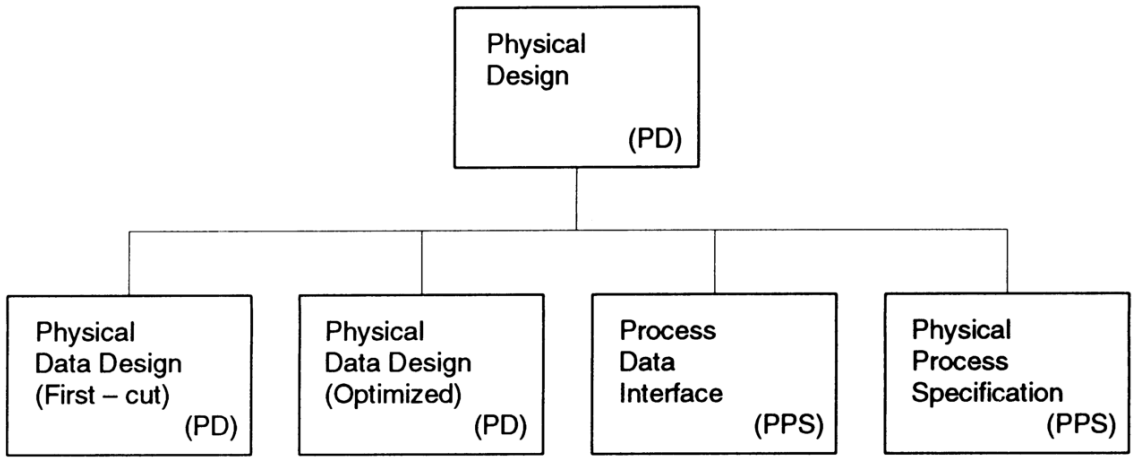


Figure B.19 — Physical Design

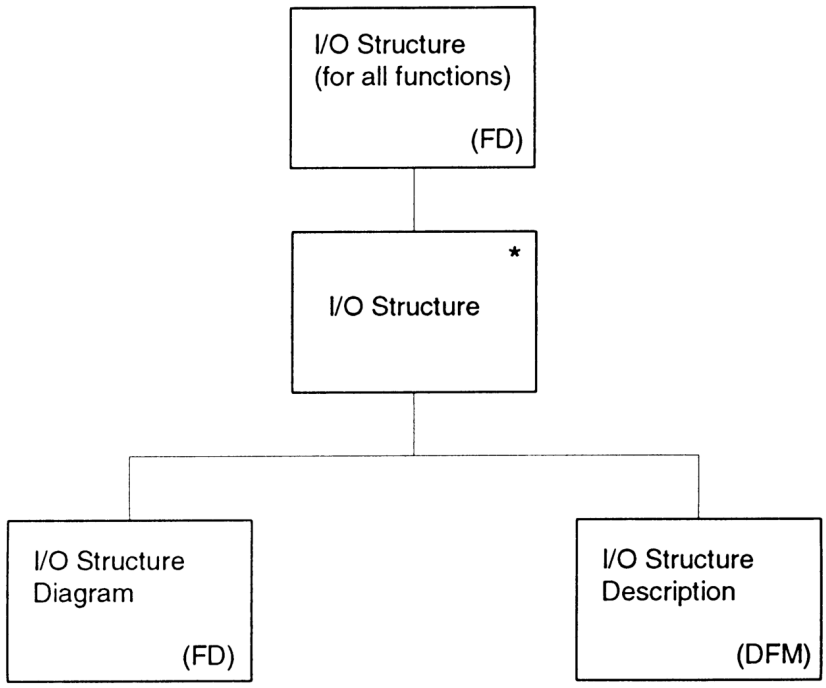


Figure B.20 — I/O Structure (for all functions)

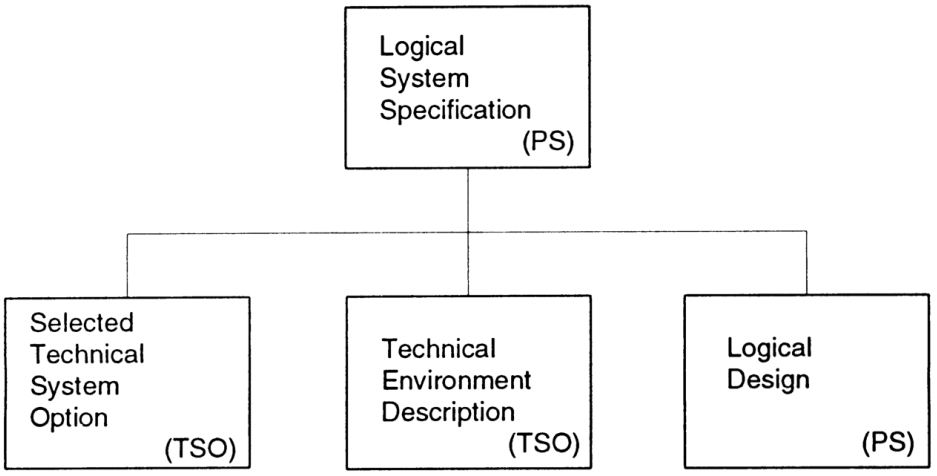


Figure B.21 — Logical System Specification

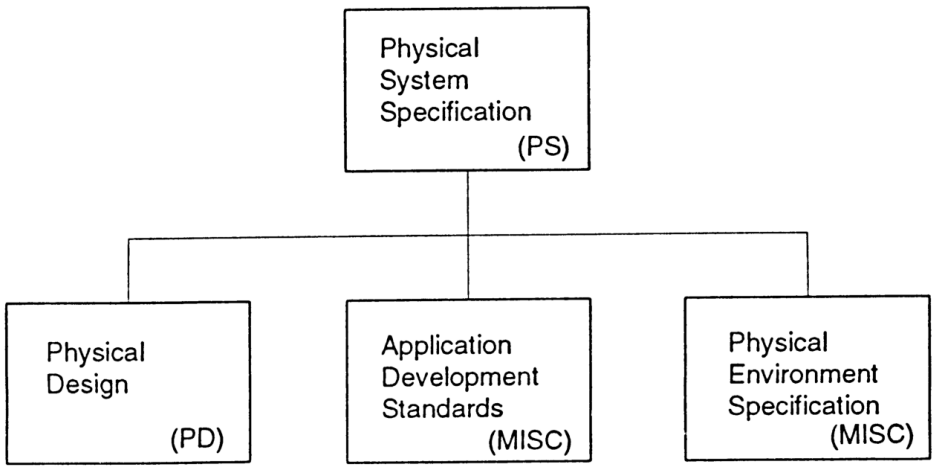


Figure B.22 — Physical System Specification

Annex C (informative) Conventions

C.1 Conventions used in 1.3

C.1.1 *Format of entries*

Each entry consists of three parts as follows:

name:	**full name for the term;**
type:	*product, object, technique, structural, model element, general;*
definition:	textual explanation of the term.

Every entry duplicates equivalent information from the glossary section of the SSADM manual [1]. It should be stressed that any divergences are not meant to preempt their resolution by the Design Authority Board. It should also be noted that not all entries in the SSADM glossary have been incorporated into this standard. Entries with (N) after them are new terms and those with (A) are amended.

C.1.2 *Conventions throughout this standard*

The following terms do use capitals:

— product names;

— Module, Stage and Step names;

— acronyms;

— the words/terms Module, Stage, Step, Project Procedures;

— Procedural Standards and Structural Standards.

The following terms do not use capitals:

— techniques;

— objects, such as enquiry function.

C.2 Conventions used in the product specifications in section 3

C.2.1 *Product groupings*

Product specifications have been grouped into conceptual headings as follows (a brief description of each of these conceptual groupings appears prior to the details for the individual products):

— Business System Options Products;

— Data Flow Modelling Products;

— Dialogue Design Products;

— Entity-Event Modelling Products;

— Function Definition Products;

— Logical Data Modelling Products;

— Logical Database Process Design Products;

— Processing Specification Products;

— Relational Data Analysis Products;

— Requirements Definition Products;

— Specification Prototyping Products;

— Technical Systems Options Products;

— Physical Data Design Products;

— Physical Processing Specification Products;

— Physical Environment Description Products;

— Miscellaneous Products;

C.2.2 *Format for product specifications*

Each product specification included in this British Standard is presented in the following format.

Name of product

Purpose. To permit full and accurate specification of a type of product. This specifies the purpose of the product or component. For clarity this may be expressed in terms of the objectives for using the product.

Composition. There are two different approaches to this element depending on the product involved as follows:

a) composition by component: showing the component structure of a product;

b) composition by product breakdown: showing that a specified product consists of a hierarchically organized set of components, each of which is itself a product. For some of these hierarchies a diagram in Annex B is included as a reference.

Completion criteria. This indicates features of the product which can be assessed to determine its adequacy and integrity. These criteria may invoke adherence to a specific notation (syntax) as well as to the relevance and meaning of the product to the specific project context (semantics). It is likely that many of these criteria can only be assessed by users' representatives in conjunction with other project personnel. In addition, many of the products are of importance in the later development stages of the information system, and have therefore to be readily available and identifiable by the relevant members of the development team. Furthermore, if products require revision and reworking, suitable configuration principles need to be established to ensure that team members are working from the same variants of products. Establishing a satisfactory and effective review and quality strategy, and configuration regime, is not addressed specifically in this standard.

In the category of miscellaneous products, some entries refer to non-SSADM products. In these instances the heading of "completion criteria" has been amended to "acceptance criteria", i.e. relating to the criteria for accepting the product as an input to the SSADM project.

NOTE 1 When amendments or updates are made to project documentation, everyone involved on the project has to use the latest version. Ensuring that this is done is a major "configuration management" problem.

NOTE 2 There are occasions when information or resources outside of the SSADM project are required for completion of particular tasks, and thus products. These could be a possible source of problems and delays which could affect the project but are outside the direct control of the SSADM project itself. Since in many cases these issues are generally applicable rather than product specific, they have not been included in the completion criteria. They may have to be considered as part of the overall project review and quality strategy; with appropriate contingency plans and strategies prepared in advance.

Status. This states whether the product is an end-stage, end-module or interim product; also whether it is delivered completed in the first instance, or evolves during further Stages as follows:

a) end stage products: products which are delivered at the end of a Stage, having been created or updated within that Stage;

b) end module products: products which are delivered at the end of a Module, having been created or updated within that Module;

c) interim products: products which are created during a Stage or Module, but which are not specifically offered as delivered products at the end of the Stage or Module itself: they are most often "in-progress" products created as part of the continuing development process, and used in the derivation of other (end-stage, end-module) products.

A further distinction is made between end-stage/end-module products which are delivered as complete at the end of the Stage or Module, and those which are delivered but expected to evolve or be the subject of further refinement in later Stages or Modules.

Presentation. This gives details of the presentation conventions to be used for graphical components of the product. It is only applicable to products with non-textual components.

COMMENTARY AND RECOMMENDATIONS ON THE ABOVE. *This comprises a brief description of the product. It also includes any general background information.*

Derivation:

This identifies the products which are used in the development of the product. In some cases the derivation may involve earlier versions of the product itself, particularly with evolving products.

Where used:

A list indicating the Stage and/or Module in which the particular product is used in the method. Entries may also indicate the specific technique with which it will be associated.

Annex D (normative) Conformity charts

D.1 Stage/Module product conformity charts

The conformity charts given in Table D.1 to Table D.23 list products by the Stage or Module in which they might be expected to be created or updated. The products are listed alphabetically, but with references to charts for associated component products listed immediately beneath them.

The charts shall be completed either by a representative of the customer, or a representative of the supplier, or both. This will clarify and confirm the expectations for both parties regarding the products to be delivered at any phase of an undertaking, the status of such products, and (if relevant) their form and presentation.

The status of a product shall be as follows.

1: end stage or end module products, delivered at the end of the stage or module; where no further refinement is anticipated except for rectification of errors or omissions.

2: end stage or end module products, delivered at the end of the stage or module; where the product is expected to evolve or be further refined in subsequent stages or modules.

3: interim products, created during a stage or module, but which are not specifically offered as delivered products at the end of the stage or module itself.

Where presentation conventions apply the letter C shall be marked in the appropriate column. Adding Y or N shall indicate the conformity of the product in this regard.

NOTE Given the inherent flexibility of SSADM version 4 [1], it is possible that additional products may be included against any Stage/Module.

Reference to a numbered section in the status column directs readers to the section for that product specification.

D.2 Product component conformity charts

For the product component charts given in Table D.8 to Table D.23 the status entry may refer to a Module or Stage, e.g. PD-1 indicates the product is delivered complete at the end of Physical Design; ST1-3,
ST3-1 indicates a product delivered as evolving in Stage 1, but complete in Stage 3.

Table D.1 — Conformity chart for Feasibility Module, Stage 0: Feasibility

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Cost/Benefit Analysis	**3.1.3**	1[a]		
Data Catalogue (see Table D.8)	**3.6.2**	2		
Data Flow Model (see Table D.9)	**3.2.4**	see **3.2.4.4**		
Feasibility Report	**3.1.5**	1		
Impact Analysis	**3.1.6**	1[a]		
Logical Data Model (see Table D.10)	**3.6.5**	2		
Non-selected Feasibility Option(s)	**3.1.4**	3		
Requirements Catalogue	**3.10.1**	see **3.10.1.4**		
Selected Feasibility Option	**3.1.4**	1		
User Catalogue	**3.10.4**	3		

[a] Delivered complete for selected option; interim product for non-selected option(s).

Table D.2 — Conformity chart for Requirements Analysis Module, Stage 1: Investigation of Current Environment

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Current Physical Data Flow Model (see Table D.9)	**3.2.4**	3		
Current Services Description (see Table D.11)	**3.10.6**	1		
Document Flow Diagrams	**3.2.5**	3		
Logical/Physical Data Store Cross-reference	**3.2.10**	3		
Process/Entity Matrix	**3.2.11**	3		
Requirements Catalogue	**3.10.1**	2		
Resource Flow Diagram	**3.2.12**	3		
User Catalogue	**3.10.4**	2		

Table D.3 — Conformity chart for Requirements Analysis Module, Stage 2: Business Systems Options

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Business Systems Options (unselected)	**3.1.1**	3		
Cost/Benefit Analysis	**3.1.3**	1[a]		
Data Catalogue (see Table D.8)	**3.6.2**	2		
Impact Analysis	**3.1.6**	1[a]		
Selected Business System Option	**3.1.2**	1		
Requirements Catalogue	**3.10.1**	2		

[a] Delivered complete for selected option; interim product for non-selected option(s).

Table D.4 — Conformity chart for Requirements Specification Module, Stage 3: Definition of Requirements

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Command Structures	**3.3.1**	1		
Data Flow Model (Required) (see Table D.9)	**3.2.4**	3		
Entity/Event Matrix	**3.4.3**	3		
Menu Structures	**3.3.7**	2		
Prototype Demonstration Objectives Document	**3.11.1**	see **3.11.1.4**		
Prototype Pathways	**3.11.2**	see **3.11.2.4**		
Prototype Result Log	**3.11.3**	see **3.11.3.4**		
Prototyping Report	**3.11.4**	1		
RDA Working Papers	**3.9.1**	3		
Report Format	**3.3.8**	1		
Requirements Specification (see Table D.12)	**3.10.7**	1		
Screen Format	**3.3.9**	1		
User Roles	**3.3.11**	1		

Table D.5 — Conformity chart for Logical System Specification Module, Stage 4: Technical System Options

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Application Style Guide	**3.16.3**	1		
Capacity Planning Input	**3.12.1**	1		
Cost/Benefit Analysis	**3.1.3**	1[a]		
Data Catalogue (see Table D.8)	**3.6.2**	2		
Impact Analysis	**3.1.6**	1[a]		
Non-Selected TSOs	**3.12.4**	3		
Requirements Catalogue	**3.10.1**	2		
Selected TSO	**3.12.5**	1		
System Description	**3.12.2**	1		
Take-on Requirements Description	**3.10.2**	1		
Technical Environment Description	**3.12.3**	1		
User Manual Requirements Description	**3.10.5**	1		

[a] Delivered complete for selected option; interim product for non-selected option(s).

Table D.6 — Conformity chart for Logical System Specification Module, Stage 5: Logical Design

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Entity Life Histories	**3.4.2**	1		C
Logical Design (see Table D.15)		1		

Table D.7 — Conformity chart for Physical Design Module, Stage 6: Physical Design

Product name	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Physical System Specification (see Table D.17)		1		

Table D.8 — Conformity chart for Data Catalogue

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Attribute/Data Item Description	**3.6.1**	[a]		
Grouped Domain Descriptions	**3.6.4**	[a]		

[a] 2 for all Stages, except 1 for Stage 6.

Table D.9 — Conformity chart for Data Flow Model

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
DFD Level 1	**3.2.2**	see **3.2.4.4**		C
Lower Level DFDs	**3.2.3**	see **3.2.3.4**		C
Elementary Process Descriptions	**3.2.6**	2 Stage 1 1 Stage 3		
External Entity Descriptions	**3.2.7**	see **3.2.7.4**		
I/O Descriptions	**3.2.8**	see **3.2.8.4**		

Table D.10 — Conformity chart for Logical Data Model

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Logical Data Structure	**3.6.6**	see **3.6.6.4**		C
Entity Descriptions	**3.6.3**	[a]		
Relationship Descriptions	**3.6.7**	see **3.6.7.4**		
[a] 2 for all Stages, except 1 for Stage 6.				

Table D.11 — Conformity chart for Current Services Description

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Context Diagram	**3.2.1**	2 Stage 0 1 Stage 1		C
Current Environment Logical Data Model (see Table D.10)	**3.6.5**	1		
Data Catalogue (see Table D.8)	**3.6.2**	2		
Data Flow Model (logical) (see Table D.9)	**3.2.4**	1		
Logical Data Store/Entity Cross-reference	**3.2.9**	see **3.2.9.4**		

Table D.12 — Conformity chart for Requirements Specification

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Data Catalogue (see Table D.8)	**3.6.2**			
Processing Specification (see Table D.13)	**3.8.1**	1		
Requirements Catalogue	**3.10.1**			

Table D.13 — Conformity chart for Processing Specification

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Effective Correspondence Diagrams	**3.4.1**	1		C
Entity Life Histories	**3.4.2**	1[a]		C
Function Definitions (set) (see Table D.14)	**3.5.3**	2 Stage 3 1 Stage 6		
Logical Data Model (required system) (see Table D.10)	**3.6.5**	see **3.6.5.4**		
User Role/Function Matrix	**3.3.10**	1		

[a] These are delivered complete at the end of RS, except for the addition of state indicators added in Stage 5.

Table D.14 — Conformity chart for Function Definition (set)

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Common (Elementary) Process Descriptions	**3.2.6**	1		
Enquiry Access Paths	**3.5.1**	1		C
Function Definitions	**3.5.2**	2 Stage 3 1 Stage 6		
I/O Structures (see Table D.23)	**3.5.4**	1		

Table D.15 — Conformity chart for Logical Design

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Command Structures	**3.3.1**	2 Stage 3 1 Stage 5		
Data Catalogue (see Table D.8)	**3.6.2**	2		
Logical Data Model (required system) (see Table D.10)	**3.6.5**	see **3.6.5.4**		
Logical Process Model (see Table D.16)	**3.7.2**	1		
Menu Structures	**3.3.7**	see **3.3.7.4**		C
Requirements Catalogue	**3.10.1**	2		

Table D.16 — Conformity chart for Logical Process Model

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Dialogues (see Table D.22)	**3.3.5**	1		
Effect Correspondence Diagrams	**3.4.1**	1		C
Enquiry Process Models	**3.7.1**	1		C
Function Definitions	**3.5.3**	1		
Update Process Models	**3.7.3**	1		C

Table D.17 — Conformity chart for Physical System Specification

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Application Development Standards (see Table D.18)	**3.16.1**	1		
Physical Design (see Table D.19)	**3.13.2**	1		
Physical Environment Specification	**3.16.7**	1		

Table D.18 — Conformity chart for Application Development Standards

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Application Naming Standards	**3.16.2**	1		
Application Style Guide	**3.16.2**	1		
Physical Design Strategy	**3.13.2**	1		
Physical Environment Classification (see Table D.21)		1		

Table D.19 — Conformity chart for Physical Design

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Physical Data Design (1st cut/optimized)	**3.13.1**	1		
Physical Process Specification	**3.14.3**	1		
Process Data Interface	**3.14.2**	1		

Table D.20 — Conformity chart for Physical Process Specification

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Data Catalogue (see Table D.8)		see **3.6.2.4**		
Function Component Implementation Map	**3.14.1**	1		
Function Definitions (set) (see Table D.14)	**3.5.2**	1		
Logical Data Model (required system) (see Table D.10)	**3.6.5**	1		
Requirements Catalogue	**3.10.1**	1		

Table D.21 — Conformity chart for Physical Environment Classification

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
DBMS Data Storage Classification	**3.15.1**	1		
DBMS Performance Classification	**3.15.2**	1		
Processing System Classification	**3.15.4**	1		

Table D.22 — Conformity chart for Dialogues

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
Dialogue Control Table	**3.3.2**	1		
Dialogue Element Description	**3.3.3**	1		
Dialogue Level Help	**3.3.4**	1		
Dialogue Structure	**3.3.6**	1		C

Table D.23 — Conformity chart for I/O Structures

Product name (components)	Reference	Status (1, 2, 3)	Conformity (A, B)	Presentation convention (Y/N)
I/O Structure	**3.5.4**	1		
I/O Structure Description(s)	**3.5.5**	1		
I/O Structure Diagram(s)	**3.5.6**	1		C

List of references (see 1.2)

Normative references

[1] *SSADM Version 4 reference manual* (produced by CCTA). Oxford: NCC Blackwell Ltd. 1990 ISBN 1-85554-004-5.

Informative references

BSI publications

BRITISH STANDARDS INSTITUTION, London

BS 0, *A standard for standards.*

BS 0-3:1991, *Guide to drafting and presentation of British Standards.*

BS 4778, *Quality vocabulary.*

BS 4778-1:1987, *International terms.*

BS 5750, *Quality systems.*

BS 5750-0, *Principal concepts and applications.*

BS 5750-0.1:1987, *Guide to selection and use.*

BS 5750-0.2:1987, *Guide to quality management and quality system elements.*

BS 5750-1:1987, *Specification for design/development, production, installation and servicing.*

BS 5750-2:1987, *Specification for production and installation.*

BS 5750-3:1987, *Specification for final inspection and test.*

BS 5750-4:1990, *Guide to the use of BS 5750-1 "Specification for design/development, production, installation and servicing", Part 2 "Specification for production and installation" and Part 3 "Specification for final inspection and test".*

BS 5750-8:1991, *Guide to quality management and quality systems elements for services.*

BS 5750-13:1991, *Guide to the application of BS 5750-1 to the development, supply and maintenance of software.*

BS 7000, *Design management systems.*

BS 7000-1:1989, *Guide to managing product design.*

BSI — British Standards Institution

BSI is the independent national body responsible for preparing British Standards. It presents the UK view on standards in Europe and at the international level. It is incorporated by Royal Charter.

Revisions

British Standards are updated by amendment or revision. Users of British Standards should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using this British Standard would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover. Tel: 020 8996 9000. Fax: 020 8996 7400.

BSI offers members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of standards.

Buying standards

Orders for all BSI, international and foreign standards publications should be addressed to Customer Services. Tel: 020 8996 9001. Fax: 020 8996 7001.

In response to orders for international standards, it is BSI policy to supply the BSI implementation of those that have been published as British Standards, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Library and its Technical Help to Exporters Service. Various BSI electronic information services are also available which give details on all its products and services. Contact the Information Centre. Tel: 020 8996 7111. Fax: 020 8996 7048.

Subscribing members of BSI are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration. Tel: 020 8996 7002. Fax: 020 8996 7001.

Copyright

BSI
389 Chiswick High Road
London
W4 4AL