

# **DFAS Analysis Standards and Guidelines**

**August 15, 1999**

# DFAS Analysis Standards and Guidelines

## Executive Summary

The purpose of the DFAS Analysis Standards is to provide analysis specifications for DFAS projects that will be using Designer/2000. These standards follow the format of the Designer 2.1 Repository.

Each of the repository entry's properties is defined as follows:

- Mandatory - Designer will not allow the repository entry to be saved unless that property is filled
- Required - this property should be defined to adequately model the application and be ready for transformation to a design model
- Not Required - this property need not be filled

# **1. Application System Properties - Each Application must have a Separate Repository System Defined**

## **1.1. Name is mandatory**

The repository contains several types of applications. It contains the latest Data Model (PUB) maintained by Headquarters and created by numerous applications such as SGL, CEFT, GETBACC, DIFS-R, DPPS, DCAS and DDRS. All applications share common data from the latest PUB. The repository also contains application work areas such as SGL\_9902\_WORK or REUSE\_WRK\_9901. In addition to work areas some applications are targeted for prototype efforts such as SGL\_9902\_PROT. Also included in the repository are applications designated as Non Standard Area (NSA). For more information refer to the *DCII Repository Object Creation/Change/Promotion Processes Version 1.0*

## **1.2. Parent is not required**

## **1.3. Title is required**

## **1.4. Authority is required**

The name of the person or organization authorizing this applications system. This field has a limit of 10 characters.

## **1.5. Owner is required**

The application owner is always CASEOWN. Applications are created via requests. The Repository Manager is the only user of Oracle Designer that can create an application. The Repository Manager will grant other repository users access to the application system. For more information on application creation see the *DCII Repository Object Creation/Change/Promotion Process Version 1.0*

## **1.6. Datawarehouse ?is mandatory**

Indicates whether or not this is a data warehouse application. The value Y assigns an additional, data warehouse specific property to some element types, for example, entity and relational table definitions.

## **1.7. Priorities is not required**

## **1.8. Constraints are not required**

## **1.9. Summary is not required**

## **1.10. Objectives are not required**

## **1.11. Description is required**

Because CASEOWN creates all applications, CASEOWN must populate the Application Properties including the Description. Application information can be sent using MSWORD to the Repository Manager. This information must be provided to the Repository Manager when the application is requested for creation. The Application Description includes the high level inputs, processing and outputs of the application being defined. A high level understanding of the interfaces for the application in context with the DCD, DCW and Non Standard areas must be included. If this information is not available, this should be stated in the notes property of the Application Definition. This information is essential for entry criteria into the analysis phase of a project. The application description should also state who the users of the system are.

## **1.12. Notes are not required**

**1.13. For the Application Definition run Repository Reports/Global/Elements and Their Application Systems Element Type: Application System**

**2. Reference Data Definition**

**2.1. Business Terminology - Will be used to store acronyms**

This object is used to store acronyms and other business terminology such as that found in the Defense Finance and Accounting Data Model (DFADM) and the Defense Finance and Accounting Process Model (DFAPM). It can also be used to define business terminology or terms that are used in an application system. A comprehensive system glossary can be produced containing these commonly used terms, together with any synonyms and any entities within that application system. The glossary can then be circulated to and agreed by everyone involved in the development project. The MS Help Generator uses the Business Terminology descriptions to generate a Glossary for the Windows Help file.

**2.1.1. Name is mandatory**

**2.1.2. Comments are not required**

**2.1.3. Description is required**

**2.1.4. Notes are not required**

**2.1.5. Some Business Terminology has Synonyms**

2.1.5.1. Name is mandatory

2.1.5.2. Description is not required

2.1.5.3. Notes are not required

**2.1.6. For Business Terminology information run Repository Reports/Global/Elements and Their Application Systems Element Type: Business Terminology**

**2.2. Documents - Will be used to store documents that contain requirements**

Various types of documents may be produced during application development, for example:

- DFAS Application Architecture
- Operational Requirements Document (ORD)
- Delivery and Acceptance Plan
- Test Plans
- Test Scripts
- Capacity Plans
- Data Migration Plan
- Business Mission Statement
- Business Vision Statement
- Business Goals
- Business Objectives for each Goal

The information that can be specified includes the document name, the location of the file that contains the document, the name of the author, the creation date, the document version number and the current status of the document.

**2.2.1. Name is mandatory**

**2.2.2. Author is mandatory**

**2.2.3. Type is required**

**2.2.4. Status is required**

**2.2.5. Format is required**

**2.2.6. Source Path is required**

**2.2.7. Ref is not required**

**2.2.8. Version is mandatory**

**2.2.9. Date Document Changed is required**

**2.2.10. Date Document Changed is required**

**2.2.11. Changed By is required**

**2.2.12. Authorization Date is not required**

**2.2.13. Authorized By is required**

**2.2.14. Comment is not required**

**2.2.15. Document Text is required**

**2.2.16. Description is required**

**2.2.17. Notes are not required**

**2.2.18. For Business Terminology information run Repository Reports/Global/Elements and Their Application Systems Element Type: Business Terminology**

### **2.3. Domains - Will be used to store common attribute information**

A domain defines a set of validation rules, format constraints and other properties that apply to a group of entity attributes, columns, Oracle object type attributes, program data constructs, module arguments or datastructure items. If you make a change to a domain, you can propagate the updates to the associated columns/attributes using the Update Columns/Attributes in a Domain utility, available from the Utilities menu. A Baseline Data Model is maintained by DFAS Headquarters Data Management. Domains are defined for almost all DDDS attribute names. All domains in the DCD may be considered global. Some application may need to create "application specific domains". Use DDDS standard naming conventions as provided by DFAS HQ/S for the following domains:

- date/ month/ year/ time attributes
- code/mnemonic/abbreviation attributes
- amount attributes
- currency attributes
- change history attributes

- description attributes
- indicator attributes
- status attributes
- numeric attributes

If they have not been previously defined, then an SCR must be generated.

Criteria for Defining Domains:

- Any attribute that ranges over a fixed set of predefined values that is less than 21 values should be associated with a static domain that describes that set of values.
- Any attribute that ranges over a dynamic set of values that is less than 21 values should be associated with a dynamic domain that describes that set of values.
- Any attribute that ranges over 20 values should be placed in a reference entity.

**2.3.1. Name is mandatory**

Example of standard domains currently maintained by HQ/S include: TY CD, EFF BEG DATE, ASGN ID, etc.

**2.3.2. Subset Of is required**

**2.3.3. Authority is not required**

**2.3.4. Format is mandatory**

**2.3.5. Average Att Length is required**

**2.3.6. Maximum Att Length is required**

**2.3.7. Att Decimal Places is required if the format is numeric**

**2.3.8. Unit Of Measure is not required**

**2.3.9. Derivation is not required**

**2.3.10. Validation Rules - not mandatory - used for testing and implementing**

**2.3.11. Datatype is required**

**2.3.12. Average Col Length is required**

**2.3.13. Maximum Col Length is required**

**2.3.14. Col Decimal Places is required if the datatype is numeric**

**2.3.15. Dynamic List ? - Mandatory - When set to Yes and the display datatype for a column is Poplist or Text List, the list of allowable values associated with the domain will be dynamic (i.e. based on a SELECT from the CG\_REF\_CODES table) rather than hard-coded in the client application.**

**2.3.16. Default is not required**

**2.3.17. Null Value is not required**

**2.3.18. Comment is not required**

**2.3.19. Description is required**

**2.3.20. Notes are not required**

**2.3.21. Some Domains have Allowable Values**

2.3.21.1. Values is mandatory

2.3.21.2. High Value is required if Allowable Value is a range

2.3.21.3. Abbreviation is not required

2.3.21.4. Meaning is required

2.3.21.5. Display Sequence is required

2.3.21.6. Description is required

2.3.21.7. Notes are not required

2.3.21.8. Some Domains have Sub Domains - standards are the same as for Domains

**2.3.22. For information on Domains run Repository Reports/Entity Relationship Modeling - Attributes in a Domain Report and Entity Definition Report From Repository Reports/Global - Domain Definition Report**

**3. Enterprise Modeling**

**3.1. Business Units - Defines the business units of an organization, and also records volumes of entities used at business units and the frequencies of business functions carried out at the business unit. - Several Business Units should be defined.**

The Application Design Transformer generates the first level of menu modules from the module business unit usages (derived from the function business unit usages). For more information refer to the *DFAS Oracle Designer Transformer Standards and Guidelines*. You can associate more than one Organization Unit with a process using the Repository Object Navigator (Business Function node, Usages: Performed by Business Units Subnode). The Process Modeler allows only one organizational unit per process, and this assignment is stored in a property called Single BPR Business Unit, which is a read-only property. The value of this property will be Yes if you assigned the process to an organizational unit on the Process Modeller. If you perform the assignment in the Repository Object Navigator, the value will be No.

The Matrix Digrammer includes critical strategy information associated with Business Units. This information should be made available for review during the entire life cycle. The matrices are:

Business Units to Objectives - Records the association between a business unit and the objective specifically set for that business unit. Typically, this is useful to those performing business planning exercises.

Business Units to Problems - Identifies which of the problems encountered by the enterprise are actually experienced by a specific business unit.

**3.1.1. Short Name is mandatory**

**3.1.2. Name is mandatory**

Examples of names of currently defined business units include:

- BACC DATA STEWARD
- BACC PROJECT OFFICER
- DCD PROJECT OFFICE
- NON STANDARD DCD PROJECT OFFICE
- SGL MAINTENANCE GROUP
- DCD DENVER GENERAL FUNDS OFFICE
- PRO FORMA ADMINSTRATOR

**3.1.3. Parent is required for non-root Business Units**

**3.1.4. Sequence in Parent is required for non-root Business Units**

**3.1.5. Comment is not required**

**3.1.6. Description is required**

**3.1.7. Notes are not required**

**3.1.8. Primary Location is not required**

**3.1.9. Primary Contact is not required**

**3.1.10. Headcount is not required**

**3.1.11. Role Cost Rate is not required**

**3.1.12. Role Cost Rate Unit is required if a Role Cost Rate is specified**

**3.1.13. Role Responsibilities is not required**

**3.1.14. Role Skill 1 is not required**

**3.1.15. Role Skill 2 is not required**

**3.1.16. Role Skill 3 is not required**

**3.1.17. Role Skill 4 is not required**

**3.1.18. Role ? is mandatory**

**3.1.19. Icon File is not required**

**3.1.20. Image File is not required**

**3.1.21. Sound File is not required**

**3.1.22. Video File is not required**

**3.1.23. Execution String is not required**

**3.1.24. Note Type is not required**

**3.1.25. Note Text is not required**

**3.1.26. Note Picture File is not required**

**3.1.27. Business Units may be associated with Objectives**

**3.1.28. Business Units may be associated with Problems**

**3.1.29. For information on Business Units run Repository Reports/Function Event Modeling - Function Definition Reports and from Repository Reports/Global - Elements and their Application Systems: element type - business units**

**3.2. Locations - If the Locations of Business Units affect their Business Functions, Locations should be defined .**

The location object records information about the geographical location at which business functions are performed. As part of Analysis standards location information should be added to the repository as required entrance criteria for analysis. If the location of a particular business function is not known this should be recorded as a Problem in Oracle Designer/2000. Certain Assumptions may also be made based on missing information. The lack of information in the Location primary access element should indicate a need for Problems and Assumptions definition within the specific application.

**3.2.1. Name is mandatory**

**3.2.2. Type is mandatory**

**3.2.3. Parent Location**

**3.2.4. Contact**

**3.2.5. Telephone Number**

**3.2.6. Address Line 1**

**3.2.7. Address Line 2**

**3.2.8. Code**

**3.2.9. Coordinate 1**

**3.2.10. Coordinate 2**

**3.2.11. Comment is not required**

**3.2.12. Description is required**

**3.2.13. Notes are not required**

**3.2.14. Business Unit Planning Items should be defined**

**3.3. Objectives - Records the objectives and aims of an organization; and also statements of business intent, which direct the activities of the organization.**

Objectives are more specific targets that help to achieve goals. For example, Goals for the DFAS Community both functional and technical include:

- Standardize/Consolidate Operations
- Standardize/Consolidate systems
- Eliminate Problem Disbursements
- Improve Customer Support

Corresponding Objectives for both communities include:

- Reduce Costs through Standardization and Consolidation
- Provide On-line access to financial information at all organization levels
- Ensure users have access to accurate, timely and relevant information
- Assure Security of all DFAS Applications
- Measure and Improve Quality of Service
- Aggressively apply new methods and technologies to improve customer service and reduce operation costs

The relationship between the goals of the technical and functional community are spelled out in the goals and objectives of each application systems. Without this information no application should move to the analysis stage. Clear definitions and agreement on the goals of all application development initiatives will increase the success rate for the effort. The Strategic Considerations Report available through the Repository Reports includes the following: details of various elements that can be defined at the strategic planning stage of enterprise modeling: assumptions, critical success factors, key performance indicators, objectives and problems.

***3.3.1. Name is mandatory***

***3.3.2. Type is mandatory***

***3.3.3. Parent Objective - only required for child objectives***

***3.3.4. Target Value is not required***

***3.3.5. Set By is not required***

***3.3.6. Unit of Measure is required if a Target Value is specified***

***3.3.7. Planning Horizon is not required***

***3.3.8. Date Required is not required***

***3.3.9. Ranking is not required***

***3.3.10. Achievement Comment is not required***

***3.3.11. Comment is not required***

***3.3.12. Description is required***

***3.3.13. Notes are not required***

***3.3.14. Objectives need not be associated with Critical Success Factors***

***3.3.15. Objectives may be associated with Business Functions***

***3.3.16. Objectives need not be associated with Key Performance Indicators***

**3.3.17. For information on Locations run Repository Reports/Enterprise Modeling /Strategic Considerations.**

**3.4. Assumptions - should be defined for any functional requirements that have assumptions.**

Records and monitors any assumptions that are made about a system; specifically those that may impact the design and implementation of tables. The recording of Assumption in the analysis stage is strongly related to the design of the database. The primary access element - Assumptions is related to the primary access element tables in order to associate between an assumption and the table definitions it influences.

**3.4.1. Name is mandatory**

**3.4.2. Type is mandatory**

**3.4.3. Made by**

**3.4.4. Ranking**

**3.4.5. Stability**

**3.4.6. Risk**

**3.4.7. Comment is not required**

**3.4.8. Description is required**

**3.4.9. Notes are not required**

**3.4.10. For information on Locations run Repository Reports/Enterprise Modeling /Strategic Considerations.**

**3.5. Problems - Records a business event or state that inhibits the progress of the enterprise towards its objective (e.g., personnel disputes, market moves, procedural or statutory changes, unresolved issues or unmade decisions).**

All problems associated with the application development initiatives should be defined prior to beginning analysis. Examples of the types of problems that may arise include: Network Performance Issues, Ownership of data issues and functional representation for gathering information. If these problems are stated up front as part of the Strategic Consideration Report proactive planning is facilitated.

**3.5.1. Name is mandatory**

**3.5.2. Type is mandatory**

**3.5.3. Parent Problem is required for child problems**

**3.5.4. Identified By is not required**

**3.5.5. Date Identified is not required**

**3.5.6. Date Solved is not required**

**3.5.7. Opportunities is not required**

**3.5.8. Cause Category is not required**

**3.5.9. Resolution Benefit is not required**

**3.5.10. Comment is not required**

**3.5.11. Description is required**

**3.5.12. Notes are not required**

**3.5.13. For information on Problems run Repository Reports/Enterprise Modeling /Strategic Considerations.**

**3.6. Critical Success Factors (CSFs) - Business events, or states, whose failure could seriously hinder the achievement of an objective.**

Critical Success Factors must be included in the entry criteria for all application development initiatives. Most businesses evaluate their success or failure by some set of performance measures. Critical Success Factor Analysis is the identification of a hierarchy of performance measures that lead to identification of critical factors and issues that will determine a business's success. Critical Success Factors are "things that must happen" if objectives are to be achieved. CSF also provide a useful baseline not only for evaluating business performance, but also for evaluating information systems' contribution to the value of the business. Specific information system statements may emerge as CSFs during analysis.

**3.6.1. Name is mandatory**

**3.6.2. Critical is mandatory**

**3.6.3. Parent**

**3.6.4. Relative Priority**

**3.6.5. Set By**

**3.6.6. Target Value**

**3.6.7. Unit of Measure**

**3.6.8. External Dependency**

**3.6.9. Date Required**

**3.6.10. Achievement Comment**

**3.6.11. Comment is not required**

**3.6.12. Description is required**

**3.6.13. Notes are not required**

**3.6.14. For information on CSFs run Repository Reports/Enterprise Modeling /Strategic Considerations.**

**3.7. Key Performance Indicators - Defines indicators that can be used to quantify or monitor the progress that is made toward achieving a set of business objectives.**

**3.7.1. Name is mandatory**

**3.7.2. Parent KPI**

**3.7.3. Responsibility**

**3.7.4. Measured By**

**3.7.5. Set By**

**3.7.6. Target Value**

**3.7.7. Unit of Measure**

**3.7.8. Comment is not required**

**3.7.9. Description is required**

**3.7.10. Notes are not required**

**3.7.11. For information on Key Performance Indicators run Repository Reports/Enterprise Modeling /Strategic Considerations.**

**3.8. Business Functions -**

**Business Function Modeling** - A business function model provides an overview of all the necessary and desired functionality of the a business that is required to achieve the business objectives. The business functions describe not only what the business does, but also what it should do. The scope of investigation is a business area, and focus is on WHAT the business had to do to meet it's objectives. There is no distinction between manual functions and functions that need to be supported by the computer system. A sound business function model remains valid as long as the business objectives do not change. Any change in the way the business in run (the how), should not affect the business functions model.

**System Function Modeling** - A system function model focuses on the functionality that must be implemented or supported by the computer system. This includes functions which: completely implement business functions

- provide support for selected steps of business functions
- provide different groups of users with different ways of performing business functions (for example and infrequent users' function and a fast-path function for regular users)
- support the way users will work with the system or to help them achieve their business functions, such as functions to support flow of work between users, like work flow queuing and management functions
- required to help administer the operation of the system such as:
  - \* perform error logging
  - \* administer or control system restart and recovery
  - \* report shortage of resources, such as database space
  - \* monitor the integrity of data
- control the flow of processing through the system, such as:
  - \* administer and support the proposed technical architecture, such as network message transmission, routing, and reception
  - \* provide processing workload scheduling: for example batch queuing, sequencing, and monitoring functions for reports and large overnight jobs
  - \* provide real time processing flow and queuing mechanisms
- distribute data across geographically dispersed systems.

Business versus System Function Modeling - Because of the technique of functional decomposition is the same in both modeling approaches the word function should be used to describe both business and system functions. The need for both business and system function models does not mean you will always end up with two separate function hierarchies. It is very common to have one function hierarchy with the higher levels representing business functions and the lower levels representing system functions. In general, the what (business functions), as opposed to the how (system functions), should only be taken down two or three levels of decomposition within a function hierarchy.

### **Cross-checking with the Data Model**

Often, in Business Requirements Definition, only the functional outlines,(for example, the top layers of the function hierarchy), are described. Then a lot of time is spent on creating the entity relationship model. However, it is impossible to create a data model without at least an implicit concept of the functions involved. For this reason, it is essential to start making the functions explicit as early as possible and not to postpone describing the functions to the last few days of requirements modeling. Many discussions on the data model are actually discussions on the functionality and vice versa.

#### ***3.8.1. Label is mandatory and should be consistently defined (e.g. numbers with decimal points to show hierarchy)***

Examples of labels currently in the RON include:

CEFT\_9902        0.0  
                  1.0  
                  1.1  
                  1.1.1  
                  1.1.2  
                  1.1.3  
                  CEFT  
                  PM\_ARCHI  
                  PM\_CONUS

GETBACC\_9902  
                  AFACDONE  
                  AFDZDONE  
                  AFMAINTX  
                  AFN1DONE  
                  ASSEMBLE S  
                  ASSEMSGL  
                  BACCOK

INTEG\_9901  
                  2.1.1  
                  2.1.2  
                  2.1.3  
                  2.1.4  
                  2.1.5  
                  2.1.6  
                  2.2.1.1

2.2.1.2  
2.2.2.1.1  
2.2.2.1.2  
2.2.2.1.3  
2.2.2.1.4  
3.1  
GET\_M31  
GET\_M36  
GET\_M37  
GET\_M39  
GET\_S11CEV  
GET\_S12CEV  
PSGL\_PT100  
PSGL\_PT110  
PSGL\_PT120  
PSGL\_PT130  
PSGL\_PT140  
PSGL\_PT150  
PSGL\_PT160  
PSGL\_PT170  
PSGL\_PT180  
PSGL\_PT190  
PSGL\_PT200

**3.8.2. Short Definition is mandatory and should consist of simple Verb-Noun phrases, for example**

- Process Vendor EFT Data
- Perform Vendor Input Data Edit
- Suspend Transaction
- Validate BACC Interdependencies / Patterns
- Post JV Transaction Details
- Enter Journal Voucher Details.

**3.8.3. Master Application is required**

**3.8.4. Master Function is required**

**3.8.5. Parent Function is required unless the function is a parent function**

**3.8.6. Sequence in Parent is required unless function is a parent function**

**3.8.7. Elementary? is required**

**3.8.8. Frequency is required**

**3.8.9. Frequency Unit is required if a Frequency is specified, a Frequency Unit must be specified also**

**3.8.10. Response Needed is required**

**3.8.11. Description is required**

**3.8.12. Notes are not required**

Functions - The notes property for functions is used for the following information:

- Enter notes of a technical nature that are important for the persons involved in the next stages in the system development process.
- Do not enter business rules here. Business rules should be expressed as separate functions.
- Use Notes to maintain the change history for the particular function.
- CODE or Procedural Logic in the notes at the Analysis Level - If DFAS decides to allow this - IT CANNOT BE A SUBSTITUTE FOR CRUD.

**3.8.13. Function Type is required for Elementary Business Functions**

**3.8.14. Intention to automate is mandatory**

**3.8.15. BPR Generate ? is mandatory**

**3.8.16. Person Cost Rate is required if BPR analysis is to be done**

**3.8.17. Person Cost Rate Unit is required if BPR analysis is to be done**

**3.8.18. Overhead Cost Rate is required if BPR analysis is to be done**

**3.8.19. Overhead Cost Rate Unit is required if BPR analysis is to be done**

**3.8.20. Total Cost Rate Not Required is required if BPR analysis is to be done**

**3.8.21. Total Cost Rate Unit is required if BPR analysis is to be done**

**3.8.22. Additional Cost Rate is required if BPR analysis is to be done**

**3.8.23. Additional Cost Rate Unit is required if BPR analysis is to be done**

**3.8.24. Prior Delay Time is required if BPR analysis is to be done**

**3.8.25. Prior Delay Time Unit is required if BPR analysis is to be done**

**3.8.26. Work Time is required if BPR analysis is to be done**

**3.8.27. Work Time Unit is required if BPR analysis is to be done**

**3.8.28. Quality Check Time is required if BPR analysis is to be done**

**3.8.29. Quality Check Time Unit is required if BPR analysis is to be done**

**3.8.30. Post Delay Time is required if BPR analysis is to be done**

**3.8.31. Post Delay Time Unit is required if BPR analysis is to be done**

**3.8.32. Total Time is required if BPR analysis is to be done**

**3.8.33. Total Time Unit is required if BPR analysis is to be done**

- 3.8.34. Measured Time 1 is not required, but is a good place to record expected processing time***
- 3.8.35. Measured Time 2 is not required, but is a good place to record expected processing time***
- 3.8.36. Measured Time 3 is not required***
- 3.8.37. Measured Time Unit is required if a Measured Time is specified***
- 3.8.38. Quality Percentage is not required***
- 3.8.39. Value Added is not required***
- 3.8.40. Value Chain Analysis is not required***
- 3.8.41. Competitive Index Value is not required***
- 3.8.42. Competitive Index is not required***
- 3.8.43. Resource Required 1 is not required***
- 3.8.44. Resource Required 2 is not required***
- 3.8.45. Resource Required 3 is not required***
- 3.8.46. Resource Required 4 is not required***
- 3.8.47. Problem Areas is not required***
- 3.8.48. Simple Rules is not required***
- 3.8.49. Entity Notes is not required***
- 3.8.50. Icon File is not required***
- 3.8.51. Image File is not required***
- 3.8.52. Sound File is not required***
- 3.8.53. Video File is not required***
- 3.8.54. Execution String is not required***
- 3.8.55. Note Type is not required***
- 3.8.56. Note Text is not required***
- 3.8.57. Note Picture File is not required***

## **FHD General Guidelines**

FHD-001: Functions must be decomposed to an elementary level.

FHD-002: All functions on the same level of decomposition must have the same level of abstraction, to the extent possible.

FHD-003: Functions should have no overlapping functionality that is not covered by an explicit common function.

FHD-004: Decomposed functions must always be fully covered by the functions on the next lower level.

FHD-005: On the lowest level, functions should easily be classified as manual, clerical, computer, or other mechanisms, but no combinations.

FHD-006: The function hierarchy should be well balanced; that is, all top-level functions should be decomposed to approximately the same number of levels.

FHD-007: Do not use more than ten subordinate functions per parent function; aim for six.

FHD-008: Create a dedicated branch for business rules that you record as a function (see Chapter 3, “Business Rule Modeling”), with sub-branches for each sub-class of business rules as follows:

- entity rules
- inter entity rules
- create rules
- update rules
- delete rules
- change event rules

FHD-009: Set check box Elementary to checked, only if the function, when executed, always fully succeeds or, if not successful, fully removes its trails. Attention: Elementary functions may be, but rarely are, decomposed.

FHD-010: Every function must be written in a clear phrase, local to the business, avoiding ambiguity as much as possible.

FHD-011: The function should not specify the “how”, but the “what” of the business. Use the current “how” of the function explicitly as an example in the function description.

FHD-012: Function definitions should have the main structure, <verb> [<adjective>] <noun>, or, depending on the language in use, [<adjective>] <noun> <verb>.

FHD-013: Avoid underscores, punctuation marks, and symbols in the function definition. Use active voice.

FHD-014: Specify frequency using the smallest possible Per unit that will form a regular basis; for example, use 800/MONTH rather than 10,000/YEAR if the function is likely to happen every now and then. Use 10,000/YEAR if a peak is likely.

FHD-015: Use Overnight, Immediate, or leave blank. Attention: This field is used by the esigner/2000 utility that creates module entries based on function definitions.

FHD-016: For every entity, there must be at least one function that allows Create, Retrieve, or Delete (the latter for the sake of completeness).

FHD-017: Functions that have the Archive check box checked should usually have the Delete check box checked as well.

### **3.9. Events - Each Business Process Diagram should both begin and end with a customer Event**

#### **Event Definition**

In the RON, it is possible to enter basic information about an event. Usually, events are very easily understood by all users. Events often form a good starting point for initially generating and identifying functions. Events can be used to group functions logically. Events function as the nodes in the function network.

- The Forms Generator does not use event information.
- Create events only if the event adds substantially to the user's understanding of the functions of the system and its structure.
- Always use explicit events to document the triggering of the following classes of business rules:
  - \* restricted relationship rules
  - \* other entity rules
  - \* change event rules
  - \* data operation rules

***3.9.1. Name is mandatory***

***3.9.2. Type is required***

***3.9.3. On Condition is not required***

***3.9.4. Date is not required***

***3.9.5. Time is not required***

***3.9.6. Frequency is not required***

***3.9.7. Frequency Unit is required if a Frequency is specified***

***3.9.8. Entity is not required***

***3.9.9. Attribute is not required***

***3.9.10. System Description is not required***

***3.9.11. Description is required***

***3.9.12. Notes are not required***

***3.9.13. Icon File is not required***

***3.9.14. Image File is not required***

***3.9.15. Sound File is not required***

***3.9.16. Video File is not required***

***3.9.17.Execution String is not required***

**3.10. Datastores - there should be at least one Datastore in each BPD**

***3.10.1.Id is mandatory***

***3.10.2.Name is mandatory***

***3.10.3.Type is not required***

***3.10.4.Comment is not required***

***3.10.5.Description is required***

***3.10.6.Notes are not required***

***3.10.7.Business Unit is required***

***3.10.8.Function is required***

***3.10.9.Store Type is not required***

***3.10.10.BPR Use in Generation ? is mandatory***

***3.10.11.Person Cost Rate is required if BPR analysis is to be done***

***3.10.12.Person Cost Rate Unit is required if BPR analysis is to be done***

***3.10.13.Overhead Cost Rate is required if BPR analysis is to be done***

***3.10.14.Overhead Cost Rate Unit is required if BPR analysis is to be done***

***3.10.15.Total Cost Rate is required if BPR analysis is to be done***

***3.10.16.Total Cost Rate Unit is required if BPR analysis is to be done***

***3.10.17.Additional Cost Rate is required if BPR analysis is to be done***

***3.10.18.Additional Cost Rate Unit is required if BPR analysis is to be done***

***3.10.19.Prior Delay Time is required if BPR analysis is to be done***

***3.10.20.Prior Delay Time Unit is required if BPR analysis is to be done***

***3.10.21.Work Time is required if BPR analysis is to be done***

***3.10.22.Work Time Unit is required if BPR analysis is to be done***

***3.10.23.Quality Check Time is required if BPR analysis is to be done***

***3.10.24.Quality Check Time Unit is required if BPR analysis is to be done***

***3.10.25.Post Delay Time is required if BPR analysis is to be done***

***3.10.26.Post Delay Time Unit is required if BPR analysis is to be done***

***3.10.27.Total Time is required if BPR analysis is to be done***

- 3.10.28.Total Time Unit is required if BPR analysis is to be done***
- 3.10.29.Measured Time 1 is required if BPR analysis is to be done***
- 3.10.30.Measured Time 2 is required if BPR analysis is to be done***
- 3.10.31.Measured Time 3 is required if BPR analysis is to be done***
- 3.10.32.Measured Time Unit is required if BPR analysis is to be done***
- 3.10.33.Frequency is not required***
- 3.10.34.Frequency Unit is required if a Frequency is specified***
- 3.10.35.Minimum Volume is not required***
- 3.10.36.Average Volume is not required***
- 3.10.37.Maximum Volume is not required***
- 3.10.38.Quality Percentage is not required***
- 3.10.39.Value Added is not required***
- 3.10.40.Value Chain Analysis is not required***
- 3.10.41.Resource Required 1 is not required***
- 3.10.42.Resource Required 2 is not required***
- 3.10.43.Resource Required 3 is not required***
- 3.10.44.Resource Required 4 is not required***
- 3.10.45.Problem Areas is not required***
- 3.10.46.Simple Rules is not required***
- 3.10.47.Icon File is not required***
- 3.10.48.Image File is not required***
- 3.10.49.Sound File is not required***
- 3.10.50.Video File is not required***
- 3.10.51.Execution String is not required***
- 3.10.52.Note Type is not required***
- 3.10.53.Note Text is not required***
- 3.10.54.Note Picture File is not required***

### **3.11. Entities Required**

The starting point for all data analysis is the DCD LOG Application maintained by the Data Management Team at DFAS Headquarters.

## **Business Data Modeling versus System Data Modeling**

The Business Data Model provides a full and detailed definition of the structure of all the data that the business areas use or generate. The System Data Model provides a full and detailed definition of the structure of all the data that the system is to store. This includes data identified within the Business Data Model that is also within the scope of the system. It also includes additional data structures (technical entities) that may be required to support any functionality within the system that is concerned with processing that is not directly visible to the end user.

The System Data Model may be produced by taking a copy of the Business Data Model and modifying it to support and facilitate the functionality of the system (described in the System Function Model). Changes are likely to include any of the following:

- removal of entities, attributes, relationships, and identities that are not going to be stored or supported by the computer system
- addition of entities, attributes and relationships to support system functionality concerned with the provision of the system's infrastructure; for example, configurable parameters, communications status, interface status, queues, error logs, and means for users to access subsets of the data that they use regularly
- addition of the surrogate keys which may be required to support the system functions
- generalization of data structures from the business data model, so that a single data structure can have slightly different uses.
- making data structures less generic within the Business Data Model where different types of an entity and dependent entities are to be processed by different functions
- simplification of the data model to simplify the system functionality

Do not try to make these structures part of the Business Data Model, as they are part of the particular projected implementation of the system. Not all entities are clearly technical or not. An entity like ALLOWED STATE TRANSITION is in the border area but still acceptable in the Business Data Model.

### **Cross-checking with Function Model**

It is critical that the entity relationship model be developed in conjunction with the function model, if one is being produced, to ensure that all the identified functions are supported by information within the ERM and that all the information within the ERM is correctly handled within the function model. The usage matrices provide a cross-checking mechanism for the ERM's relationship to the function model.

#### ***3.11.1.Name is mandatory***

Examples of standard DFAS entity names include:

- GL ACT
- GL ACT CD PAIR
- PF ELEMENT
- PF CRITERIA
- OL
- OL DETAIL

For the entire list see the DCD LOG Application in the Repository

#### ***3.11.2.Short Name is mandatory***

**3.11.3.Plural is mandatory**

**3.11.4.Type Of is not required - Subtype Entities should have either a sibling Subtype or a Relationship**

**3.11.5.Initial is required**

**3.11.6.Average is required**

**3.11.7.Maximum is required**

**3.11.8.Annual Growth Rate (%)is required**

**3.11.9.Datawarehouse Type is not required**

**3.11.10.Description is required**

**3.11.11.Notes are not required**

The notes property for entities is used for the following information:

- Change History of an entity should be recorded in the Notes, stating the date, the reason for the change, and the change itself.
- Recording Legacy System Information
- Communication issues and action items among team members
- Recording Subtype, Supertype resolution information

**3.11.12.Attributes - Each Entity or Subtype must have at least one Attribute**

3.11.12.1.Name is mandatory

3.11.12.2.Sequence in Entity is not required

3.11.12.3.Domain is required if the Attribute is derived from a Domain

3.11.12.4.Format is required

3.11.12.5.Average Length is required

3.11.12.6.Maximum Length is required

3.11.12.7.Decimal Places is required if the Format is numeric

3.11.12.8.Optional ? is mandatory

3.11.12.9.On Condition is not required

3.11.12.10.Units is not required

3.11.12.11.Default is not required

3.11.12.12.Null Value is not required

3.11.12.13.Derivation is not required

3.11.12.14.Percent Used - Initial is required

3.11.12.15.Percent Used - Average is required

- 3.11.12.16. Authority is not required
- 3.11.12.17. Responsible is not required
- 3.11.12.18. Validation Rules
- 3.11.12.19. Comment is not required
- 3.11.12.20. Description is required
- 3.11.12.21. Notes is not required
- 3.11.12.22. Sequence in Sort Key is not required
- 3.11.12.23. Sort Sequence Order is not required
- 3.11.12.24. Some Attributes have Allowable Values
  - 3.11.12.24.1. Value is mandatory and is either the valid value or the low value if no range is specified
  - 3.11.12.24.2. High Value is required
  - 3.11.12.24.3. Abbreviation is not required
  - 3.11.12.24.4. Meaning is required
  - 3.11.12.24.5. Display Sequences required
  - 3.11.12.24.6. Description is required
  - 3.11.12.24.7. Notes are not required

#### General ERD Guidelines

ERD-001: Every conceptual model must be accompanied by a graphical representation of that model.

ERD-002: Make at least one diagram showing all entities, relationships, and arcs for the entire application.

ERD-003: Make separate diagrams for every sub-system. Start from a copy of the diagram of the entire system and remove the obsolete entities, thus keeping the basic layout intact.

#### **Drawing Conventions**

The most convenient way to represent the entity information is in the form of one or more entity relationship diagrams, and the drawing conventions are explained in that light.

##### Attributes

ERD-006: All attributes that are part of the first unique identifier should be listed inside the softbox representing the entity to which they belong.

##### Sub-types

ERD-007: Sub-types should: form a complete set of mutually exclusive classes (a partition of the super-type); sub-types never come alone

- only represent one classification at a time
- include only the attributes unique to sub-types: sub-types inherit the attributes of the super-type automatically

- attach only to relationships specific to sub-types: sub-types inherit the relationships of the super-type automatically

### Super-types

ERD-008: Super-types should:

- include only those attributes common to all its sub-types
- take part only in those relationships that are valid for all its sub-types

### Relationship Lines

ERD-009: Relationship lines should:

- not cross entity softboxes
- avoid crossing other relationship lines, to the extent that this is possible
- be drawn using short, straight lines and right angles, to the extent that this is possible
- be drawn one to many, from top to bottom and left to right, to the extent that this is possible
- possess relationship names at both ends that allow reading in either direction.

### Layout

ERD-010: Avoid making drastic changes to the basic layout of an entity relationship (ER) diagram that has been around for some time. Make sure the floor plan of the first layout of the overall ER diagram will be very similar to the completed one. If someone drastically changes the floor plan of the ER diagram you used to understand, it will take you quite some time to reach the same level of understanding of the new layout.

ERD-011: Give the core entities of your system a central place in your diagram. Take for example the entity ORDER LINE or PRODUCED LOT.

ERD-012: Place entities that are often referred to, but that are not specific to the business, somewhere at the border of your diagram.

ERD-013: Try to avoid the use of arcs. It is usually worthwhile to replace the arc with sub-types.

ERD-012: Do not draw implied sub-types as sub-types, unless they have a right of their own to exist. An implied sub-type of an entity is a sub-type that exists as an implication of the entities to which it refers. As there are SERVICES and PRODUCTS to which the ORDER LINES can refer, there are—by implication—two kinds of ORDER LINES. This kind of distinction cannot be denied, but it often makes no sense to model it. Always bear in mind that the goal of representing entity or functional information in some kind of diagram is to add clarity to the model.

ERD-013: ER diagrams should always show a legend stating:

- diagram name
- application (sub)system
- author

- date

### **3.11.13. Relationships - Entities should have at least one relationship, however there will be certain entities that are standalone**

Relationships should always be named since there can be various relationships between two entities, and their purposes must be known. There are many different relationships possible between two entities; for example, between the entities PERSON and COMPANY the relationship is currently working for may be of interest, but so is the relationship is owner of. You must give each end of a relationship a name, an optionally, and a degree (1 or many).

#### **Optionally and Degree**

The terms “many-to-many”, “m:m”, “m:n” all mean the same thing. The type of relationship that occurs most often is as follows:

```
m:1 >>----- - - - - (mandatory m to optional 1)
```

Usually, about 80% of the relationships in a normal model are of the above type, while about 20% are likely to be as follows:

```
m:1 > - - - - - - - - (optional m to optional 1)
m:1 >----- (mandatory m to mandatory 1)
1:1 ----- - - - - (mandatory 1 to optional 1)
1:1 - - - - - - - - - (optional 1 to optional 1)
m:n >- - - - - - - - < (optional m to optional n)
```

It is very unlikely that you will find correct examples of the following:

```
m:n >----- - - - - < (mandatory m to optional n)
m:n >-----< (mandatory m to mandatory n)
1:m ----- - - - - < (mandatory 1 to optional m)
1:1 ----- (mandatory 1 to mandatory 1)
```

It is a good practice to convert as many of the non-m:1 relationships as you can to m:1 relationships; non-m:1 relationships are often incorrect.

3.11.13.1.From Entity is mandatory

3.11.13.2.From Relationship Name is mandatory and should fit the following template sentence: [A|An] <From Entity Name> [may be|must be] <From Relationship Name> [one and only one|one or more] <To Entity Name>.

3.11.13.3.From Minimum Cardinality is mandatory

3.11.13.4.From Maximum Cardinality is not required for "Many" Relationships

3.11.13.5.From In Arc is not required

3.11.13.6.From Transferable ? is mandatory

3.11.13.7.From Average is not required

3.11.13.8.From Minimum is not required

3.11.13.9.From Maximum is not required

3.11.13.10.From Description is required

3.11.13.11.From Notes are not required

- 3.11.13.12.To Entity is mandatory
- 3.11.13.13.To Relationship Name is mandatory and should fit the following template sentence: [A|An] <To Entity Name> [may be|must be] <To Relationship Name> [one and only one|one or more] <From Entity Name>.
- 3.11.13.14.To Minimum Cardinality is mandatory
- 3.11.13.15.To Maximum Cardinality is not required for "Many" Relationships
- 3.11.13.16.To In Arc is not required
- 3.11.13.17.To Transferable ? is mandatory
- 3.11.13.18.To Average is not required
- 3.11.13.19.To Minimum is not required
- 3.11.13.20.To Maximum is not required
- 3.11.13.21.To Description is required
- 3.11.13.22.To Notes are not required

**3.11.14.Unique Identifiers - Entities that do not have Unique Identifiers**

**3.11.15.3.11.15. Usages - Used to create the CRUD matrix that links the entities to the functions**

- 3.11.15.1. Used by Business Function - each entity must be associated with one or more lowest level business function. To use any of the entities, they must be attached to an elementary business function that will be responsible for creating, retrieving, updating, or deleting (CRUD) instances of that entity. Each entity must therefore have a usage of at least one elementary function. At least one of the Create? Retrieve ? Update? And Delete? properties must be set to yes
  - 3.11.15.1.1. Create ? is mandatory
  - 3.11.15.2. Retrieve ? is mandatory
  - 3.11.15.3.3.11.15.1.3. Update ? is mandatory
  - 3.11.15.4.3.11.15.1.4. Delete ? is mandatory
  - 3.11.15.5.3.11.15.1.5. Archive ? is mandatory
  - 3.11.15.6.3.11.15.1.6. Other ? is mandatory
  - 3.11.15.7.3.11.15.1.7. Comment is not required
  - 3.11.15.8.3.11.15.2. Attributes - each of the attributes in the entity must also be associated with each of the elementary business functions that the attribute?s parent entity is associated with.

For each attribute, there must be one elementary business function which will be responsible for inserting, retrieving, updating, and nullifying (IRUN) attributes. The easiest way to make sure that the attributes are associated with the functions is to use the utility menu item: Create Function Attribute Matrix.