

## **6 Diagramming Guidelines**

---

## 6.1 GENERAL

### 6.1.1 Introduction

This section of the standards document specifies standard practices for elements and characteristics of Diagramming. By implementing consistent diagramming practices confusion will be minimized when interpreting diagrams and attempting to understand modelled business processes.

### 6.1.2 Objectives

The intent of this section is to establish Diagramming standards and guidelines that will accomplish the following:

- Increase consistency
- Improve readability
- Improve pattern recognition.

### 6.1.3 Applicability

This document applies to all DCII development.

### 6.1.4 References

Some layout methodologies presented here are based on information from the Oracle book CASE\*Method Entity Relationship Modelling by Richard Barker.

### 6.1.5 Document Organization

This document is organized into four distinct sections:

- Process Model Diagrams
- Function Hierarchy Diagrams
- Entity Relationship Diagrams
- Server Model Diagrams

The use of the Data Flow Diagramming tool is not recommended and will not be a standard practice within DCII.

### 6.1.6 Display a Legend on all Diagrams

DIA-01 Under *File-> Summary Information*, check the following to be displayed:

- Diagram
- Title
- Created
- Last Modified
- Author
- Application

DIA-02 Display the legend in the upper left-hand corner of most diagrams. For Business Process Model diagrams, place the legend in the swim lane for the *Unspecified* business unit.

- DIA-03 The diagram title should be indicative of the component being modeled and should be understandable by business users. Due to size constraints, the diagram name may be somewhat cryptic.

Example:

Title = Interview Applicants Process  
Diagram Name = INTERVIEW APPLICANTS

### **6.1.7 Delete Unneeded Diagrams**

- DIA-04 If a diagram is no longer needed, delete it from the Repository.

### **6.1.8 Consolidating Diagrams**

- DIA-05 Consolidate Diagrams frequently using *Edit->Requery*
- DIA-06 In general, it is a good practice to consolidate a diagram each time you open it to ensure that you are working with current information. At a minimum, diagrams should be consolidated when there is a change to the Designer Repository relative to business functions, flows, stores, events (triggers) and business units. If you wish to keep historical views of diagrams, save them under another name and do not consolidate them.
- DIA-07 There is no option to consolidate Server Model Diagrams. They are automatically consolidated upon opening. If you wish to keep historical views of diagrams, you must print them on paper (before a change occurs) as there is no way to open one without seeing the underlying repository changes.

---

## 6.2 Business Process Model (BPM) Diagrams

These guidelines should be implemented when creating Business Process Model Diagrams via the Oracle Designer Business Process Modeler Diagrammer.

### 6.2.1 Multiple Modular Diagrams

DIA-08 Rather than creating one large business process diagram, create many sub-diagrams. This will eliminate the need to maintain one large diagram that will over time become unreadable and unmanageable. Also, each process step of a business process diagram will appear as a function on a function hierarchy diagram.

### 6.2.2 Avoid Overcrowding

DIA-09 Do not crowd many processes onto a single business process diagram. In general, one diagram should contain no more than six to nine process steps.

DIA-10 If a single diagram contains more than nine process steps, consider decomposing one of the more complex process steps into a new diagram.

### 6.2.3 BPM Diagram Layout Convention

In this section, please refer to the example diagram: Figure BPM-1.

---

#### 6.2.3.1 Diagram Element Inclusion:

DIA-11 Diagrams should include:

- Major process steps (i.e., no functions)
- Flows (i.e., data, material or temporal)
- Flow names
- Process labels
- Triggers and outcomes
- Stores (data and material)

---

#### 6.2.3.2 Process Step Placement

DIA-12 Place process steps in the business units' swim lane that performs it.

DIA-13 When possible, place process steps to the right of a dependent process step rather than under it. This will reduce the swim lane height.

DIA-14 Place process steps that are associated with a trigger one extra grid-snap to the right of the business unit. This will make the printout more readable.

DIA-15 Before creating a new process step, always use the *Edit->Include* facility first to prevent duplicate processes.

---

#### 6.2.3.3 Drawing Flows

DIA-16 Face arrowheads to the right if at all possible.

DIA-17 Avoid line crossings if at all possible.

DIA-18 Before creating a new flow, always use the *Edit->Include* facility first to prevent creation of duplicate flows.

---

**6.2.3.4 Flow name placement**

---

DIA-19 When flow names are used, place flow names in the middle of the flow.

---

**6.2.3.5 Label placement**

---

DIA-20 Place the label of the process above and to the left of the process step using this method:

Copy label to the annotation text field, set annotation type to 'TEXT' in multimedia tab.

Check Display Annotation in *Options->Customize->Graphics* menu item.

---

**6.2.3.6 Material Store placement**

---

DIA-21 Before creating a new material store, always use the *Edit->Include* facility first to prevent creation of duplicate material stores.

DIA-22 Place material stores in the Business Units swim lane to show that they are responsible for the material.

---

**6.2.3.7 Data Store placement**

---

DIA-23 Before creating a new datastore, always use the *Edit->Include* facility first to prevent creation of duplicate datastores.

DIA-24 Datastores should be placed in the swim lane corresponding to the organization unit to which the stores belong. Place datastores that belong to multiple organization units in the *Unspecified* swim lane.

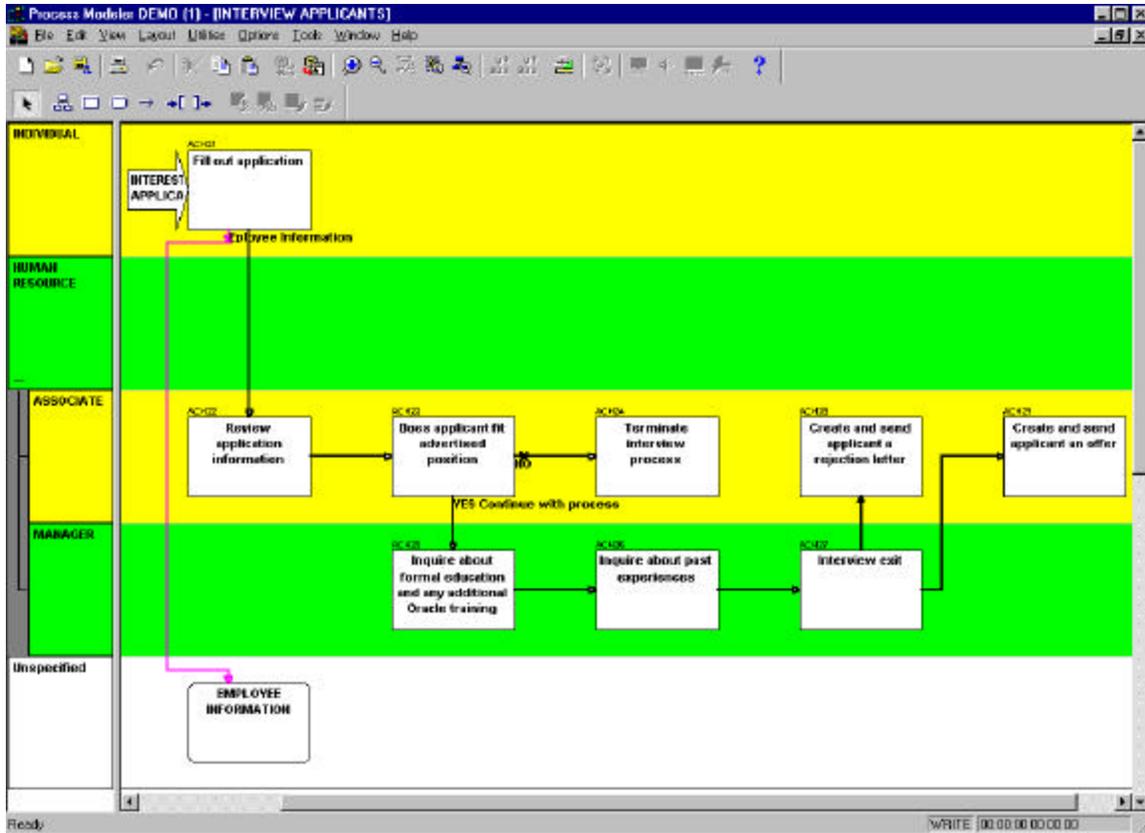


Figure BPM-1.

### 6.2.4 Color Coding Diagrams

DIA-25 Use color coding or shading to distinguish the following:

- Process steps that are decomposed (open down) into sub-diagrams
- Different type of flows (Data flows from Material flows)
- Alternating business units and/or swim lanes
- Do not use colors on the *Unspecified* business unit or its swim lane

### 6.2.5 Annotations

DIA-26 Use annotation in the form of notes or graphics to facilitate communication.

### 6.2.6 Specific Types

DIA-27 Set the specific type for all objects in the BPM whenever the type is known.

### 6.2.7 Display Modes

DIA-28 Display Diagrams in the following modes:

**Symbol Mode:** Use Symbol mode for simple diagramming. This mode is appropriate for presentations to the non-technical business community.

**Enhanced Symbol Mode:** Use Enhanced Symbol mode for flowcharting. This mode is appropriate for presentations to a more technical audience.

**Iconic Mode:** Use Iconic mode to display associated icons for process steps, stores, and flows. This mode is appropriate for upper management and end-user presentations.

## 6.2.8 Visualizing a process

DIA-29 The BPM Diagrammer in Designer has a number of other options that can be used to enhance the presentation of a diagram. Below are some additional recommendations on the use of these features.

### 6.2.8.1 Images

---

DIA-30 Use images only if it facilitates communication.

### 6.2.8.2 Video

---

DIA-31 Use video clips only when it facilitates communication.

### 6.2.8.3 Sound

---

DIA-32 Use sound (digital recordings) only when it facilitates communication.

### 6.2.8.4 Animation

---

DIA-33 Use animation only when it facilitates communication.

## 6.2.9 Events

DIA-34 Each Business Process Diagram should both begin and end with a customer Event

### 6.2.9.1 Event Definition

---

DIA-35 In the Repository Object Navigator, it is possible to enter basic information about an event. Usually, all users very easily understand events. 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.

DIA-36 The Forms Generator does not use event information.

DIA-37 Create events only if the event adds substantially to the users' understanding of the functions of the system and its structure.

DIA-38 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

Mandatory	Required	Not Required
Name	Type	On Condition

	Description	Date
		Time
		Frequency
		Frequency Unit
		Entity
		Attribute
		System Description
		Notes
		Icon File
		Image File
		Sound File
		Video File
		Execution String

---

## 6.3 Function Hierarchy Diagrams (FHD)

These guidelines should be implemented when creating Function Hierarchy Diagrams via the Oracle Designer Function Hierarchy Diagrammer.

### 6.3.1 Multiple Modular Diagrams

- DIA-39 Rather than creating just one function hierarchy diagram, create many sub-diagrams.
- DIA-40 Create one diagram that will function as a view of the entire enterprise.
- DIA-41 Create a diagram for each of the major processes that are sub-functions of the root function.

### 6.3.2 Avoid Overcrowding

- DIA-42 Do not crowd many processes onto a single function hierarchy diagram. In general, one diagram should contain no more than twenty functions.
- DIA-43 If a single diagram, other than one used to present a consolidated view of the enterprise, contains more than twenty functions, consider decomposing one of the more complex process steps into a new diagram.

### 6.3.3 FHD Layout Conventions

- DIA-44 This section lists some layout conventions that are helpful when creating diagrams. These conventions aid in making the diagrams easier for you to print and the business community to understand. The three layout styles are shown in figures FHD-1 through FHD-3.

**Horizontal Layout** is the default layout. This style shows each level of the hierarchy in a horizontal row.

**Vertical Layout** displays each level of the hierarchy in a vertical line, with each level indented slightly to the right of the one above it.

**Hybrid Layout** is a combination of vertical and horizontal layouts that shows parent functions in a horizontal layout and elementary (childless) functions in a vertical layout. This style is the best of the three options for reading and printing multilevel hierarchies.

- DIA-45 In most cases, use the hybrid option as it uses the diagram space most efficiently. But this is really most dependent on how deep your function model is. If it is shallow and wide, often the horizontal layout will also look good. In general the vertical style should be avoided unless the model (or model segment) is very small.

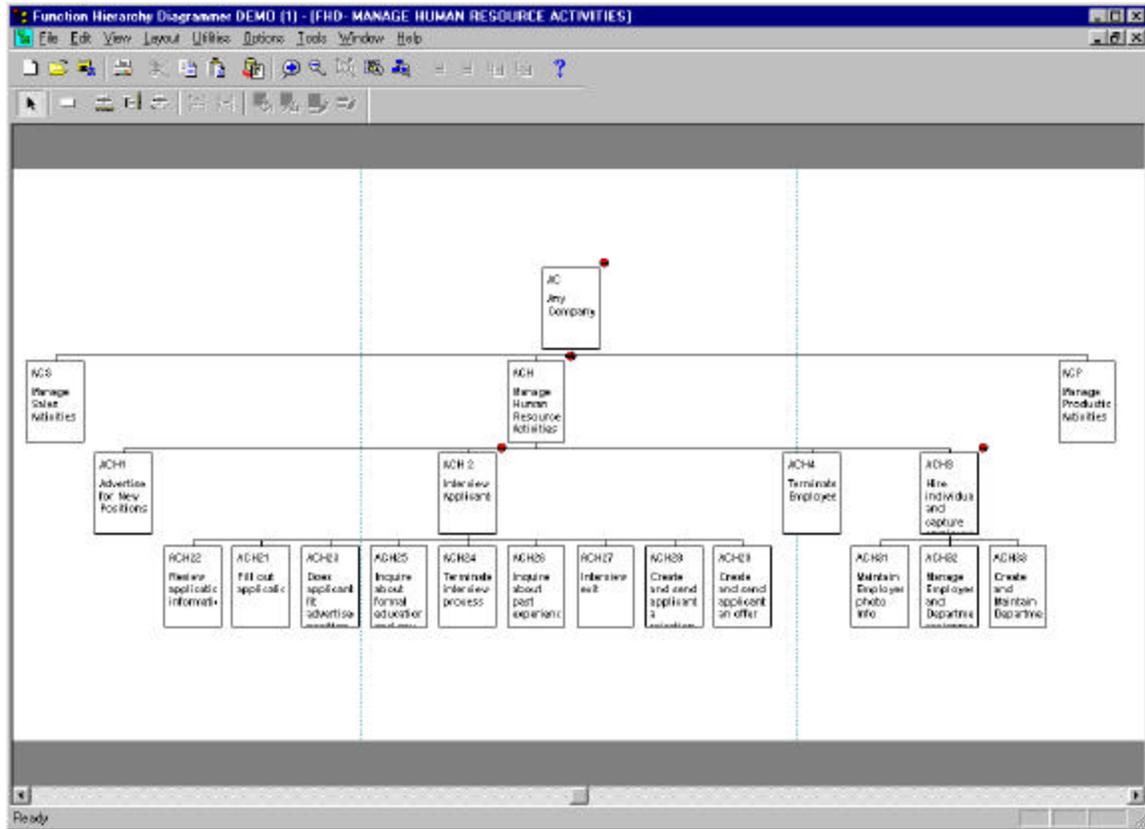


Figure FHD-1. Example FHD displayed in horizontal layout style

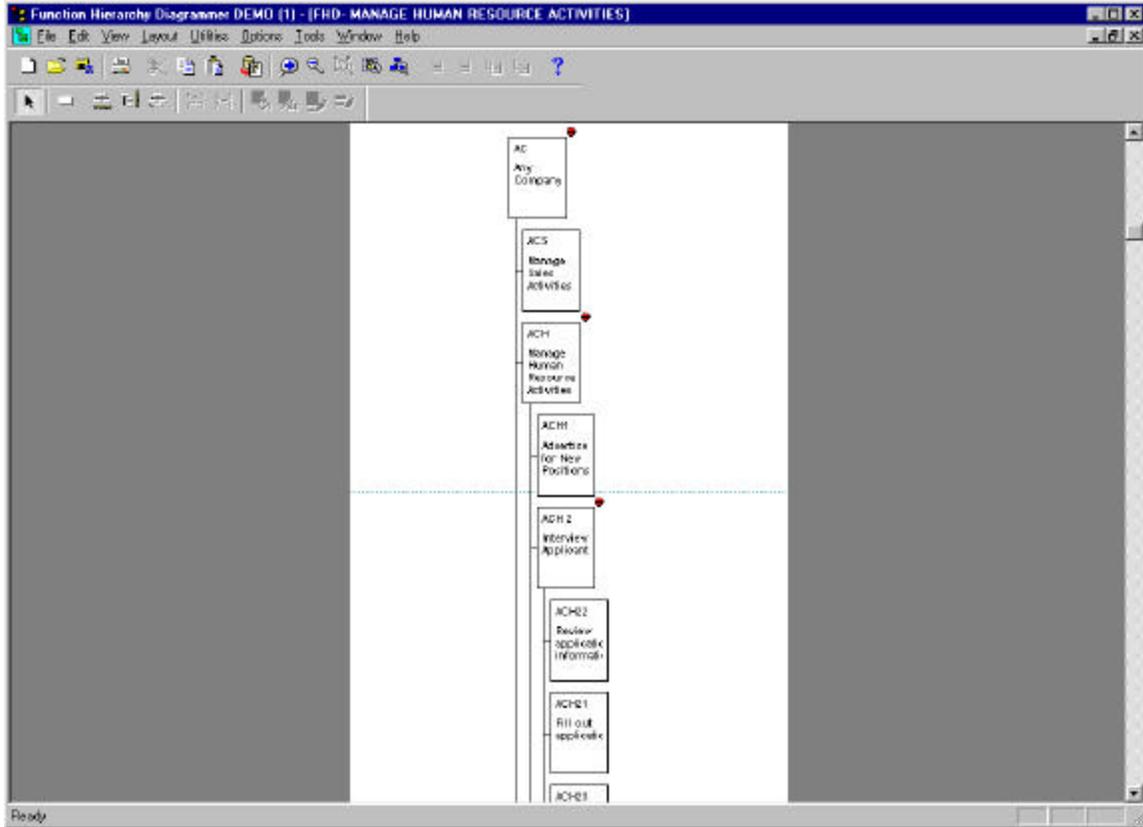


Figure FHD-2 Example FHD displayed in vertical layout style

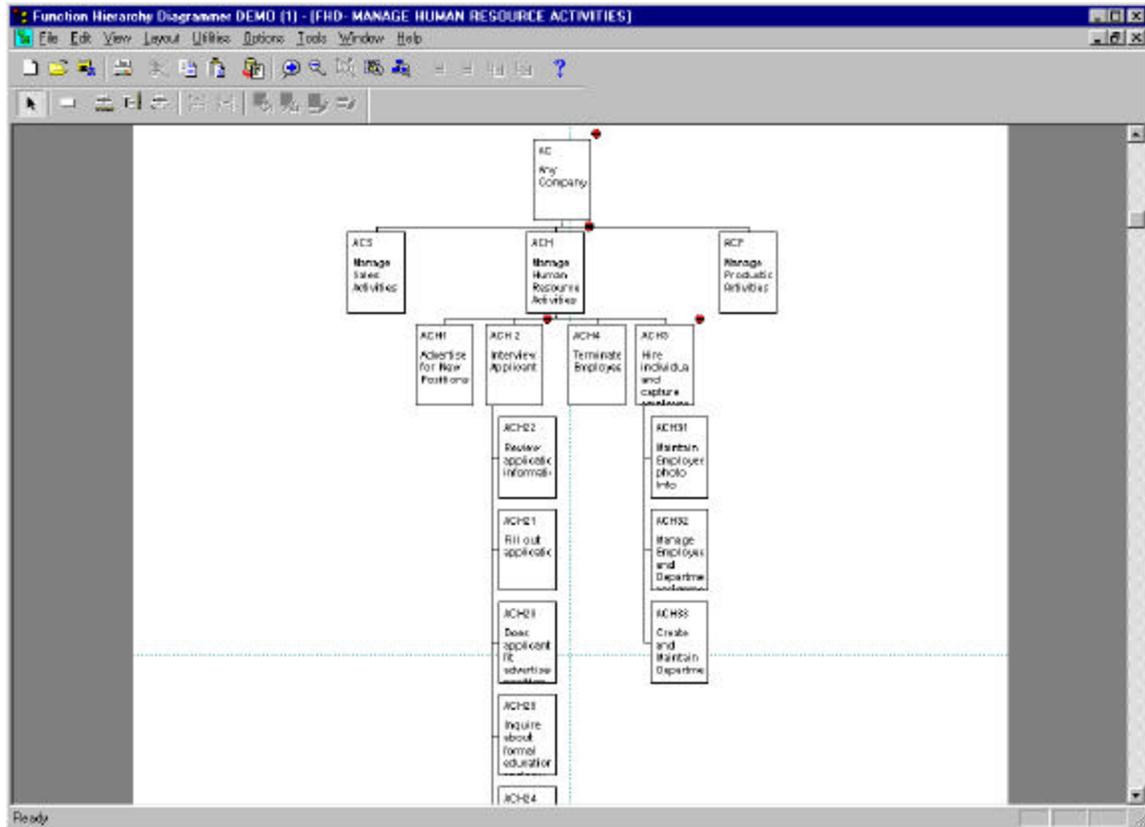


Figure FHD-3 Example FHD displayed in hybrid layout style

### 6.3.4 Color Coding Diagrams

DIA-46 Use color coding or shading to distinguish the following:

Elementary functions, functions that have been decomposed to their lowest level

External functions, functions that are performed by external organizations but may have an effect on our organizations processes

Functions that are the basis of other diagrams (i.e. functions that are decomposed further.)

### 6.3.5 General Guidelines

DIA-47 The function hierarchy should be well balanced; that is, all top-level functions should be decomposed to approximately the same number of levels.

DIA-48 Do not use more than ten subordinate functions per parent function; aim for six.

DIA-49 Create a dedicated branch for business rules that you record as a function 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

---

## 6.4 Entity Relationship Diagrams (ERD)

DIA-50 These guidelines should be implemented when creating Entity Relationship Diagrams via the Oracle Designer Entity Relationship Diagrammer.

### 6.4.1 Multiple Modular Diagrams

DIA-51 Rather than creating one large diagram, create many smaller diagrams for the purposes of maintaining entity relationships. This will eliminate the need to maintain one large diagram that will over time become unreadable, unmanageable, and overwhelming to non-technical users. Create larger diagrams, as necessary, for the purpose of creating consolidated views of the overall database.

### 6.4.2 Diagram Correspondence to Business Model

DIA-52 Each diagram should correspond to a logical section of the overall business model. They could represent a specific business area or logical data view (In data warehousing terms these would be subject areas.) For example; If you have a high level function called “Capture Customer Information”, create a diagram that will depict the data needed to support that business function.

DIA-53 Do not include entities that are not needed to support that high level function. The entity CONTRACT should appear on a diagram called “Customer Contract” along with all the other entities to support the creation and administration of a Contract, but it may not be needed on a diagram showing customer invoicing.

### 6.4.3 Avoid Overcrowding

DIA-54 Do not crowd many entities onto a single diagram. In general, one diagram should contain no more than 15 to 20 entities. This does not apply to diagrams created to present a consolidated overview of the database.

DIA-55 If a single diagram contains more than 15 entities, consider a logical refinement (e.g., the CUSTOMER INFO diagram would be subdivided into two diagrams: CUSTOMER INFO1 and CUSTOMER INFO2).

### 6.4.4 ERD Layout

In this section, please refer to the example diagram, Figure ERD-1.

Following are the guidelines to be used for determining the layout of an ER diagram:

---

#### 6.4.4.1 Entity Placement

DIA-56 Place fundamental entities to the right (preferred) or bottom of the diagram.

DIA-57 Size entities to reduce clutter and provide clear paths for relationships.

DIA-58 Before creating a new entity, always use the *Edit->Include* facility first to prevent creating duplicate entities and relationships.

**6.4.4.2 Drawing Relationships**

- DIA-59 Show crow’s feet facing to the left (West, preferred) or up (North).
- DIA-60 Avoid line crossings if at all possible.
- DIA-61 Keep relationship lines horizontal (preferred) or vertical; do not angle them.
- DIA-62 Minimize the use of bent lines (straight lines are preferred).

**6.4.4.3 Relationship name placement**

- DIA-63 Place relationship names in a clockwise pattern.

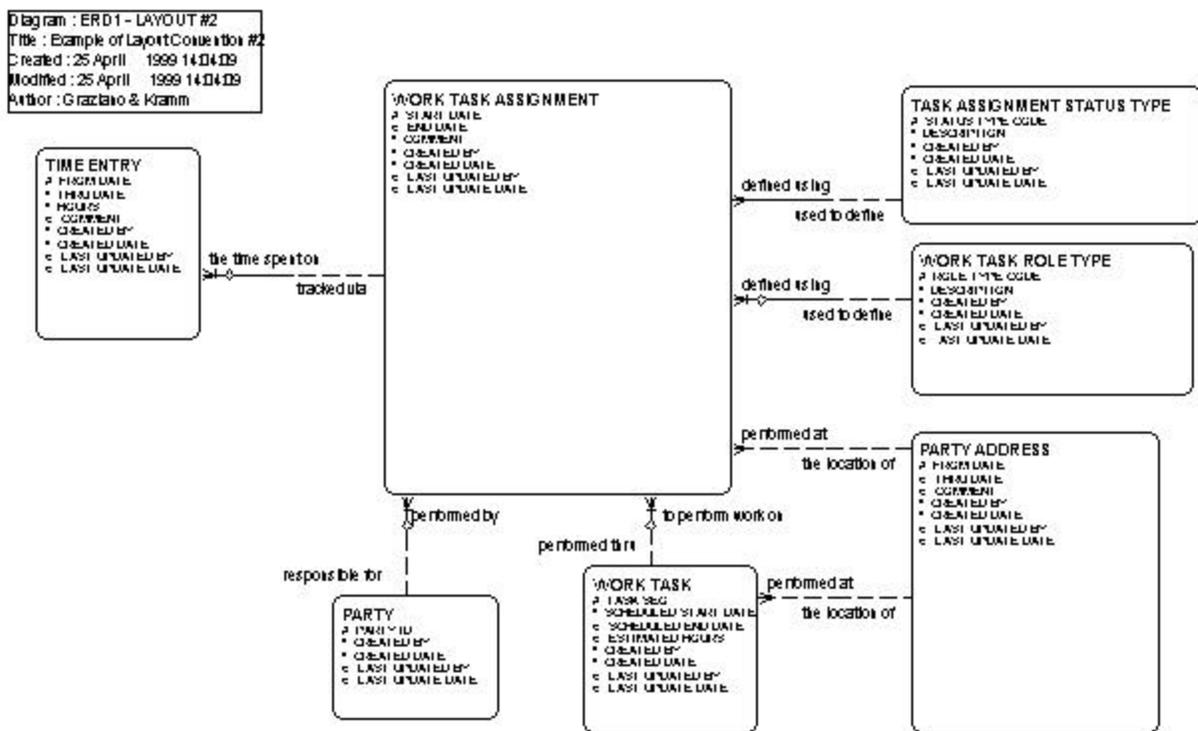


Figure ERD-1

**6.4.5 Drawing Relationships to Sub-types**

- DIA-64 When drawing relationships to a sub-type entity, ensure that the line(s) between the sub-type and other entities are clearly connected to the sub-type and not its super-type.
- DIA-65 Use of sub-types implies that they will have attributes that are specific to them. If this is not the case, do not use sub-type constructs where simple examples will do.

## 6.4.6 Color Coding Diagrams

DIA-66 Use color coding or shading to distinguish the following:

- Entities shared from other applications
- Entities that appear on more than one diagram
- Code type entities
- Subject area groupings

## 6.4.7 View Options

DIA-67 Strategy phase diagrams should include:

- Fundamental and transaction entities (i.e., no codes)
- Relationship lines
- Relationship names

DIA-68 Strategy phase diagrams may also include (where defined):

- Primary UID attributes
- Primary UID relationship indicators

DIA-69 Analysis phase diagrams should include (where they exist):

- All entities including code types and intersection entities
- All attributes
- All relationships
- Relationship names
- Primary UID relationship indicators
- Relationship transferability indicators
- Hidden arcs

## 6.4.8 General ERD Guidelines

DIA-70 Every conceptual model must be accompanied by a graphical representation of that model.

DIA-71 Make at least one diagram showing all entities, relationships, and arcs for the entire application.

DIA-72 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.

## 6.4.9 Drawing Conventions

DIA-73 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.

### 6.4.9.1 Attributes

---

DIA-74 All attributes that are part of the first unique identifier should be listed inside the softbox representing the entity to which they belong.

### 6.4.9.2 Sub-types

---

DIA-75 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

#### **6.4.9.3 Super-types**

---

DIA-76 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

#### **6.4.9.4 Relationship Lines**

---

DIA-77 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

#### **6.4.9.5 Layout**

---

DIA-78 Avoid making drastic changes to the basic layout of an entity relationship diagram (ERD) 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.

DIA-79 Give the core entities of your system a central place in your diagram.

DIA-80 Place entities that are often referred to, but that are not specific to the business, somewhere at the border of your diagram.

DIA-81 Try to avoid the use of arcs. It is usually worthwhile to replace the arc with sub-types.

DIA-82 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.

---

## 6.5 Server Model Diagrams (SMD)

DIA-83 These guidelines should be implemented when creating Server Model Diagrams via the Oracle Designer Design Editor.

### 6.5.1 Multiple Modular Diagrams

DIA-84 Rather than creating one large diagram, create many smaller diagrams. This will eliminate the need to maintain one large diagram that will over time become unreadable, unmanageable, and overwhelming to non-technical users. If you need to answer questions such as, what tables have foreign keys to other tables, use the Design Editor (usages nodes) or the Table Definition report. Do not rely on one large diagram.

### 6.5.2 Diagram Correspondence to Business Model

DIA-85 Each diagram should correspond to a logical section of the overall business model. They could represent a specific business area or logical data view. (In data warehousing terms these would be subject areas.) For example; If you have a high level function called “Capture Customer Information”, create a diagram that will depict the data structures needed to support that business function.

DIA-86 Do not include entities that are not needed to support that high level function. A table CONTRACTS should appear on a diagram called “Customer Contract” along with all the other tables and views to support the creation and administration of a Contract, but it may not be needed on a diagram showing customer invoicing.

### 6.5.3 Avoid Overcrowding

DIA-87 Do not crowd many objects onto a single diagram. In general, one diagram should contain no more than 15 to 20 tables or views.

DIA-88 If a single diagram contains more than 15 objects, consider a logical refinement (e.g., a CUSTOMER INFO diagram would be subdivided into two diagrams: CUSTOMER INFO1 and CUSTOMER INFO2).

### 6.5.4 Server Model Diagram Layout

DIA-89 In this section, please refer to the example diagram, Figure SMD-1.

DIA-90 Following are the guidelines for to be used for determining the layout of a SM diagram:

#### 6.5.4.1 Object Placement

---

DIA-91 Place fundamental objects to the right of the diagram.

DIA-92 Size objects so all columns can be seen and provide clear paths for foreign keys.

DIA-93 Always use the *Edit->Include* facility first to prevent creation of duplicate objects and constraints.

**6.5.4.2 Drawing Foreign Key Constraints**

- DIA-94 Show crow’s feet facing to the left (West, preferred) or up (North).
- DIA-95 Avoid line crossings if at all possible.
- DIA-96 Keep lines horizontal (preferred) or vertical; do not angle them.
- DIA-97 Minimize the use of bent lines (straight lines are preferred).

**6.5.4.3 Constraint name placement**

- DIA-98 For development purposes, it is useful to display the foreign key constraint names, and column datatype and length properties. This can be done within the "Options - Customize" menu.

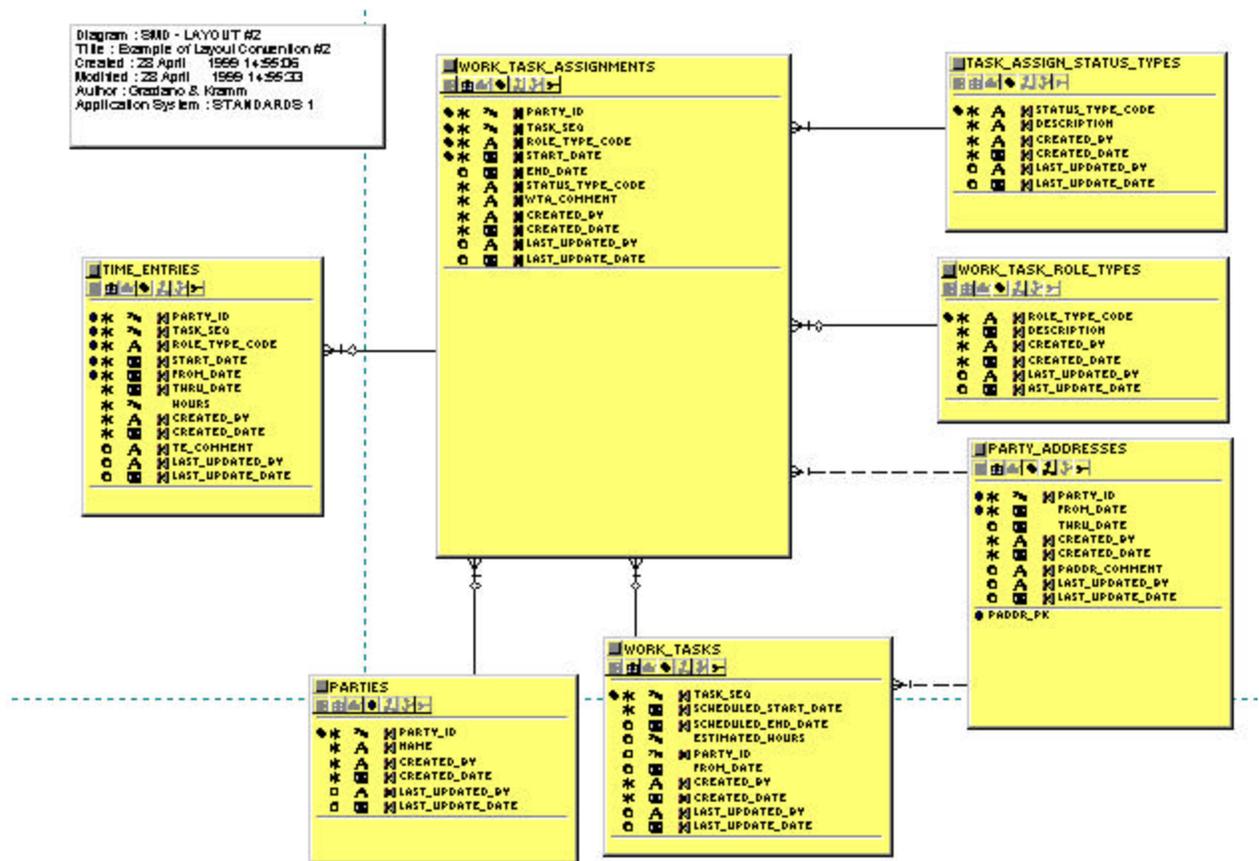


Figure SMD-1

**6.5.5 Color Coding Diagrams**

- DIA-99 Use color coding or shading to distinguish the following:

- Tables
- Views
- Snapshots

- Clusters
- Objects shared from other applications
- Objects that appear on more than one diagram
- Code type tables or views
- Subject area groupings

### **6.5.6 View Options**

DIA-100 Each Server Model diagram created for a logical area should include (where they exist):

- All tables and views within that logical area
- All columns
- All foreign key constraints
- All column detail indicators