Part 8:

Business Processes in Oracle Designer

References:

- Barker/Longman: CASE*Method, Function and Process Modelling, Addison-Wesley, 1992, ISBN 0-201-56525-0, ca. $63.
- Koletzke/Dorsey: Oracle Designer Handbook, 2nd Edition. ORACLE Press, 1998, ISBN 0-07-882417-6, ca. $40.
- A. Lulushi: Inside Oracle Designer/2000. Prentice Hall, 1998, ISBN 0-13-849753-2, ca. $50.
- Oracle/Martin Wykes: Designer/2000, Release 2.1.1, Tutorial. Part No. Z23274-02, Oracle, 1998.
- Online Help of the Oracle Designer.
- Scheer: Business Process Engineering: Reference Models for Industrial Enterprises. Springer Verlag, 1998, ISBN 3540638679.
- Scheer: ARIS: Business Process Modeling, 2nd Edition. Springer Verlag, 1998, ISBN 3540644385.

Objectives

After completing this chapter, you should be able to:

- explain the relationship between modeling data and tasks (usages of data).
- enumerate the basic constructs of business process diagrams and function hierarchy diagrams. Read/write such diagrams.
- decide which Designer Tool is best suited for a certain task.

  Tools mentioned in this chapter: Process Modeler, Function Hierarchy Diagrammer, Matrix Diagrammer, Application Design Transformer, Design Editor.

The time is not sufficient for a detailed study. I recommend that you work through the tutorial shipped with Oracle Designer.

Overview

1. Process Modeler
2. Function Hierarchy Diagrammer
3. Matrix Diagrammer
4. Application Design Transformer
5. Design Editor

Introduction (1)

- The data is collected in order to support some real-world tasks by means of application programs.
- So we should model also these tasks:
  - First, we must document which programs are needed (they will be developed as part of the project).
  - Second, we can check whether the collected data is sufficient to support the tasks.
  - Third, we can check whether the programs are complete to manipulate the collected data.

    E.g. it would normally be a problem when there is no program to insert data into a table. But static look-up tables may be filled directly with SQL statements.

Introduction (2)

- ER-Diagrams model only the static structure of the data.
- Now we want to model also dynamic aspects, and how activities in the real world are using the data.
- Note that data is nevertheless relatively independent of programs:
  - New programs may later be developed which work on the same data.
  - A DBMS supports ad-hoc queries directly in SQL.
- This is an important difference to other software engineering projects where data is only seen in the context of programs.

Introduction (3)

- Not every task can be supported by programs:
  - When we develop process diagrams, we only model what is done or needs to be done.
  - Later, we decide which tasks are supported by programs or even automated.
- You should only describe <u>what</u> is done. If you describe too much <u>how</u> it is done, you lose implementation options.

  The employees should have some freedom.
  So don't go into too much detail describing the process.

Process Diagrams (1)

- A process diagram in the Oracle Process Modeler contains:
  - Process Steps: Tasks/Activities in a business process.

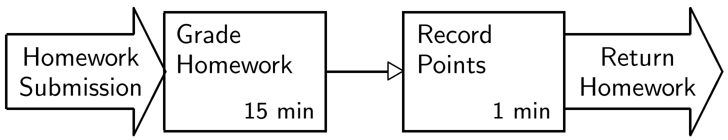
    E.g. "grade a homework", "record points".
  - Flow: An edge between process steps, describing the movement of information.

    Flows can describe the flow of data, material, or time-based dependencies.
  - Trigger: An event that initiates a business process.

    E.g. "customer calls", "end of the month reached".
  - Outcome: Event that results from the business process.

    E.g. "Grades delivered to registrar", "student satisfied".
  - Store: Collection of information or materials.

Process Diagrams (2)



This diagram contains:

- Trigger: "Homework Submission".
- Process Steps: "Grade Homework" and "Record Points".
- Flow: Between the two process steps.
- Outcome: "Return Homework".

Information About Process Steps (1)

Properties of Process Steps:

- Whether this task is going to be supported by programs.
  "To be automated".
- Type of the process step.
  Data Entry, Decision Point, External, Internal, Report, Process Step (this is the default).
- The time needed to perform the task.
  You can differentiate between prior delay, work, quality check, post delay, and also store two measured times.
- Cost (e.g. cost of the person performing the task per hour).
- Frequency (how often is this task performed per time unit).

Information About Process Steps (2)

- Resources needed to perform the task.
  But there is no distinction between resources which are used up and resources which can be used multiple times. E.g. the manual lists "a fax machine" and "100 tonnes of potatoes" as examples.
- Yield
  What percentage of the product passes the quality check.
- Multimedia information (for animation and illustration).
  You can select an icon for the process step, and associate sound, video, an image or a program call with it.
- Various texts
  Description, Notes, Value Chain Analysis, ...

Hierarchy of Process Steps

- There is a hierarchy of process steps:
  - You can refine a process step as a process diagram.
  - Each diagram describes a higher level process step.
- When you create a new process diagram, you get a dialog box for selecting the parent process step.
  You can also choose to create a new root process.
- The parent process is shown in the title bar of the diagram.
- With "`File`→`Open Up`" you open the diagram for the parent process of the current diagram.
- After selecting a process step in a diagram, you can use "`File`→`Open Down`" to open its diagram.

Organization Units (1)

- The Process Modeler also allows to model organization units (business units, departments, maybe single persons).
  E.g. the registration office, the instructor, the GSA, etc.
- Each process step (task) can be assigned to an organization unit which is responsible for performing it.
- The process modeler shows the organization units on the left of the diagram. Each organization unit owns a horizontal stripe ("swim lane") of the diagram.
  "Swim lanes" are delimited by thin horizontal lines or shown in different colors (`Options`→`Customize`→`Graphics`).
  The size of a swim lane is changed by selecting the organization unit and pressing shift and cursor up/down.

Organization Units (2)

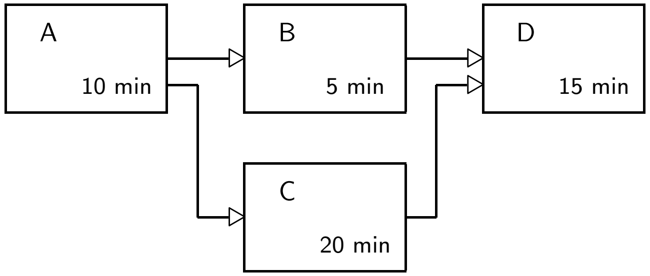
- Organization units also form a tree. E.g. if something has to be signed by the department head, he/she might be seen as a subunit of the department.

  You create a new root organization unit by clicking on the "tree" symbol in the drawing palette and then clicking into the "Unspecified" unit which is always shown at the bottom. If you click into an existing unit, a new child unit is created. The "`View`" menu contains "`Drill Up Organization`" and "`Drill Down Organization`".

- Koletzke/Dorsey state that nontechnologists can instantly relate to organization units and that this type of depiction is not provided by many other CASE tools.

Flows (1)

- Flows express a dependency: E.g. here "B" can only be started after "A" is finished, and the same for "C". "D" can only be started after both, "B" and "C" are done.



- So the total task will take at least 45 min.

Flows (2)

- The Process Modeler can compute the "critical path", i.e. those process steps which determine the duration of the overall task and for which any delay would immediately result in a delay of the complete task.

  You must select the first process step, from which you want to compute the critical path. Then choose the menu item "`Utilities`→`Calculate Critical Path`". You can choose whether the computed time should be stored for the parent process step in the repository.

- You can select an "AND" or "OR" logic for flows. This is relevant if a process step has more than one incoming flow.
- "OR" flows are shown with a crossing bar:

Flows (3)

- The Oracle documentation says:
  - "Selecting the OR option button specifies that the destination is activated as soon as either this flow or another flow (or a group of flows that are ANDed together) is completed."
  - "Selecting the AND option button specifies that the destination is activated only when all flows marked AND are completed."
- I used an OR flow and two AND flows. Then one of the AND flows determined the critical path although the OR flow came from a much shorter step (?).

  Probably, at the destination one should have conditions like "`(X AND Y) OR Z`" or "`X AND (Y OR Z)`". This cannot be expressed by only assigning AND or OR to the flows.

Flows (4)

- You can attach conditions to flows: Only if this condition is met, control/information/material will actually be passed.
- E.g. you can have a process step of type "Decision Point" with two outgoing flows, one with the condition "class closed", and the other with condition "seats available".
- You can also define the flow percentage (how often the process follows this path as opposed to another.

  E.g. "seats available" happens in 95% of the cases, and "class closed" in 5%.
- You can also store times, frequencies, costs, resources, multimedia information and texts for flows.

  This is the same as for process steps (see above).

More on Diagram Element Data

Information about Triggers and Outcomes:

- Type: System (e.g. inventory stock goes below minimum), Time (e.g. first of month), Other (e.g. Student comes).
- Frequency: e.g. 3 per hour.
- Text: description or notes.

Validity Checking:

- It seems that very little or none cross checking is done.
- E.g. you can define that a trigger happens 10 times per hour, but the process step which it triggers happens only once per hour. (Also flows are not checked for consistency.)

More on Process Diagrams

Process Diagrams can be shown in three modes:

- Iconic: Process steps are shown as icons.
- Symbol: As above.
- Enhanced Symbol: Different graphical representations of different process step types (e.g. data entry, decison point).

Drawing Tips:

- There is an invisible grid which cannot be changed.
- The size of process steps can be changed.
- The "paper" is extended when you drag an object to the right.
- To create or remove dog-legs, use shift-click.

Diagram Animation

- The process modeler can animate diagrams in iconic mode.

  You select a start process step and then choose "`Utilities`→`Animation`→`Start`".
- You can define a time step, e.g. 5 Minutes. Whatever you defined as a time step will be compressed to half a second animation time.

  During the time this task is performed, the Process Modeler will cycle through three icon files, e.g. `BIKE1.BMP`, `BIKE2.BMP`, `BIKE3.BMP`.
- You can choose that certain flows will not participate in the animation. They will be shown crossed out on the diagram.

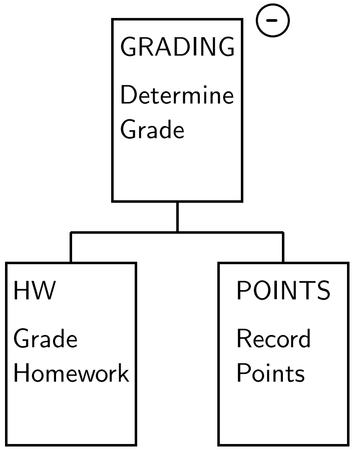
Overview

1. Process Modeler
2. Function Hierarchy Diagrammer
3. Matrix Diagrammer
4. Application Design Transformer
5. Design Editor

Function Hierarchy Diagrammer (1)

- Functions shown in the Function Hierarchy Diagrammer are the same repository objects as process steps (see above).
- So when you have created process diagrams, you can create function hierarchy diagrams simply by selecting a root process step from which to start.
- Then the tree of this process step, its child process steps, the grandchild process steps etc. will be shown.
- Flows are not contained in these diagrams.
- Function hierarchy diagrams are intended to show the refinement starting with a main business function and decomposing it into subtasks, their subtasks, and so on.

Function Hierarchy Diagrammer (2)



Function Hierarchy Diagrammer (3)

- You can choose a layout: Horizontal, Vertical, Hybrid.
  - The picture on the previous page shows horizontal layout: Children of the root are placed in one horizontal line.
- The minus or plus sign in the red circle allows to expand/collapse a subtree (i.e. show/hide child functions).
  - You must double-click in this sign.
- You can also select the root node shown with "`Set Focus`".
  - The focus functions are in the "`File`" menu.
- If you change the size of one function box, all are changed in the same way (i.e. all boxes must have the same size).
- One diagram can contain multiple root nodes.

Function Hierarchy Diagrammer (4)

- You can change the sequence of the child nodes and reparent them.

  Drag the function box you want to move. When you are above another function box, the mouse cursor turns into a symbol showing the action: If you drop a function A in the right third of another function B, A will become the right brother of A. If you drop A in the left third of B, it will become the left brother. If you drop A it in the center of B, A will become child of B.
- If you delete a business function, the entire subtree below it will be deleted, too (cannot be undone).
- You can also create new functions (only symbol in palette).

Function Properties (1)

- The properties of business functions are a bit different than the properties of process steps, although both are two different views of the same object.
- Functions correspond more to modules of the application system to be developed, whereas process steps refer to the real world.
- The frequency of the function is recorded (how often it is called per hour), and whether an immediate response is required (maybe it can be executed overnight).
- You can store that when this function is completed, it will trigger other functions (this is not the same as a flow).

Function Properties (2)

- The most important part of the function properties are the entity and attribute usages.
- You can specify which entity types are accessed by this function.
- For each such entity type, you can check whether instances are created, retrieved (in queries), updated, deleted, archived.

  Also “other” usages can be recorded (whatever that can be).
- You can also store for single attributes, whether they will be inserted, retrieved, updated, nullified, or archived.

  “Insert” means that they are actually specified in an insert (i.e. not null), “nullify” means that they are set to null.

Overview

Matrix Diagrammer (1)

- The Matrix Diagrammer allows to view information from the repository in form of a 2D-matrix (table/spreadsheet).

  Some 3D-tables are also supported, with the third dimension implemented as tabs.

- E.g. one can choose business functions as rows and entity types as columns.

  There are 66 possible selections for the information in the rows. Once you have selected the row entries, the column selection box shows only associated repository object types.

- In the matrix entries, you can display for each business function and entity type whether the function creates / retrieves / updates / deletes entities of this type (CRUD).

Matrix Diagrammer (2)

- You can also select what exactly to show in the row and column headings (e.g. the entity name or short name), and how to display the information in the matrix entries.
- When you click into a matrix cell, the property palette allows to view and change all properties of this association.

  In the same way, you can get the property palette of the row and column headings.

- Viewing the matrix needs a large monitor and high resolution (the iconic mode might be slightly better).
- This matrix allows you to check that your business functions are complete (e.g. entities of all types can be created) and that you have no superflous (unused) entities.

Overview

1. Process Modeler
2. Function Hierarchy Diagrammer
3. Matrix Diagrammer
4. Application Design Transformer
5. Design Editor

Application Design Transformer (1)

- The purpose of the Application Design Transformer is to create a first-cut application design out of your business function specifications.
- An application program or part of an application program is called a module in Oracle Designer.

  Often, a module is the implementation of one business function. But business functions with the same data usages might be merged into one module, and business functions with complex data usages might be split. E.g. an Oracle Developer Form would be one module.

- Oracle Designer distinguishes application modules (short: modules) and menu module networks (short: menus).

Application Design Transformer (2)

- If you haven't specified data usages for a business function, it will be mapped into a manual module.

  I.e. no code will be generated. If you want to generate an application program, you must specify which entities are used by a business function. Also, you must generate the tables with the Database Design Transformer before you use the Application Design Transformer.

- A module will only be created for leaf functions, which are not further decomposed.

  This is only true if none of their ancestors is declared as elementary or is a common function (appearing in more than one place in the hierarchy). Modules are then created for the elementary or common function.

Application Design Transformer (3)

- If a business function only retrieves data (no insert, update, delete), it will be mapped to a "Report Module".

  Then by default code for Oracle Developer Reports is generated. However, you can select the language used for Reports. If you specified the reponse needed as "immediate", a screen module (form) will be created instead.

- If its response is specified as "Overnight" and it is not "retrieve only", it will be implemented by a "Utility Module".

  By default, PL/SQL code is generated for utility modules.

- In all other cases: "Screen Module".

  This means by default that code for Oracle Developer Forms is generated.

Application Design Transformer (4)

- "Merging Modules": Business functions can be implemented by a single module if they
  - have the same value for the response needed,
  - the same input parameters,
  - belong to the same organization unit,
  - and have similar data usages

    The "Merge Granularity" can be: "Identical Entities", "Identical Entities and Usages", "Identical Attributes".

- "Splitting Modules": If you have business functions associated with more than one organization unit, they will be implemented by one module for each organization unit.

Application Design Transformer (5)

- Module arguments (parameters) are derived from the dataflows to and from the business functions.
- A module argument is created for each attribute or data item in the dataflow (unless to/from a datastore).
- Note that data can be assigned to flows only in the Dataflow Diagrammer (or with the Repository Object Navigator).
- We didn't discuss the Dataflow Diagrammer here, but it is similar to a combination of the Process Modeler and the Function Hierarchy Diagrammer.

  But without the multimedia and animation capabilities, and without the relation to organization units.

Application Design Transformer (6)

- Generated modules are first only “candidate modules”.
- You must manually accept or reject them before you can continue (e.g. create menus or implement the modules).
- You can do this in the Design Editor or Repository Object Navigator by changing the property “Candidate?” to “No”.
  You can delete candidate modules which you do not need.
- After that, you can produce a menu module network with the Application Design Transfomer.
- The menu groups modules first by application system, then by organization unit, and then by type (sceen, report, utility).
  You also need to change their candidate status manually.

Overview

1. Process Modeler
2. Function Hierarchy Diagrammer
3. Matrix Diagrammer
4. Application Design Transformer
5. Design Editor

Design Editor (1)

- You use the Design Editor once the analysis phase is complete, so you have (at least) entity relationship diagrams and function hierarchy diagrams.
- You then use the Database Design Transformer to generate a relational schema (Tables).
- This can be refined with the Design Editor.
- When you are done, you can use the Design Editor to generate DDL files or to create the tables immediately.
- In the same way, you use the Application Design Transformer to generate Modules (application programs).

Design Editor (2)

- These are also refined by using the Design Editor.
  At least you must remove the candidate mark.
- Then code e.g. for Oracle Developer Forms, Reports, Oracle WebServer, Visual Basic and C++ can be generated.
- You can set a large number of generator preferences.
  In this way you can influence the generated forms and ensure a common look and feel.
- The Design Editor also contains the reverse engineering (design capture) utilities (for existing databases, forms, etc.).
  As well as the “Logic Editor” for creating PL/SQL, the “Module Network Viewer”, and allows to edit information for DB administration and distribution/replication.