


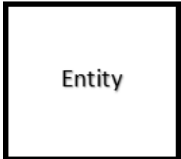



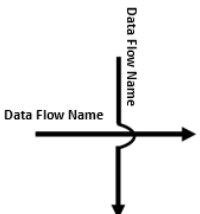
St. Xavier's College (Autonomous), Ahmedabad

Department of Computer Science

Guidelines for various Diagrams used in SDP

Data Flow Diagram Guidelines:

1. Use your don symbols to draw DFD

Symbol	Naming convention	Example
	<ul style="list-style-type: none"> Process name identifies a specific function and consists of a verb (and an adjective , if necessary) followed by a singular noun 	<ul style="list-style-type: none"> Apply rent payment Calculate commission Assign final grade Verify order Fill order Deposit payment
	<ul style="list-style-type: none"> Only external entities that provide data to the system or receive output from the system Entity Name should be a singular form of a department , outside organization , other information system or person 	<ul style="list-style-type: none"> Customer Bank Employee Payroll department Student Student records system Instructor
	<ul style="list-style-type: none"> Name of Data store (table name) appears between the lines Table Name is plural name consisting of a noun and adjectives , if needed Exceptions to the plural name rules are collective nouns that represents multiple occurrences of objects (eg. Gradebook) Data store has atleast one incoming and one outgoing data flow and is connected to a process symbol with a data flow A data store should be connected to atleast one process There is an exception – a data store has no input data flow if it contains fixed reference data that is not updated by the systme (e.g. Taxtable directly connected or downloaded from RBI) 	<ul style="list-style-type: none"> Students Accounts receivable Products Daily payments Purchase orders Outstanding checks Insurance policies Employees
	<ul style="list-style-type: none"> Consist of a single data item (student ID)or it could include a set of data (ID number , name , registration date for a specific class) Data Flow Name appears above, below or alongside the line Consists of a singular noun and an adjective , if needed. Do not cross lines. Overlapping arrows is an exceptional case for the tool you use to draw DFD. (in certain tools, automatically it draws cruve on overlaping place.) 	<ul style="list-style-type: none"> Deposit Invoice payment Student grade Order Commission
		
 <p style="text-align: center;">Overlapping Arrows</p>		

2. Create DFDs:
 - a. Step 1 : Context level diagram (process 0)
 - i. Place the system entities around the perimeter of the page and use data flows to connect the entities to the central process.
 - ii. Data stores are not shown in the context diagram.
 - b. Step 2 : Level 1 DFD (level 1 , it is preferred to have only 9 process, if the system is large processes can go up to 11)
 - c. Step 3 : lower-level diagrams (sub levels , sub processes)
 3. On lower-level diagrams with multiple processes, you should not have more than nine process symbols.
 4. To avoid crossing lines – simply duplicate an entity or data store.
 5. When duplicating a symbol on a diagram, make sure to document the duplication to avoid possible confusion.
 6. An asterisk, next to the symbol name and inside the duplicated symbols signifies that they are duplicated on the diagram.
 7. Provide a unique name and reference number for each process.
 8. All processes should be numbered using a decimal notation consisting of the parent's reference number, a decimal point, and a sequence number within the new diagram.
 9. If the parent process is 1 then reference numbers for child processes would be 1.1 , 1.2 and 1.3 likewise.
 10. If process 1.3 decomposed further then it would be 1.3.1 , 1.3.2 and 1.3.3 likewise.
 11. Do the balancing which ensures that the input and output data flows of the parent DFD are maintained on the child DFD.
 12. Process name, Entity name and table name shouldn't be same.
 13. Generation of report won't be considered as a separate process.
-

ER Diagram Guidelines:

1. Identify all the entities in the system. An entity should appear only once in a particular diagram. Create rectangles for all entities and name them properly.
 2. Identify relationships between entities. Connect them using a line and add a diamond in the middle describing the relationship.
 3. Add attributes for entities. Give meaningful attribute names so they can be understood easily.
 4. Use crow's foot notation to show cardinality and ordinality
-

Data Dictionary Guidelines:

1. List out all the table names
 2. Create data dictionary for all tables, which should include – Table Name , Table Description , All attributes , data type , length , constraint and attribute's short description.
-

Evaluation Parameters:

1. Dressing and overall appearance
2. Behaviour and Language
3. Understanding of concept
4. Presentation design and delivery

Use Case Diagram Guidelines:

- Actors :
 1. Primary actors are placed in the top-left corner of the diagram.
 2. Actors are external objects; hence actors are always shown outside of the system boundary box of a use-case diagram.
 3. Actors are always named with singular, business –relevant nouns.
 4. One actor can be related with one or more use-case.
 5. External system as an actor is shown with notation <<system>>.
 6. Actors never interact with each other directly, but they interact through the system.
 - Use-Case :
 1. Minor steps of functionality should not be considered as use-case. Functionality as a whole must be considered as a valid use-case.
 2. Use-case names must start with a verb.
 3. Use-case name should be identified from business terminology and not the technical term.
 - Relationships:
 1. If an actor is involved in the use-case logic, a relationship is indicated between the actor and the use-case.
 2. Actor use case relationship is not shown with an arrowhead.
-

Activity Diagram Guidelines:

1. One activity diagram should be drawn for each use-case. No two use-cases flow should be mixed into a single activity diagram. An activity diagram can also be drawn for a system or an actor as a whole.
 2. Always an activity diagram should reflect business flow rather than system flow.
 3. Only one initial state element should be drawn in an activity diagram.
 4. In case of swimlanes, the initial state should be placed in the first swimlane.
 5. As a thumb rule, it should be avoided to have more than five swimlanes in a single activity diagram.
 6. An initial state should be connected directly to “Action/Activity” element of the activity diagram and not to any other element.
 7. The initial state must be connected to only one action/activity element and not to multiple action/activity elements.
 8. On every transition, leaving a decision end, a guard condition must be specified.
 9. Every fork must have a corresponding join.
 10. A fork must have only one entering transition and two or more leaving transitions.
 11. A join must have two or more entering transitions and only one leaving transition.
 12. It is always better to provide a name/label for the initial and final states.
-

Class Diagram Guidelines:

1. During analysis, a domain analysis class diagram will be drawn first.
2. Identify, responsibilities of each domain class in domain class diagrams.
3. Visibility (private, protected and public) is indicated only on design models.
4. Design class diagrams should follow languages naming conventions.
5. An association class is modelled on the analysis diagram and the association name is not specified in that case.

6. Draw dashed line from the centre of association joining two classes to the association class
 7. The class name should be singular
 8. The class name, with which the client is comfortable, should be selected, rather than semantically accurate terminology.
-

State Chart Diagram Guidelines:

1. A statechart diagram may be either one shot life cycle type or continuous loop type.
 2. Objects for which one shot life-cycle statechart diagram is drawn have fixed life – from START state till END state.
 3. Objects for which a continuous loop statechart diagram is drawn, do not have an end state.
 4. A separate statechart diagram must be drawn for each object, showing various states of that object throughout its life cycle within the context of the system.
 5. A statechart diagram is drawn for only those classes showing interesting or complex internal behaviour.
 6. The class whose behaviour does not differ based on the state, the statechart diagram for those classes is trivial and of no much use.
 7. While drawing a statechart diagram, first find out the initial state and final state. After this, the intermediate states during the life of an object are identified. The states of an object can be identified by looking at the boundary values of its attributes.
 8. Always place the initial state in the top-left corner and the final state in the bottom right corner.
 9. The state name should be simple and written in present tense
-

Sequence Diagram Guidelines:

1. For each use-case, one sequence diagram is drawn:
 1. For each use-case, identify the object which initiates the task and that will be the actor for a sequence diagram.
 2. Find out the interface (form or screen) through which an actor interacts with the system and that will be the boundary object for a sequence diagram.
 3. Find out the object which does not have the interface and control the use-case and that will be the controller object for sequence diagram.
 4. Find out the object which stores and manages information, usually the database table and that will be the entity object for sequence diagram.
 2. Always have one boundary class per actor/use case pair, one control class per use-case.
 3. Ordering of message sequence is always shown from left to right.
 4. An actor must be the same as specified in a use-case diagram.
 5. An actor can have the same name as a class.
 6. Primary actors must be specified on the left-most side of the diagram.
 7. Reactive system actors must be specified on the right most side of your diagram.
 8. Proactive system actors must be specified on the left-most side of your diagram.
 9. An object can call itself recursively. An arrow commencing and ending at itself denotes this.
-
-