

OnLine Shopping

1. Statement: *Customer Order Process*

A *computer manufacturer* offers the possibility of purchasing computers via the internet. The customer can select a computer on the manufacturer's web page. The computers are classified into servers, desktops, and portables. The customer can select a standard configuration or can build a desired configuration online. The configurable components (such as memory) are presented as picklists of available options. For each new configuration, the system can calculate price.

To place an order, the customer must fill out the shipment and payment information. Acceptable payment methods are credit cards and checks. Once the order has been entered, the system sends a confirmation e-mail message to the customer with details of the order. While waiting for the arrival of the computer, the customer can check the order status online at any time.

The back end order processing consists of the steps needed to verify the customer's credentials and payment method, to request the ordered configuration from the warehouse, to print an invoice, and to request the warehouse to ship the computer to the customer.

2. User Requirements

- R1. The customer uses the manufacturer's online shopping web page to view the standard configuration of the chosen server, desktop, or portable computer. The price is also shown.
- R2. The customer chooses to view the details of the configuration, perhaps with the intention of buying it as is or to build a more suitable configuration. The price for each configuration can be computed at the customer's request.
- R3. The customer may choose to order the computer online or may request that the salesperson contact him/her to explain order details, negotiate the price, etc. before the

order is actually places.

- R4. To place the order, the customer must fill out an online form with shipment and invoice address, and with payment details (credit card or check).
- R5. After the customer's order has been entered into the system, the salesperson sends an electronic request to the warehouse with details of the ordered configuration.
- R6. The details of the transaction, including an order number and customer account number, are e-mailed to the customer, so that the customer can check the status of the order online.
- R7. The warehouse obtains the invoice from the salesperson and ships the computer to the customer.

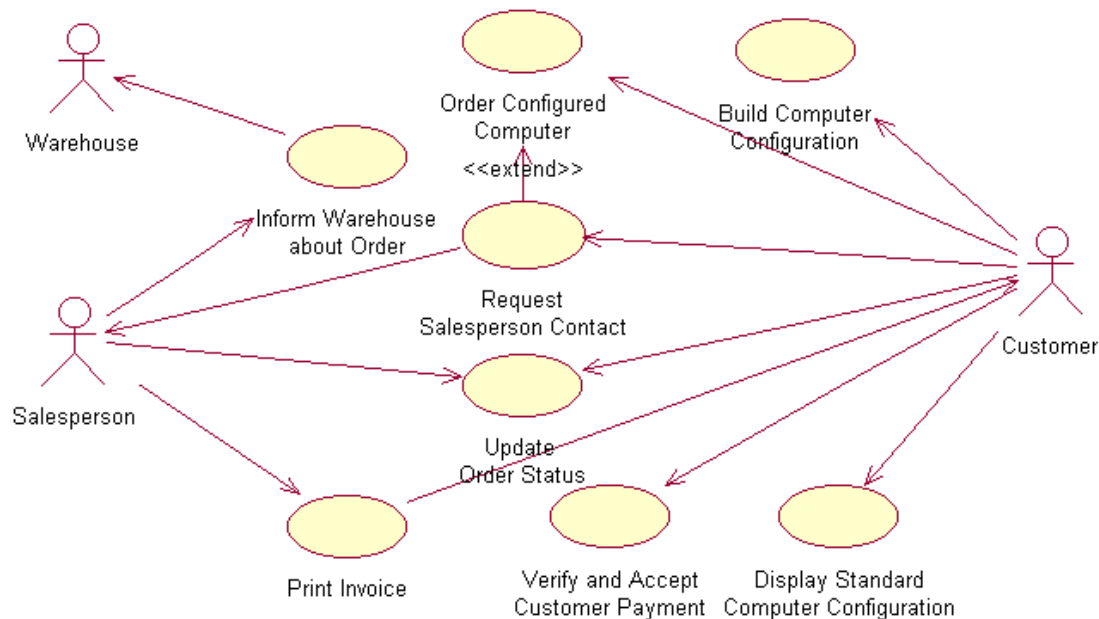
3. Actors and Use Cases

- An *actor* is an entity external to the system that interacts with the system to stimulate it with input events, or to receive useful results from it.
- A *use case* is a narrative document that describes the sequence of events of an actor using a system to complete a process.
 - Use cases can be either derived from the identification of tasks of the actor, or determined from direct analysis of function requirements.
 - There may be some use cases that do not directly interact with actors, but generalize or specialize main use cases.

Req#	Actor	Use case
R1	Customer	Display Standard Computer Configuration
R2	Customer	Build Computer Configuration
R3	Customer, Salesperson	Order Configured Computer, Request Salesperson Contact
R4	Customer	Order Configured Computer, Verify and Accept Customer Payment
R5	Salesperson, Warehouse	Inform Warehouse about Order
R6	Salesperson, Customer	Order Configured Computer, Update Order Status
R7	Salesperson, Warehouse	Print Invoice

4. Use Case Diagram

- Use case diagrams model a systems's behavior, showing its functionality and its interactions with the actors.



5. Documenting Use Cases

- Each use case has to be described in a *flow of events* document, whose typical structure is as the following.
 - *Brief description*
 - *Actors* involved
 - *Preconditions* necessary for the uses case to start
 - *Detailed description* of flow of events that includes:
 - *Main flow* of events that can be broken down to show *subflows*, which can be further divided into smaller subflows
 - *Alternative flows* to define exceptional situations
 - *Postconditions* that define the state of the system after the use case ends

Use case		Order configured computer
Brief description		This use case allows a <i>Customer</i> to enter a purchase order, providing a shipment, invoice address, and payment details.
Actors		<i>Customer</i>
Preconditions		<i>Customer</i> points an Internet browser to the computer manufacturer's order entry web page. The page displays the details of a configured computer together with its price.
Main flow	A1	The use case begins when the <i>Customer</i> decides to order the configured computer by choosing the <code>Continue</code> (or similarly named) function when the order details are displayed on the screen.
	A2	The system requests that the <i>Customer</i> enter the purchase details, including: name of the salesperson (if known), shipment details (customer's name and address), invoice details (if different from shipment details), a payment method

		(credit card or check), and any comments.
	A3	The <i>Customer</i> chooses the <code>Purchase</code> (or similarly named) function to send the order to the manufacturer.
	A4	The system assign a unique order number and a customer account number to the purchase order and it stores the order information in the database.
	A5	The system e-mails the order number and the customer number to the <i>Customer</i> , together with all order details, as the confirmation of the order's acceptance.
Alternative flows	A6	The <i>Customer</i> activates the <code>Purchase</code> function before providing all mandatory information. The system displays an error and it requests that the missing information be supplied.
	A7	The <i>Customer</i> chooses the <code>Reset</code> (or similarly named) function to revert to an empty purchase form. The system allows the <i>Customer</i> to enter the information again.

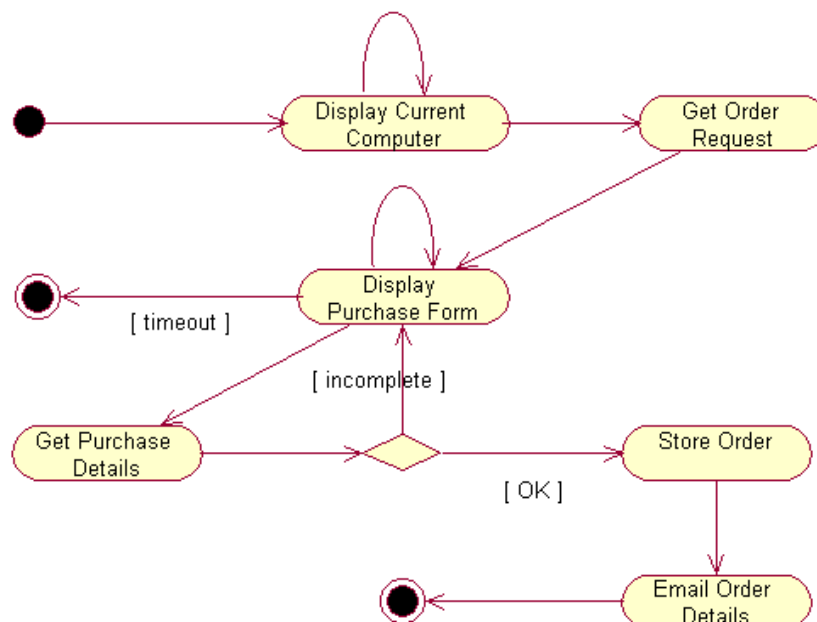
Postconditions	If the use case was successful, the purchase order is recorded in the system's database. Otherwise, the system's state is unchanged.
----------------	--

6. Activities

No.	Activity state
1	Display Current Configuration; Get Order Request
2	Display Purchase Form
3	Get Purchase Details
4	Store Order
5	Email Order Details
6	Get Purchase Details; Display Purchase Form
7	Display Purchase Form

7. Activity Diagram

- *Activity diagrams* model the workflow of a business process and the sequence of activities in a process.



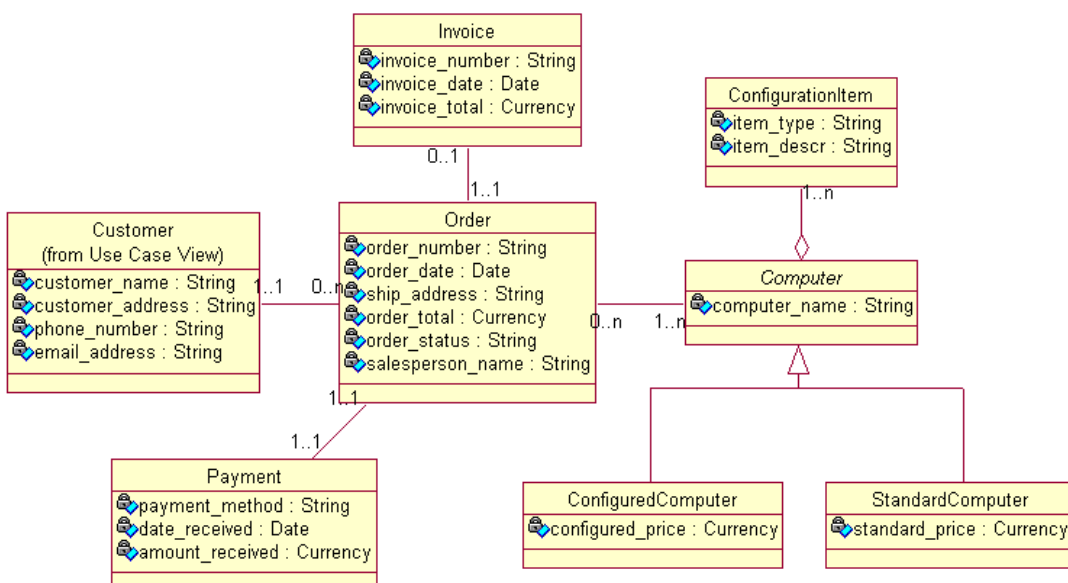
8. Entity Classes

- *Entity classes* are long-lived (persistent) entities that define the essence of a system.

R#	Entity class
R1	Customer, Computer (StandardConfiguration, Product)
R2	Customer, ConfiguredComputer (ConfiguredProduct, ConfigurationItem), ConfigurationItem
R3	Customer, ConfiguredComputer, Order, Salesperson
R4	Customer, Order, Shipment, Invoice, Payment
R5	Customer, Order, Salesperson, ConfiguredComputer, ConfigurationItem
R6	Order, Customer, OrderStatus
R7	Invoice, Shipment

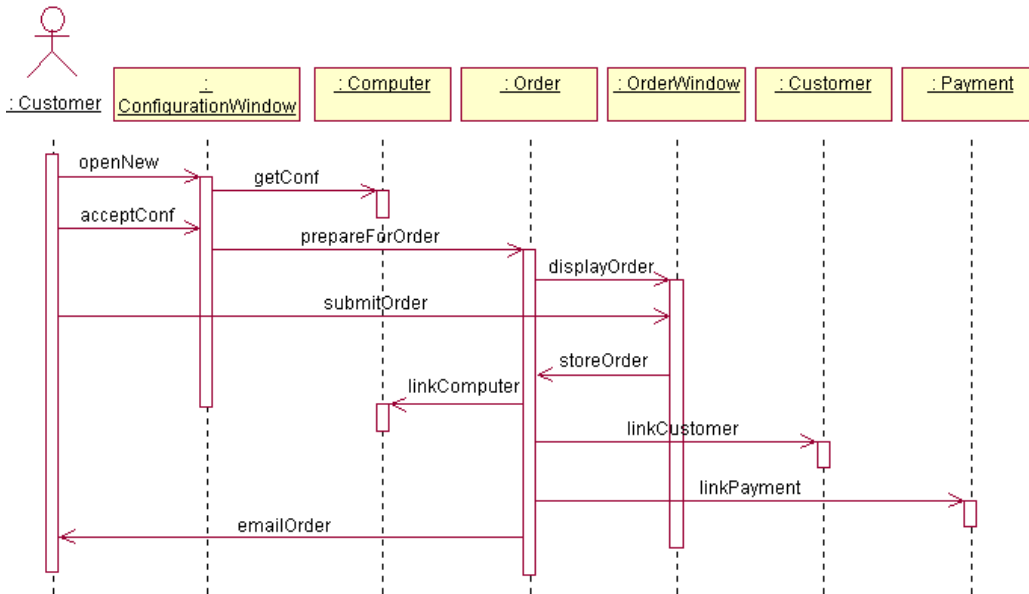
9. Conceptual Model

- A *conceptual model* illustrates entity classes in the problem domain and their relationships.



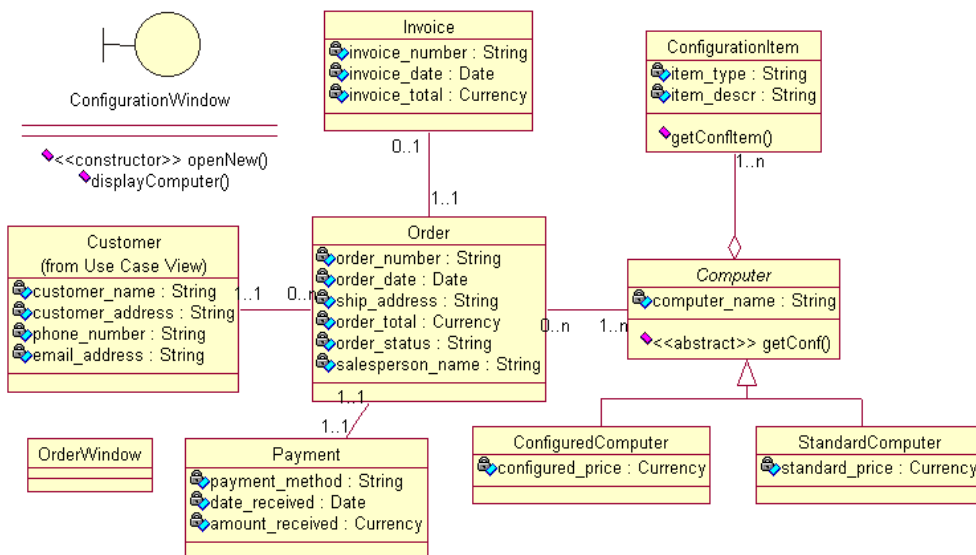
10. Sequence Diagrams

- A *sequence diagram* illustrates object interactions arranged in a time sequence.



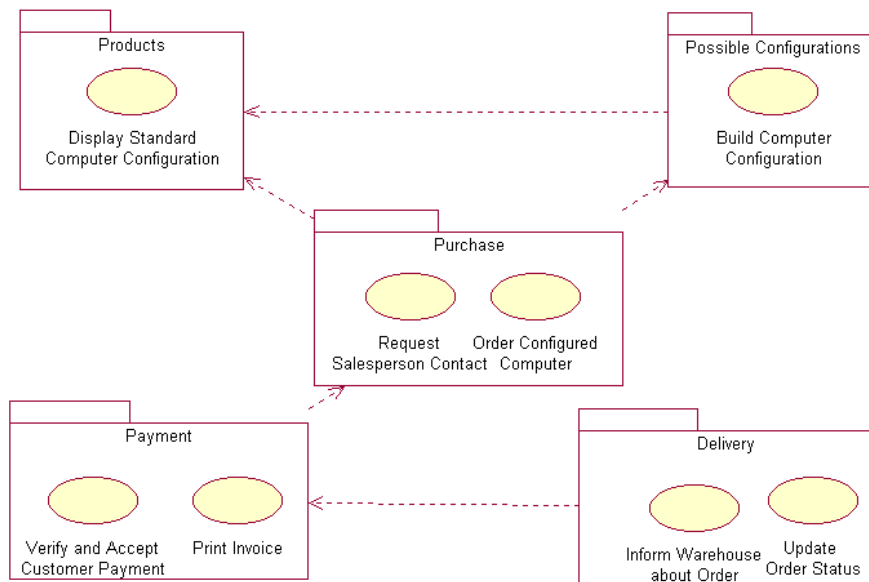
11. Class Diagram

- *Class diagrams* show the relationships among and details about each class.



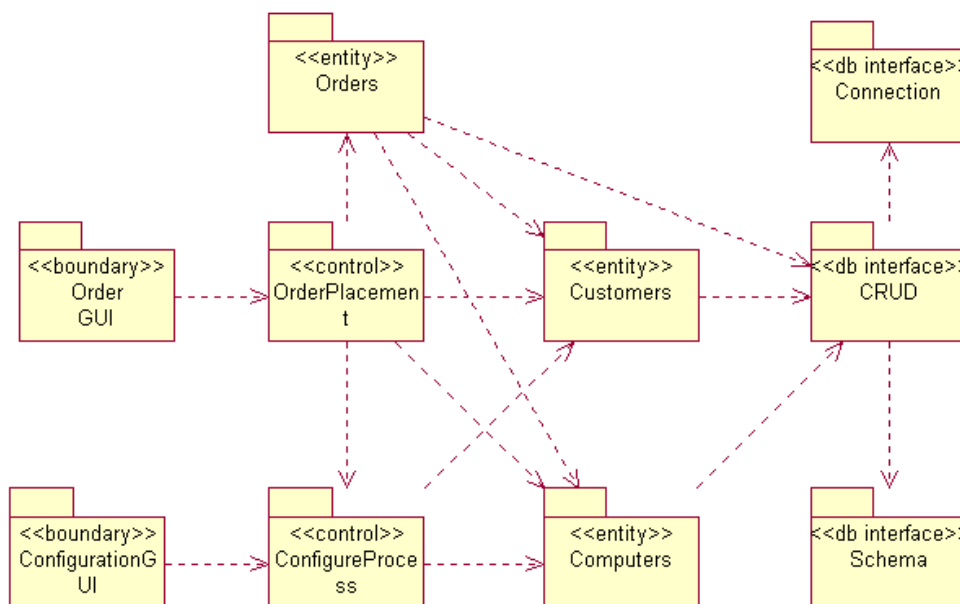
12. Use Case Packages

- *Use case packages* are used in the analysis to identify main clusters of use cases.



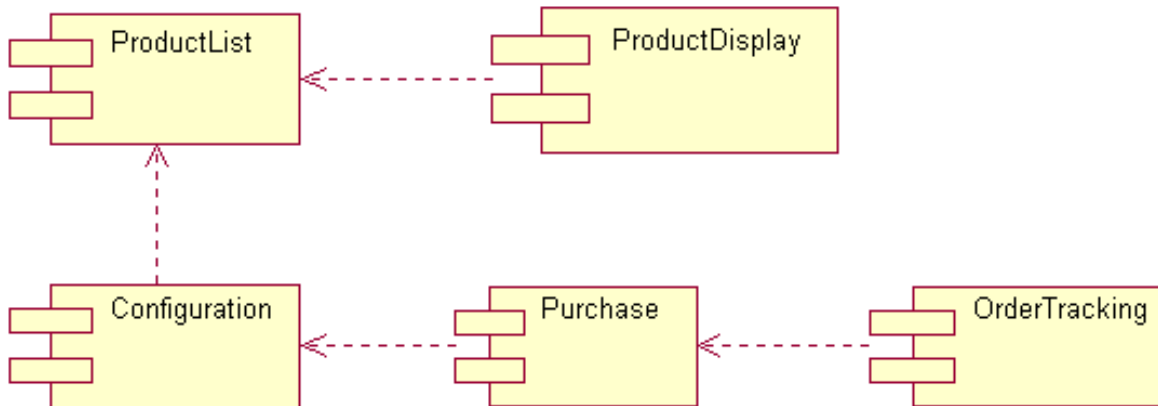
13. Class Packages

- *Class Packages* are used in the design to cluster classes.



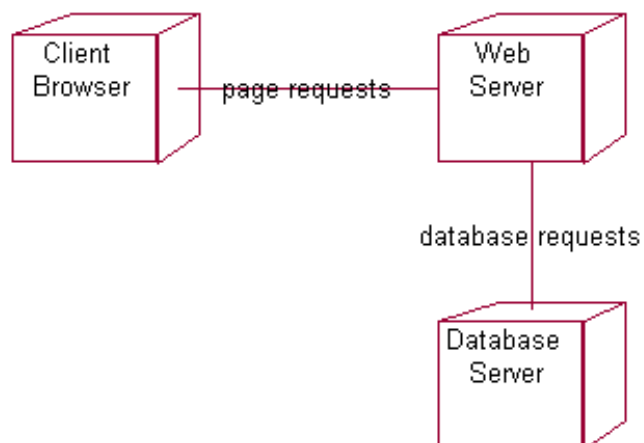
14. Component Diagram

- *Component diagrams* provide a physical view of the system. Each *component* is a cohesive functional unit with clear interfaces so that it becomes a replaceable part of the system.



15. Deployment Diagram

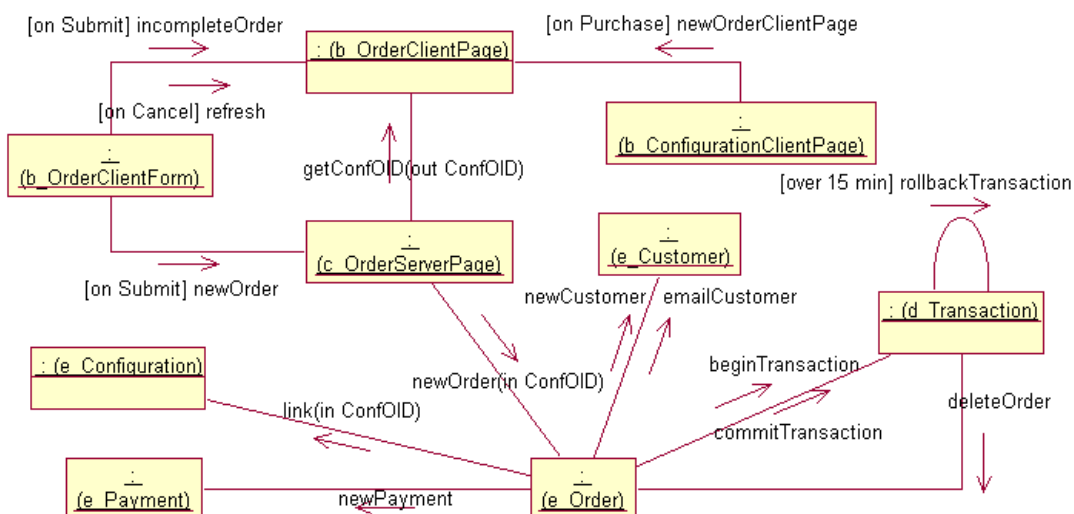
- The *deployment diagram* shows the mapping of processes to hardware. It is most useful in a distributed architecture environment where applications and servers may be at different locations.



16. Collaboration Diagrams

- *Collaboration diagrams* provide a view of the interactions or structural relationships between objects in the current model.
 - *Collaborations* define the realization of use cases and the realization of more intricate operations as simple operations do not have to be modeled as collaborations.
 - The design of collaborations leads invariably to the *elaboration* (modifications and extensions) of existing class diagrams and to the production of new collaboration diagrams (or the elaboration of existing sequence diagrams).

- Other kinds of diagram, in particular statechart diagrams, may also need to be developed or elaborated.



17. Statechart Diagrams

- *Statechart diagrams* are used to model the dynamic behavior of individual classes or objects.

