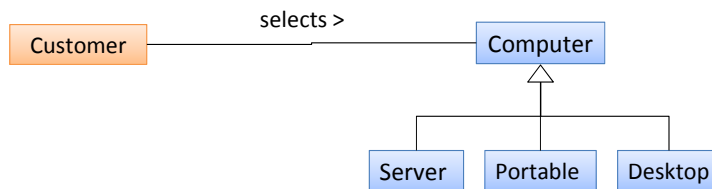


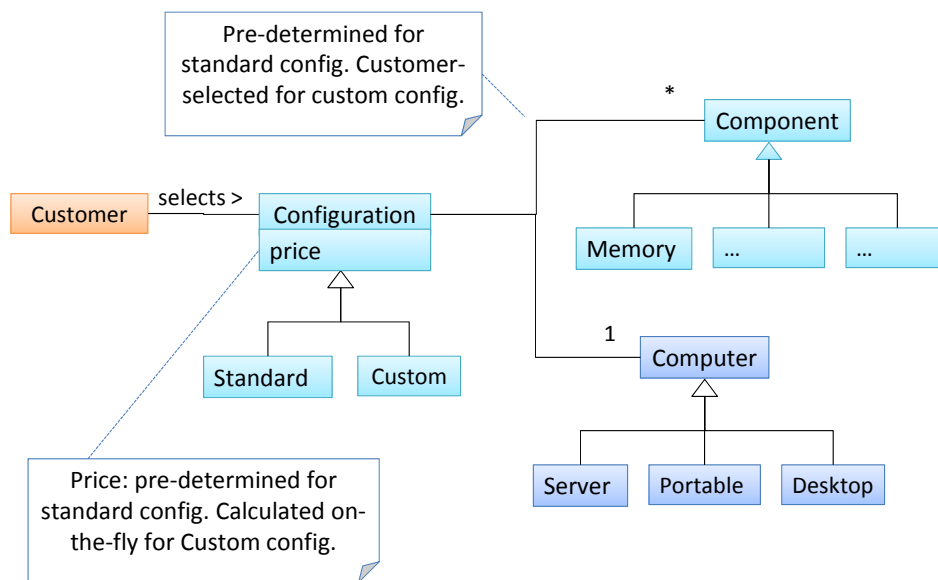
Building a domain model for a computer store

In this example, we build a domain model to fit a given description of how a computer store operates.

A computer manufacturer offers the possibility of purchasing computers via internet. The customer can select a computer on the manufacturer's web page. The computers are classified into servers, desktops, and portables.

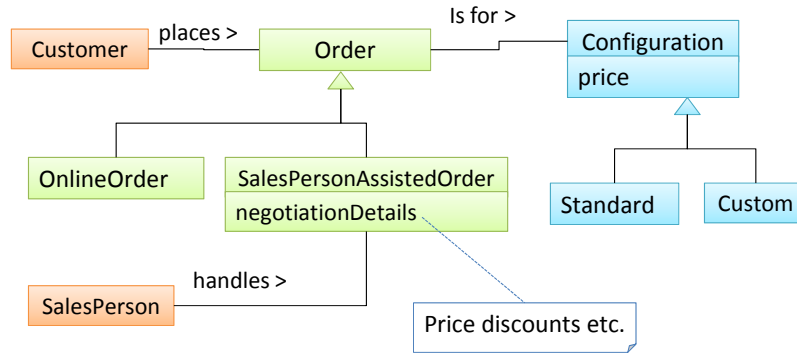


The customer can select a standard configuration or can build a desired configuration online. The configurable components such as memory are presented as lists of available options. For each new configuration, the system can calculate price. Price is shown to the customer when he chooses a standard configuration. Price is also computed when customer customizes a configuration.

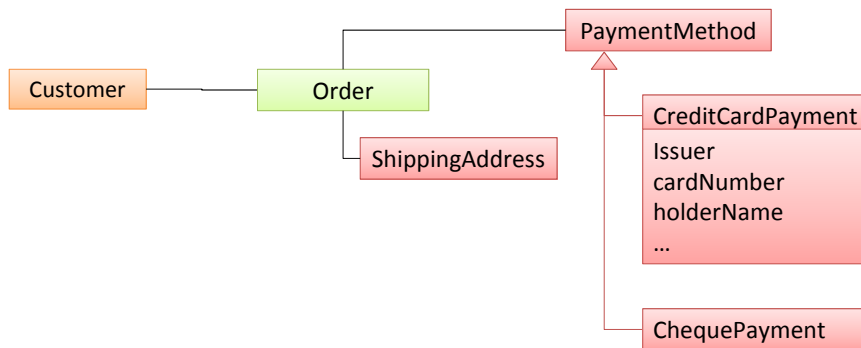


[Note how we ignore UI elements from the domain model. The UI is part of the existing solution, not part of the problem domain]

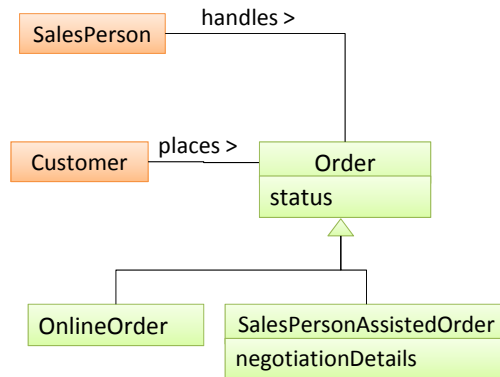
Customer may choose to order the computer online or may request that salesperson contact him/her to explain order details, negotiate price etc before the order is actually placed.



To place an order, the customer must fill out the shipment and payment information. Acceptable payment methods are credit cards and cheques.



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. After customer's order is entered, the salesperson sends an electronic request, an invoice, to warehouse with details of the ordered configuration. The warehouse ships the computer to the customer.



[Notes:

1. it appears that even online orders are subsequently handled by a salesperson. So we replace *SalesPerson—SalesmanAssistedOrder* link with a *SalesPerson—Order* link.
2. We have not added Invoice as a class because it appears to be a temporary entity derived of existing entities. It is ok to include it too.

]