Old Dominion University
Dept. of Computer Science

Asst Solution: Mail Order - early analysis

Steven Zeil

Last Revision Date: February 24, 2005

1. Question 1: Candidate Classes and Responsibilities

For the initial list, mark up the description, looking for *noun phrases* and verb phrases .

> *Fancy Oriental Objects (FOO), Incorporated* is a *mail-order company* specializing in overpriced *knick-knacks* and *gimcracks* imported from Eastern Asia. *FOO* processes *orders* via a number of regional *warehouses* . Each *warehouse* has responsibility for inventory management and order processing . Because *demand* of some *products* varies by *region* , the *inventory* at the *warehouses* may vary considerably. Also, the *products* offered tend to vary considerably from *year* to year.
>
> *Customer* *orders* are received by the *central mail-order facility (MOF)* . The *MOF* checks the *list of products ordered* against the *available stock* in the various *warehouses* and divides the order among the warehouses, sending a *packing slip* to each warehouse. The *packing list* tells the warehouse which *products* from the total *order* that warehouse should ship to the *customer* .
>
> The *MOF* prepares a single *invoice* for each customer order, indicating the *items to be shipped* to fulfill the order, their expected *shipping date* , and various *charges* to the customer including the *price* of the *items* , *shipping and handling costs* , and *sales taxes* . This invoice is mailed separately to the customer.
>
> FOO would like to have a common inventory- and order-tracking system across all of its warehouses, to replace the current amalgamation of manual processing and low-level computer tools (spreadsheets, word processors) used to support this process.

1) So the candidate classes are: Fancy Oriental Objects (FOO) Incorporated, mail-order company, knick-knacks, gimcracks, FOO, orders, warehouses, demand, products, region, inventory, year, Customer, central mail-order facility (MOF), MOF, list of products ordered, available stock, packing slip, packing list, invoice, items to be shipped, shipping date,

charges, price, items, shipping and handling costs, and sales taxes.

Some of these appear to be synonyms (e.g., “knick-knacks” and “gimcracks” are simply derogatory terms for the “products” offered by the company, “MOF” is simply an abbreviation for “mail-order facility”). Some are named instances (objects) of a more generic class (e.g., FOO is a mail-order company). Others (e.g., mail-order company and MOF) might appear to be synonyms when taken in isolation, but when taken in the context of the problem description, it becomes clear that these have a distinct meaning in the “application vocabulary”. Finally, there are a few terms that appear synonymous, but questions that some teams put to the domain expert resulted in a clarification (e.g., “products” and “items” are related, but not identical. A product is something that gets listed in a catalog. An item is the physical object. For example, if a customer orders 10 Acme “Great Chinese Philosopher” Figurines, that’s 10 items but only one product.)

Some of these terms are also probably unimportant to the model (e.g., “gimcrack”, “year”)

2) In marking up the verb phrases, I ignored verbs that simply said one thing “is” another, or “is composed of” others. Those kinds of statements are capturing relationships between classes, which are important to the URL diagrams, but do not directly affect responsibilities except by suggesting attributes.

The candidate responsibilities are: specializing in, process (order), inventory management, order processing, vary, receive (order), check stock/inventory, divide order, send (packing slip), ship (items), fulfill (order), mail (invoice)

1.1. Grading Criteria

Completeness: Are all candidate classes and responsibilities included?

Some teams read that whole description and still only come up with 3-5 candidate classes. Many forgot to give a list of candidate responsibilities.

You can’t argue that the 3-5 classes you chose are the only important ones. We won’t know that for sure until much later in the analysis. That’s why these are called *candidate* classes.

Correctness: Part of the OO approach is to always use terminology that is natural to the application domain. This facilitates communication both within the development team and between the developers and the domain experts and users.

If, for example, the description refers to “customers, don’t call them “users, “people, etc. In some cases the terminology is unclear. People needed to ask the domain expert. (In a few cases, even the expert wasn’t sure.)

It’s bad enough to invent terminology, but it’s even worse to be so vague about it that no one could possibly know what you mean. “Weasel words” is my own term for words like “status”, “information”, “data”, etc., that we are tempted to use as names to “weasel out” of the effort of coming up with a correct, precise term. E.g., when I see something like “get order info”, it’s pretty clear that the author had no clue just what that info really was.

Defensibility: This is analysis. You are building a model of how the assessment world behaves. Don’t introduce things that aren’t mentioned and that therefore might not exist in that world.

A particularly bad form of unsupported classes are ones that are introduced just because people felt that the eventual program would need them. People who introduced classes for things like “inventory databases”, for example, were engaging in a late stage of design rather than an early stage of analysis. There’s no mention of such a thing in the description. So we have no assurance that such a thing exists in the real world. (That’s not a trivial objection: does every warehouse have its own database or is there a single centralized one at the MOF? If you can’t answer that (and we can’t from the information given), how are you going to model it?

A database may, indeed, prove to be part of the implementing data structure for one of the classes. But this is analysis, and the focus is on modeling the real world.

2. Question 2: CRC cards

Mail Order Company	
get/set MOF	Mail Order Facility
get/set warehouses	Warehouse

The “get/set” indicates that these are, in effect, attributes. The plural noun “warehouses and “performances” indicate that there appear to be more than one of each. Some people might prefer to say “list of warehouses, “set of warehouses, etc., to indicate this more explicitly.

I have no card for FOO itself, because the statement “(FOO) is a mail-order company” indicates that it is an object of this Mail Order Company class.

Customer Order	
get/set Customer	Customer
get/set ProductOrders	ProductOrder
split (ProductOrders): PackingSlip	

ProductOrder?	
get/set Product	Product
get/set Quantity ordered	

ProductOrder is marked with a question mark. It’s obvious that we need a term for the line items in a customer order (what product and how many items of that product). The noun phrase “list of Products Ordered” served as the justification for this choice of name, but is really pretty thin evidence. One could easily argue that the description does not give a term for this, so we should mark this as a future question for our domain experts, “What is the name you use for . . . ?”.

Warehouse	
check Inventory(Product, #): (shipping) Date	Product
manage inventory	Date
process partial order (PackingSlip)	Item
	PackingSlip

“Manage inventory” probably means supporting operations to add and remove items as they are received/shipped. There’s really not enough info provided to be more specific, but we can make note of that for later stages of analysis. We might also choose to say simply that a warehouse has a set of Items as its attribute.

“process” is dangerously close to being a weasel word (see Question 1 above), but checking the description shows that this was the exact term used.

Product	
get/set price	Price
get/set product ID#	
get/set shipping weight	

The description implies that price is known for any ordered products. The other two attributes listed here were revealed by the domain expert upon questions from multiple teams. Products will probably have other attributes (e.g., names, catalog descriptions), but we’d need to get the info from a domain expert. (Note - it’s OK to list reasonable guesses at attributes as long as you mark them with “?” or otherwise indicate that this is a guess to be validated later by consultation with the domain experts.)

Customer	
get/set name	Address
get/set shipping Address	

Mail Order Facility	
process customer order (Order)	Order
	Warehouse
	Packing Slip
	Invoice
	Customer

What's interesting here is what's *not* listed in the responsibilities. We don't list splitting an order into packing slips, sending the packing slips to the warehouses, sending the invoice to the customer, because those are all actions that would be taken by the MOF in response to receiving a customer order. In other words, those are operations that the "function body" of "process customer order" calls on other objects. Hence they are the responsibilities of those other objects.

Packing Slip	
get/set Customer	Customer
get/set ProductOrders	ProductOrder

Invoice	
get/set Customer	Customer
get/set ProductOrders	ProductOrder
get/set Charges	Charge
get/set ShippingDate	Date

Charge	
get/set amount	Currency

The Charge card would be stacked on top of a deck made up of the next 3:

Price	
get/set ProductOrder	ProductOrder
get/set amount	Currency

ShippingAndHandling	
get/set amount	Currency

SalesTax	
get/set amount	Currency

2.1. Grading Criteria

Notation: Are cards properly divided into 3 sections (C, R & C). Do the contents of the three sections appear appropriate?

Completeness: Are all major classes and responsibilities accounted for?

Correctness: As in Question 1, proper naming is important.

In addition, some teams were marked down for “*reversed responsibilities*”. A responsibility is something that a class promises to do in response to a “call” from another object. When you see a statement like “the MOF sends the invoice to the customer”, that does not mean that “send invoice” is a responsibility of the MOF, because no other object in the system is sending the MOF a message telling it to send the invoice. The statement clearly indicates that this is something the MOF does, possibly as a single step in a method for responding to some other message. The responsibility for accepting this message might belong to the Customer, or to the Invoice (or to some as-yet-unidentified class, e.g., MailRoom).

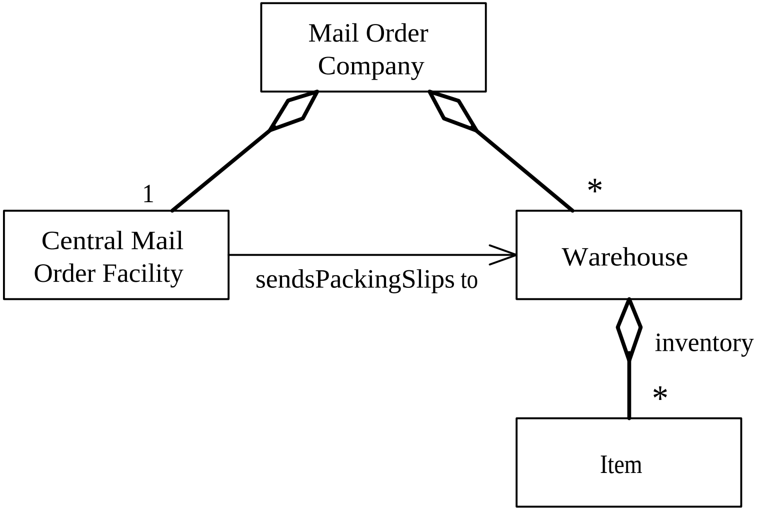
Defensibility: As in Question 1. “Premature design” is an increasing tempting danger as we move into the CRC cards.

Expressiveness: Although CRC cards aren’t supposed to be pretty, they should not be so vague that no one can glean useful information from them.

3. Question 3 - UML Class Relationship Diagrams

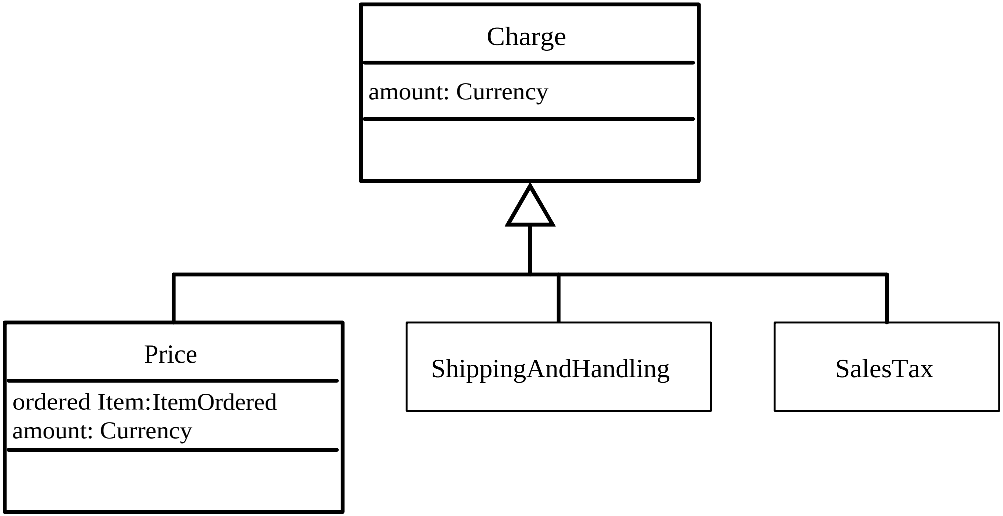
Clearly, there are many possibilities here.

The different kinds of relationships suggest some possibilities. For example, there is a physical/ownership structure in this world:

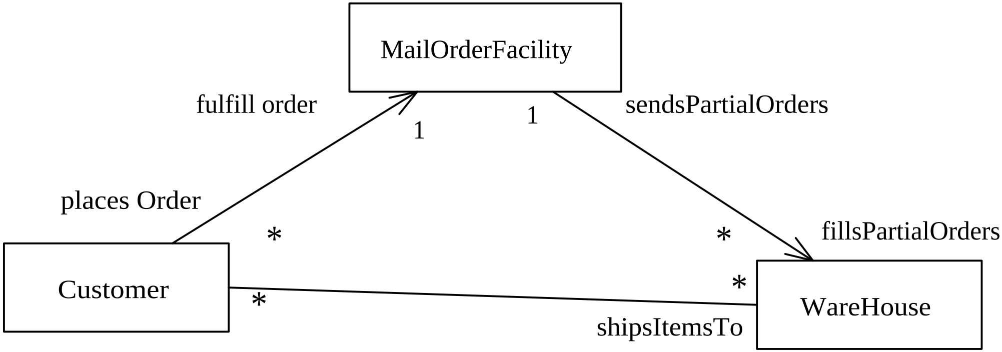

Mail Order Company
1
*
Central Mail Order Facility
sendsPackingSlips to
Warehouse
inventory
*
Item

The association between the MOF and the Warehouses indicates directionality - we can tell from the description that the MOF can reach/contact the warehouses as necessary. The “sends packing slips to” name for that association is pretty weak, but the description doesn’t really provide us with enough info on the nature of the structural relationship to some up with a more appropriate name.

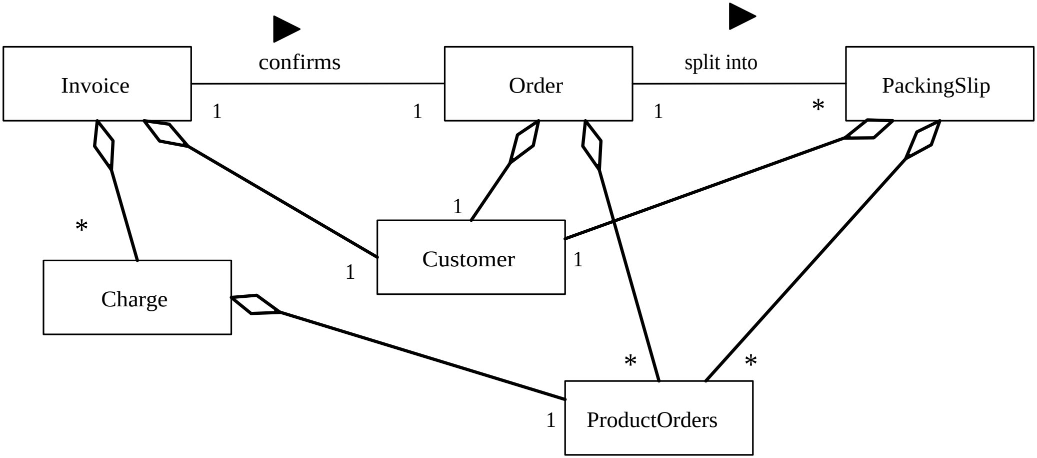
We can also take note of the one apparent inheritance hierarchy in the model:



But I would hope that at least one diagram would address either central responsibilities for order processing:



or the rather complex inter-relationship among the documents maintained by the system:



3.1. Grading Criteria

Notation: Are the diagram elements technically correct?

Completeness: Did students provide three diagrams with examples of three different kinds of associations? Are associations decorated with association/role names and cardinality when appropriate?

Correctness: Is the diagram an accurate portrayal of information from the model?

Defensibility: Is the information in the diagram justified by the available information?

Expressiveness: Does each diagram make a coherent statement about the model? Is there enough information provided for the diagram to convey useful information?