

APPENDIX

F THE YOURDON PRESS CASE STUDY

F.1 INTRODUCTION

No discussion of systems analysis would be complete without at least one example to illustrate the various modeling tools and techniques discussed in this book. Unfortunately, almost any case study is likely to be either entirely fictitious or a grossly oversimplified and "sanitized" version of a real-world situation. Also, it is difficult to find an example that illustrates both a business- and scientific-oriented application.

This case study describes the requirements for computerization of the information processing activities for YOURDON Press. On the one hand, it is very indicative of a real-world publishing activity that operated for approximately 10 years. Indeed, one of the things I want to show in this case study is that things are not always done for rational reasons (including the formation of companies and the initiation of many systems development projects!), and that most systems have to deal with a lot of annoying unpleasant "glitches" in the real world.

On the other hand, YOURDON Press has now joined the ranks of fictitious examples, for it was acquired by Prentice-Hall in 1986, and its information processing activities have been subsumed into Prentice-Hall's.¹ Thus, this case study describes what YOURDON Press' information processing requirements *would* have been had it continued on as an independent publisher.

The following sections provide a brief background of the YOURDON Press operation, the environmental model of the system, the behavioral model, and the user implementation model.

¹ Prentice-Hall's information processing activities, meanwhile, are being subsumed into its new parent company, Simon & Schuster. And Simon & Schuster is part of an even larger company, Gulf + Western, which all goes to show that systems are almost always part of larger systems.

F.2 BACKGROUND

To understand the workings of YOURDON Press, it is necessary to spend some time explaining the larger context of the corporation within which it existed: YOURDON inc. Without YOURDON inc., there would have been no YOURDON Press; though without YOURDON Press, it is fairly clear that YOURDON inc. would not have achieved the success that it has.

YOURDON inc. was formed as an outgrowth of independent consulting and lecturing activities that I had been carrying out for a number of years in the late 1960s and early 1970s. The company was formed in April 1973 because my accountant told me that a corporation offered some tax advantages that would not be available to me as a self-employed consultant. Notwithstanding this practical tax advice, the new corporation did not really begin conducting any business until April Fool's Day, 1974.

As is true in most companies (and most data processing projects!), one of the first activities was to think of a proper company name. My wife and I, who served as the company's sole stockholders, directors, officers, and employees, were rather fond of the name "Artichokes And Other Fur-Bearing Animals, Inc.," but decided that it would not fit onto our stationery. We finally settled on the name "Superprogrammers, Limited," and filed the appropriate papers with the State of New York to establish the name. About two weeks later, just as we were about to place some advertisements for our first series of seminars on structured programming, the state informed us that our company name had not been approved: it was too close to the name of an already extant company. When we investigated, we found that the other company was named "Supermarket Products, Inc."² Out of desperation, we quickly chose a name that we were reasonably certain would not be duplicated by any other company: my own name. Thus was YOURDON inc. born.

The company's initial activities were professional seminars on advanced programming techniques and on-line systems design aimed at veteran programmers and systems analysts in large organizations and government agencies. The seminars involved about 20 hours of classroom lectures and were accompanied by a few hundred pages of class notes; the class notes for the seminar on advanced programming techniques eventually became a textbook: *Techniques of Program Structure and Design*, published by Prentice-Hall in 1975.

Because of the large number of seminar attendees, it proved economical to print the class notes in moderate volume and to bind the pages together; thus, they bore some resemblance to a book, though some of the pages were printed upside down and other pages fell out of the book at the slightest provocation. Nevertheless, some of the seminar attendees asked to purchase additional copies of the class notes, and thus, as a sideline, YOURDON inc. found itself in the business of selling "books."

² When we investigated, we found that Supermarket Products was located somewhere on the periphery of New York City and was chiefly involved in importing bananas from Guatemala. We didn't see what this had to do with computers or why the state felt that our name would impinge on poor old Supermarket Products, but we decided not to fight the bureaucracy.

However, YOURDON inc. concentrated primarily on training activities: the number of distinct training courses grew to approximately 50 by the mid-1980s, and the company has now trained some 250,000 data processing professionals throughout the United States and in over 30 other countries. Professional consulting activities also began to grow, and many of the company's technical staff now serve as consultants, project leaders, and systems analysts on major systems development projects in client companies throughout North America and Europe. And in the mid-1980s, the company entered the CASE market, with an analyst toolkit product of the nature described in Appendix A. In 1987, YOURDON inc. had offices in 8 cities, with a staff of approximately 150 people.

YOURDON Press began as a division of YOURDON inc. in 1976 with the softcover publication of three books: *Structured Design*, by Yourdon and Constantine; *Learning to Program in Structured COBOL*, by Yourdon, Gane, and Sarson; and *How To Manage Structured Programming*, by Yourdon. As with so many other real-world business operations, this happened without a great deal of planning or organized thinking: the books seemed like a good way of popularizing the structured techniques concepts being developed and marketed in YOURDON inc.'s training seminars.

The first three books were produced on a simple IBM Selectric typewriter and were bound in 8.5- by 11-inch sheets; all of this predated the days of convenient typesetting and desktop publishing. Advertising was rather modest, consisting of a few *Computerworld* ads and mailings to YOURDON's seminar customer list. Sales were equally modest; indeed, for the first several years of its existence, YOURDON Press represented only a tiny fraction of the company's overall revenues.

Consequently, the information system surrounding the early YOURDON Press activities was modest and entirely manual in nature. Orders were taken over the phone or by mail, but credit card orders were not accepted. Invoices were typed by hand on four-part invoice forms, and orders were individually packed by hand. The inventory was stored in one of the world's most elegant warehouse spaces: windowed offices of the 38th floor of YOURDON inc.'s offices at 1133 Avenue of the Americas, looking out over all of Manhattan.

Automation arrived at YOURDON inc. in the spring of 1976, in the form of a second-hand PDP-11/45 minicomputer and a mysterious operating system called UNIX.³ A few months later, a phototypesetter, two dozen terminals, and the TROFF typesetting package were added. This immediately facilitated typeset production of YOURDON Press textbooks and eventually led to the automation of several aspects of YOURDON inc.'s training business and general accounting activities. But the YOURDON Press operational activities, the activities that could be considered as an "information system," continued to operate in a manual fashion for several more years.

³ UNIX is not so mysterious now, of course, but in the mid-1970s hardly anyone outside Bell Laboratories and a few universities had heard of it. Neither I nor most of my colleagues at YOURDON were all that prescient; we owed our decision, which we later came to appreciate very much, to the urgings of Dr. P.J. Plauger, who had joined the company from Bell Labs in 1975. Plauger is widely known for books co-authored with Brian Kernighan, including *The Elements of Programming Style* (Reading, Mass.: Addison-Wesley, 1973) and *Software Tools* (New York: McGraw-Hill, 1976).

In 1980, a limited number of computerized applications were developed for YOURDON Press, using the convenient pipeline features of the UNIX operating system. Between 1980 and 1985, the C programming language and a number of UNIX shell scripts were used to gradually add a number of simple programs for order processing, sales reports, shipping labels, and various accounting reports. Though these programs were easy to develop and reasonably reliable to operate, they were developed on a piecemeal basis, similar to what one often sees today in an EDP organization where the end users have access to spreadsheets, report generators, and fourth generation programming languages. They were also rather limited; for example, if the details of an order needed to be modified subsequent to its entry, the system was unable to accomplish it. Instead, the standard UNIX text editor was used to modify the order, which was stored in the computer as a simple ASCII text string, terminated by an end-of-line character.

One of the most difficult activities in the day-to-day operation of YOURDON Press was the task of producing an up to date statement showing all a customer's orders, payments, book returns, and credits for a given time period. Equally difficult was the process of reconciling these activities (which took place as interactions between the customers and YOURDON Press administrative personnel) with the financial records maintained by YOURDON inc.'s accounting department. For various reasons, YOURDON Press and the Accounting Department always seemed to be "out of synch" with one another. This was further complicated by the fact that YOURDON inc.'s London office had their own inventory of books and did their own shipping and invoicing independently of the New York office; prices were quoted in pounds sterling rather than dollars and were generally somewhat higher than the prices quoted from the New York office.⁴ Once every quarter, when financial statements had to be prepared, long, frustrating, mind-numbing meetings took place in which computer printouts produced by the Accounting Department were manually compared with computer printouts produced by YOURDON Press in order to reconcile differences. Tempers flared; people shouted insults, obscenities, and various epithets at one another; blunt objects were sometimes thrown back and forth across the room. It was not a pleasant activity to look forward to each quarter.

Thus, by 1986, it was evident that a full-scale *system* would have to be developed if YOURDON Press was going to continue to grow; initial planning began for the new system. However, it was also evident that a substantial amount of capital would be required to continue growing the business, not only for additional computer equipment, but also to modernize the typesetting equipment (which was now obsolete) and enlarge the editorial and marketing activities of the division. It was finally decided

⁴ The issue of separate inventories, and sales from separate offices was looming on the horizon as a larger and larger problem. Each of the various YOURDON offices insisted that they needed to have a small, local inventory to service the walk-in customers who wanted to be able to get a book right away, rather than waiting several days (or weeks) for a shipment from Galactic Headquarters. And the Canadian office argued that it needed its own pricing structure (i.e., prices quoted in Canadian dollars rather than U.S. dollars) and its own marketing/advertising campaign to appeal to the Canadian market in a different way than the U.S. market. In some cases, the remote offices would simply give the book to the customer and ask the central New York office to generate the invoice. In other cases, the customer would pay for the book on the spot, and ask for a receipt. Sales from the London office accounted for roughly 10% of the overall YOURDON Press revenues, while sales from the other offices accounted for less than 1% of the overall YOURDON Press revenues.

that it would make more sense to have the publishing operation acquired by a larger organization, and this led to the merger with Prentice-Hall. Thus, the system models described below represent what the requirements would have been if YOURDON Press had continued to operate as an independent business.

The planning for a new information system also coincided with a series of organizational changes within YOURDON Press and the rest of YOURDON inc. From its inception in 1974 until approximately 1983, the company had the organizational structure shown in Figure F.1.

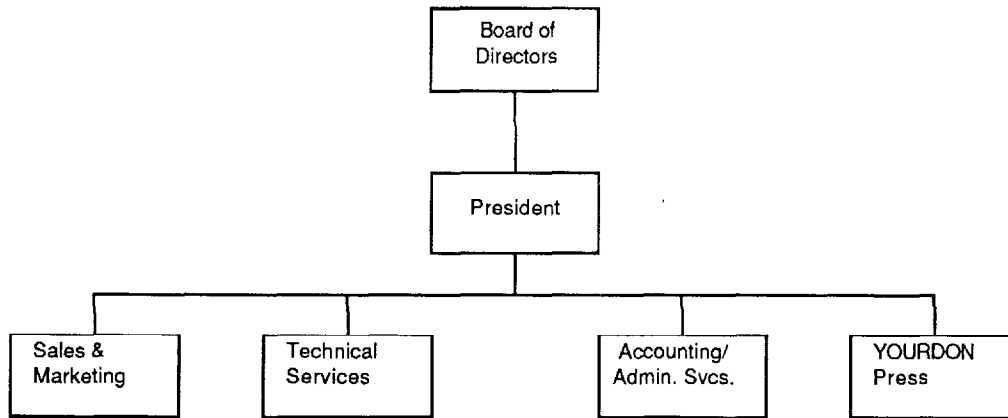


Figure F.1: Organizational structure of YOURDON inc., 1974-1983

Between 1984 and 1986, the company shifted to more of a regional organization, and added a new division for its software product, as shown in Figure F.2.

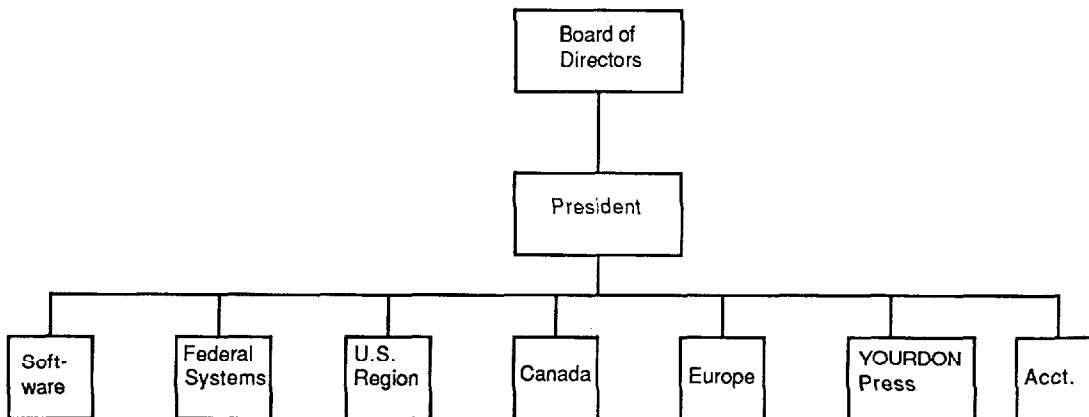


Figure F.2: Organizational structure of YOURDON inc., 1984-1986

And during this period, YOURDON Press gradually developed the organizational structure shown in Figure F.3.

As part of this reorganization, the YOURDON Press shipping operations were moved out of the elegant office space occupied by the rest of the staff, and into a warehouse in beautiful downtown Yonkers, New York. Thus, there was a physical separation of some 20 miles between the people entering orders and the people packing books into boxes and sending the orders to the customers.

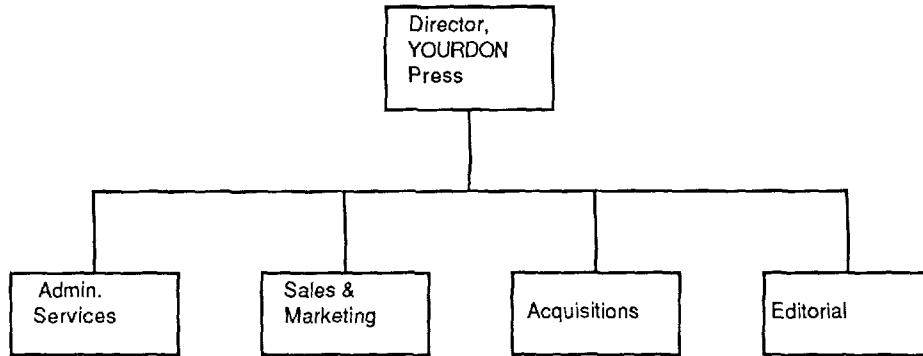


Figure F.3: Organizational structure of YOURDON Press

The four major groups within YOURDON Press had the following responsibilities:

- Administrative services was responsible for most of the day to day interactions between YOURDON Press and the customers. Thus, this group accepted orders; produced invoices; received payments, discussed book returns and credits with customers; interacted with the warehouse to arrange for shipments of books; and interacted with the Accounting Department, as discussed above.
- Sales and marketing was responsible for producing catalogs of the various YOURDON Press books; running ads in computer magazines and trade journals; sending promotional brochures to various mailing lists; and soliciting sales via telephone calls to large corporate purchasers of technical computer books.
- Acquisitions was responsible for finding new authors and new books. This part of YOURDON Press carried on all discussions with authors up to the point where a final manuscript was delivered by the author.
- Editorial was responsible for taking a final manuscript and turning it into a published book. This involved not only the copyediting of the book, but also interactions with printers to obtain proposals for the initial printing of the book. Editorial was also responsible for the artwork and production of the book cover, as well as the inside contents.

It should be kept in mind that YOURDON Press was a fairly small operation compared to such well-known publishing operations as McGraw-Hill, Harcourt Brace, Prentice-Hall, and Random House. An idea of the scale of the operation can be gleaned from the following statistics:

- YOURDON Press had approximately 50 books in its list; typically 4 to 6 news titles were added to the list each year.
- The books were written by approximately two dozen authors, and the acquisition group interacted with approximately 200 potential authors, that is, individuals who expressed interest in writing a book, but who had not yet actually finished writing anything.
- YOURDON Press processed approximately 50 orders a day.
- The average order was approximately \$100, which typically represented three to four books. Some orders, of course, were for an individual book; some orders were larger. Wild celebrations and cheers broke out whenever an order larger than \$5000 was received.
- Approximately 50,000 books per year were shipped.

Aside from its small scale, though, YOURDON Press operated in much the same way that other, larger publishers do. Sales were made via written orders, telephone orders, or walk-in orders (i.e., a customer coming into the YOURDON inc./YOURDON Press offices to purchase a book). Payment could be made in cash (which was rare), by check, or by credit card. As a matter of policy, orders under \$100 had to be prepaid; larger orders, especially those from book stores and companies, generally required an invoice.

To understand the publishing business, one must also be familiar with the concept of *returns*. If an individual customer or corporation felt that a book was not suitable for their needs or if it was received in damaged condition, they could generally return the book and ask for a refund. This would normally happen within a matter of days after the shipment was received by the customer. Book stores, on the other hand, had the privilege of returning up to half of the books in an order within a year of the date that the order was placed; this is common within the publishing industry, because book stores often don't know in advance how much demand there will be for a book and want to avoid getting stuck with inventory that they can't sell.

F.3 THE ENVIRONMENTAL MODEL

F.3.1 The Statement of Purpose

The purpose of the YOURDON Press Information System (YPIS) is to maintain information needed to sell books to customers. This includes order entry, invoicing, generation of shipping documents, inventory control, and production of royalty reports and accounting reports.

F.3.2 The Context Diagram

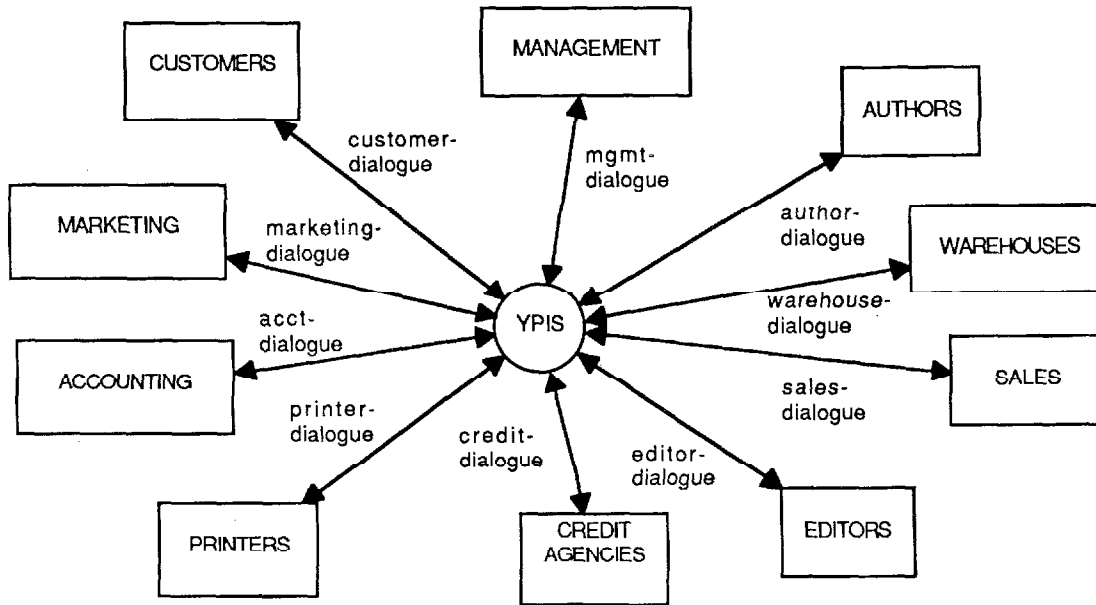


Figure F.4: Context diagram for the YPIS system

F.3.3 The Event List

The event list for YPIS consists of 40 events. Most of the events are flow-driven, though most of the events involving the Accounting Department are temporal. The events are listed next; temporal events are marked with a T notation following the description of the event.

1. Customer orders book (this includes special rush orders, too).
2. Customer sends payment.
3. Customer asks for book information (price, etc.).
4. Customer asks permission to return a book.
5. Customer asks about status of a book order.
6. Customer asks about status of an invoice.
7. Customer needs (monthly) statement. (T)
8. Customer asks for credit memo.

9. Customer wants refund check.
10. Accounting needs (daily) cash receipts. (T)
11. Accounting needs (daily) revenue reports. (T)
12. Accounting needs (monthly) net revenue report. (T)
13. Accounting needs (quarterly) author royalty report. (T)
14. Accounting needs (monthly) inventory data. (T)
15. Accounting needs (monthly) sales commission report. (T)
16. Management sets new credit limit for a customer.
17. Accounting needs (monthly) aged accounts receivable report. (T)
18. Printer offers quotation for print (or reprint) order.
19. Management authorizes a print order.
20. Printer advises exact print quantity and delivery date.
21. Printer sends invoice for print job.
22. Management asks for quotation on print order.
23. Marketing asks for mailing labels from customer database.
24. Marketing needs statistics on book sales.
25. Marketing needs in-stock date for new titles.
26. Editors announce new book title (date ready for printer).
27. Authors need quarterly royalty report. (T)
28. Warehouse needs shipping data and mailing labels. (T)
29. Warehouse receives books from printer.
30. Warehouse receives book returns from customer.
31. Warehouse conducts (monthly) physical inventory.
32. Warehouse makes shipment of book order to customer.
33. Warehouse announces that a book is out of stock.

34. Acquisition department announces new book project.
35. Salesperson places order on behalf of customer.
36. Marketing declares that a book is out of print.
37. Customer announces a change of address.
38. Author announces a change of address.
39. Customer elects to join agency plan.
40. Invoices need to be sent to customer. (T)

F.4 THE BEHAVIORAL MODEL

F.4.1 The Preliminary Behavioral Model: Dataflow Diagrams

Each of the 40 events listed in Section F.3.3 has an associated dataflow diagram. Of course, the logistics of printing a book make it unwieldy, to say the least, to connect all 40 diagrams together into a single, composite diagram representing the entire system. As we pointed out in Chapter 19, this is the sort of exercise that requires a very large sheet of paper—or several small sheets of paper taped together. I leave that as an exercise for the reader.

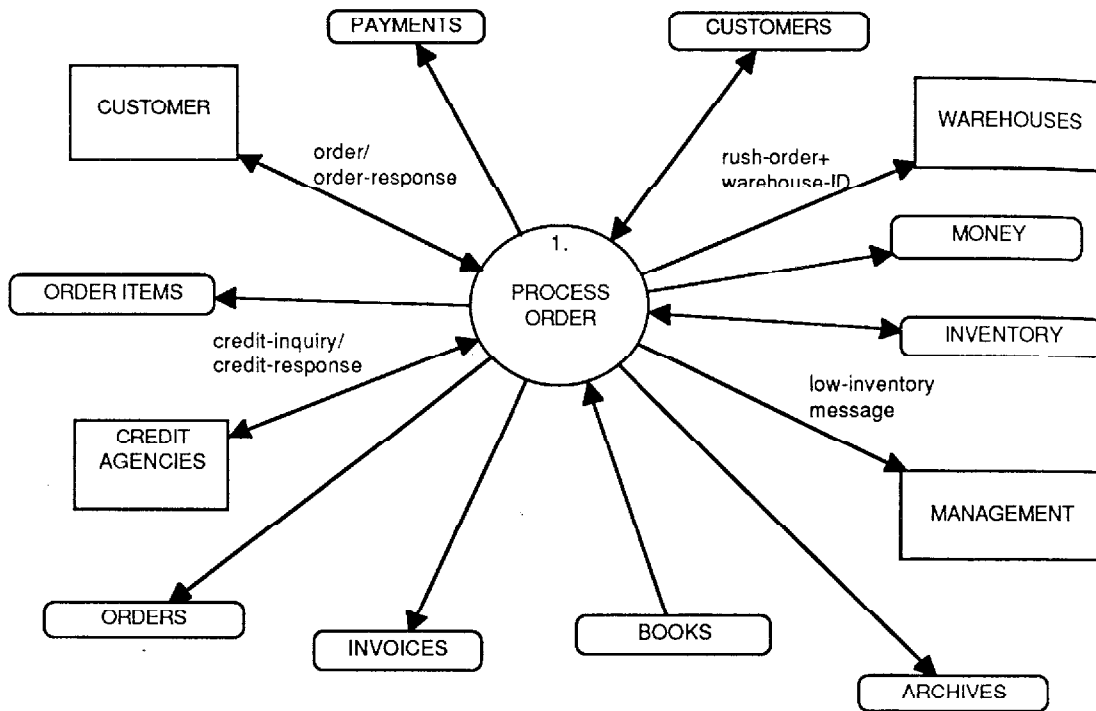
The diagrams were drawn with Version 2.0 the **Design** software package, available from Mota Systems Inc., in Cambridge, Mass. While it does not represent a full-fledged CASE toolkit, it is more sophisticated than most simple graphics packages; and it has the advantage of running on a Macintosh computer, which was used for the preparation of this book. To accommodate the **Design** program, I have shown stores in the DFDs with the notation given in Figure F.5.



Figure F.5: Notation for stores in the YOURDON Press case study

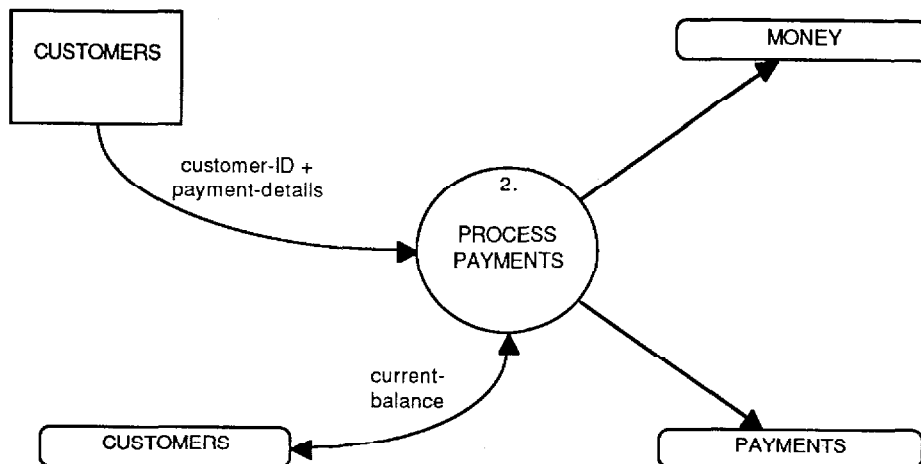
As I drew the preliminary DFDs, I kept notes on errors that I found and changes that I suddenly found that I had to make in other parts of the model; these notes are itemized below each DFD. The reason for doing this is to emphasize that in a real-world project the systems analyst rarely draws a perfect DFD the first time; after thinking about the system and after follow-up interviews with the user, it is inevitable that errors will be found either in the DFD being examined or in some other part of the system model.

No attempt was made to create an organized data dictionary as the preliminary behavior model was developed. After the initial DFD model was created, process specifications were sketched out to see if there were any obvious errors; many such errors are shown as comments on the following pages. A leveled set of DFDs was then created and the data dictionary was then developed.

Event 1: Customer orders book.**Notes**

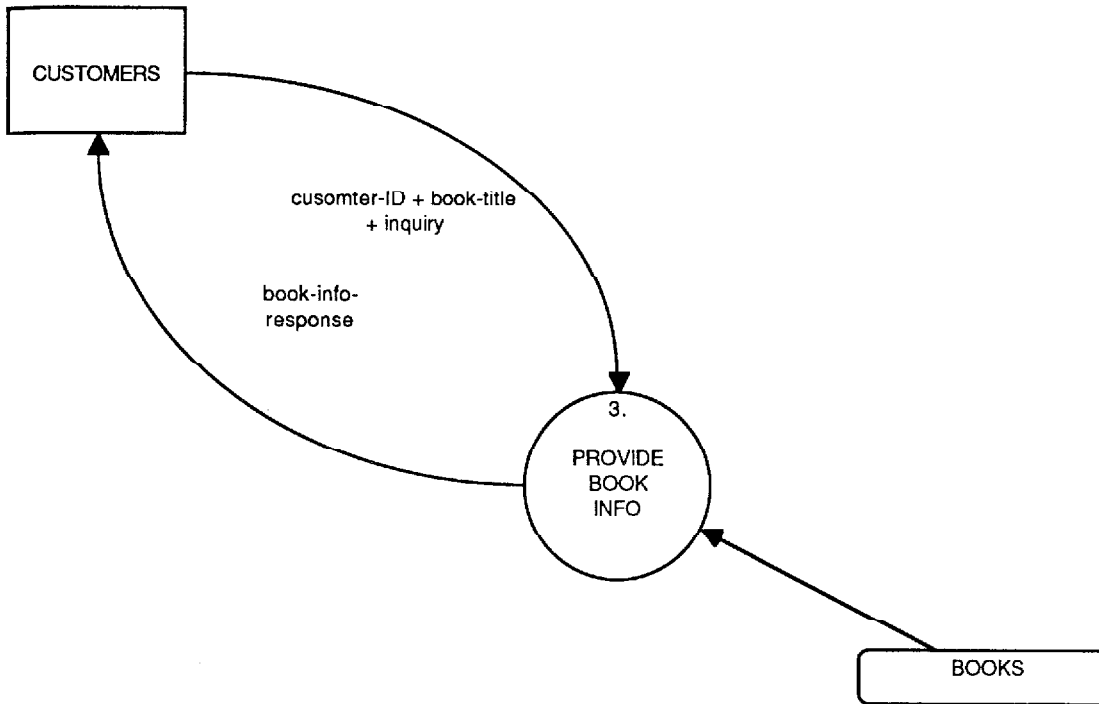
1. After the first version of this diagram was drawn, I remembered that credit card orders normally require an authorization if the amount is above some preset limit. YOURDON Press accepted orders paid with Mastercard, Visa, and American Express; hence the interface to the terminator labeled "CREDIT AGENCIES."
2. Some further thinking about the credit situation made it obvious that the definition of **customer** in the **CUSTOMERS** store would have to include **credit-limit** as a field. It also became evident that there was a required event to change the customer's credit limit (event 16), which had not been obvious before.
3. Note that orders are *not* shipped on a one-at-a-time basis, with the exception of rush orders. Details of a rush order are sent immediately to the warehouse; all other orders are simply stored in the **ORDERS** store. As a separate event (event 28), the warehouse receives mailing labels and shipping instructions for a group of orders (typically one day's worth of orders). I had forgotten the rush orders in the initial version of the diagram.

4. When drawing this DFD, I also realized the need for an **ARCHIVES** store, which is a copy of the customer's original written order (or, in the case of a telephone order, the salesperson's order form), plus a copy of the invoice that was generated for the order. The invoice copy is not necessary in an essential model (since it could be regenerated), but the other documents are necessary in case of a subsequent dispute with the customer, and in case of audits or investigations from the tax authorities, and so on.
5. Note that the order can be received by mail, by phone, or in person. We don't show this in the DFD above, since these are all transporter functions.
6. Note that the system does not reorder books from the printer automatically. Instead, management is informed at various times that inventory has fallen below a preset threshold. This can occur as a result of event 1, as well as several other events.
7. Orders may be received from new customers (especially new book stores or companies that will be doing ongoing business with YOURDON Press). Hence a new record will have to be created in **CUSTOMERS** with the standard discount rate, and so on. This is the reason for the double-headed arrow between bubble 1 and the **CUSTOMERS** store.

Event 2: Customer sends payment.**Notes**

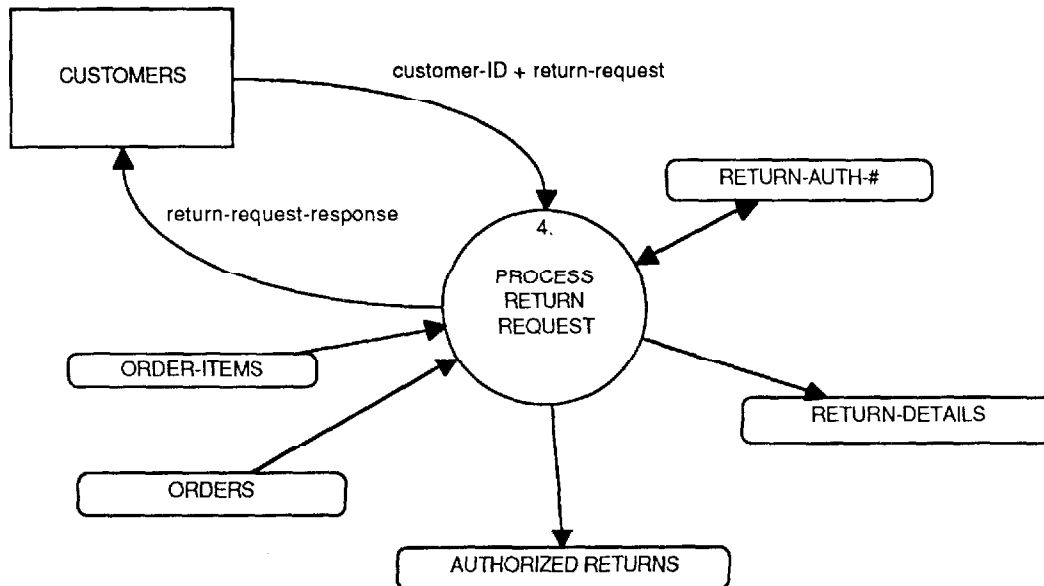
1. Payment may cover several different invoices, but it's not always clear which invoice(s) is/are involved. Sometimes the customers don't identify the invoice they are paying; sometimes they identify invoices that have already been paid; sometimes they reference nonexistent invoice numbers.
2. Sometimes it's not even clear where the payment is coming from. This is particularly true in the case of book store chains: XYZ book store in city A may be owned by a conglomerate, PQR, in city B. If a random check arrives from PQR Corp., addressed to YOURDON Press, we may not be able to determine which invoice or even which company is involved. Payments of this kind are put in an accounting category called unapplied cash. The assumption is that if we continue sending overdue invoices to XYZ book store they will call us and tell us that the invoice was paid by PQR.
3. There is no guarantee that the payment will be for the exact amount of the invoice. Some payments are high or low by a small, random amount. Some customers try to avoid paying the sales tax or the shipping and handling fees, this usually results in payments that are one or two dollars too low.

Event 3: Customer asks for book information.



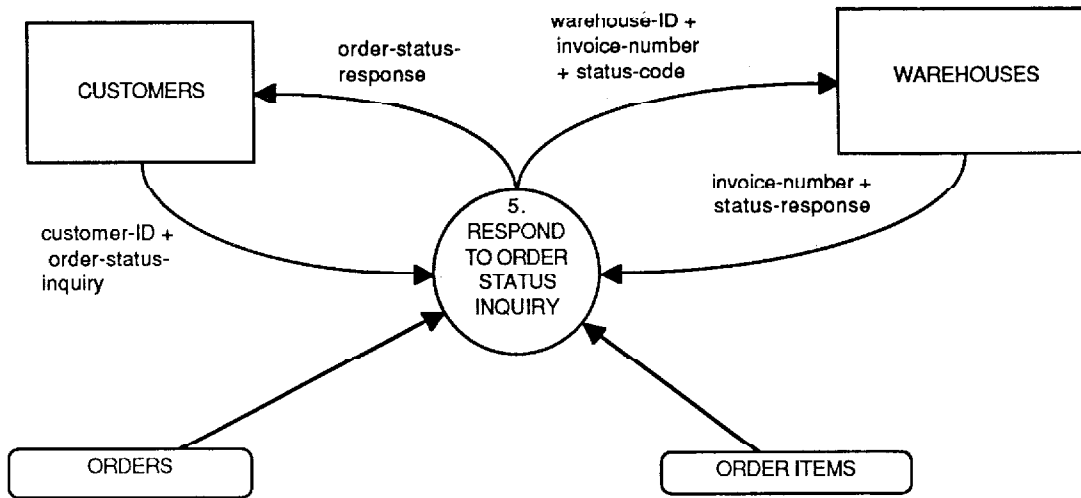
Notes

1. The customer is generally asking for such things as the price of the book, or when a new book is expected to be in stock, or the schedule for volume discounts.

Event 4: Customer asks permission to return a book.**Notes**

1. Customers are supposed to get approval from YOURDON Press before returning the books. They don't always do it.
2. Actual returns arrive later on (event 30) and may or may not match the requested return that has been authorized here.
3. Note that a requested return has to be matched up with an original order.

Event 5: Customer asks about status of a book order.



Notes

1. Shipment of a customer's order may have been delayed because of a backlog in the warehouse or because the book is out of stock. It is this potential delay that typically leads to a customer inquiry.
2. If the customer decides to cancel the order at this point, it is treated as a separate event (event 8).
3. Another possibility is that the order has not been received by YOURDON Press (either because it was lost by the Post Office or somewhere within the mailroom at the customer's office or at YOURDON's office).
4. The order may have been received by YOURDON Press and processed; indeed, it is possible that the warehouse even shipped the order, but it may have been lost by the Post Office (or other shipping agency) en route to the customer. This is treated in the same way (e.g., the customer may decide to cancel the order at this point or may ask for a credit and place the order again).
5. While developing this DFD, I realized that invoices have not been sent to the customer (shipping of the books by the warehouse and mailing of the invoice by the main YOURDON Press office are two separate events). Thus, we need a separate store for invoices, and a temporal event to cause the invoices to be sent.

F.4.2 The Final Behavioral Model: Dataflow Diagrams

The initial behavioral model shown on the last several pages was transformed into a leveled set of DFDs. Upward leveling produced the Figure 1 diagram shown on the next page; it is sufficiently complex that I didn't show the details of all the inputs and outputs to each bubble. Subsequent figures then show which events were grouped together. In one case, a single event (event 26) was not leveled upward, and appears as process 5 in Figure 1. And in one case (event 1) additional downward leveling was required because of the complexity of the processing.

Figure 0: The top-level DFD

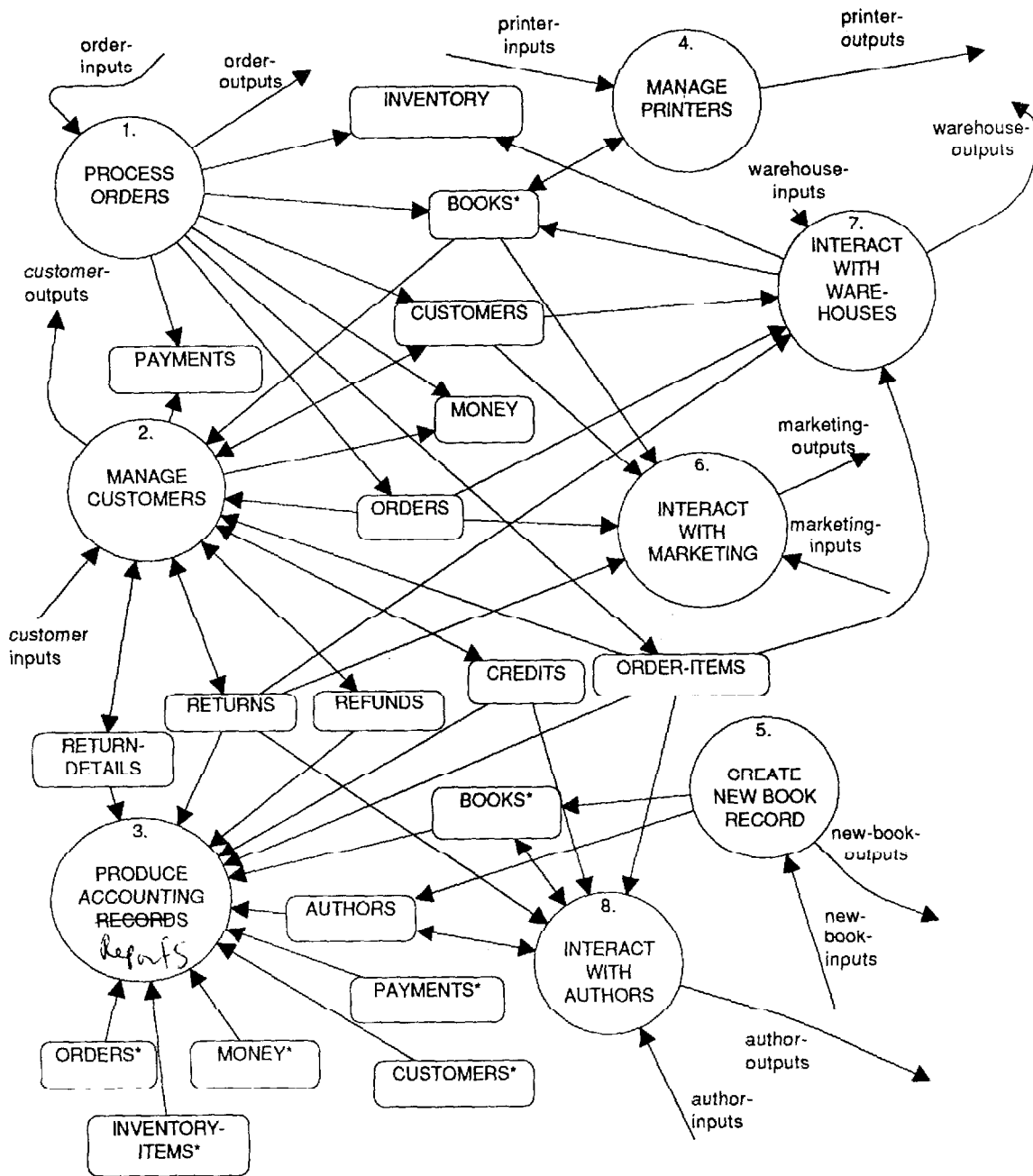


Figure 1: Process orders

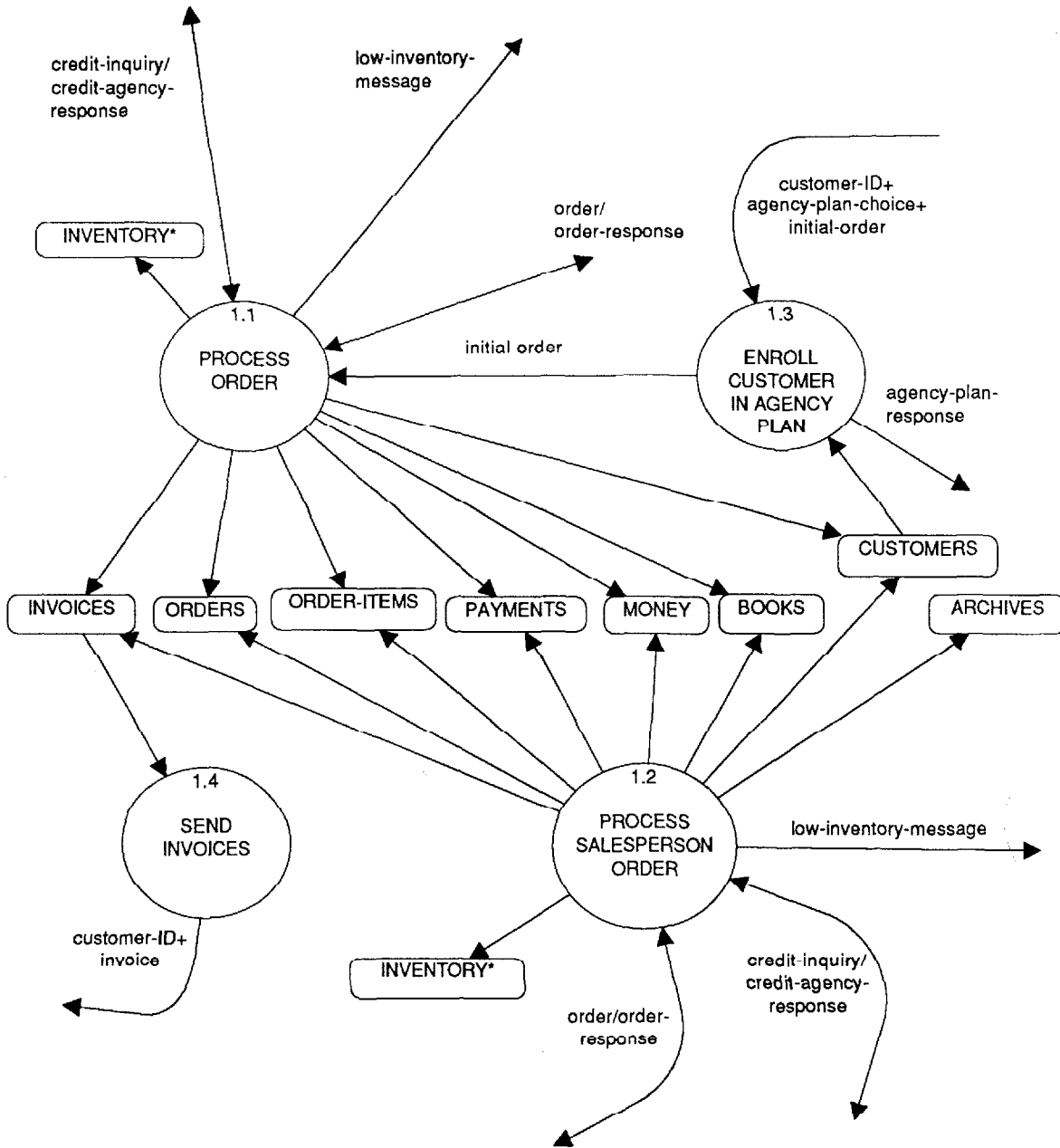


Figure 1.1: Process orders

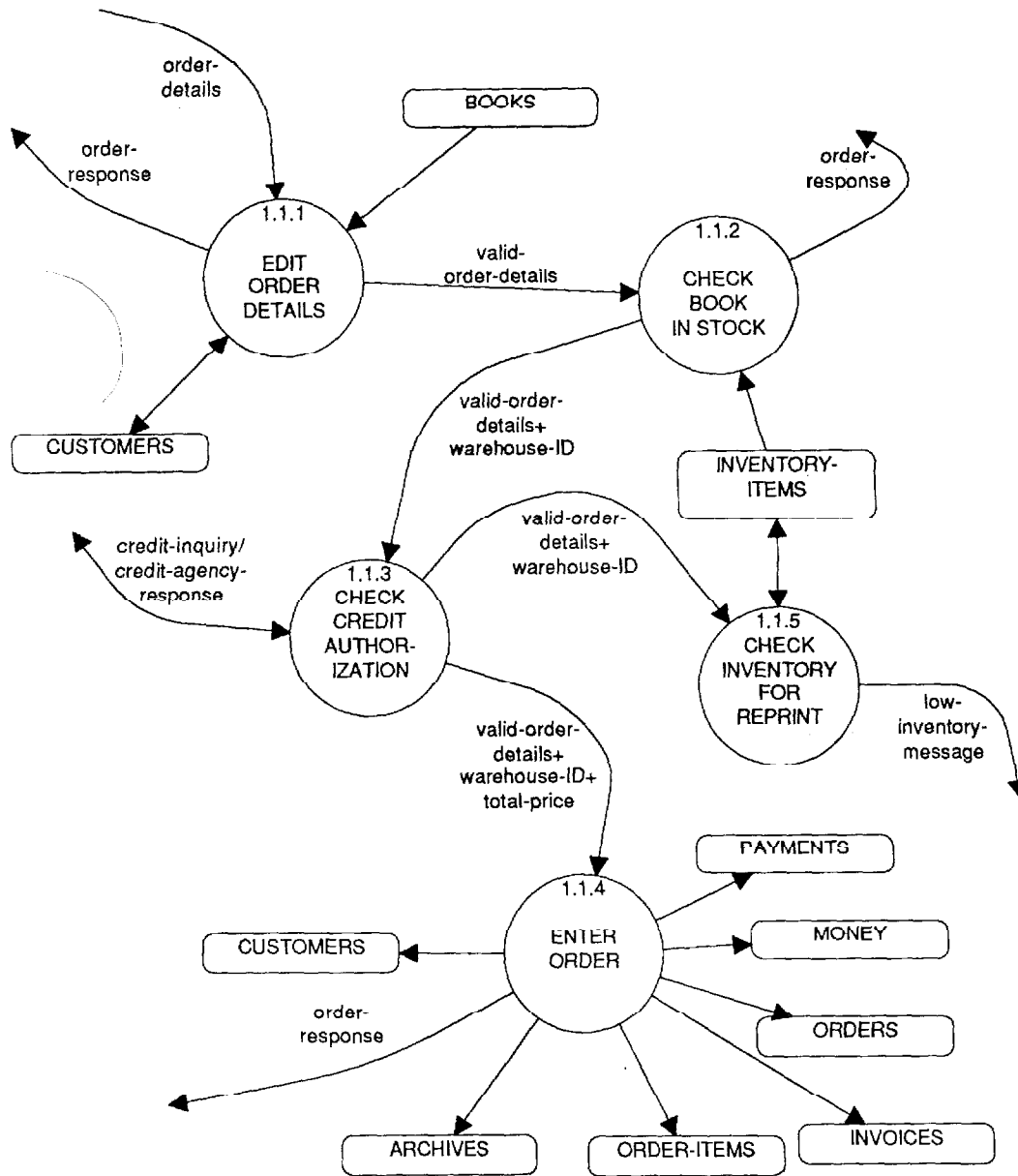


Figure 2: Manage customers

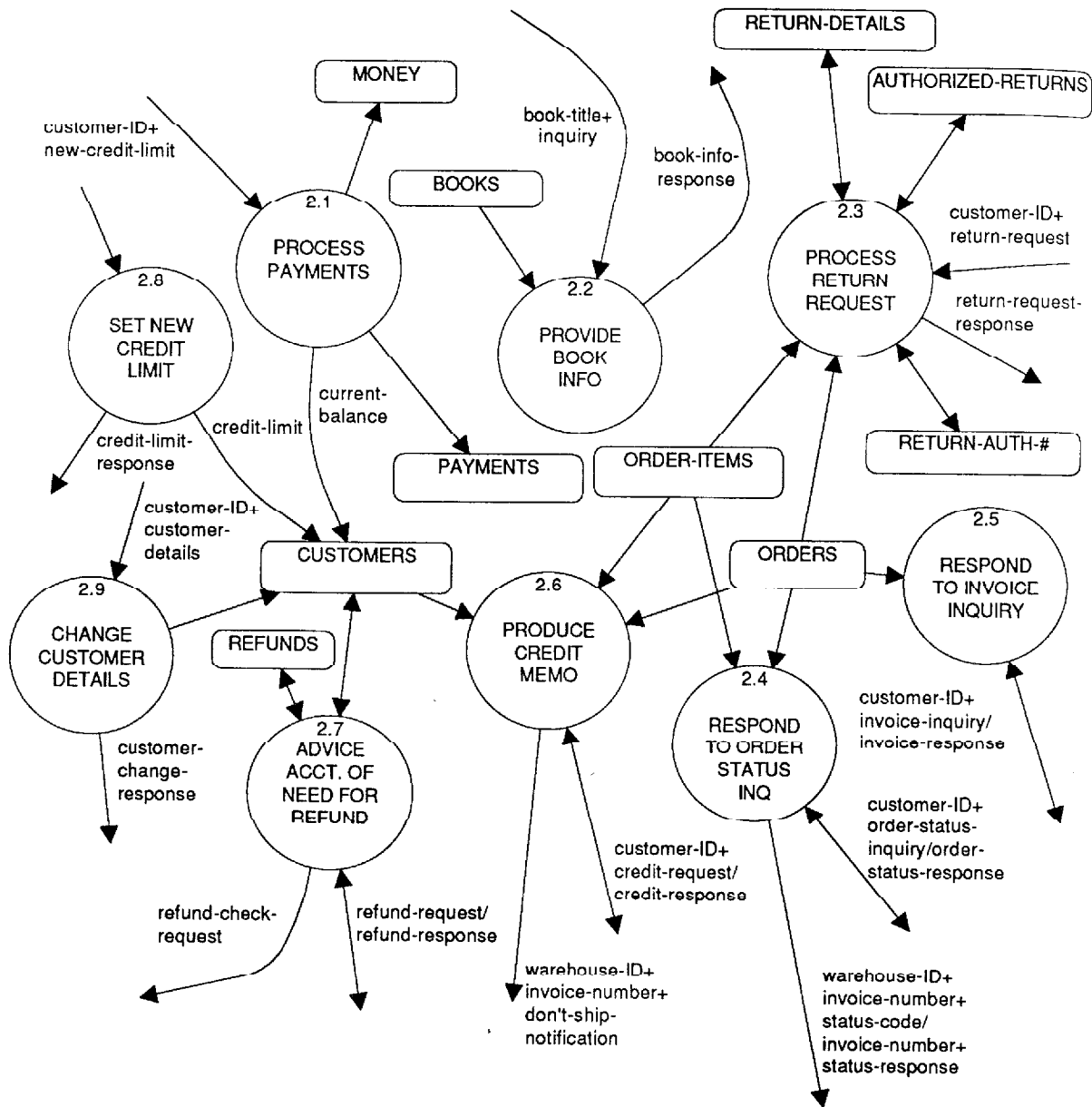


Figure 3: Produce accounting reports

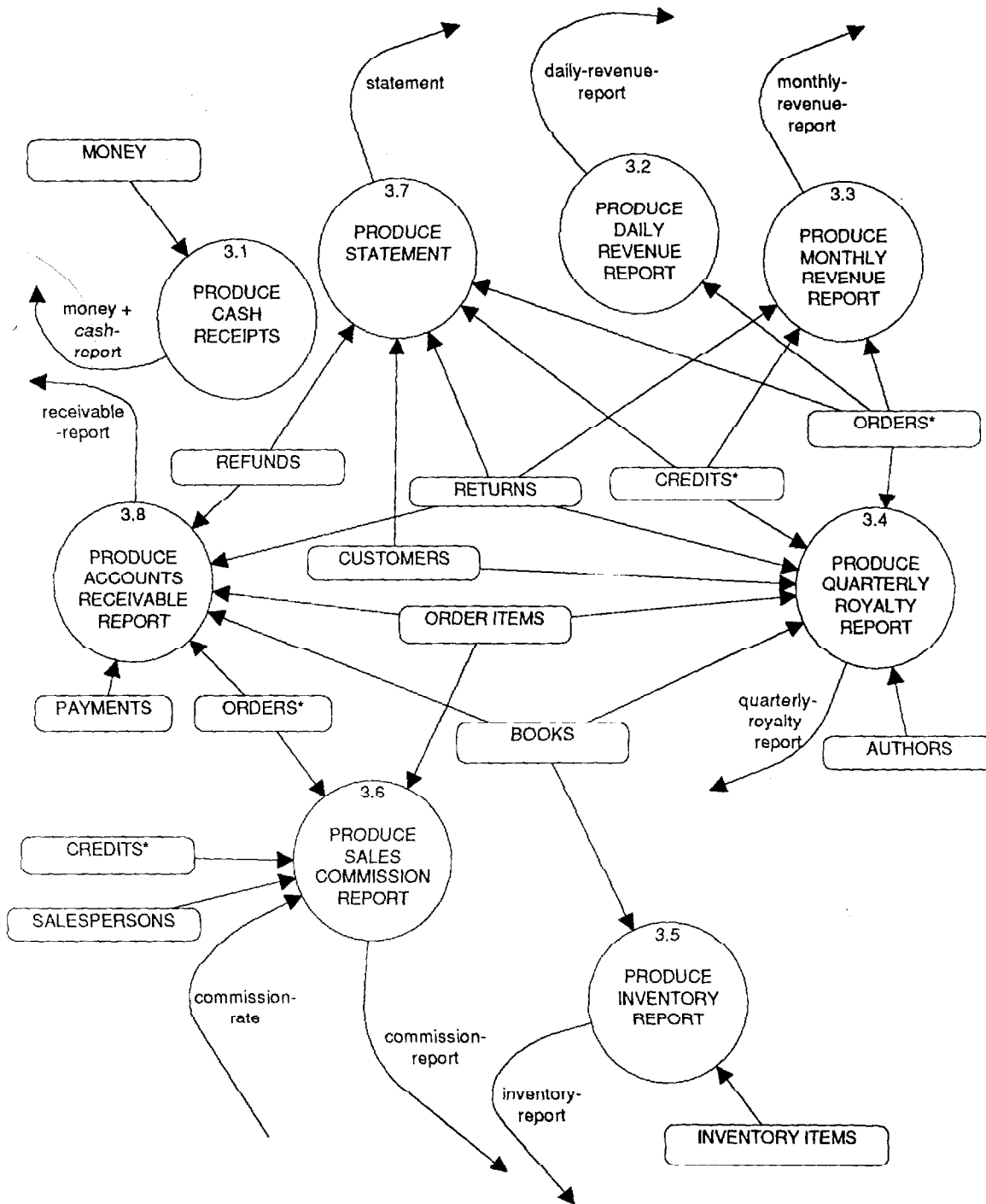


Figure 4: Manage printers

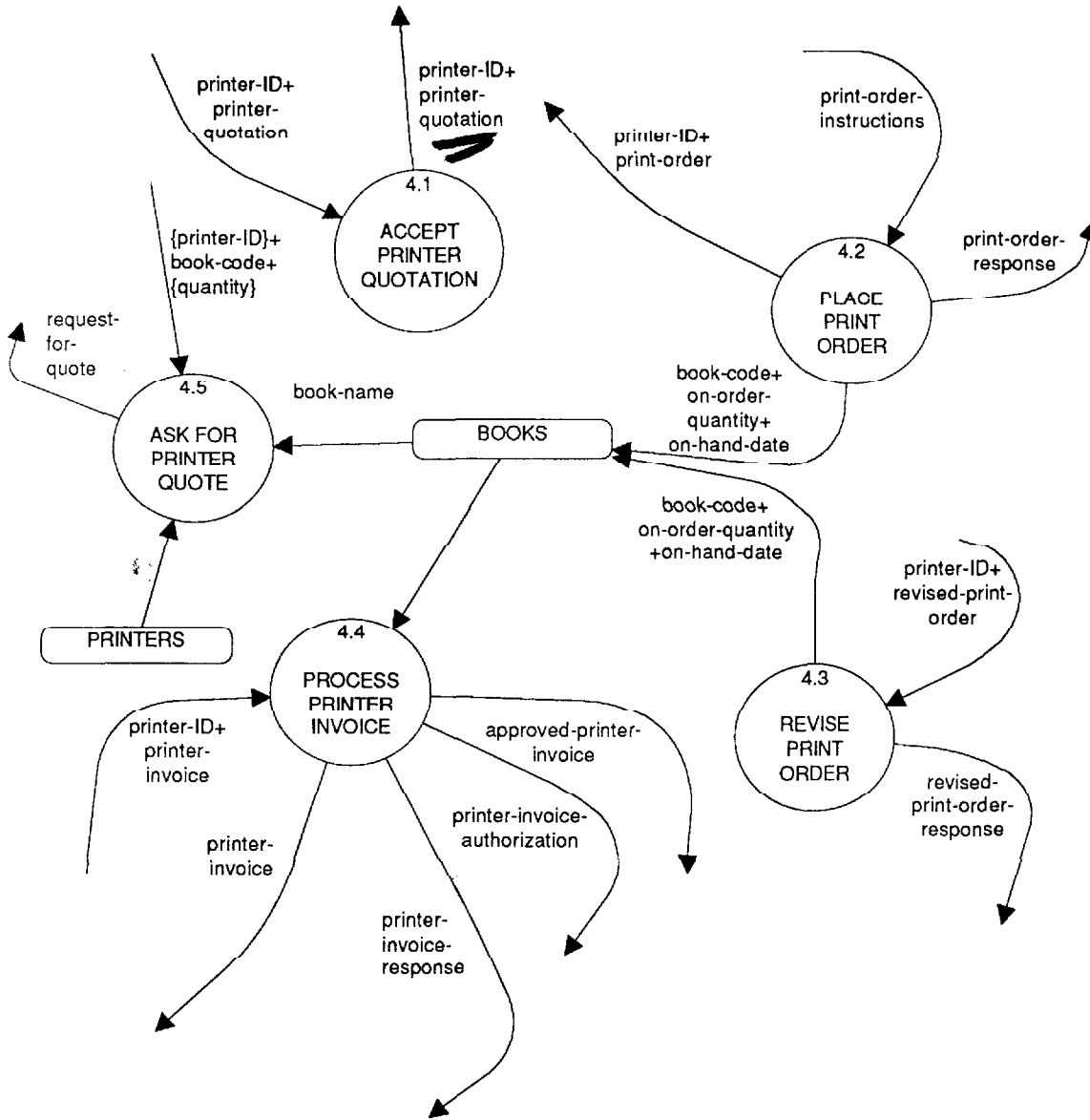


Figure 6: Interact with marketing

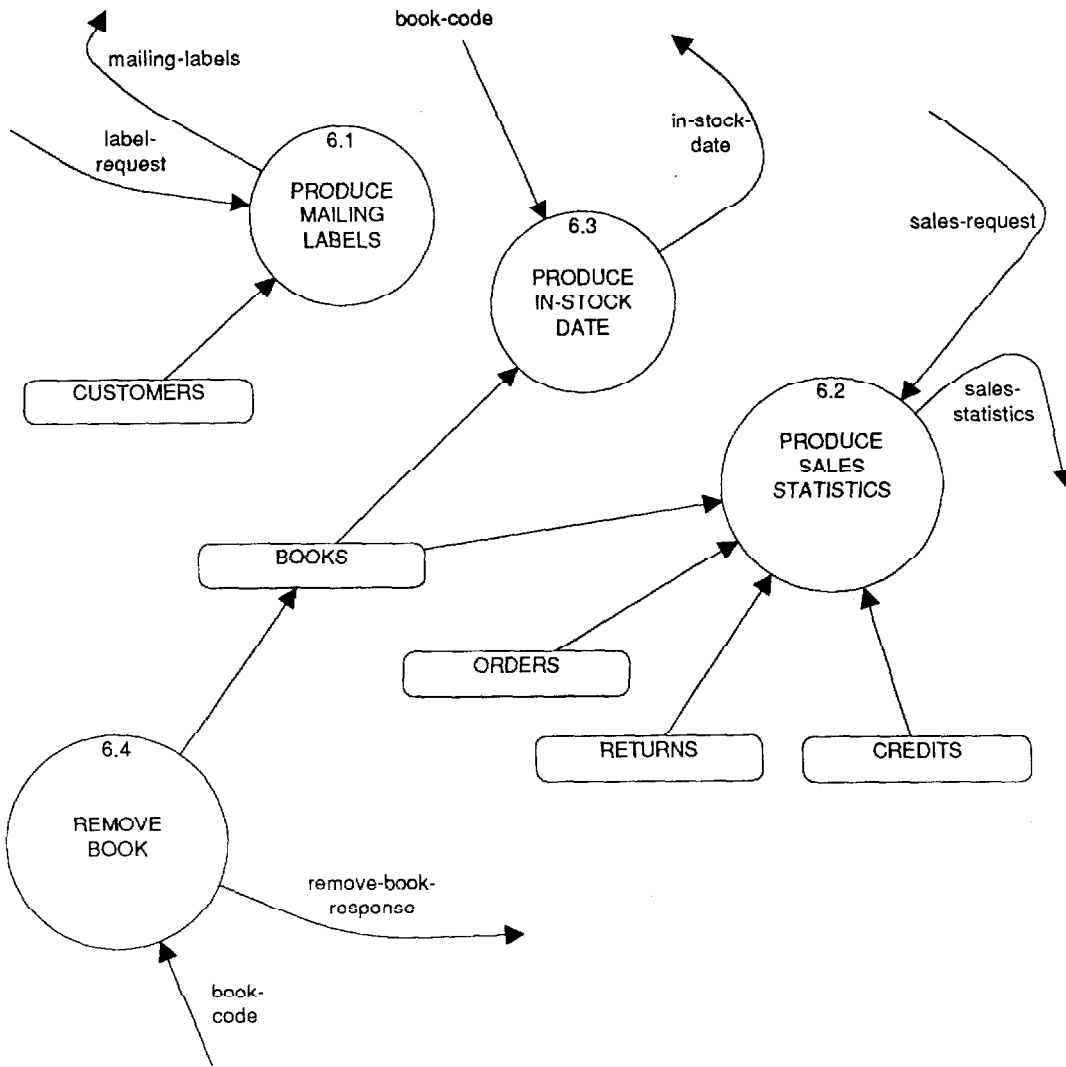


Figure 7: Interact with warehouses

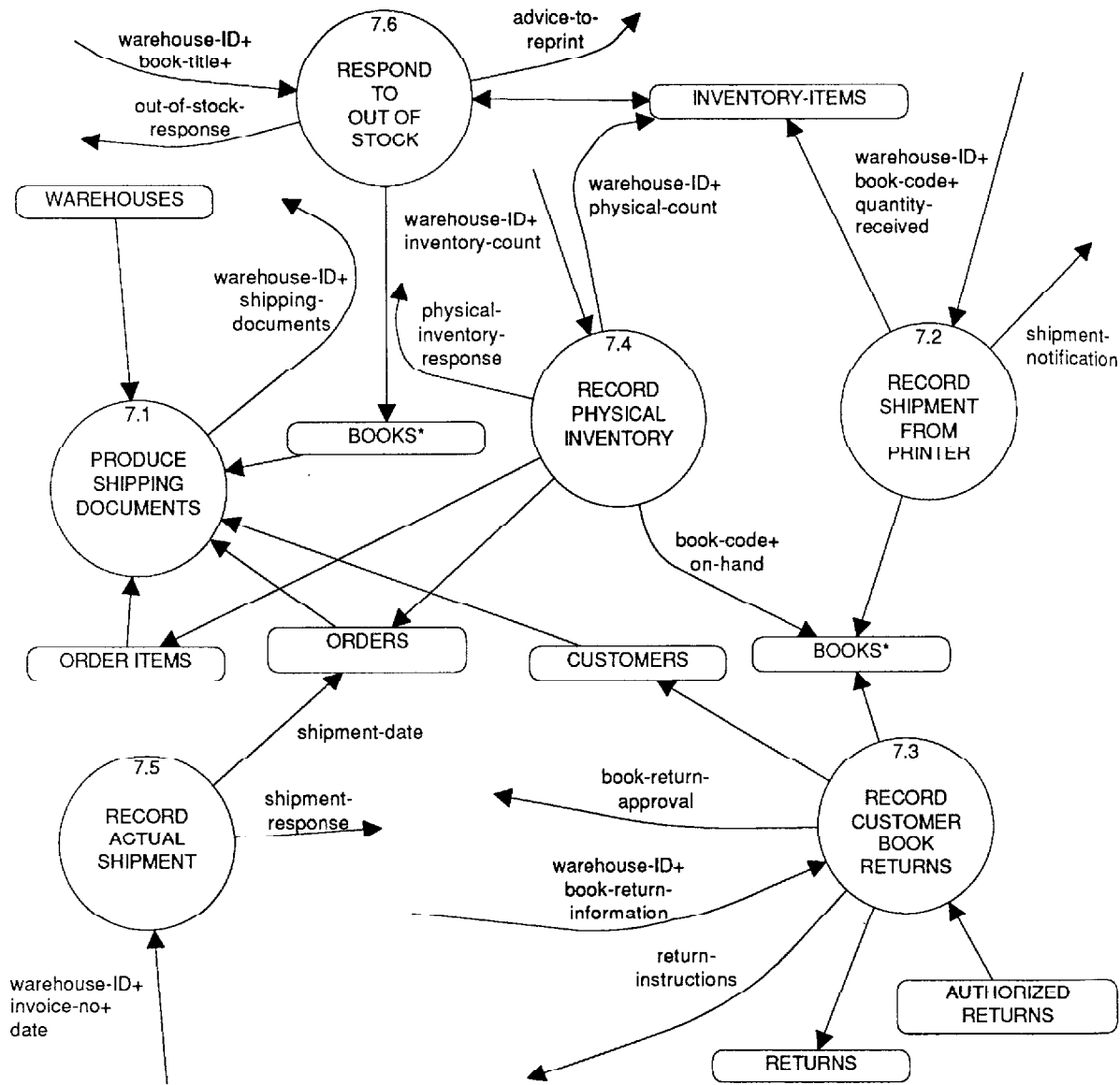
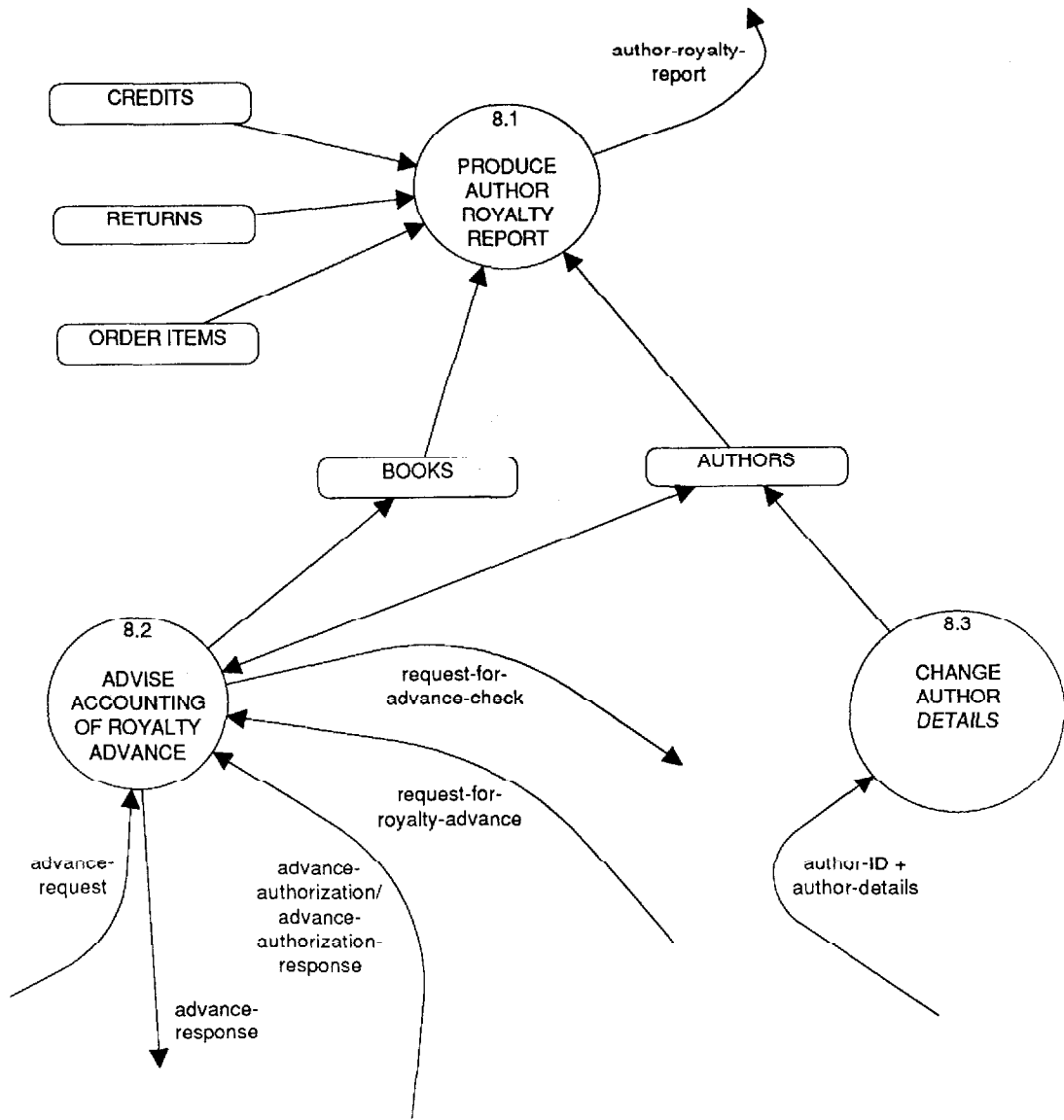


Figure 8: Interact with authors



F.4.3 The Data Dictionary

The data dictionary is organized in the fashion described in Chapter 10. Self-defining terms (data elements whose meanings are sufficiently well known that no explicit definition is required) are entered with a definition of **. See, for example, the definition of "country" and "state."

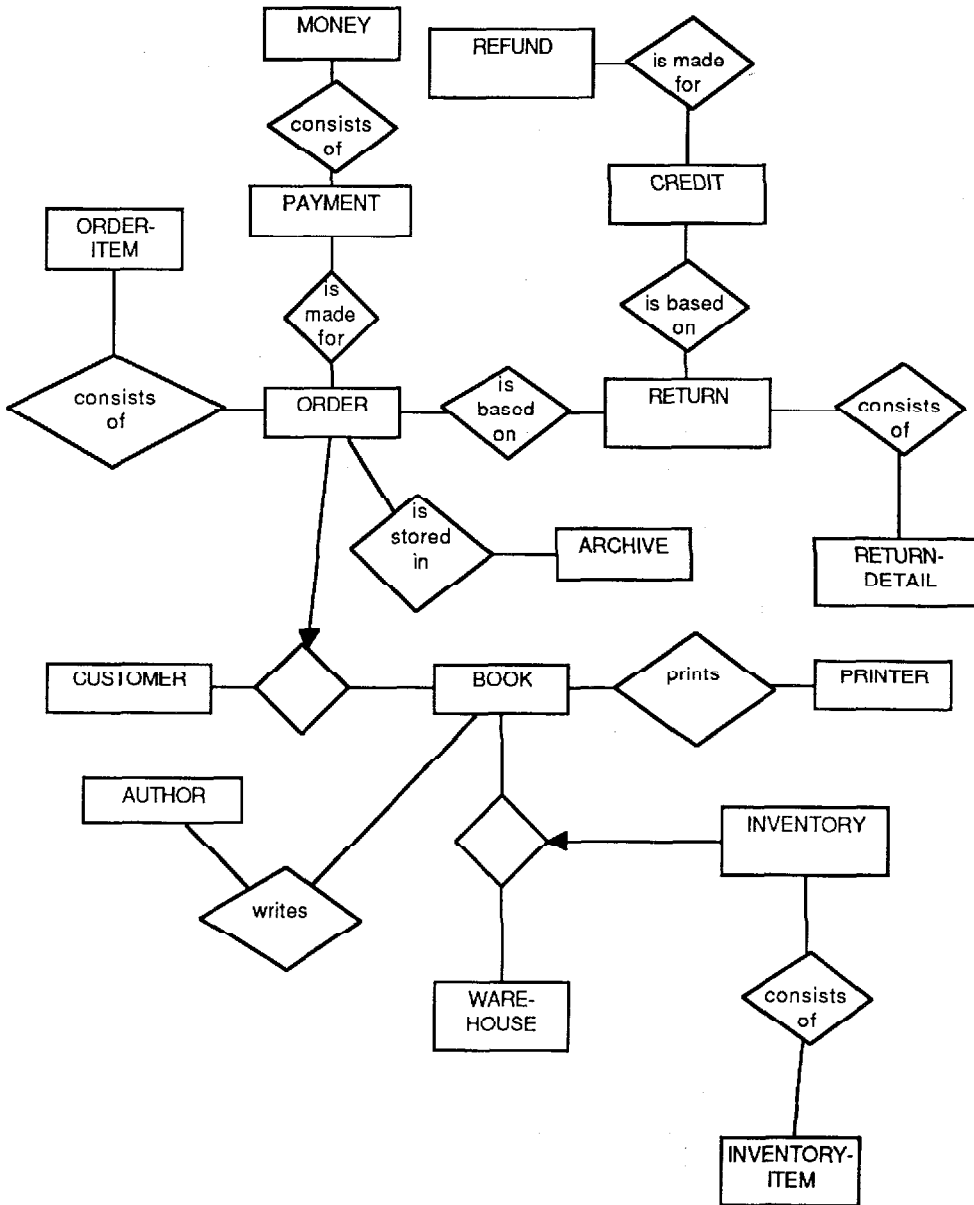
advance-amount	=	*amount of money requested for a royalty advance* *units: dollars*
advance-authorization-response	=	*response from management to a request for a royalty advance for an author* ["Yes" "No"]
advance-response	=	*response to acquisitions editor when he makes a request for an royalty advance for an author* ["No such author" "No such book" "Advance approved" "Advance denied"]
agency-plan-choice	=	*customer's choice of which level of agency plan to join* *range: 0-4*
agency-plan-level	=	*code to indicate what level of "standing orders" the customer has chosen for forthcoming YOURDON Press books* *range: 0-4*
alphabetic-character	=	*a letter in the English alphabet* **
alphanumeric-character	=	*either a number or a letter or a punctuation mark* [alphabetic-character numeric-digit punctuation-mark]
approved printer-invoice	=	*a printer invoice which has been approved by mgmt* book-code + invoice-amount
as-of-date	=	*the date that a physical inventory was taken* **
AUTHORS	=	{author}
author	=	*information maintained about each author* @author-ID + author-details + royalty-balance
author-details	=	courtesy-title + first-name + last-name + street-address+ city + postal-code + (country) + phone-number

author-ID	=	* identification of each YOURDON Press author Social Security numbers are not used because not all authors are American citizens* last-name + first-name
author-royalty-report	=	*report for authors showing royalties earned or lost on sales, credits, and returns of each book during a three-month period* {book-copies-total +book-revenue-total + book-royalty-total}
AUTHORIZED-RETURNS	=	{authorized-return}
authorized-return	=	*information about a group of books that YOURDON Press has authorized a customer to return for credit* @return-auth-# + return-details
back-order-OK	=	*an indication of whether a customer will place an order even if there are not enough books currently in stock* ["Yes" "No"]
BOOKS	=	{book}
book	=	*information maintained about a YOURDON Press book* @book-code + book-title + author-ID + total-in-stock + on-order-quantity + on-hand-date + royalty-rate + out-of-print-indicator + reorder-threshold
book-code	=	*a numeric code the identifies each book* 1{numeric-digit}
book-code-request	=	*message to management asking for a book-code to be assigned to a new book* "Please assign new book code for following book:"
book-copies-total	=	*total number of copies of a single book sold, taking into account returns and credits, during a three-month period* **
book-info-response	=	*information about the title, price, etc. of a book* book
book-return-approval	=	*response to customer when warehouse notifies the system that book returns have been received* ["This return is not authorized" "This return is OK"]

book-return-information	=	*information about a group of books that have been returned to the warehouse by a customer* (customer-ID) + (customer-name) + {book-code + quantity-returned} + return-auth-#
book-revenue-total	=	*total revenues earned for sales of a single book during a three-month period, taking into account credits and returns* *units: dollars*
book-royalty-total	=	*total royalties earned (or lost) from sales or returns or credits on a single book during a three-month period* *units: dollars*
book-title	=	*the full title of a YOURDON Press book* 1{alphanumeric-character}
commission-quantity	=	*the amount of commission paid to a salesperson for an individual book order; calculated in process 3.6* **
commission-rate	=	*the commission rate paid to salespeople for book sales. This is expressed as a fraction* *range: 0 - 0.25*
commission-report	=	*sales commission report* {salesperson-ID+invoice-number + commission-quantity} + total-commission}
company-name	=	*the name of a company or organization* **
country	=	*the name of a country, e.g., "Canada"* **
courtesy-title	=	*the prefix in front of a person's first name* ["Mr." "Mrs." "Ms." "Miss" "Dr." "Prof."]
CREDITS	=	{credit}
credit	=	*an individual credit given to a customer because of a problem with an order* @invoice-number + customer-ID + credit-date + book-code + credit-book-quantity + credit-amount
credit-agency-response	=	*response from a credit card agency to a request for charge authorization* ["Yes" "No"]

year-code = *the last two digits of the current year, e.g., "88"
if the current year is 1988*
2{numeric digit}2

F.4.4 The Entity-Relationship Diagram



F.4.5 The Process Specifications

PROCESS 1.1.1: EDIT ORDER DETAILS

BEGIN

IF **tentative-customer-ID** = "new"

customer-ID = next available customer-ID

credit-limit = standard credit limit

current-balance = 0

agency-plan-level = 0

customer = **customer-ID** + **customer-name** + (**company-name**) +

customer-address + **current-balance** + **credit-limit** +

agency-plan-level

 APPEND new **customer** record to **CUSTOMERS**

ELSE

 FIND **customer** in **CUSTOMERS** with **customer-ID** = **customer-ID**
 in **order-details**

 IF record cannot be found

order-response = "No such customer"

 DISPLAY **order-response**

 EXIT

 note: this means the order will not be processed any further

 ENDIF

DO WHILE there are more **order-items** in **order-details**

 FIND **book** in **BOOKS** with **book-code** = **book-code** in **order-item**

 IF record cannot be found

order-response = "No such book"

 DISPLAY **order-response**

 EXIT

 note: this means the order will not be processed any further

 ENDIF

END DO

IF **sales-tax-rate** is out of range

order-response = "Illegal sales tax rate"

 DISPLAY **order-response**

 EXIT

 note: this means the order will not be processed any further

ENDIF

IF **shipping-charges** are out of range

order-response = "Illegal shipping charges"

 DISPLAY **order-response**

 EXIT

 note: this means that order will not be processed any further

ENDIF

valid-order-details = **customer-ID** + {**order-item**} + **sales-tax-rate** +

shipping-charges + **payment-type** + (**order-payment**) +

 (**credit-card-number**) + **back-order-OK** + **order-type**

DISPLAY (to process 1.1.2) **valid-order-details**

END

```

PROCESS 1.1.2: CHECK BOOK IN STOCK
BEGIN
IF back-order-OK = "Yes"
    DISPLAY (to bubble 1.1.3) valid-order-details + "YONKERS"
ELSE
    IF order-type = "Walk-in"
        DO WHILE there are more order-items in valid-order-details
            FIND inventory-item in INVENTORY-ITEMS with book-code
                = book-code in valid-order-details and
                warehouse-ID = the location where this walk-in
                order was received
            READ inventory-item record
            IF inventory-quantity < quantity-ordered
                order-response = "Not enough books to fill your order"
                EXIT
                *note: this means no further processing of this order
            ENDIF
        END DO
        DISPLAY (to bubble 1.1.4) valid-order-details + warehouse-ID
            (of the warehouse at the location where this walk-in order
            was received)
    ELSE
        enough-books = "Yes"
        REPEAT UNTIL there are no more warehouses in WAREHOUSES or
            enough-books = "Yes"
            *note that this means we will examine at least one warehouse*
            enough-books = "Yes"
            DO WHILE there are more order-items in valid-order-details
                FIND inventory-item in INVENTORY-ITEMS with book-code
                    = book-code in valid-order-details and
                    warehouse-ID = warehouse-ID of current
                    warehouse record
                READ inventory-item record
                IF inventory-quantity < quantity-ordered
                    enough-books = "No"
                ENDIF
            END DO
        END REPEAT
        IF enough-books = "No"
            order-response = "Not enough books to fill your order"
            DISPLAY order-response
        ELSE
            DISPLAY (to bubble 1.1.4) valid-order-details +
                warehouse-ID of current warehouse record
        ENDIF
    ENDIF
ENDIF
END

```



```

PROCESS 1.1.3: CHECK CREDIT AUTHORIZATION
BEGIN
total-price = 0
REPEAT UNTIL there are no more order-items in valid-order-details
ADD (quantity-ordered * unit-price * discount) to total-price
END REPEAT
MULTIPLY total-price by (1 + sales-tax-rate)
ADD shipping-charges to total-price
credit-OK = "Yes"
CASE payment-type of
CASE payment-type = "Cash" or payment-type = "Check"
IF total-price > order-payment
order-response = "Purchase price exceeds amount paid"
DISPLAY order-response
*note: this means no further processing of the order will*
*take place at this point*
credit-OK = "No"
ENDIF
CASE payment-type = "Credit card"
credit-inquiry = "Request for authorization" + total-price
DISPLAY (to credit card agency) credit-inquiry
ACCEPT (from credit agency) credit-agency-response
IF credit-agency-response = "No"
order-response = "Credit request denied"
DISPLAY order-response
credit-OK = "No"
*note: this means no further processing of the order will*
*take place at this point*
ENDIF
CASE payment-type = "Bill me"
FIND customer in CUSTOMERS with customer-ID = customer-ID
in valid-order-details
READ customer record
IF current-balance + total-price > credit-limit
order-response = "Order exceeds your credit limit"
DISPLAY order-response
credit-OK = "No"
*note: this means no further processing of the order will*
*take place at this point*
ENDIF
END CASE
IF credit-OK = "yes"
DISPLAY (to bubble 1.1.4) valid-order-details + total-price + warehouse-ID
DISPLAY (to bubble 1.1.5) valid-order-details + warehouse-ID
ENDIF
END

```

PROCESS 1.1.4: ENTER ORDER

```

BEGIN
DO WHILE there are more order-items in valid -order-details
    CREATE order-item record from next order-item in valid-order details
    APPEND order-item record to ORDER-ITEMS
END DO
CREATE order record from valid-order-details and warehouse-ID
APPEND order record to ORDERS
CREATE invoice record from valid-order-details
APPEND invoice record to INVOICES
IF payment-type = "Cash" or "Check" or "Credit card"
    CREATE money record from valid-order-details
    APPEND money record to MONEY
    CREATE payment record from valid-order-details
    APPEND payment record to PAYMENTS
ENDIF
APPEND valid-order-details to ARCHIVES
order-response = "Order accepted"
DISPLAY order-response
END

```

PROCESS 1.1.5: CHECK INVENTORY FOR REPRINTING

```

BEGIN
DO WHILE there are more order-items in valid-order-details
    FIND inventory-item in INVENTORY-ITEMS with book-code = book-code in
        order-item and warehouse-ID matching warehouse-ID provided as
        input to this process
    READ inventory-item record
    SUBTRACT quantity-ordered from inventory-quantity
    *Note: this could result in a negative inventory; it simply means that the*
    *warehouse won't be able to fill the order until a reprint arrives*
    WRITE inventory-item record
    FIND book in BOOKS with book-code = book-code in order-item
    READ book record
    SUBTRACT quantity-ordered from total-in-stock
    WRITE book record
    IF total-in-stock < reorder-threshold
        low-inventory-message = book-code + total-in-stock +
            "time to reprint"
        DISPLAY low-inventory-message
    ENDIF
END DO
END

```

PROCESS 1.2: PROCESS SALESPERSON ORDER

At the present time, the policy for processing a salesperson order is the same as that for processing a normal customer order. See Figure 1.1 for details.

PROCESS 1.3: ENROLL CUSTOMER IN AGENCY PLAN

Precondition-1.

There is a **customer** in **CUSTOMERS** which matches **customer-ID** with
agency-plan-level = 0 and with **agency-plan-choice > 0**
and **agency-plan-choice** less than or equal maximum agency level.

Postcondition-1.

agency-plan-level in **customer** is set to **agency-plan-choice**.

PROCESS 1.4: SEND INVOICES

BEGIN

DO WHILE there are more invoices in **INVOICES** RFAD next **invoice** DISPLAY **invoice**

END DO

END

PROCESS 2.1: PROCESS PAYMENTS

BEGIN

IF **customer-ID** is present FIND record in **CUSTOMERS** with matching **customer-ID**

IF record cannot be found

 WRITE **payment-details** to **PAYMENTS** with **customer-ID =**
 "unapplied cash"

ELSE

 SUBTRACT **total-amount** from **current-balance** WRITE record to **CUSTOMERS** WRITE **payment-details** to **PAYMENTS**

ELSE

 WRITE **payment-details** to **PAYMENTS** with **customer-ID =**
 "unapplied cash" WRITE today's date + **customer-ID** + **payment-details** to **MONEY**

ENDIF

END

PROCESS 2.2: PROVIDE BOOK INFO

BEGIN

FIND **book** record in **BOOKS** with matching **book-title****book-info-response** = contents of entire **book** recordDISPLAY **book-info-response**

END

PROCESS 2.3: PROCESS RETURN REQUEST

BEGIN

FIND **order** in **ORDERS** that matches **invoice-number** in **return-request**

IF record is not found

return-request-response = "Cannot find this order" DISPLAY **return-request-response**

ELSE

```

READ order record
IF shipment-date is more than one year ago
    return-request-response = "Books were shipped more than a
        year ago"
    DISPLAY return-request-response
ELSE
    everything-OK = "yes"
    REPEAT UNTIL no more return-items in return-details
        FIND order-item in ORDER-ITEMS that matches
            invoice-number in return-request and
            book-code in return-item
        IF no record is found
            DISPLAY "This book was not part of the order"
            everything-OK = "no"
        ELSE
            READ order-item record
            IF quantity-to-return in return-item is more than
                half of quantity-ordered in order-item
                return-request-response = "Can't
                    return that many books"
                DISPLAY return-request-response
                everything-OK = "no"
            ENDIF
        ENDIF
    END-REPEAT
    IF everything-OK = "yes"
        READ return-auth-# from RETURN-AUTH-#
        return-request-response = "Return is OK," + "Please
            identify actual return with" + return-auth-#
        DISPLAY return-response-request
        WRITE return-details, return-auth-# to
            AUTHORIZED-RETURNS
        ADD 1 to return-auth-#
        WRITE return-auth-# to RETURN-AUTH-#
    ENDIF
ENDIF
ENDIF
END

```

PROCESS 2.4: RESPONSE TO ORDER STATUS INQUIRY

```

BEGIN
FIND order in ORDERS with matching invoice-number
IF record cannot be found
    order-status-response = "No order exists for that invoice number"
    DISPLAY order-status-response
ELSE
    READ order record in ORDERS with matching invoice-number
    order-status-response = order-date + {order-items} + shipment-date
    DISPLAY order-status-response

```