

**Name:** INVENTORY-EXCEPTION-REPORT  
**Aliases:** REORDER-REPORT. PURCHASING-REPORT  
**Description:** Group item representing daily list of parts to be reordered. Sent to purchasing. See back of card for structure.

**Format:** Group item.

**Location:** output to printer

**Name:** PART-NUMBER  
**Aliases:**  
**Description:** Key field that uniquely identifies a specific part in inventory.

**Format:** Alphanumeric; 8 characters.  
PIC X(8).

**Location:** INVENTORY-EXCEPTION-REPORT  
INVENTORY  
REORDER

**Name:** REORDER-QUANTITY  
**Aliases:**  
**Description:** The number of units of a given part that are to be reordered at a single time.

**Format:** numeric; 5 digits.  
PIC 9(5)

**Location:** INVENTORY-EXCEPTION-REPORT  
INVENTORY  
REORDER

# Data Dictionary

|               |                                    |             |
|---------------|------------------------------------|-------------|
| <u>COURSE</u> |                                    |             |
| I.D.          | Course identification number       | STRING [10] |
| Course Name   | Formal Course Name                 | STRING [30] |
| HWK_TOT       | Total homework points possible     | INTEGER     |
| HWK%          | Weight percent for homework scores | INTEGER     |
| TEST_TOT      | Total test points possible         | INTEGER     |
| TEST%         | Weight percent for test            | INTEGER     |
| PROJECT       | Project points possible            | INTEGER     |
| PROJECT%      | Weight percent for project         | INTEGER     |

|                |                                       |             |
|----------------|---------------------------------------|-------------|
| <u>STUDENT</u> |                                       |             |
| STUDENT_STAT   | H(Hearer, P(Pass/Fail), W(Withdrawn)) | CHAR        |
| ID             | Student Number                        | LONGINT     |
| L_NAME         | Last Name                             | STRING [30] |
| F_NAME         | First Name                            | STRING [20] |
| FINAL          | Score on final                        | INTEGER     |
| PROJECT        | Score on Project                      | INTEGER     |
| %              | Overall Percentage                    | INTEGER     |
| SEM_GRADE      | Actual Grade Assigned                 | CHAR        |

|                 |  |         |
|-----------------|--|---------|
| <u>HOMEWORK</u> |  |         |
| HWK#            | Indicates the hwk number score applies | INTEGER |
| SCORE           | Indicates score on homework            | INTEGER |

|             |   |         |
|-------------|---|---------|
| <u>TEST</u> |   |         |
| TEST#       | Indicates the test number score applies | INTEGER |
| SCORE       | Indicates score on test                 | INTEGER |

INVENTORY FILE: Main Key = Inventory Item Number; Alternate Key = Item Description

|                |                            |                         |                          |
|----------------|----------------------------|-------------------------|--------------------------|
| <u>LAYOUT:</u> | Inventory Item Number      | Item Name               | Item Description         |
|                | Item Category              | Price Structure         | Price Code               |
|                | Commission Code            | Miscellaneous Item Flag | Average Cost per Unit    |
|                | Cost at Last Purchase      | Quantity on Order       | Quantity on Hand         |
|                | Reorder Level              | Dead Item Flag          | Quantity Purchased       |
|                | Quantity Committed         | Warehouse Location Code | G/L Account Number       |
|                | Order Unit of Measure      | Price Unit of Measure   | Conversion Factor        |
|                | Quantity Sold Year-to-Date | Quantity Sold Last Year | Sales Year-to-Date       |
|                | Sales Last Year            | Expenses Year-to-Date   | Expenses Last Year       |
|                | New Price Structure        | New Price Code          | New Price Effective Date |

CUSTOMER FILE: Main Key = Customer Number; Alternate Key = Customer Name

|                |                            |                    |                           |
|----------------|----------------------------|--------------------|---------------------------|
| <u>LAYOUT:</u> | Customer Number            | Customer Name      | Customer Address          |
|                | Customer Telephone Number  | Point of Contact   | Salesman Number           |
|                | Balance Method             | Statement Cycle    | Credit Approver           |
|                | Credit Limit               | Credit Rating      | Financial Charge Applies? |
|                | Partial Shipment Accepted? | Terms Code         | Tax Code                  |
|                | Sales Period-to-Date       | Sales Year-to-Date | Expenses Period-to-Date   |
|                | Expenses Year-to-Date      | Account Balance    | Unposted Balance          |
|                | Last Payment Amount        | Last Payment Date  |                           |

VENDOR FILE: Main Key = Vendor Number; Alternate Key = Vendor Name

|                |                        |                     |                       |
|----------------|------------------------|---------------------|-----------------------|
| <u>LAYOUT:</u> | Vendor Number          | Vendor Name         | Vendor Address        |
|                | Terms                  | Due Dates           | Discount Dates        |
|                | Discount Percentage    | Last Purchase Date  | Lead Time             |
|                | Purchases Year-to-Date | Purchases Last Year | Discount Year-to-Date |
|                | Discount Last Year     |                     |                       |

ORDER FILE: Main Key = Order Number; Alternate Key = Customer Number

|                |                        |                                     |                     |
|----------------|------------------------|-------------------------------------|---------------------|
| <u>LAYOUT:</u> | Order Number           | Customer Number                     | Shipping Address    |
|                | Billing Address        | Order Date                          | Salesman            |
|                | Customer PO Number     | Ship Via                            | Ship Date           |
|                | Terms                  | Tax Percentage                      | Discount Percentage |
|                | Profit Center          | Cash Received                       | Check Number        |
|                | Number of Shipments    | [Inventory Item Number              | Quantity Ordered    |
|                | Price                  | Discount                            | Quantity Shipped    |
|                | Quantity Back Ordered] | NOTE: [ ... ] for each item ordered |                     |

ITEM RESOURCE FILE: Main Key = Vendor Item Number & Vendor Name

Alternate Key = Inventory Item Number

|                |                    |                     |                       |
|----------------|--------------------|---------------------|-----------------------|
| <u>LAYOUT:</u> | Vendor Item Number | Vendor Name         | Inventory Item Number |
|                | Cost Per Unit      | Discounts in Effect |                       |

BACK ORDER FILE: Main Key = Order Number & Inventory Item Number

Alternate Key = Inventory Item Number & Order Date & Order Number

|                |                       |            |              |
|----------------|-----------------------|------------|--------------|
| <u>LAYOUT:</u> | Inventory Item Number | Order Date | Order Number |
|                | Back Order Quantity   |            |              |

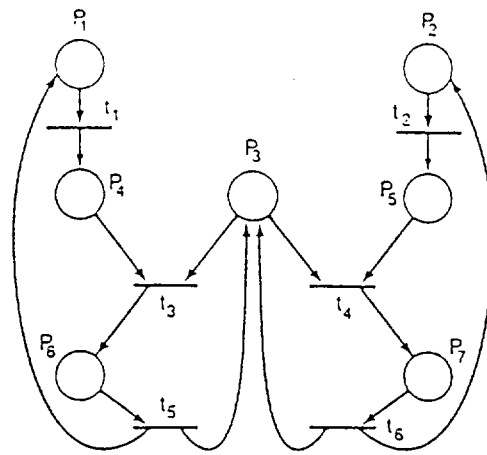


Figure 5.17 A Petri net.

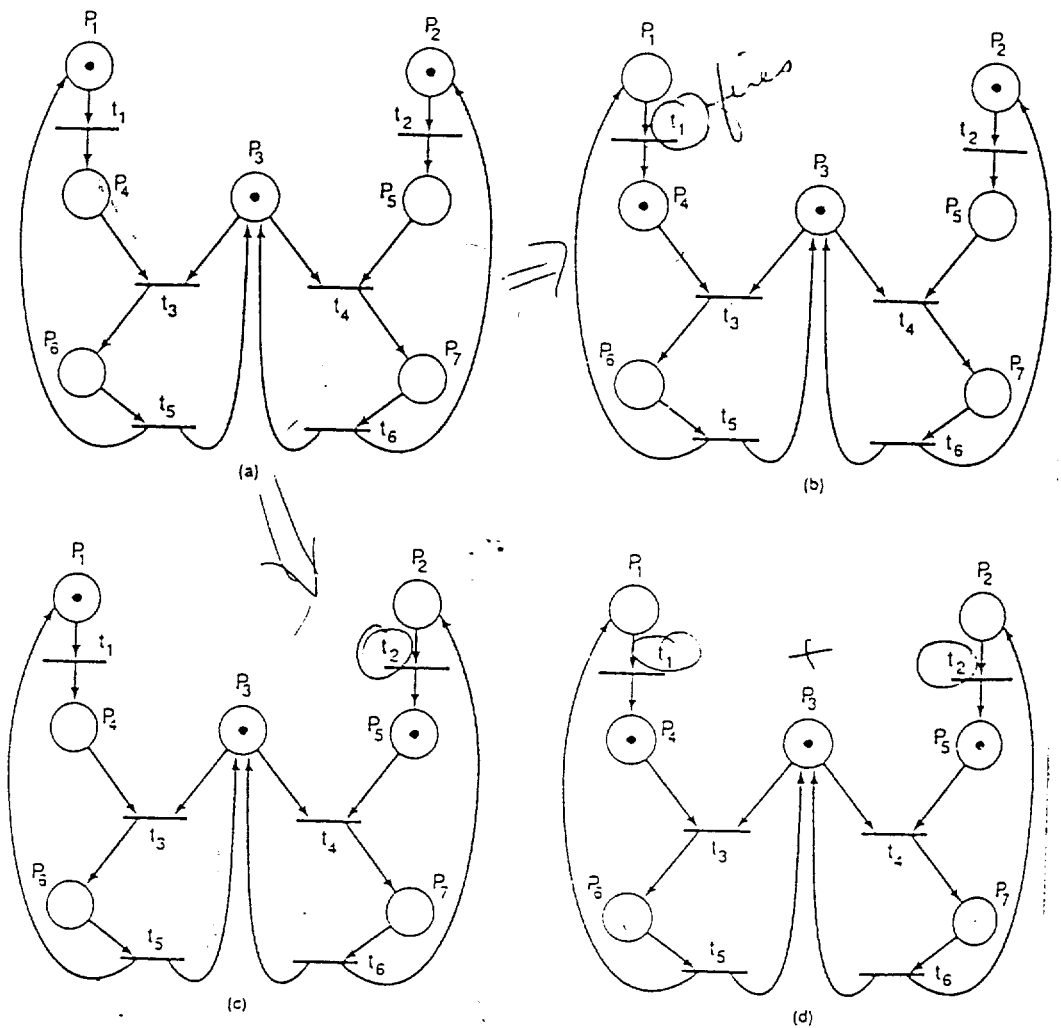


Figure 5.18 Evolution of a Petri net.  
 (a) Initial marking. (b)  $t_1$  fires from initial marking.  
 (c)  $t_2$  fires from initial marking. (d)  $t_1$  and  $t_2$  fire from initial marking.

*asynchronous  
 flow*

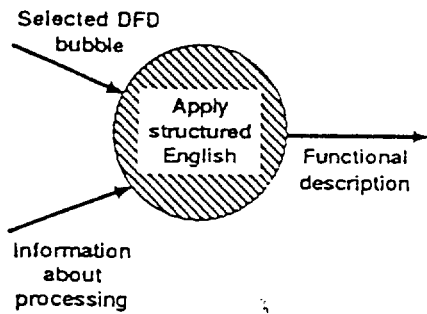


FIGURE 5.12  
Generating a functional description.

```

procedure: Apply Structured English
  select DFD bubble, call it target.bubble;
  do while target.bubble needs further refinement
    if target.bubble is multifunctional
      then decompose as required;
      select new target.bubble;
    else no further refinement needed;
    endif
  enddo
  use procedure statement to name target.bubble;
  describe sequence of process tasks;
  repeat until all process tasks have been described
    case of processing task logic:
      case: process task includes a sequence
        state sequence of steps;
      case: process task is a condition
        use if-then-else to describe condition;
      case: process task includes repetition
        use do while or repeat until to describe;
      case: process task includes a number of cases
        use case of to describe;
    endcase
  endrep
  review structured English that has been generated;
  modify as required using procedure Apply Structured English;
end procedure
  
```

## DESCRIPTION OF FUNCTIONAL SUBSYSTEMS

### INVENTORY MANAGEMENT

INPUT : orders transmitted from stores  
manual adjustments (CRT input)  
receipt reports (from warehouse - via CRT)

OUTPUT : stock listing  
adjustment audit report  
reorder lists by product line and/or  
vendor  
inventory level report showing EOQ,  
reorder point safety stock, etc.

### ORDER ENTRY SUBSYSTEM

INPUT : orders input from stores  
credit authorization/override (via CRT)  
shipping adjustments (via CRT)  
credit limit information from A/R

OUTPUT : picking ticket/packing slip  
invoices  
listing of orders over credit limit  
orders pending for over 48 hours  
orders shipped today  
update transaction file for A/R  
update file for G/L with today's sales

### ACCOUNTS RECEIVABLE

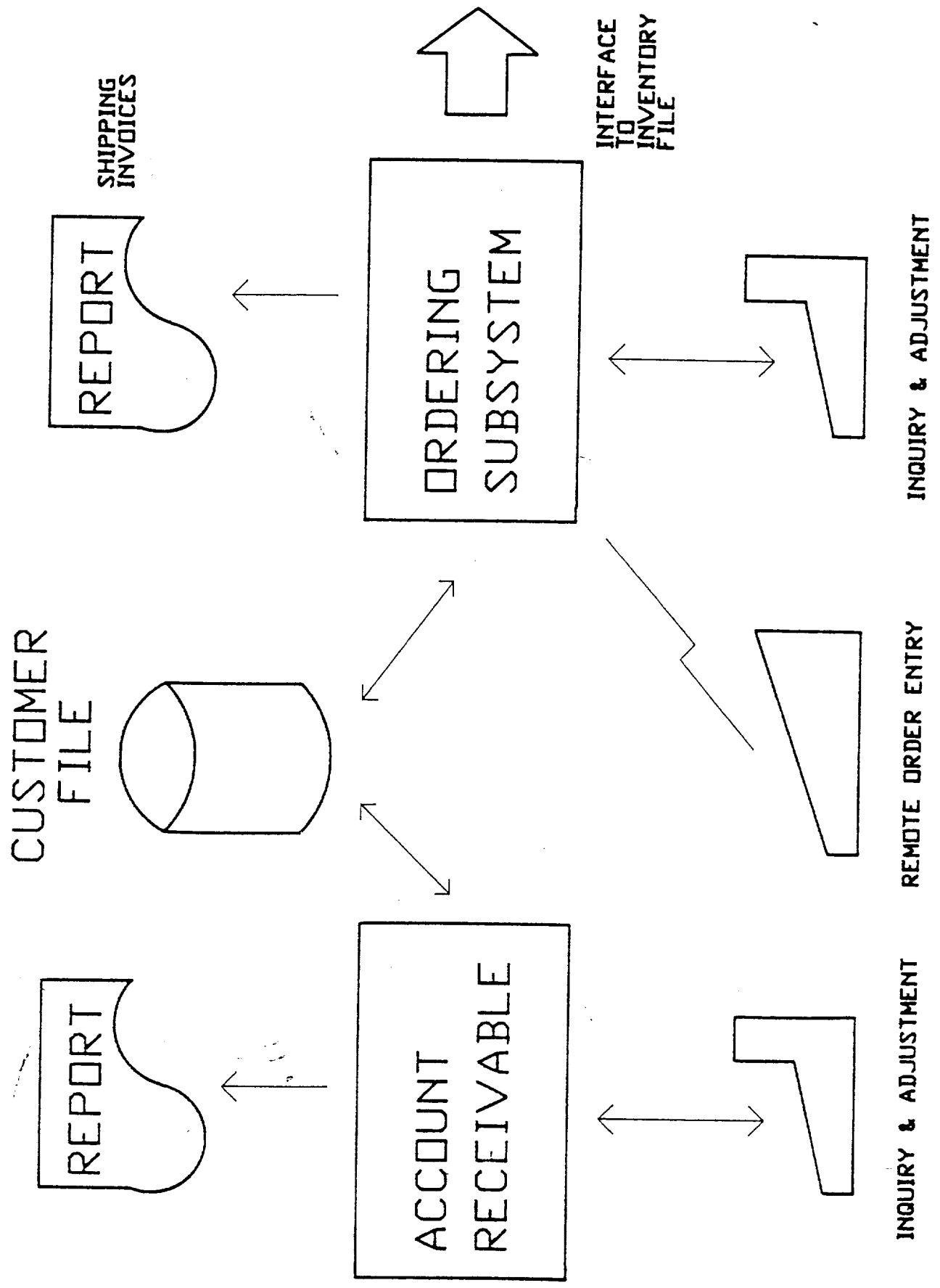
INPUT : invoice totals from order entry  
credit/debit adjustments (via CRT)  
payment postings (A/R clerks via CRT)

OUTPUT : customer statements  
credit/debit adjustment memos  
collection notices  
aged account listing  
trial balance  
aged trial balance  
Daily payment/adjustment audit register  
Daily receipt totals to post to G/L

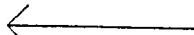
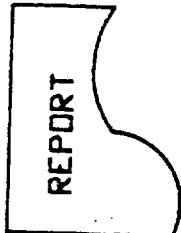
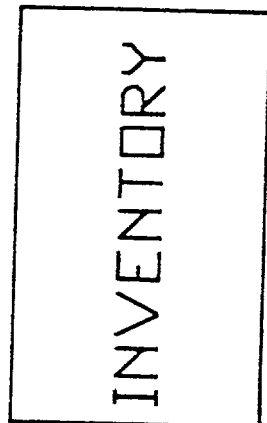
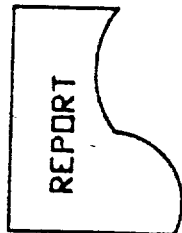
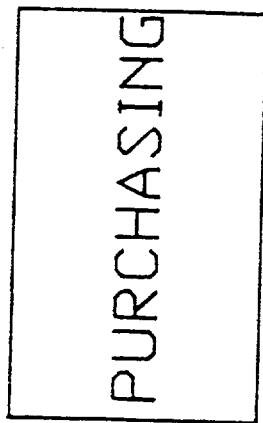
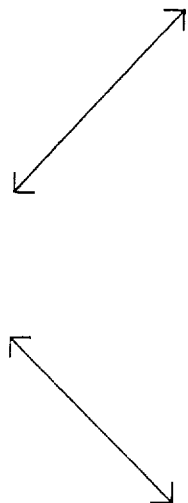
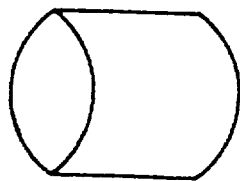
### PURCHASING SUBSYSTEM

INPUT : payment authorizations  
purchase order requests  
information on vendors

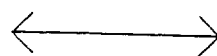
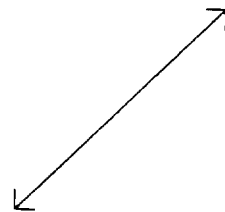
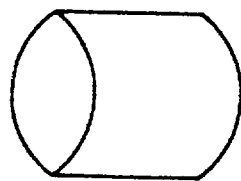
OUTPUT : purchase orders (3-part form)  
purchases by vendor  
Purchases by buyer  
vendor data



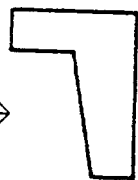
INVENTORY  
FILE



VENDOR  
FILE



ENTRY & INQUIRY



UPDATE & INQUIRY



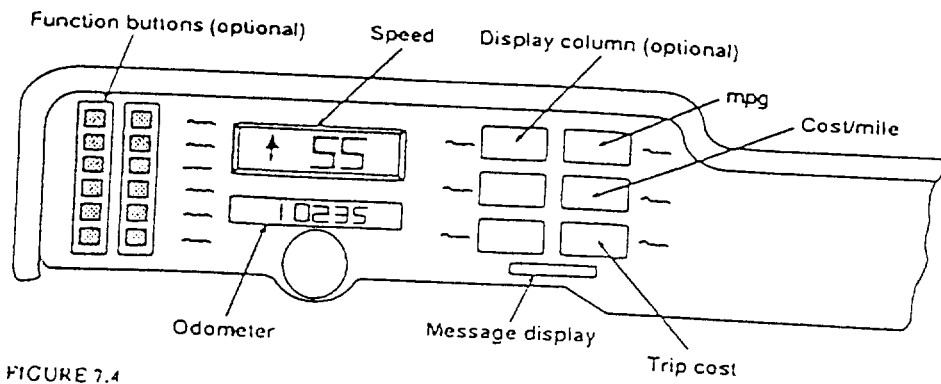
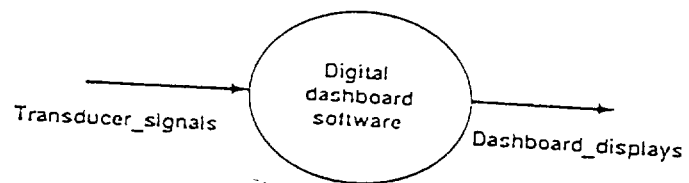
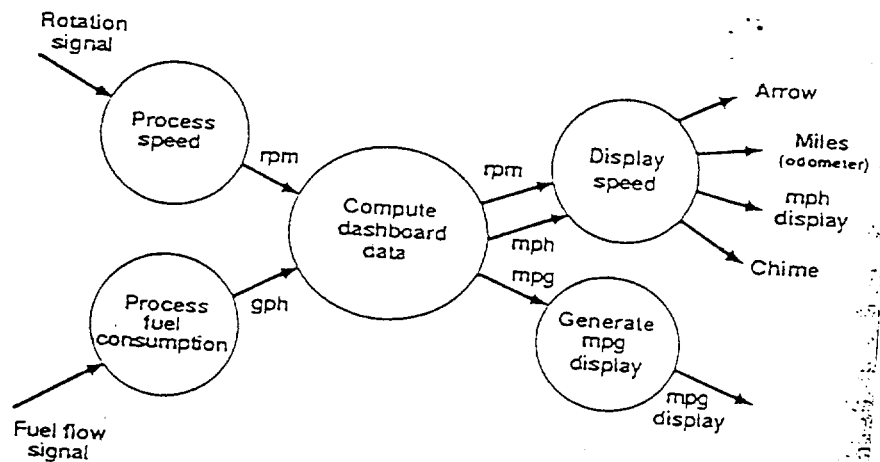


FIGURE 7.4  
A design example.



Data dictionary  
 transducer\_signals=rotation\_signal + fuel\_flow\_signal  
 dashboard\_displays=mph\_display + arrow + mpg\_display + odometer + chime  
 rotation\_signal = ...

(a)



(b)

FIGURE 7.5  
 (a) Digital dashboard software—level 01 DFD. (b) Digital dashboard software—level 02 DFD.

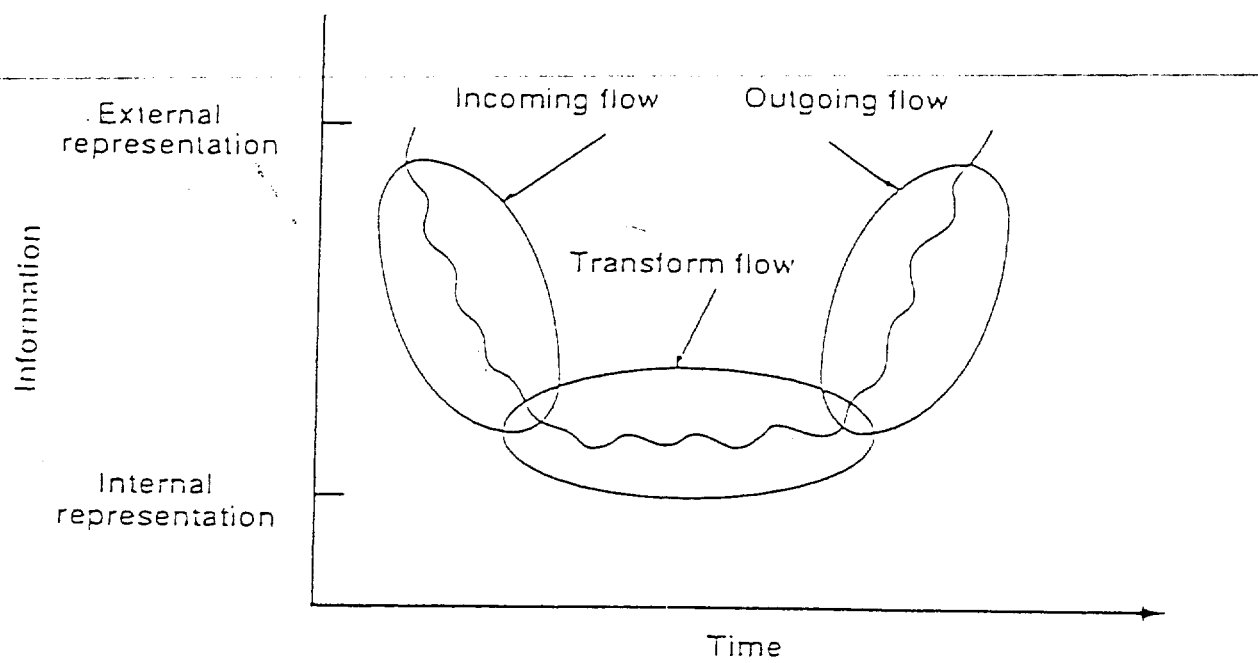


FIGURE 7.1  
Flow of information.

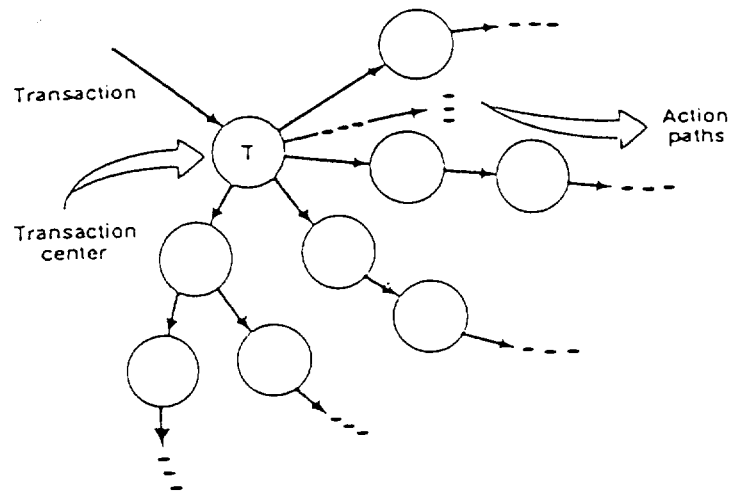


FIGURE 7.2  
Transaction flow.

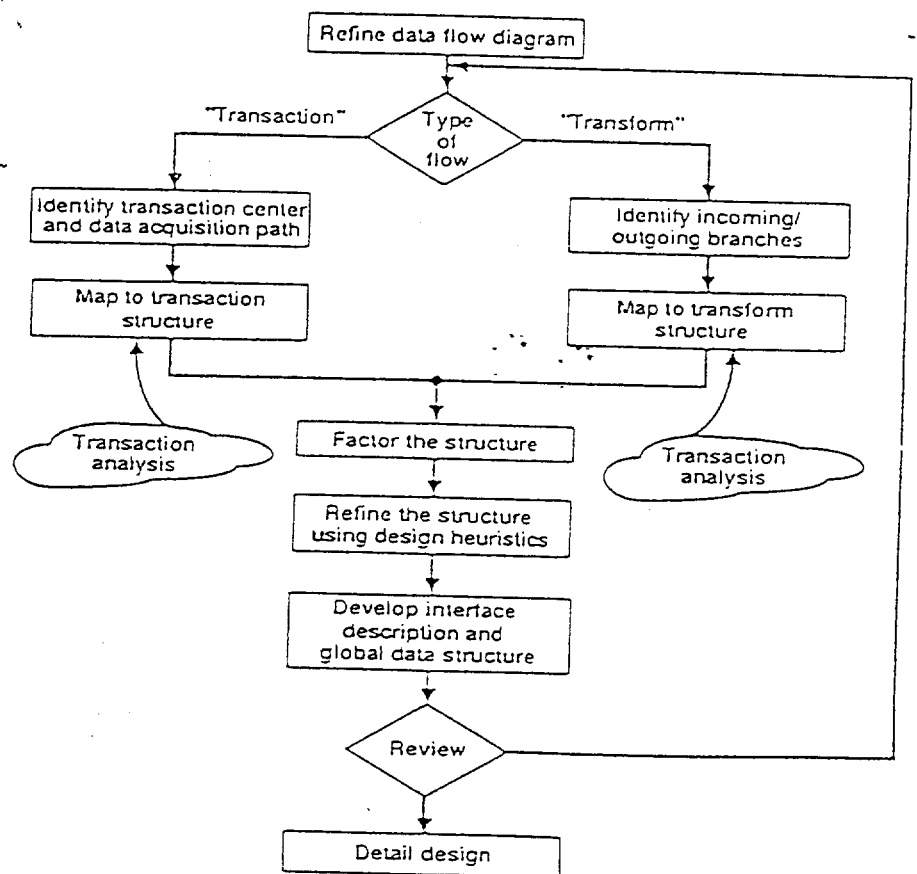


FIGURE 7.3  
Data flow-oriented design.

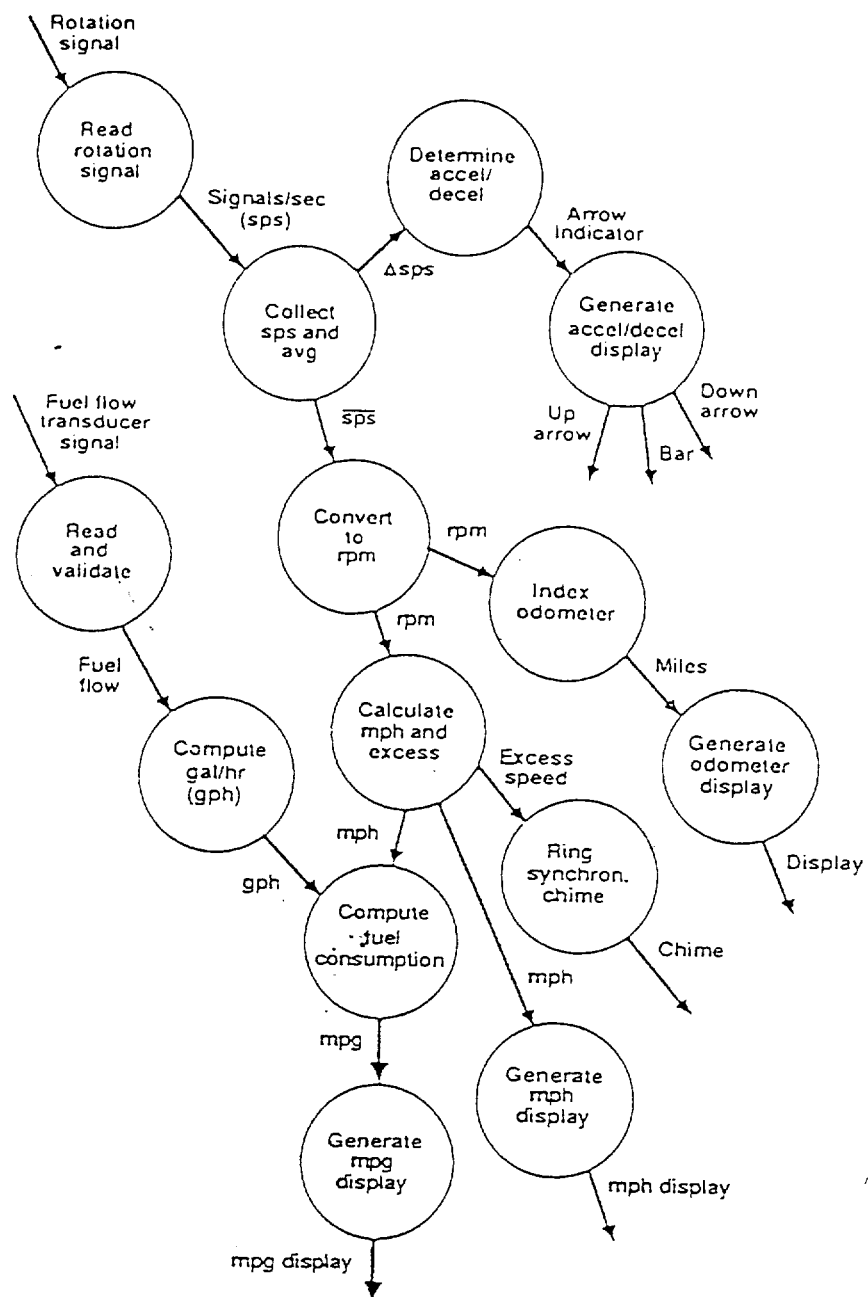


FIGURE 7.5  
(c) Digital dashboard software — level 03 DFD.

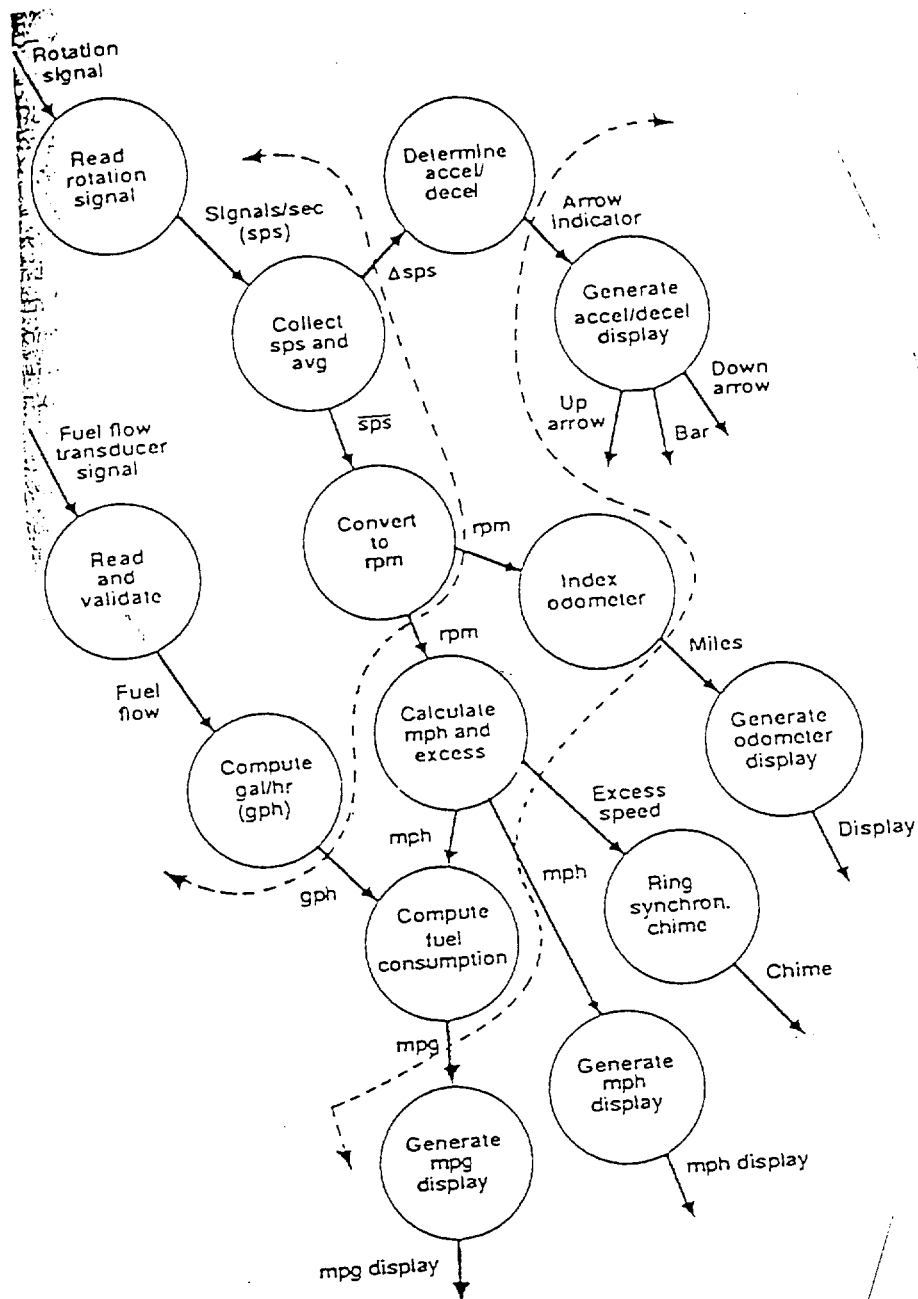


FIGURE 7.6  
Digital dashboard software—level 03 DFD with flow boundaries.

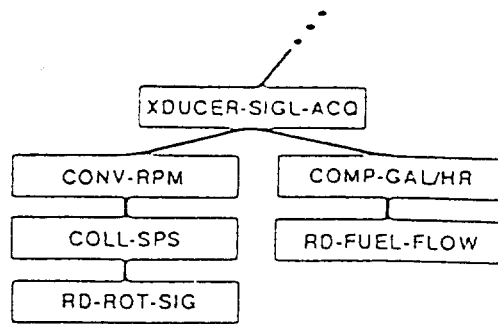


FIGURE 7.10  
Unrefined incoming structure.

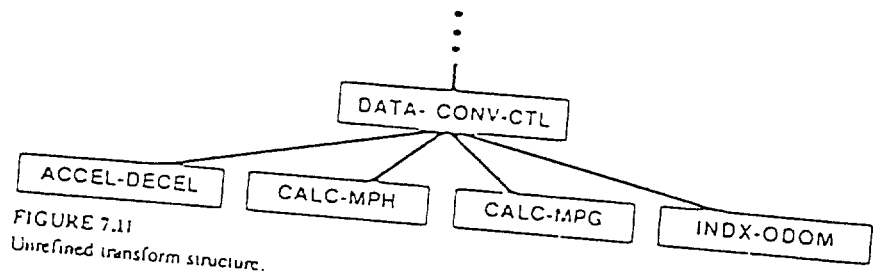


FIGURE 7.11  
Unrefined transform structure.

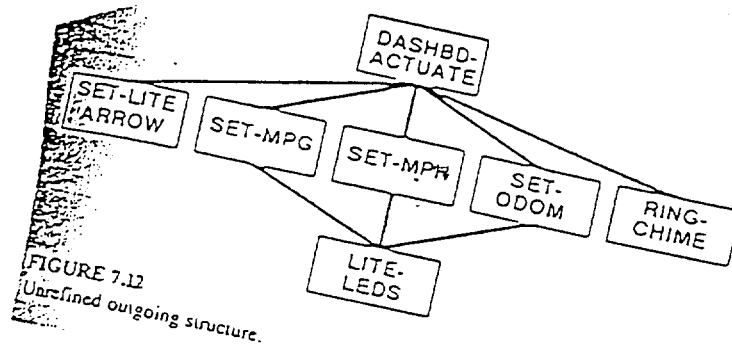


FIGURE 7.12  
Unrefined outgoing structure.

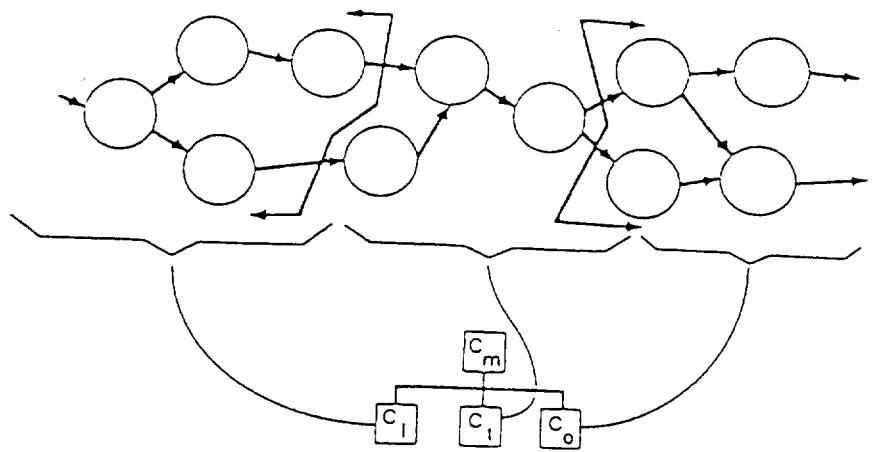


FIGURE 7.7  
Transaction mapping.

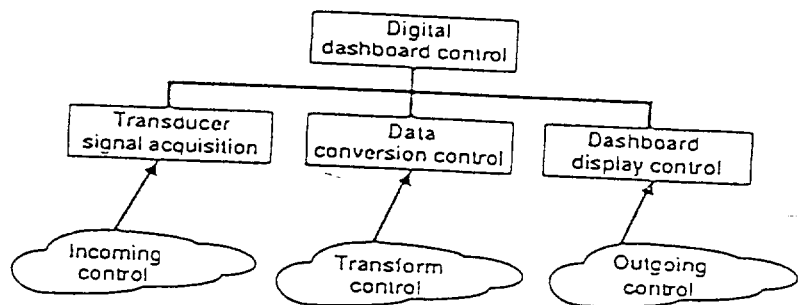


FIGURE 7.8  
First-level factoring.

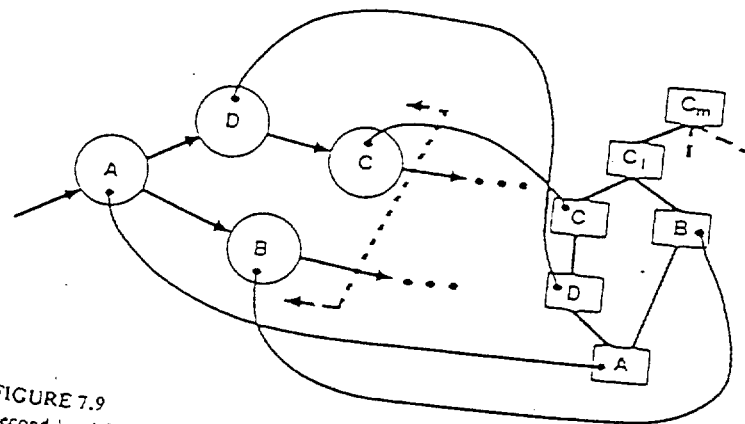


FIGURE 7.9  
Second-level factoring.

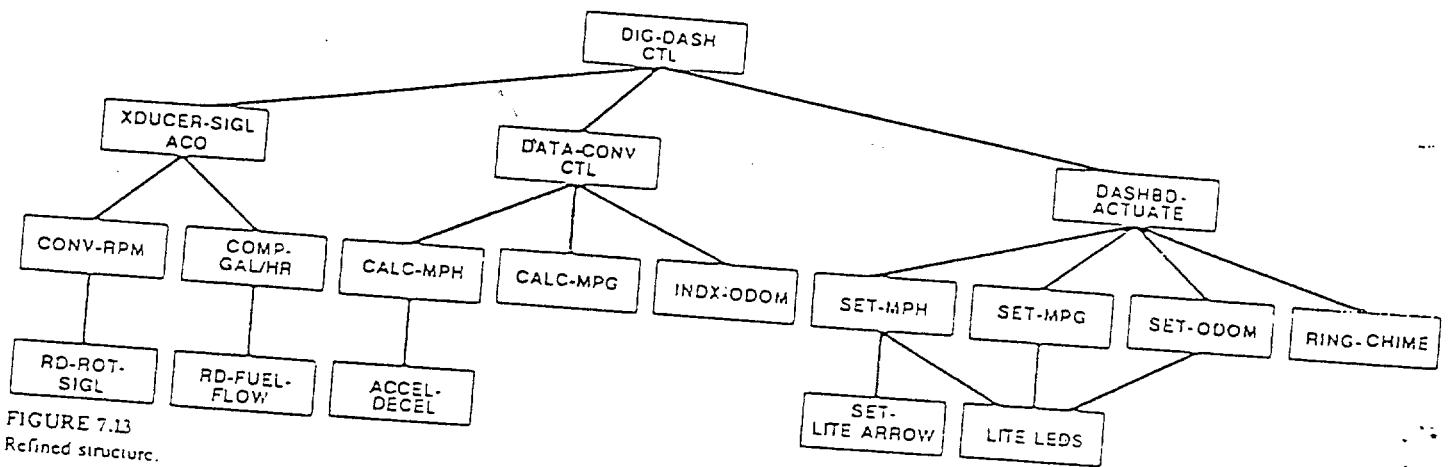
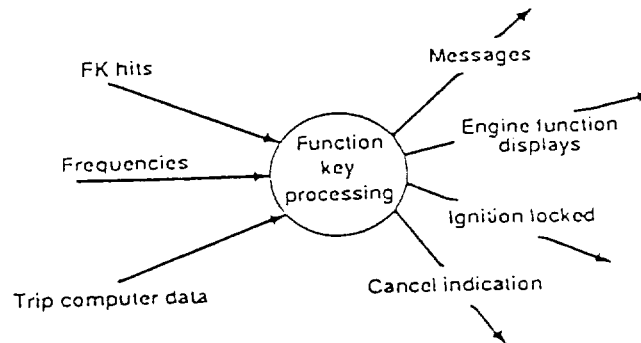


FIGURE 7.13  
Refined structure.



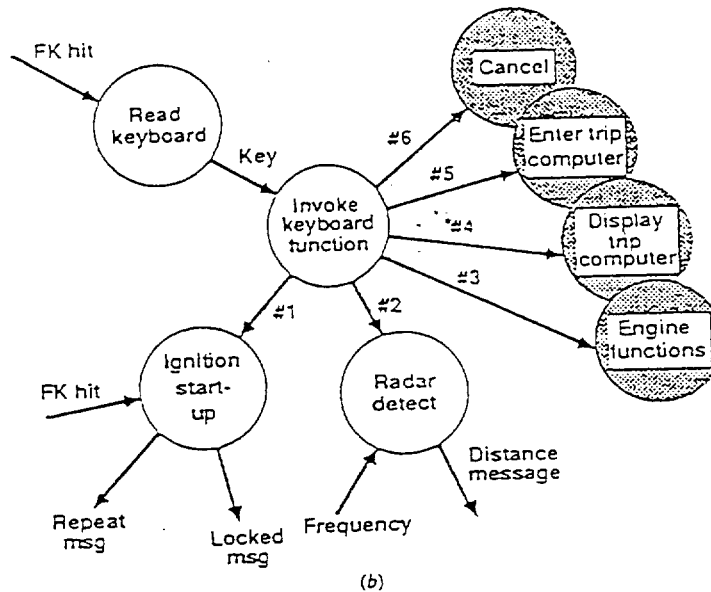
| Button | Function selected  |
|--------|--|
| 1      | Initiate ignition start-up sequence  |
| 2      | Activate radar detector (only available where legal!)                                  |
| 3      | Activate display of various engine functions including oil pressure, temperature, etc. |
| 4      | Display trip computer data   |
| 5      | Enter trip computer data   |
| 6      | Cancel previous key (clear)  |



Data dictionary

FK hits = [ menu select | ignition code | keyed data ]  
 menu select = [ 1 | 2 | 3 | 4 | 5 | 6 ]  
 and so on ...

(a)



(b)

FIGURE 7J4

(a) Dashboard function key processing—level 01 DFD. (b) Dashboard function key processing—level 02 DFD.

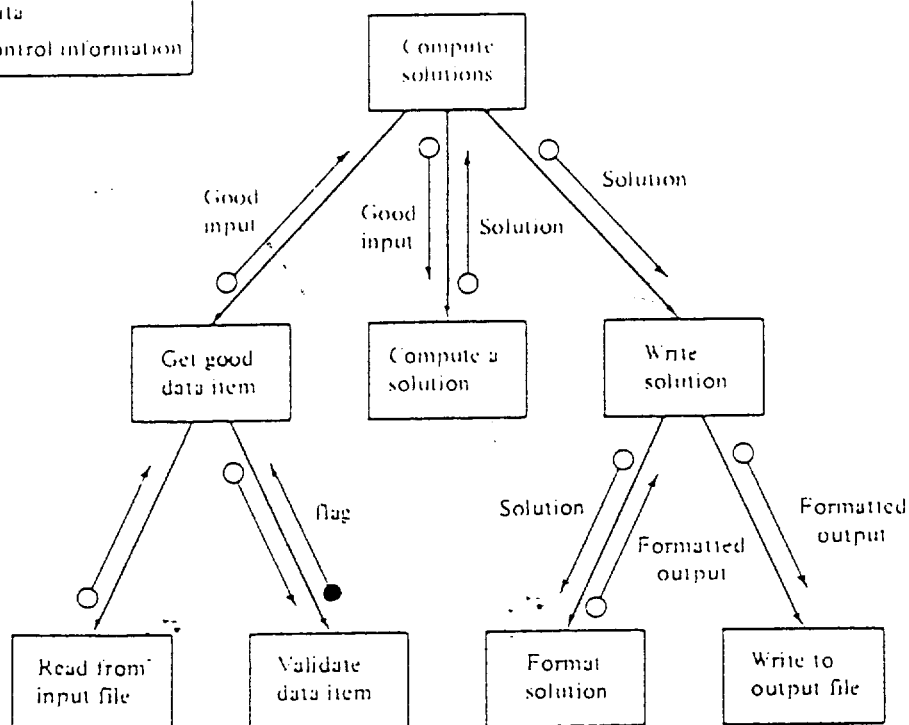
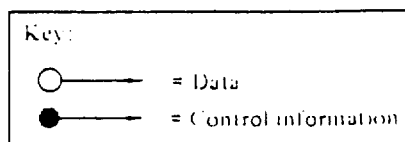
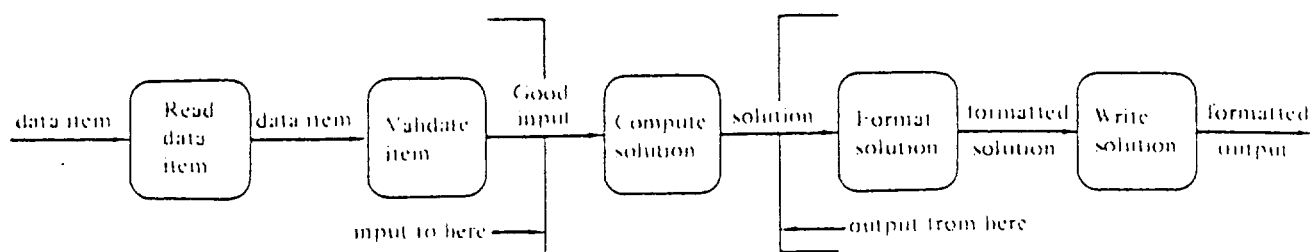


Figure 5.14 Conversion of a transform-centered data flow diagram into an input, process, output structure chart.

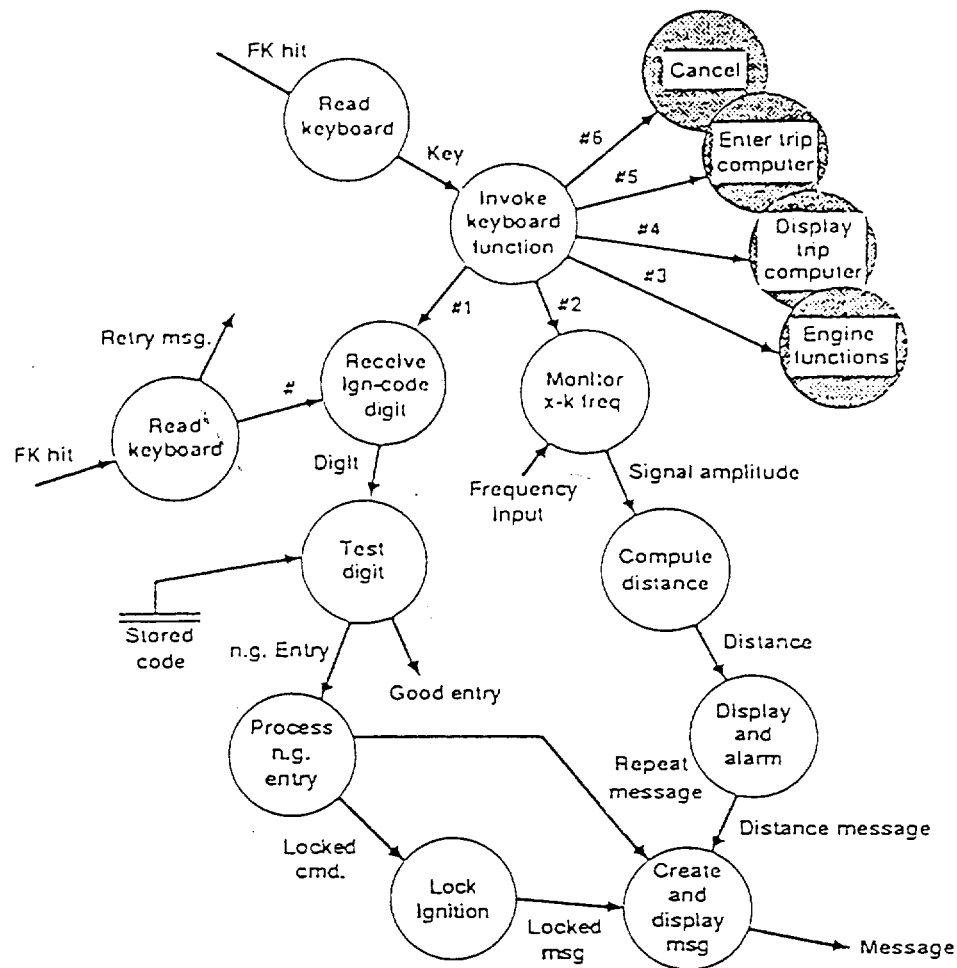


FIGURE 7J4  
(c) Dashboard function key processing—level 03 DFD.

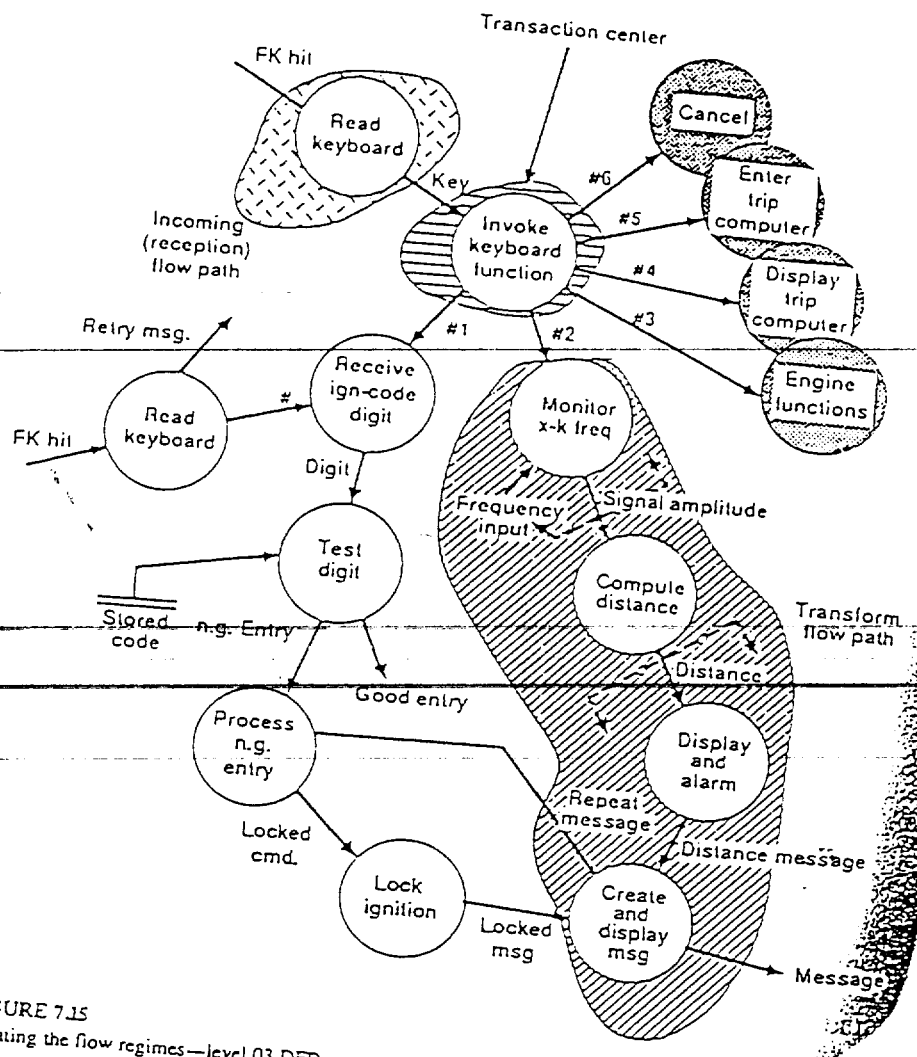


FIGURE 7.15  
Isolating the flow regimes—level 03 DFD.

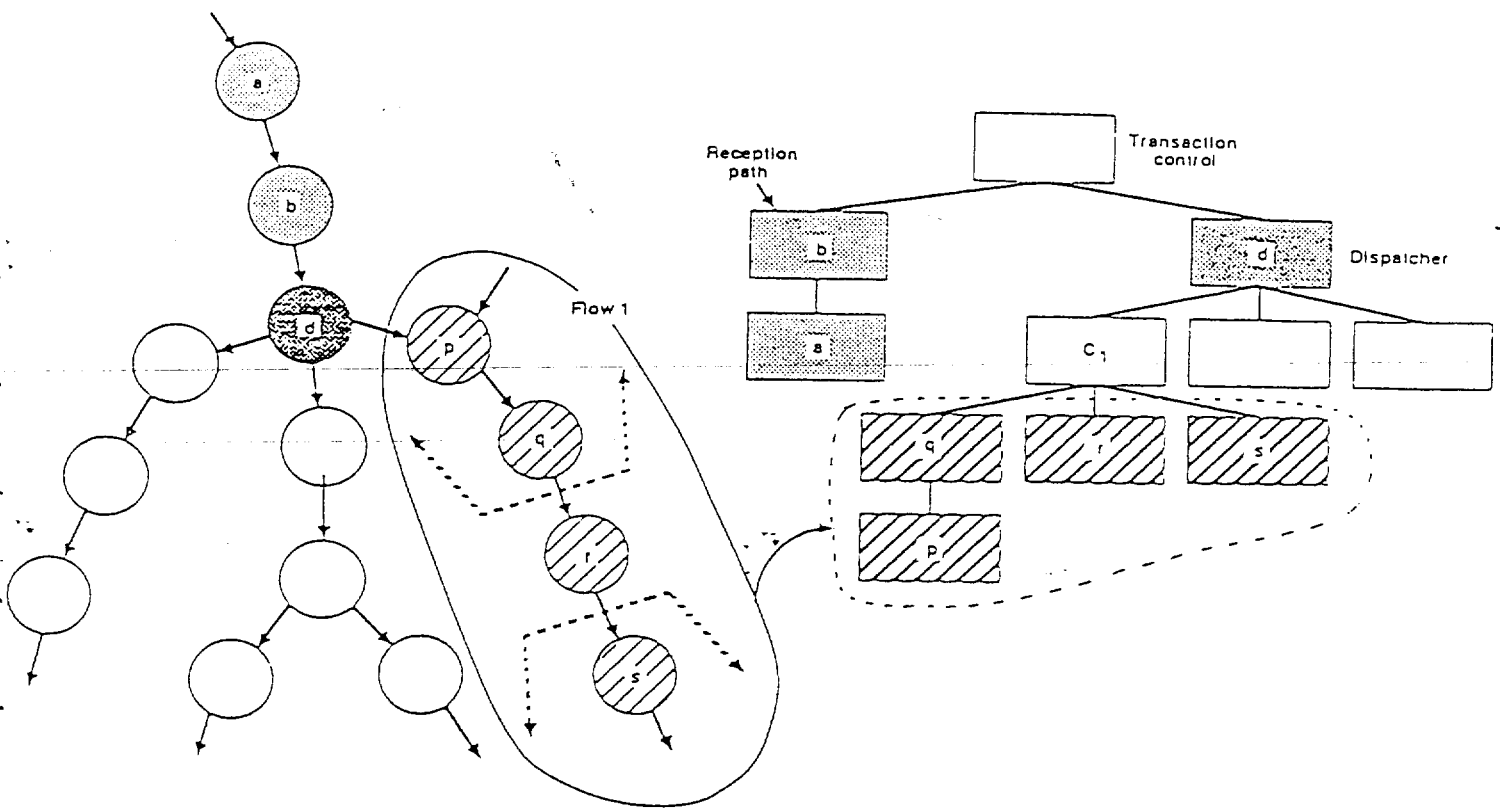


FIGURE 7.16  
Transaction mapping.

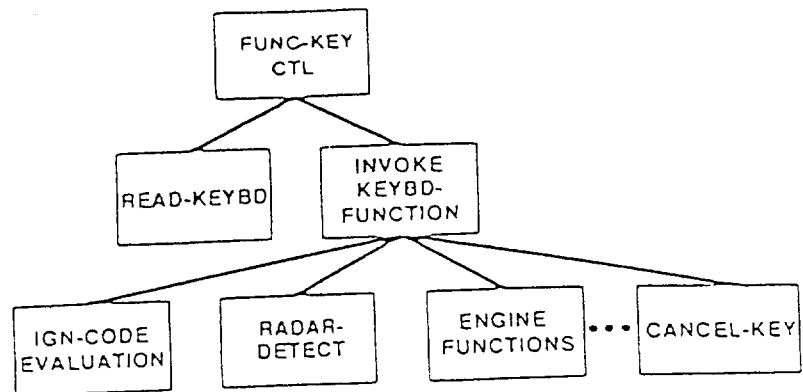


FIGURE 7.17  
Map the transaction structure.

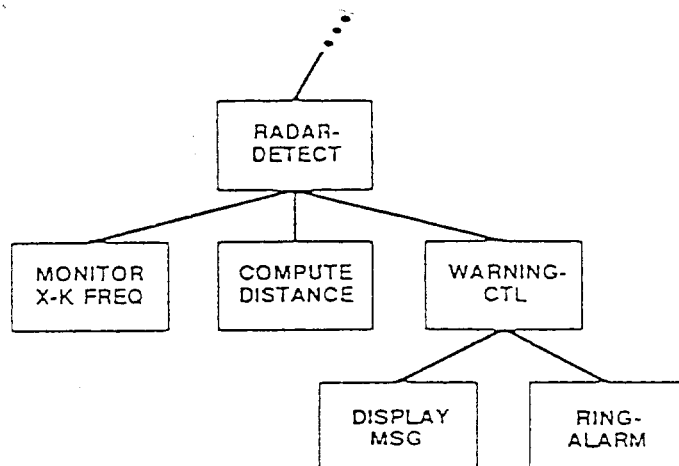


FIGURE 7.18  
Develop action path structure.

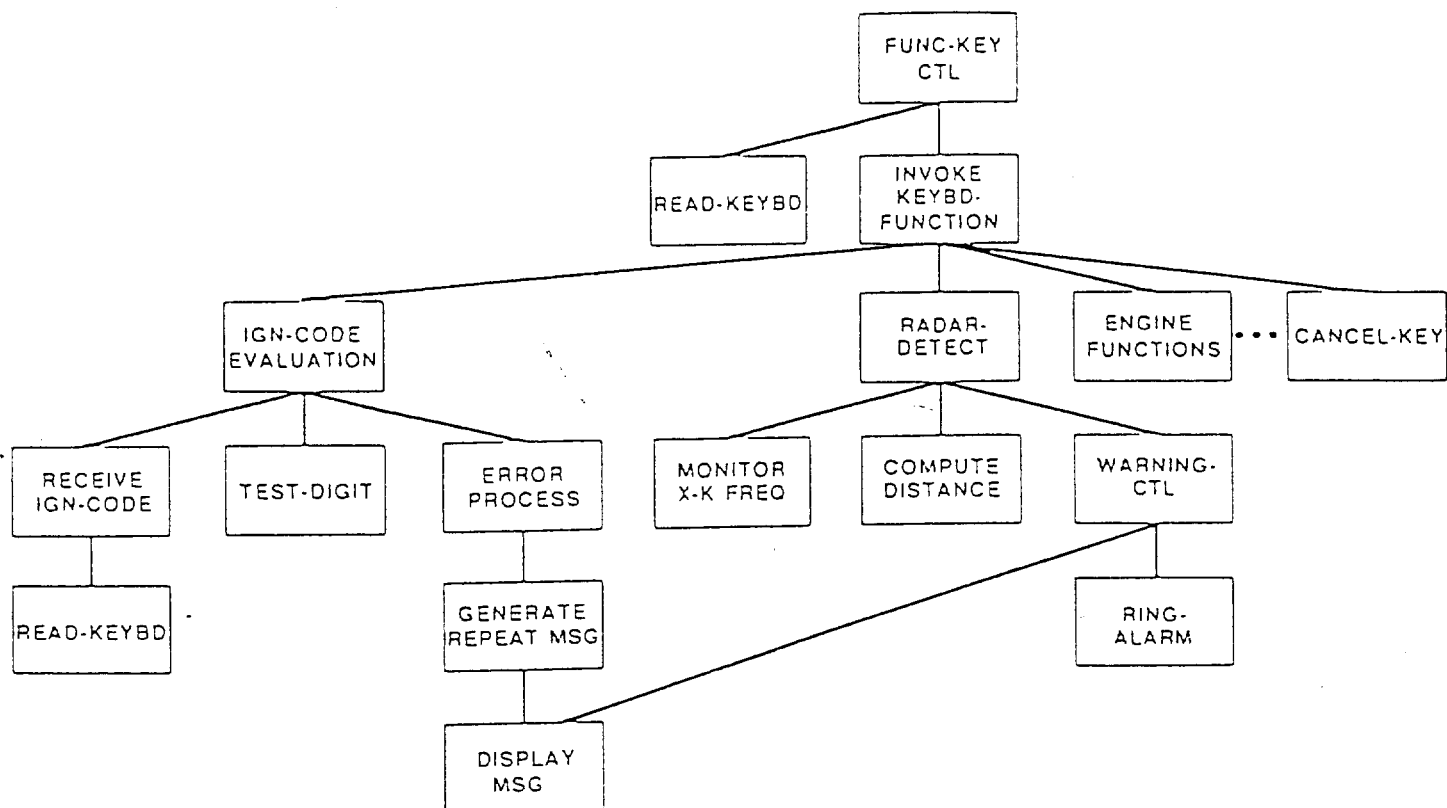


FIGURE 7.19  
Dashboard function key processing — program structure.



## OBJECT-ORIENTED DESIGN

- o ORIGINS-INFORMATION HIDING
- o DESIGN CONCEPTS
  - OBJECTS, OPERATIONS, MESSAGES
  - CLASSES, INSTANCES, INHERITANCE
  - OBJECT DESCRIPTIONS
- o DESIGN METHODS
- o FORMALIZING
  - OBJECTS/ATTRIBUTES
  - OPERATIONS, OBJECTS
  - PROGRAM COMPONENTS
  - INTERFACES
  - GRAPHICAL REPRESENTATIONS
  - IMPLEMENTATION



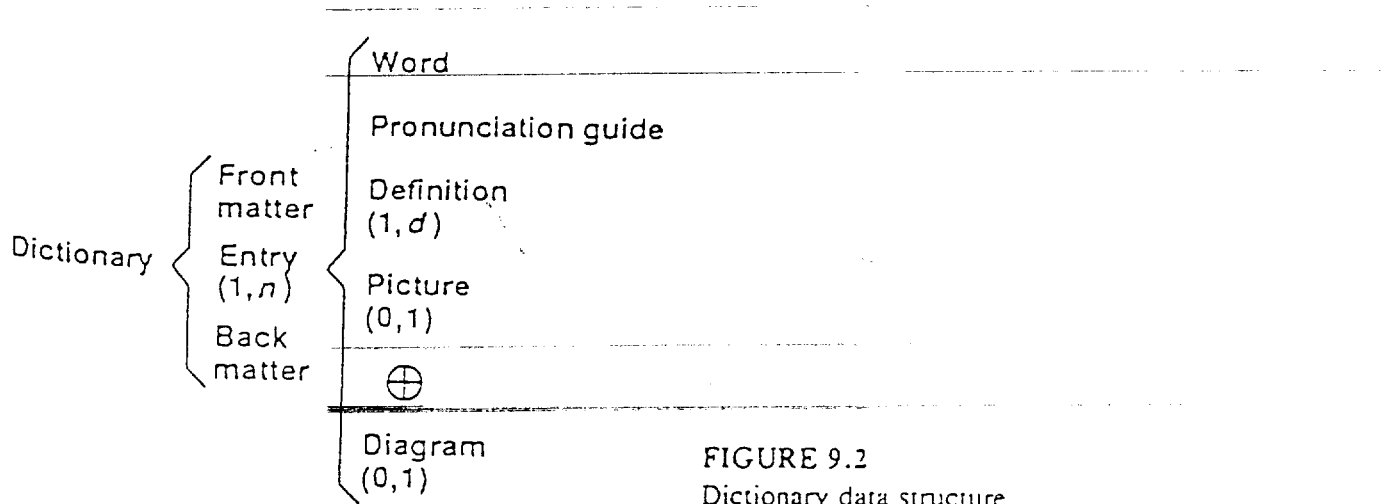
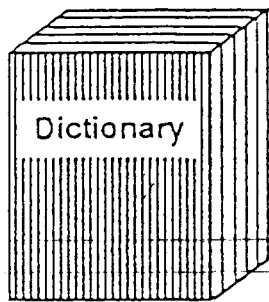


FIGURE 9.2  
Dictionary data structure.



Real-world object

Message:

Find - word (word)

Print - definition (word)

Interface

Dictionary  
data structure

Operations

Add - word

Delete - word

Find - word

Print - definition

⋮

Software realization of object

FIGURE 9.1  
Objects.

1. Define the problem.
2. Develop an informal strategy for the software realization of the real-world problem domain.
3. Formalize the strategy using the following substeps:
  - a. Identify objects and their attributes
  - b. Identify operations that may be applied to objects
  - c. Establish interfaces by showing the relationship between objects and operations.
  - d. Decide on detailed design issues that will provide an implementation description for objects.
4. Reapply steps 2, 3, and 4 recursively until a complete design is created.

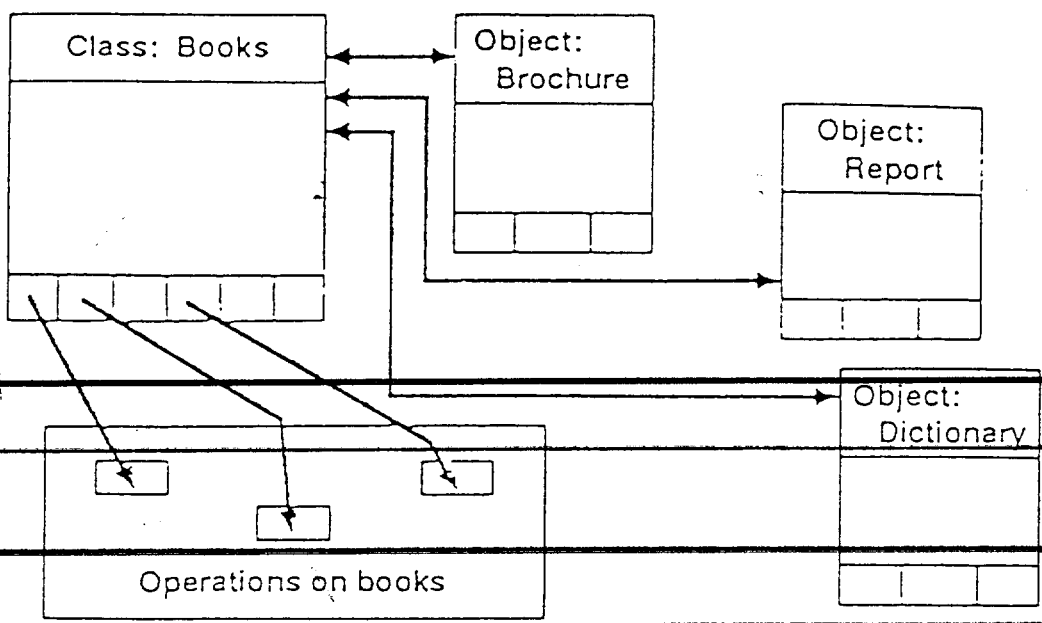


FIGURE 9.3  
Objects, classes, and inheritance.

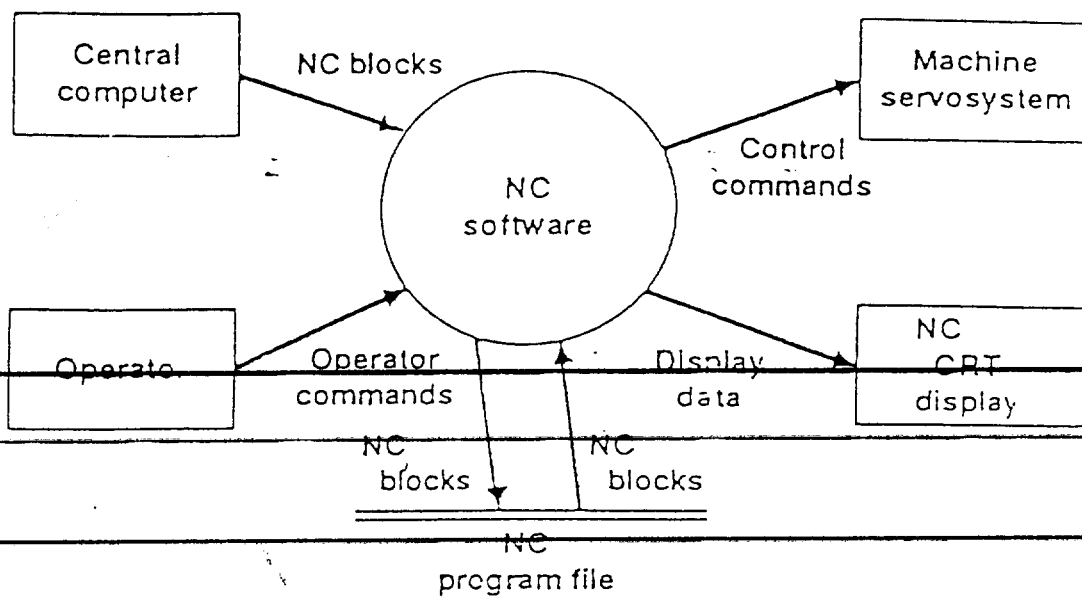


FIGURE 9.4

Analysis representation—an NC system.

NC.blocks = n.code + (g.code) + x + y + z + (s.code) + eob

operator commands = [insert.block|display.pgm|exec.pgm]

NC.program.file = program.number + {NC.blocks}\*

control commands = position.cmds + special.cmds

display.data = listing of NC program file + (echo of inserted NC block from operator)

n.code = N + three-digit number

g.code = G + two-digit number between 01 and 99

x = X + four-digit coordinate for x axis

y = Y + four-digit coordinate for y axis

z = Z + four-digit coordinate for z axis

s.code = S + two-digit number between 00 and 32

eob = ascii control character for end of block

... and so on.

A numerical control (NC) for machine tools is to be developed. The NC is "programmed" with machine instructions and produces control commands for the machine tool's servosystem. In addition the NC contains a CRT display and keyboard for operator interaction.

A central computer transmits NC blocks to a numerical control that contains software which reads each NC block and stores it in an NC program file. NC blocks are read from the NC program file and decomposed into control words for position and special functions. Control words are processed and encoded into position control commands and special control commands that are sent to the machine servosystem. Operator commands are input to the NC software via a keyboard interface. Operator commands enable the operator to insert an NC block into an existing NC program file, to display an NC program file on a CRT, and to execute an NC program.

A central computer transmits NC blocks to a numerical control that contains software which reads each NC block and stores it in an NC program file. NC blocks are read from the NC program file and decomposed into control words for position and special functions. Control words are processed and encoded into position control commands and special control commands that are sent to the machine servosystem. Operator commands are input to the NC software via a keyboard interface. Operator commands enable the operator to insert an NC block into an existing NC program file, to display an NC program file on a CRT, and to execute an NC program.