Hierarchy-Input-Output-Processinf Chart

Home

5th form - Course Outline and SBA

---

form-5O

form-5O---online-tests

Upper 6th-form---computer-science

Upper 6th IA Guideline

Lower 6th - Computer Science

Lower 6th IA Guidelines

Lower 6th - Online Tests

Comment/Questions Forum

Teacher Evaluation

Wolmer's High School for Girls

Computing Education Department

Upper sixth Teacher: Mrs. McCallum-Rodney

HIPO Technique

Purpose

The HIPO (Hierarchy plus Input-Process-Output) technique is a tool for planning and/or documenting a computer program. A HIPO model consists of a hierarchy chart that graphically represents the program's control structure and a set of IPO (Input-Process-Output) charts that describe the inputs to, the outputs from, and the functions (or processes) performed by each module on the hierarchy chart.

Strengths, weaknesses, and limitations

Using the HIPO technique, designers can evaluate and refine a program's design, and correct flaws prior to implementation. Given the graphic nature of HIPO, users and managers can easily follow a program's structure. The hierarchy chart serves as a useful planning and visualization document for managing the program development process. The IPO charts define for the programmer each module's inputs, outputs, and algorithms.

In theory, HIPO provides valuable long-term documentation. However, the "text plus flowchart" nature of the IPO charts makes them difficult to maintain, so the documentation often does not represent the current state of the program.

By its very nature, the HIPO technique is best used to plan and/or document a hierarchically structured program.

Inputs and related ideas

During the analysis stage of the system life cycle (Part IV), the analyst creates logical models using such tools as data flow diagrams and entity-relationship diagrams. Given those models as a base, the analyst then identifies several alternative solutions during the high-level system design stage (Part V) using such tools as system flow diagrams to document them. The alternatives usually identify, at a black box level, one or more programs. HIPO is a tool for planning and/or documenting the programs.

The HIPO technique is often used to plan or document a structured program. A variety of tools, including pseudocode and structured English , can be used to describe processes on an IPO chart. System flowcharting symbols are sometimes used to identify physical input, output, and storage devices on an IPO chart.

Concepts

A completed HIPO package has two parts. A hierarchy chart is used to represent the top-down structure of the program. For each module depicted on the hierarchy chart, an IPO (Input-Process-Output) chart is used to describe the inputs to, the outputs from, and the process performed by the module.

The hierarchy chart

Table 1 summarizes the primary tasks to be performed by an interactive inventory program. Figure 1 shows one possible hierarchy chart (or visual table of contents) for that program. Each box represents one module that can call its subordinates and return control to its higher-level parent.

Table 1 A Set of Tasks to Be Performed by an Interactive Inventory Program

---

1.0 Manage inventory

2.0 Update stock

2.1 Process sale
2.2 Process return
2.3 Process shipment

3.0 Generate report

3.1 Respond to query
3.2 Display status report

4.0 Maintain inventory data

4.1 Modify record
4.2 Add record
4.3 Delete record

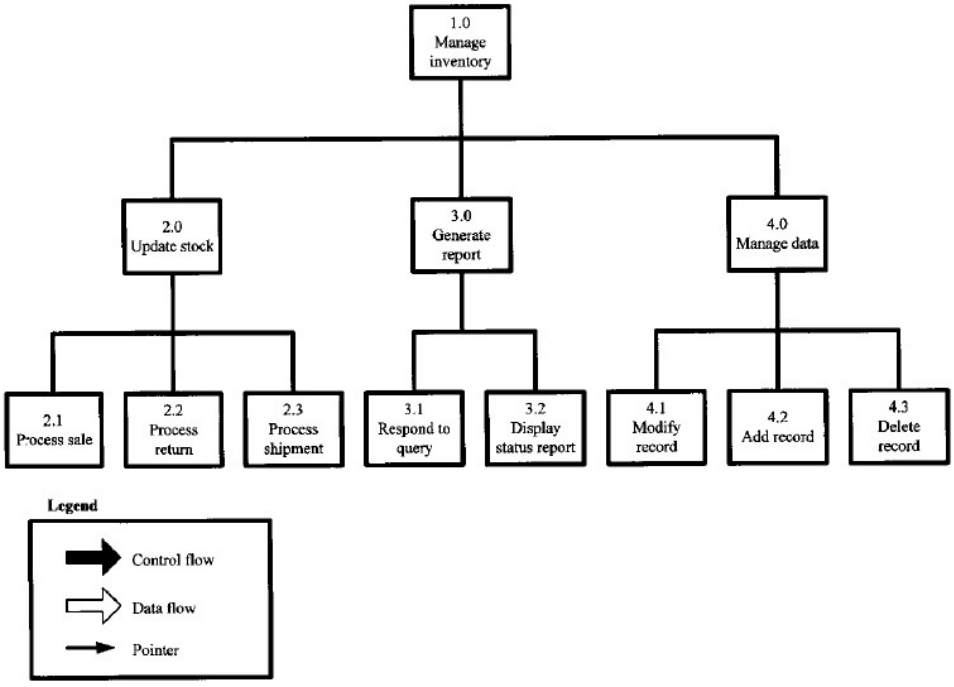

Figure 1 A hierarchy chart for an interactive inventory control program.

At the top of Figure 1 is the main control module, *Manage inventory* (module 1.0). It accepts a transaction, determines the transaction type, and calls one of its three subordinates (modules 2.0, 3.0, and 4.0).

Lower-level modules are identified relative to their parent modules; for example, modules 2.1, 2.2, and 2.3 are subordinates of module 2.0, modules 2.1.1, 2.1.2, and 2.1.3 are subordinates of 2.1, and so on. The module names consist of an active verb followed by a subject that suggests the module's function.

The objective of the module identifiers is to uniquely identify each module and to indicate its place in the hierarchy. Some designers use Roman numerals (level I, level II) or letters (level A, level B) to designate levels. Others prefer a hierarchical numbering scheme; e.g., 1.0 for the first level; 1.1, 1.2, 1.3 for the second level; and so on. The key is consistency.

The box at the lower-left of Figure 1 is a legend that explains how the arrows on the hierarchy chart and the IPO charts are to be interpreted. By default, a wide clear arrow represents a data flow, a wide black arrow represents a control flow, and a narrow arrow indicates a pointer.

The IPO charts

An IPO chart is prepared to document each of the modules on the hierarchy chart.

64.4.2.1 Overview diagrams

An overview diagram is a high-level IPO chart that summarizes the inputs to, processes or tasks performed by, and outputs from a module. For example, Figure 2 shows an overview diagram for process 2.0, *Update stock*. Where appropriate, system

flowcharting symbols are used to identify the physical devices that generate the inputs and accept the outputs. The processes are typically described in brief paragraph or sentence form. Arrows show the primary input and output data flows.

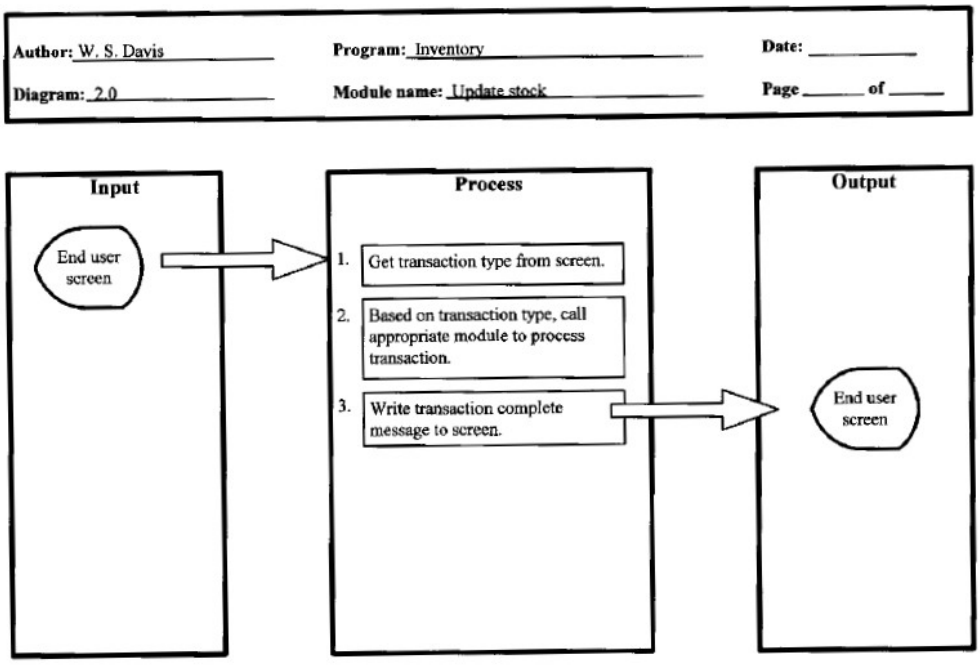

Figure 2 An overview diagram for process 2.0.

Overview diagrams are primarily planning tools. They often do not appear in the completed documentation package.

Detail diagrams

A detail diagram is a low-level IPO chart that shows how specific input and output data elements or data structures are linked to specific processes. In effect, the designer integrates a system flowchart into the overview diagram to show the flow of data and control through the module.

Figure 3 shows a detail diagram for module 2.0, *Update stock*. The process steps are written in pseudocode. Note that the first step writes a menu to the user screen and input data (the transaction type) flows from that screen to step 2. Step 3 is a case structure. Step 4 writes a *transaction complete* message to the user screen.

The solid black arrows at the top and bottom of the process box show that control flows from module 1.0 and, upon completion, returns to module 1.0. Within the case structure (step 3) are other solid black arrows.

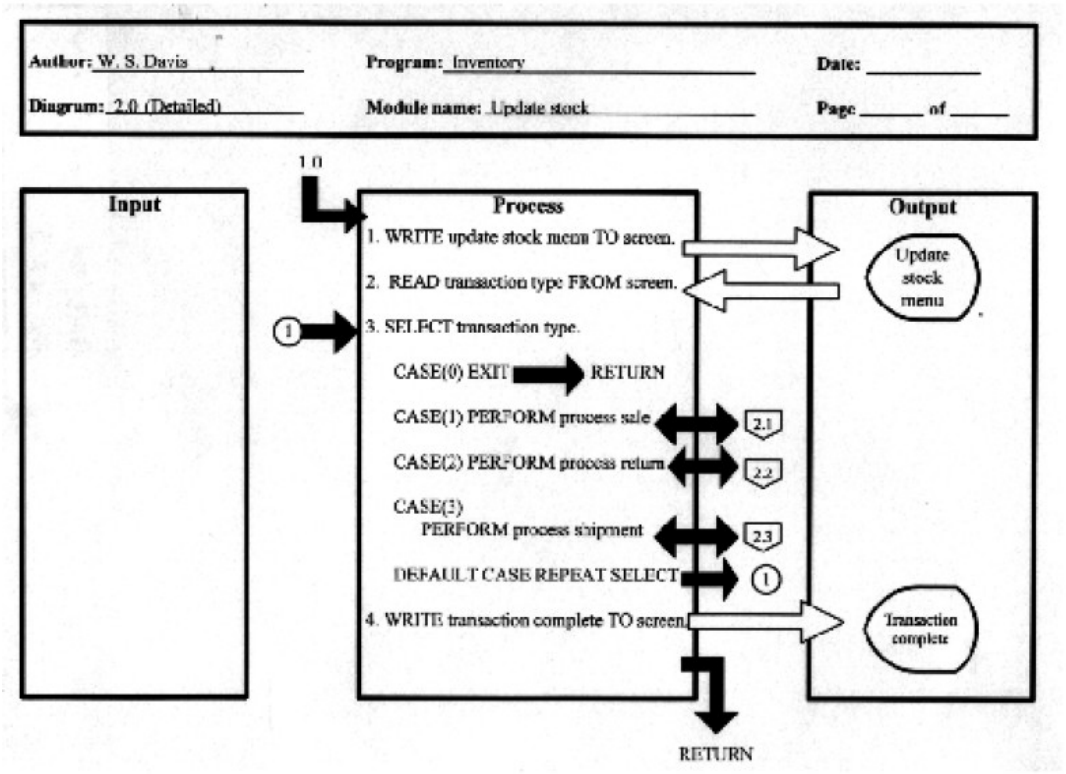

Figure 3 A detail diagram for process 2.0.

Following case 0 is a return (to module 1.0). The two-headed black arrows following cases 1, 2, and 3 represent subroutine calls; the off-page connector symbols (the little home plates) identify each subroutine's module number. Note that each subroutine is documented in a separate IPO chart. Following the default case, the arrow points to an on-page connector symbol numbered 1. Note the matching on-page connector symbol pointing to the select structure. On-page connectors are also used to avoid crossing arrows on data flows.

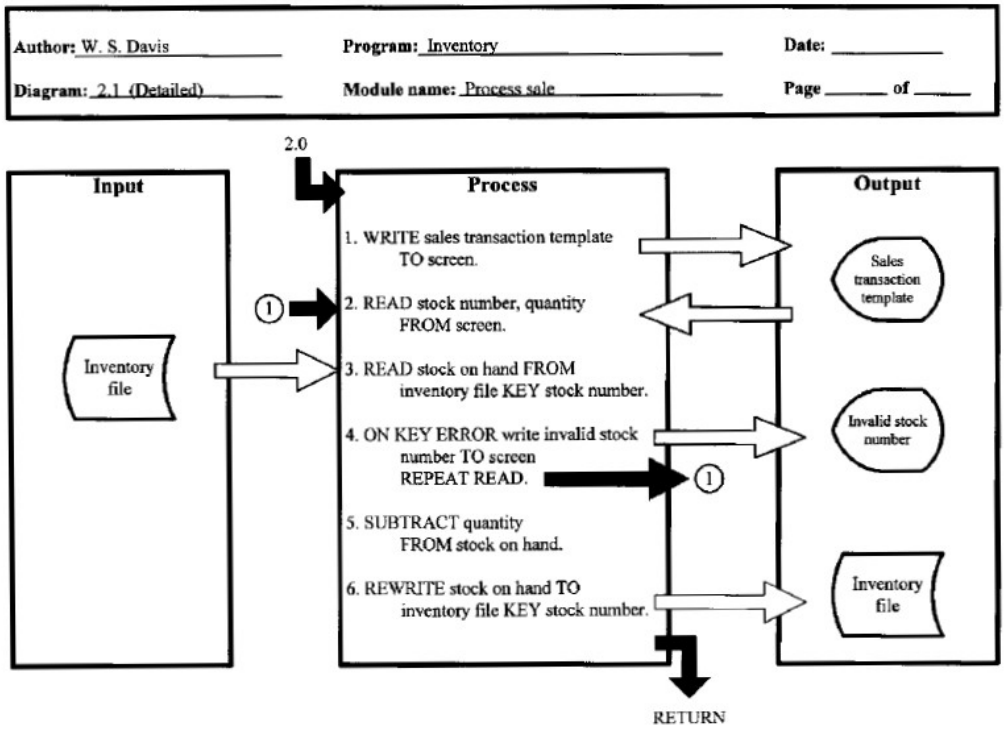

Figure 4 A detail diagram for process 2.1.

Often, detailed notes and explanations are written on an extended description that is attached to each detail diagram. The notes might specify access methods, data types, and so on.

Figure 4 shows a detail diagram for process 2.1. The module writes a template to the user screen, reads a stock number and a quantity from the screen, uses the stock number as a key to access an inventory file, and updates the stock on hand. Note that

the logic repeats the data entry process if the stock number does not match an inventory record. A real IPO chart is likely to show the error response process in greater detail.

Simplified IPO charts

Some designers simplify the IPO charts by eliminating the arrows and system flowchart symbols and showing only the text. Often, the input and out put blocks are moved above the process block (Figure 5), yielding a form that fits better on a standard 8.5 × 11 (portrait orientation) sheet of paper. Some programmers insert modified IPO charts similar to Figure 5 directly into their source code as comments. Because the documentation is closely linked to the code, it is often more reliable than stand-alone HIPO documentation, and more likely to be maintained.

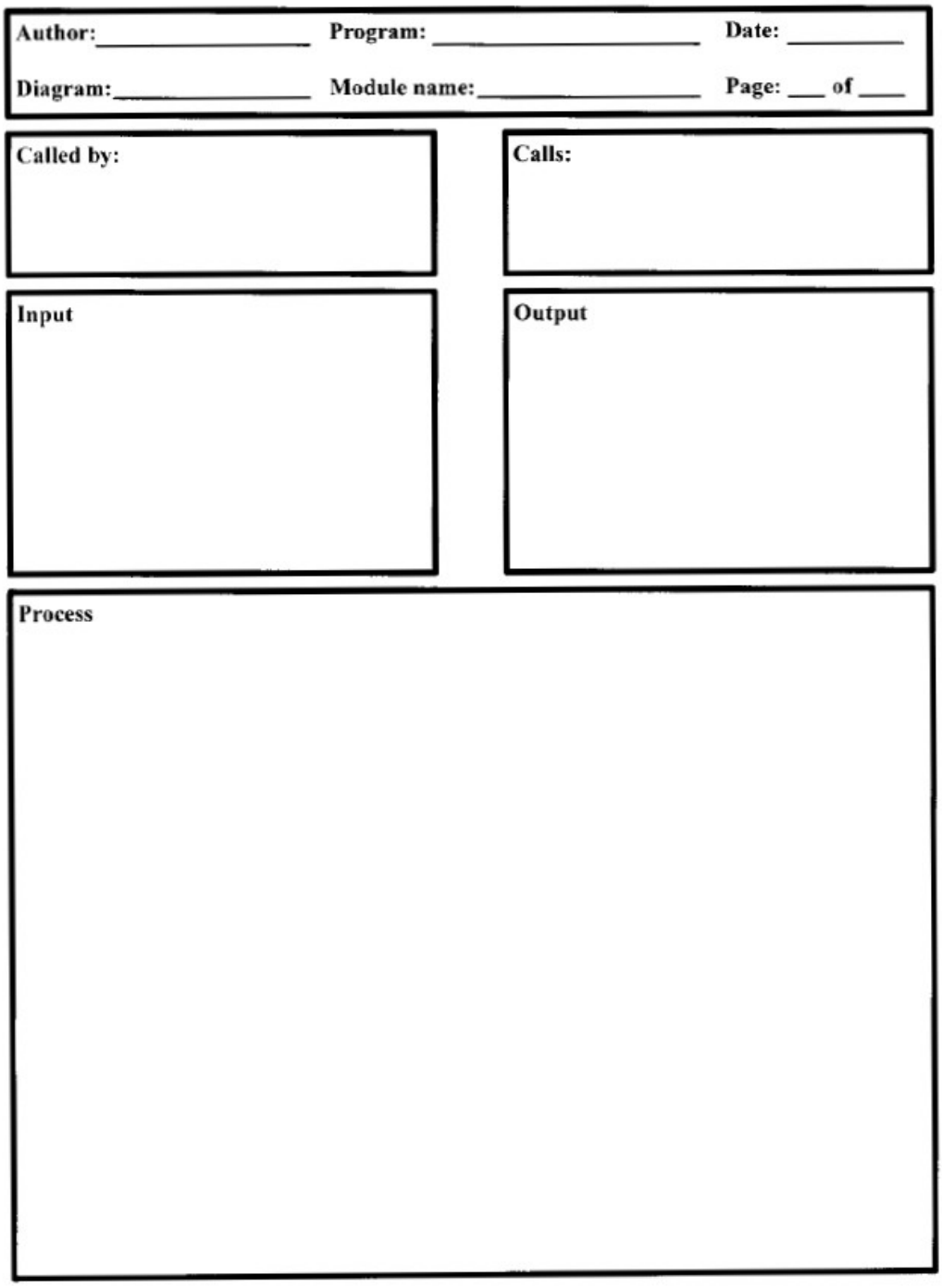

Figure 5 A simplified IPO diagram

Key terms

Detail diagram —
: A low-level IPO chart that shows how specific input and output data elements or data structures are linked to specific processes.

Hierarchy chart —
: A diagram that graphically represents a program's control structure.

HIPO (Hierarchy plus Input-Process-Output) —
: A tool for planning and/or documenting a computer program that utilizes a hierarchy chart to graphically represent the program's control structure and a set of IPO (Input-Process-Output) charts to describe the inputs to, the outputs from, and the functions performed by each module on the hierarchy chart.

IPO (Input-Process-Output) chart —
: A chart that describes or documents the inputs to, the outputs from, and the functions (or processes) performed by a program module.

Overview diagram —
: A high-level IPO chart that summarizes the inputs to, processes or tasks performed by, and outputs from a module.

Visual Table of Contents (VTOC) —
: A more formal name for a hierarchy chart.

TEST on HIPO (Tuesday Oct 9, 2012)

A program is required by a company to read an employee's number, pay rate and the number of hours worked in a week. The program is then to validate the pay rate and the hours worked fields and, if valid, compute the employee's weekly pay and print it along with the input data.

VALIDATION: According to the company's rules, the maximum hours an employee can work per week is 60 hours, and the maximum hourly rate is $25.00 per hour. If the hours worked field or the hourly rate field is out of range, the input data and an appropriate message is to be printed and the employee's weekly pay is not calculated.

WEEKLY PAY CALCULATIONS: Weekly pay is calculated as hours worked times pay rate. If more than 35 hours are worked, payment for overtime hours worked is calculates at time-and-a-half.

Required:

1. Construct a hierarchy chart
2. Construct an IPO chart for the module relating to validation

HOME-WORK ASSIGNMENT - Due Friday October 19, 2012

The Acme Spare Parts Company wants to produce a product orders report from its product orders file. Each recordon the file contains the product number of the item ordered, the product description, the number of units ordered, the retail price per unit, the freight charges per unit, and the packaging cost per unit.

Your algorithm is to read the product orders file, calculate the total amount due for each product ordered and print these details on the product orders report.

The amount due for each product is calculated as the product of the number of units ordered and the retail price of the unit. A discount of 10% is allowed on the amount due for all orders over $100.00. The freight charges and packaging costs per unit must be added to this resulting value to determine the total amount due.

Develop the:

- Hierarchy Chart
- Structure Chart
- One module's IPO chart