DFD Rules and Guidelines

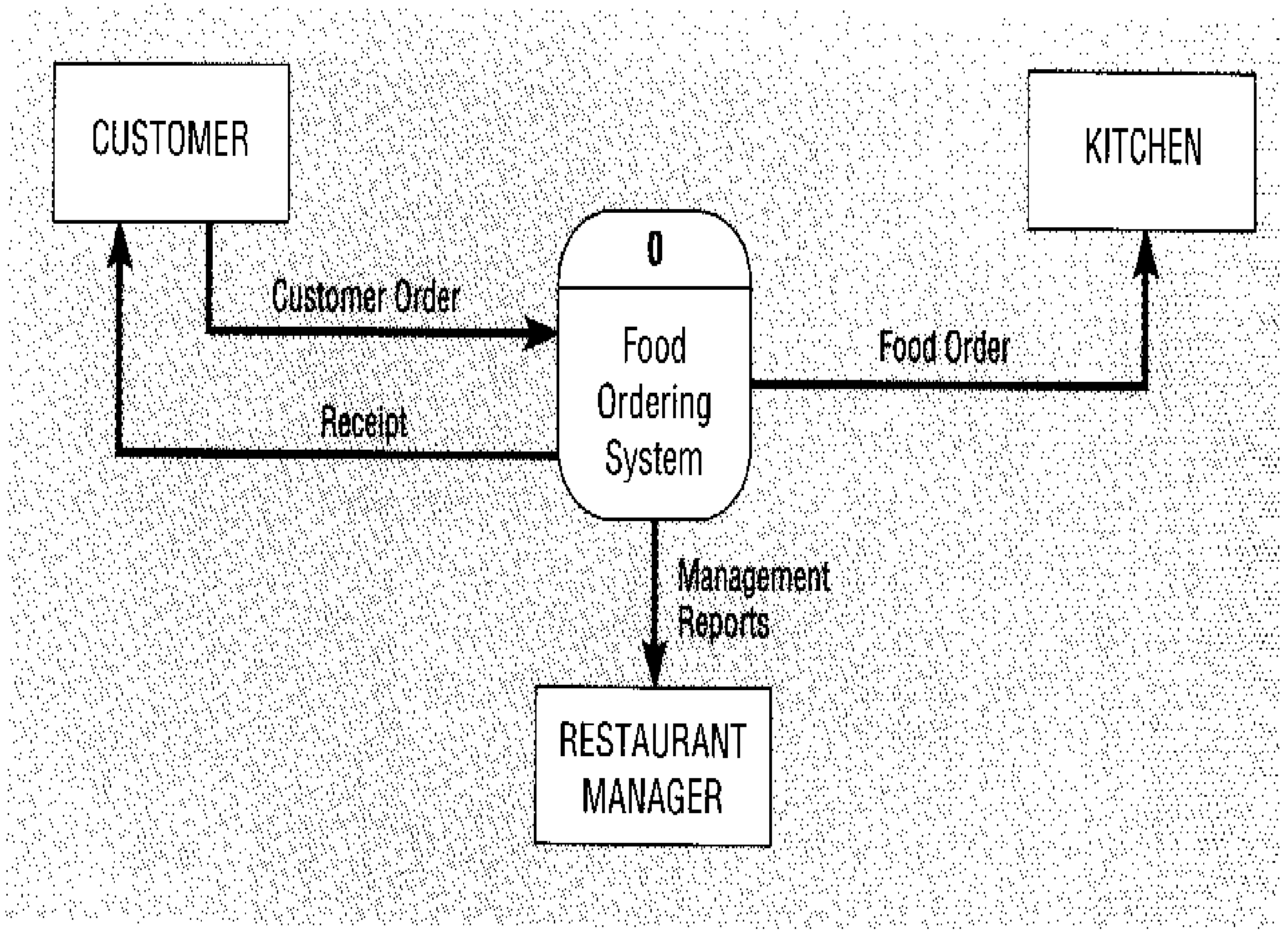
Yong Choi

BPA

CSUB

DFD example - Hoosier Burger’s food ordering system I

Figure 8-4
Context diagram of Hoosier Burger’s food ordering system



* One process (level 0 - the whole system)

* No data store

DFD example - Hoosier Burger's food ordering system II

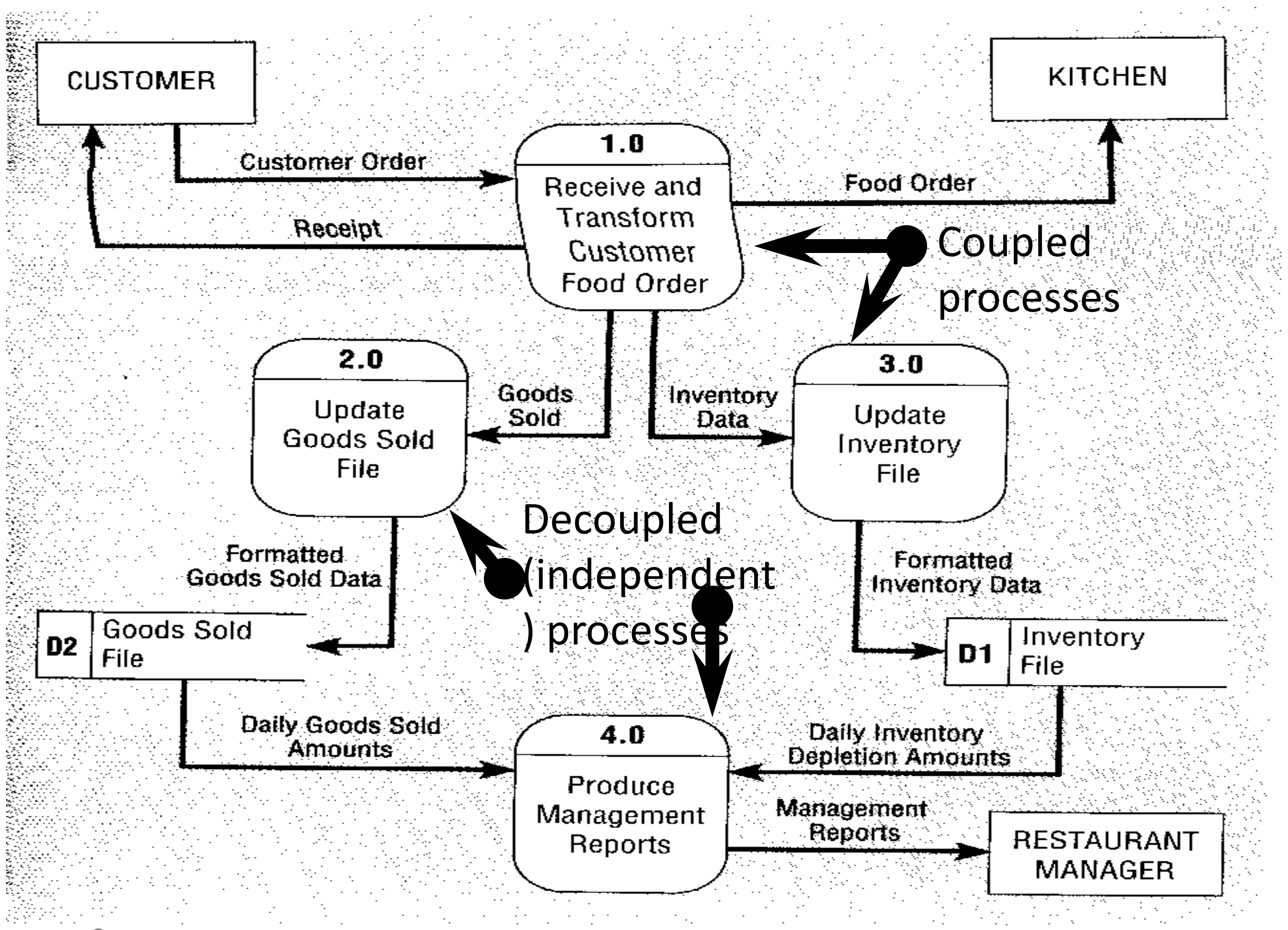


Figure 8-5
Level-0 DFD of Hoosier Burger's food ordering system

Represent the major processes & data stores of the level-0 whole-system process of the context diagram

DFD guidelines & rules I

- Starting with context diagram, DFDs are refined and decomposed from level to level, with more detail at each lower level
- Process's input & output are different
- Unique descriptive names to all objects
  - But the same objects (and names) may appear at various levels
  - To minimize clutter a data store (or even dataflow) may be repeated even on the same diagram
  - Process names usually start with a verb

DFD guidelines & rules II

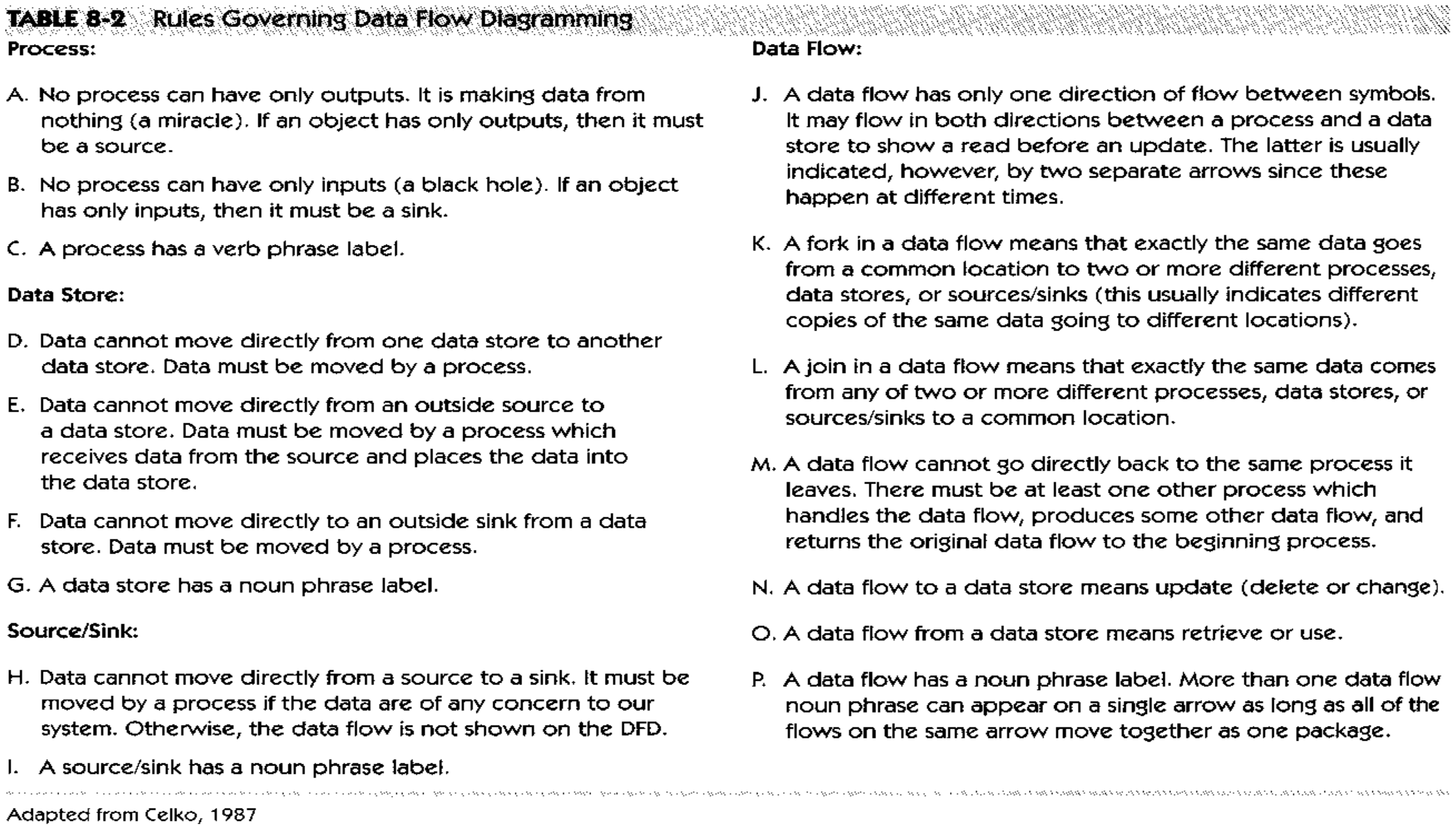


TABLE 8-2 Rules Governing Data Flow Diagramming

Process:

A. No process can have only outputs. It is making data from nothing (a miracle). If an object has only outputs, then it must be a source.

B. No process can have only inputs (a black hole). If an object has only inputs, then it must be a sink.

C. A process has a verb phrase label.

Data Store:

D. Data cannot move directly from one data store to another data store. Data must be moved by a process.

E. Data cannot move directly from an outside source to a data store. Data must be moved by a process which receives data from the source and places the data into the data store.

F. Data cannot move directly to an outside sink from a data store. Data must be moved by a process.

G. A data store has a noun phrase label.

Source/Sink:

H. Data cannot move directly from a source to a sink. It must be moved by a process if the data are of any concern to our system. Otherwise, the data flow is not shown on the DFD.

I. A source/sink has a noun phrase label.

Data Flow:

J. A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however, by two separate arrows since these happen at different times.

K. A fork in a data flow means that exactly the same data goes from a common location to two or more different processes, data stores, or sources/sinks (this usually indicates different copies of the same data going to different locations).

L. A join in a data flow means that exactly the same data comes from any of two or more different processes, data stores, or sources/sinks to a common location.

M. A data flow cannot go directly back to the same process it leaves. There must be at least one other process which handles the data flow, produces some other data flow, and returns the original data flow to the beginning process.

N. A data flow to a data store means update (delete or change).

O. A data flow from a data store means retrieve or use.

P. A data flow has a noun phrase label. More than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Adapted from Celko, 1987

DFD guidelines & rules III

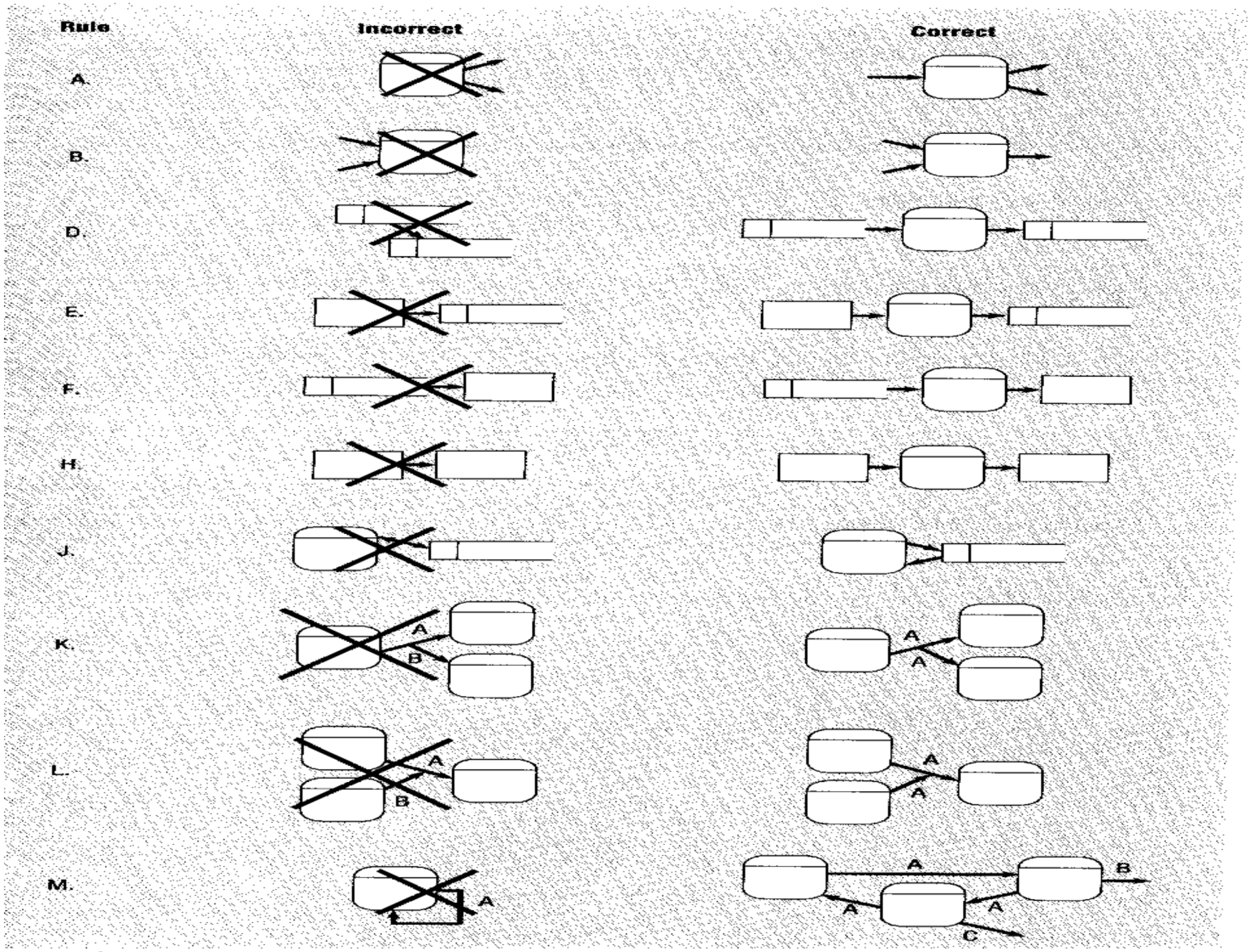


Figure 8-6
Incorrect and correct ways to draw data flow diagrams

DFD (functional) decomposition

- An iterative hierarchical process of refining the details of a system, creating a set of charts at lower and lower levels, in which a process at a certain level is explained on the next level in greater detail
- Primitive DFD: the lowest level DFD, where no process can (or it is useful to) be broken any further

DFD example - Hoosier Burger's food ordering system III

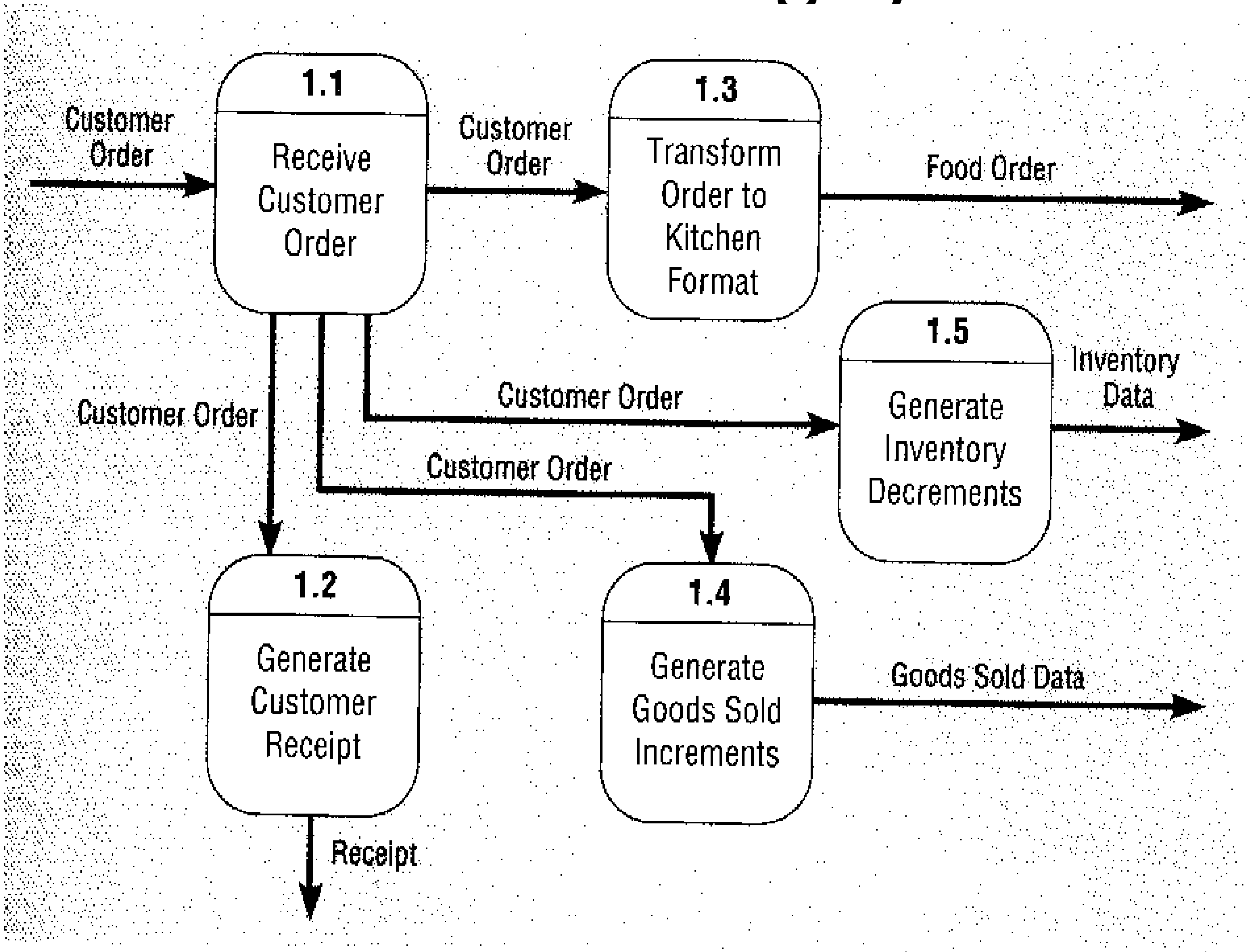


Figure 8-7
Level-1 diagram showing the decomposition of Process 1.0 from the level-0 diagram for Hoosier Burger's food ordering system

* Hierarchical notation

* No sources or sinks

DFD example - Hoosier Burger's food ordering system IV

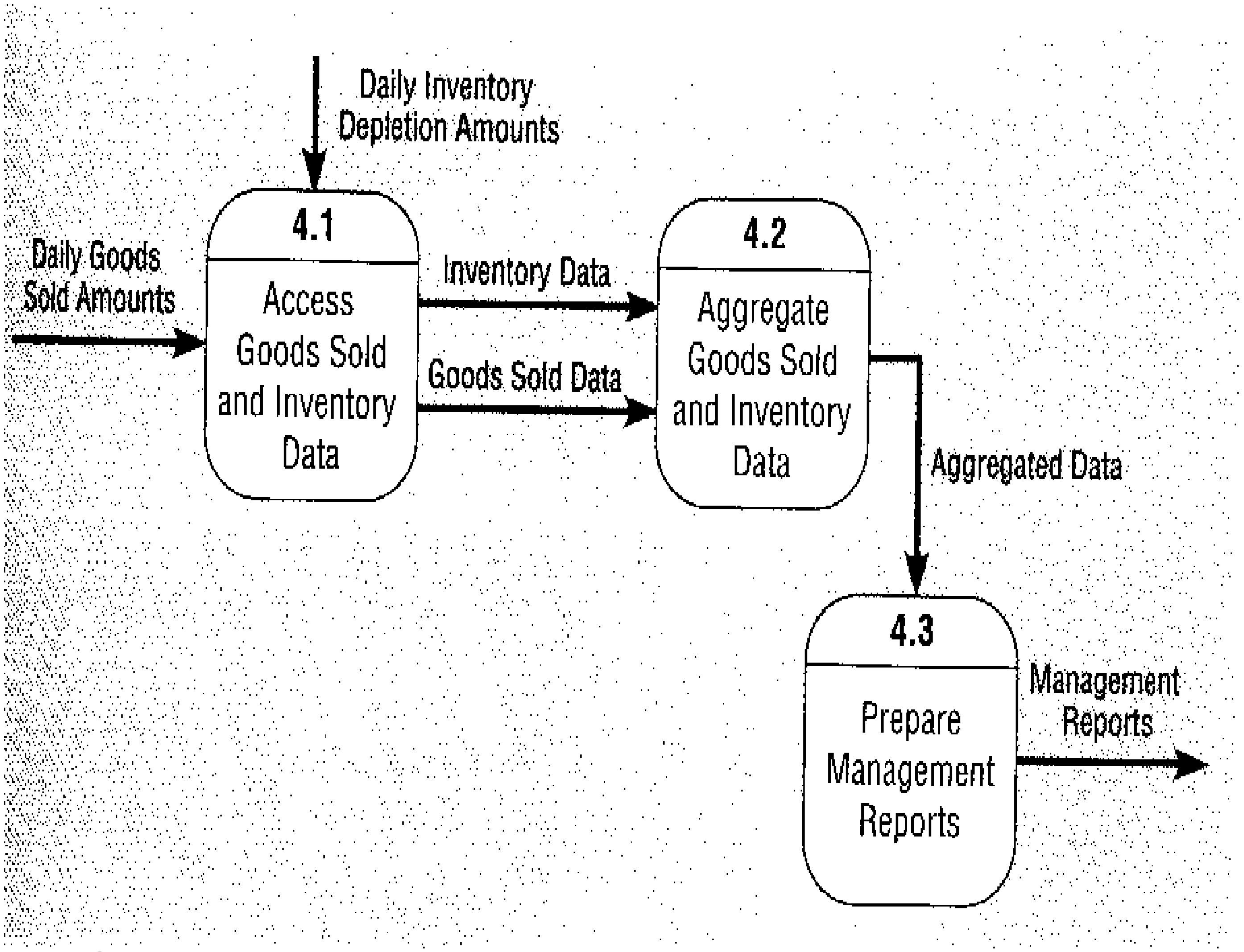
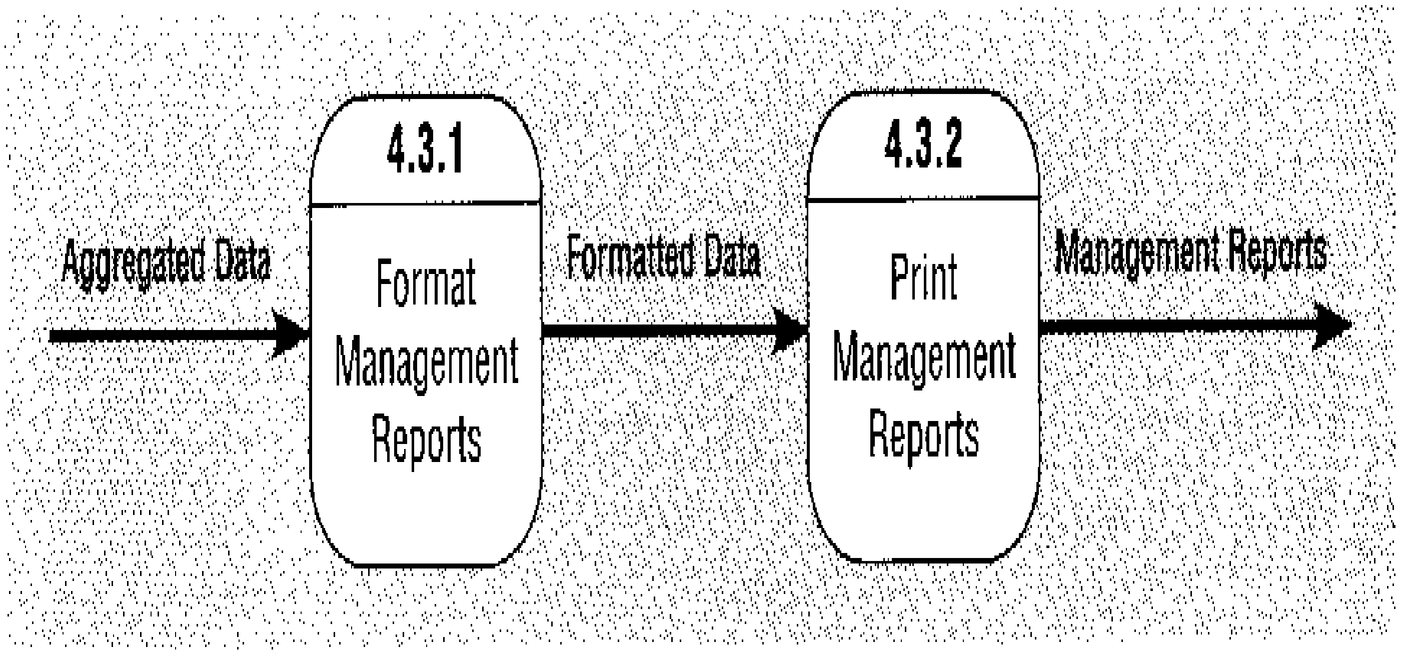

Daily Inventory Depletion Amounts
Daily Goods Sold Amounts
4.1
Access Goods Sold and Inventory Data
Inventory Data
Goods Sold Data
4.2
Aggregate Goods Sold and Inventory Data
Aggregated Data
4.3
Prepare Management Reports
Management Reports

Figure 8-8
Level-1 diagram showing the decomposition of Process 4.0 from the level-0 diagram for Hoosier Burger's food ordering system

* No need to decompose processes 2.0 & 3.0 (singular logical action)

DFD example - Hoosier Burger's food ordering system V

Figure 8-9
Level-2 diagram showing the decomposition of Process 4.3 from the level-1 diagram for Process 4.0 for Hoosier Burger's food ordering system



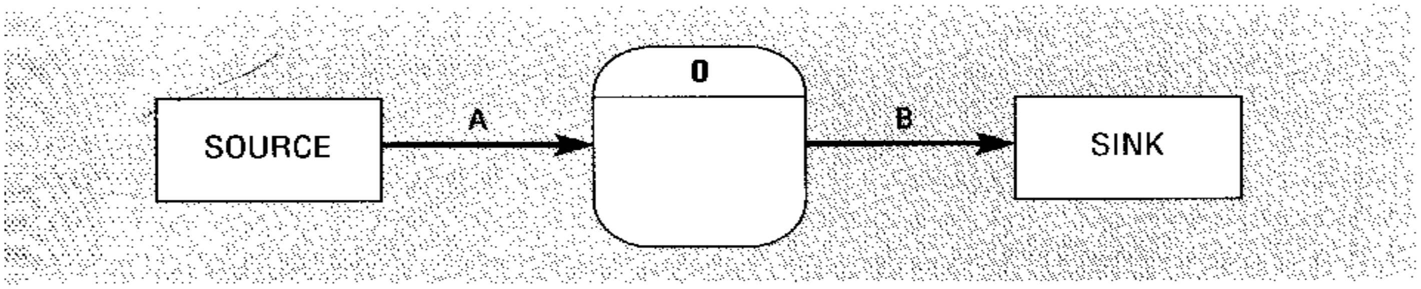
Similar decomposition of other level-1 processes can be done, as need dictates

DFD balancing I

- Conservation of inputs and outputs when a process is decomposed
  - A decomposed process must have the same inputs and outputs as the non-decomposed process from which it was derived

DFD balancing II

- An unbalanced example:



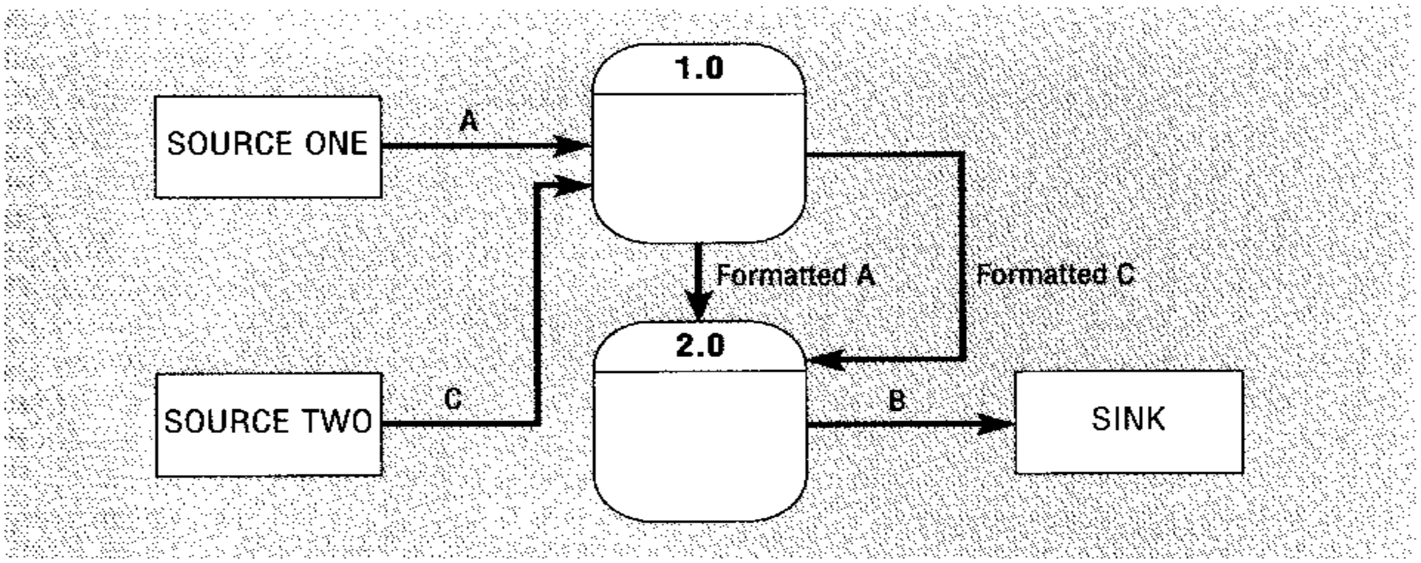


Figure 8-10
An unbalanced set of data flow diagrams

(a) Context diagram

(b) Level-0 diagram

DFD balancing III

- But a composite dataflow may be split ...
  - e.g., “payments and coupons” leading into process 1.0 may be split into:
    - “payments” leading into 1.1
    - “coupons” leading into 1.2
  - But all data must be conserved between levels

DFD guidelines & rules IV

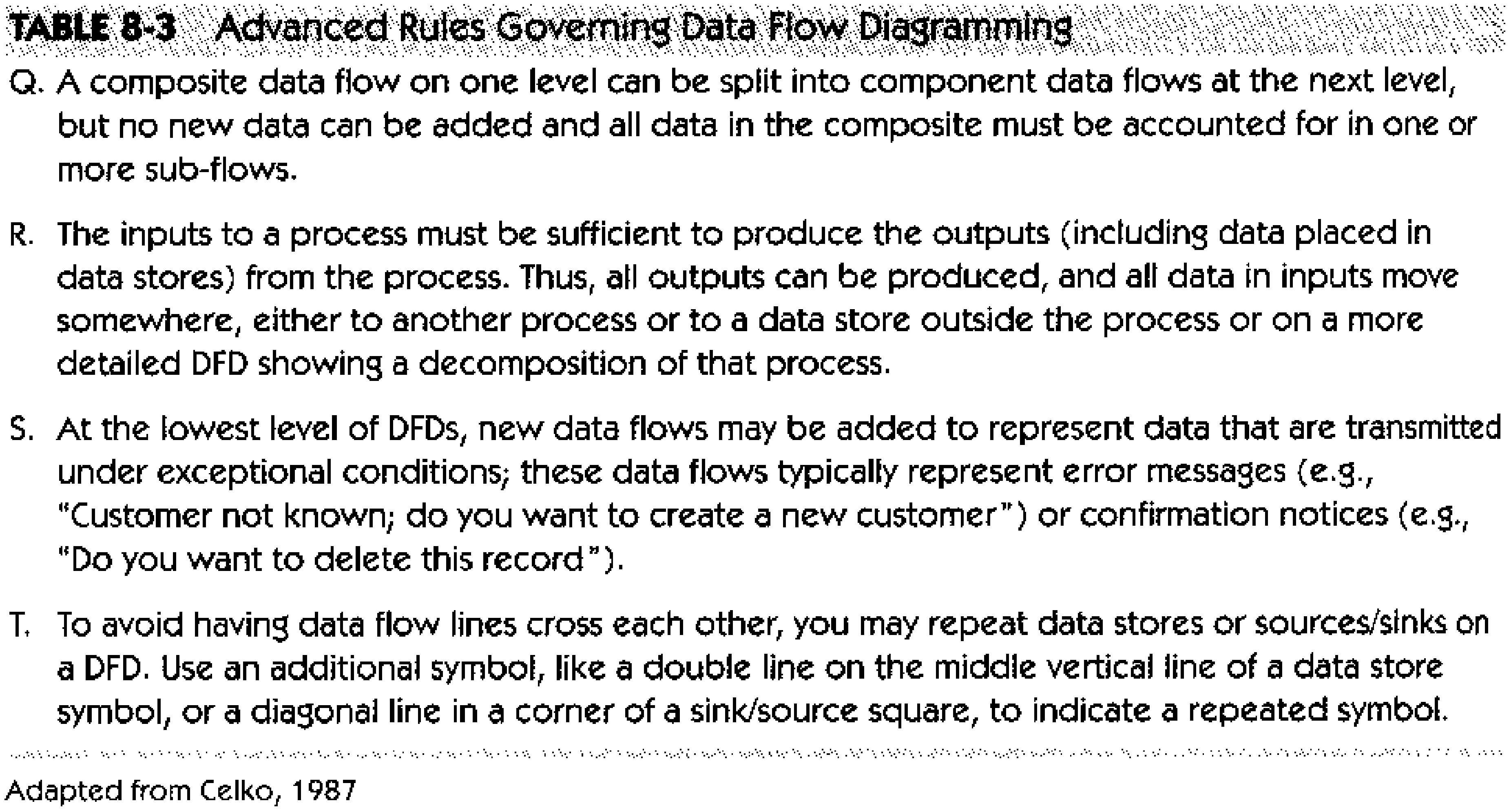


TABLE 8-3 Advanced Rules Governing Data Flow Diagramming

Q. A composite data flow on one level can be split into component data flows at the next level, but no new data can be added and all data in the composite must be accounted for in one or more sub-flows.

R. The inputs to a process must be sufficient to produce the outputs (including data placed in data stores) from the process. Thus, all outputs can be produced, and all data in inputs move somewhere, either to another process or to a data store outside the process or on a more detailed DFD showing a decomposition of that process.

S. At the lowest level of DFDs, new data flows may be added to represent data that are transmitted under exceptional conditions; these data flows typically represent error messages (e.g., "Customer not known; do you want to create a new customer") or confirmation notices (e.g., "Do you want to delete this record").

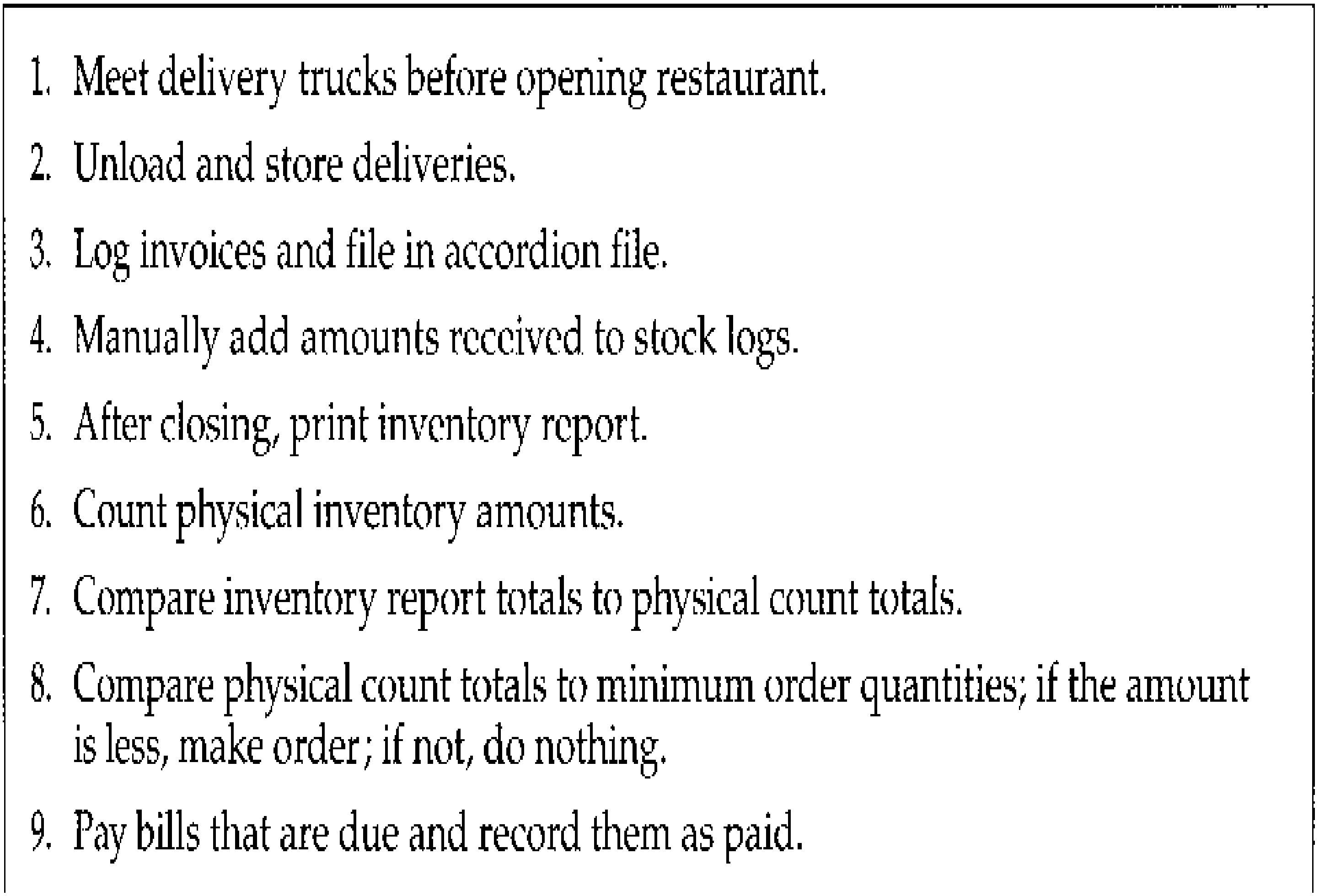
T. To avoid having data flow lines cross each other, you may repeat data stores or sources/sinks on a DFD. Use an additional symbol, like a double line on the middle vertical line of a data store symbol, or a diagonal line in a corner of a sink/source square, to indicate a repeated symbol.

Adapted from Celko, 1987

DFD types I

- Current physical DFD
  - Process labels include location and technology
    - Names of people
    - Names of computer and other physical systems
  - Same with data stores and dataflows

DFD types IIa



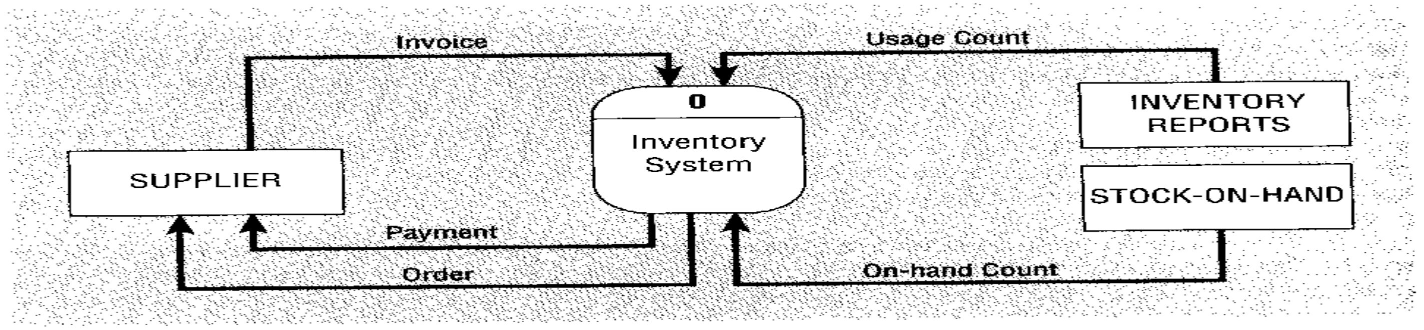
1. Meet delivery trucks before opening restaurant.
2. Unload and store deliveries.
3. Log invoices and file in accordion file.
4. Manually add amounts received to stock logs.
5. After closing, print inventory report.
6. Count physical inventory amounts.
7. Compare inventory report totals to physical count totals.
8. Compare physical count totals to minimum order quantities; if the amount is less, make order; if not, do nothing.
9. Pay bills that are due and record them as paid.

Figure 8-12
List of activities involved in Bob Mellankamp's inventory control system for Hoosier Burger

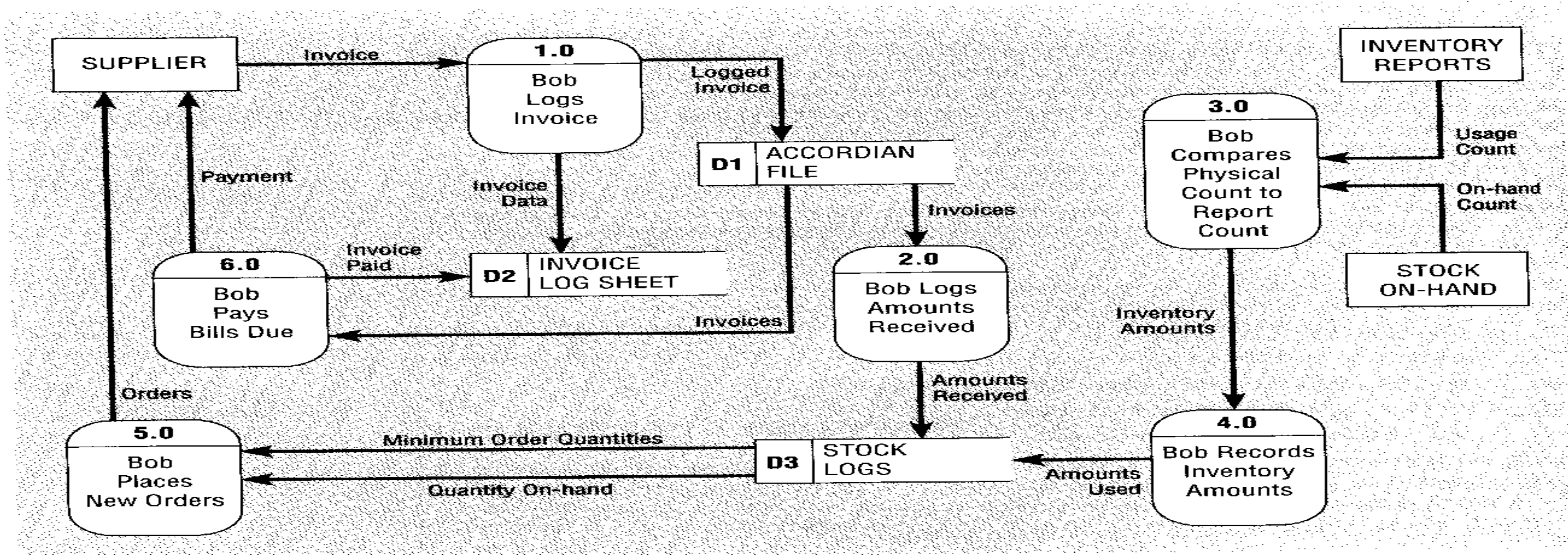
DFD types IIb

Figure 8-13
Hoosier Burger's current physical inventory control system

(a) Context diagram



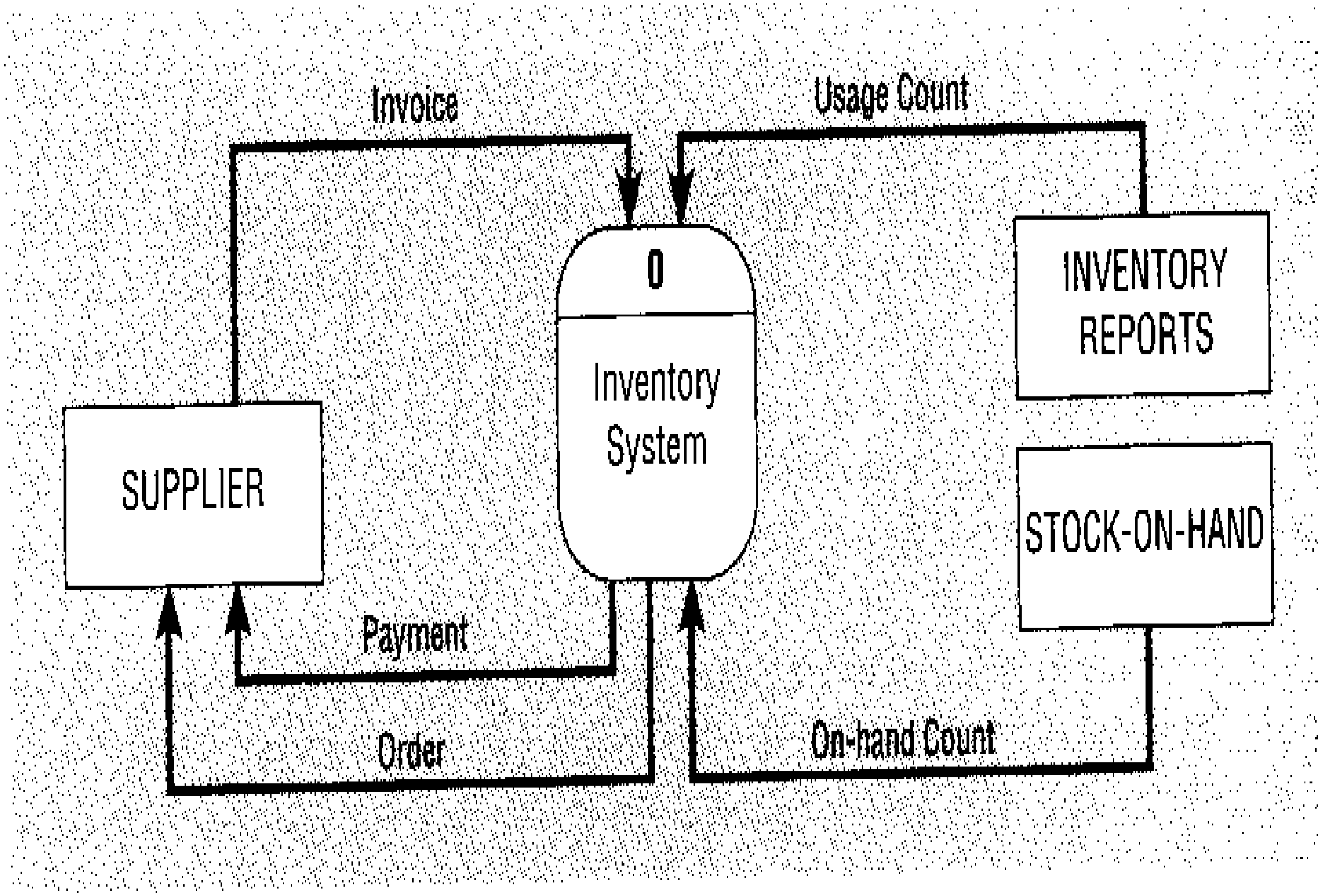
(b) Level-0 data flow diagram



DFD types IIc

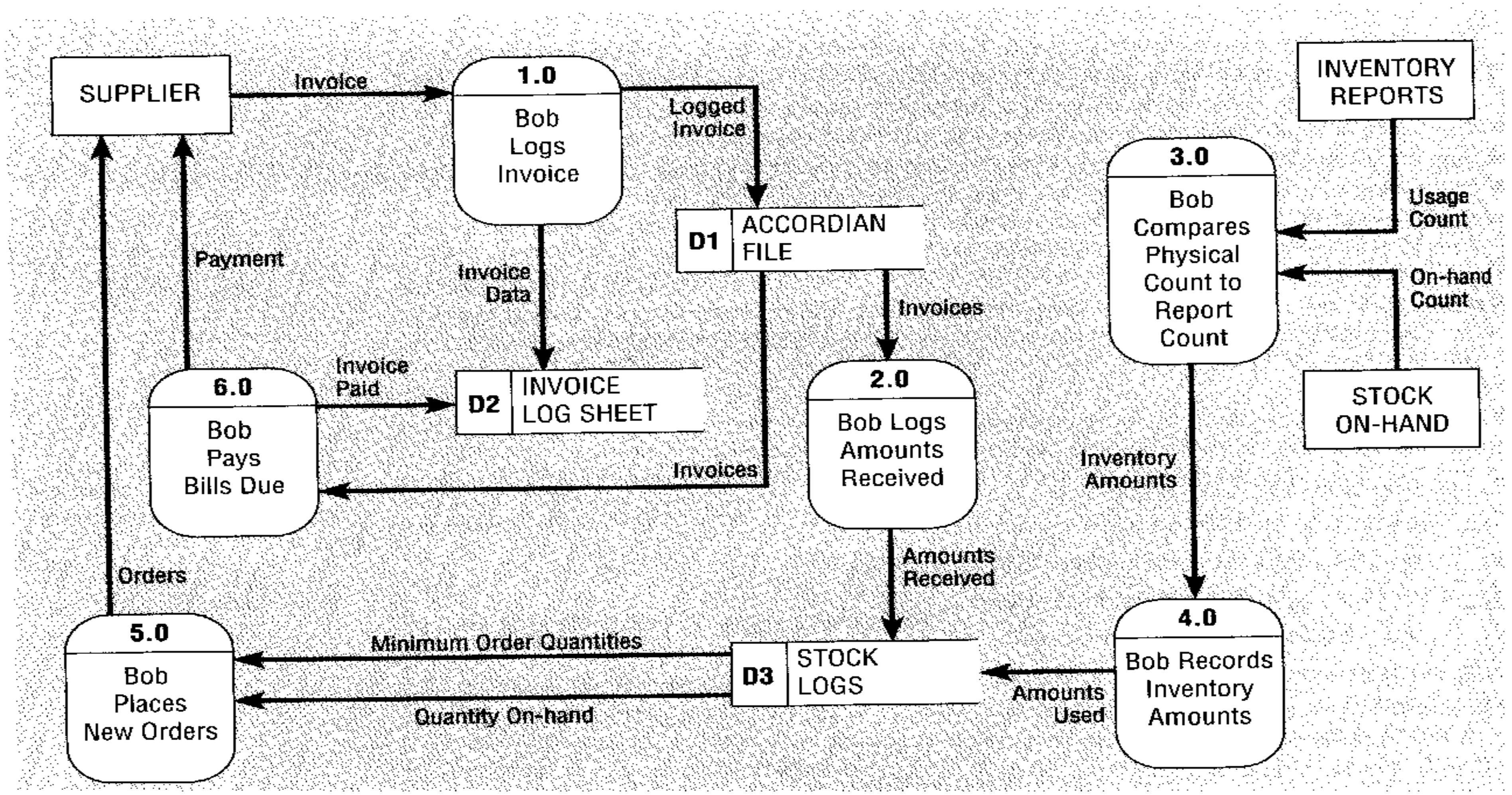
Figure 8-13
Hoosier Burger's current physical inventory control system

(a) Context diagram



DFD types IId

(b) Level-0 data flow diagram

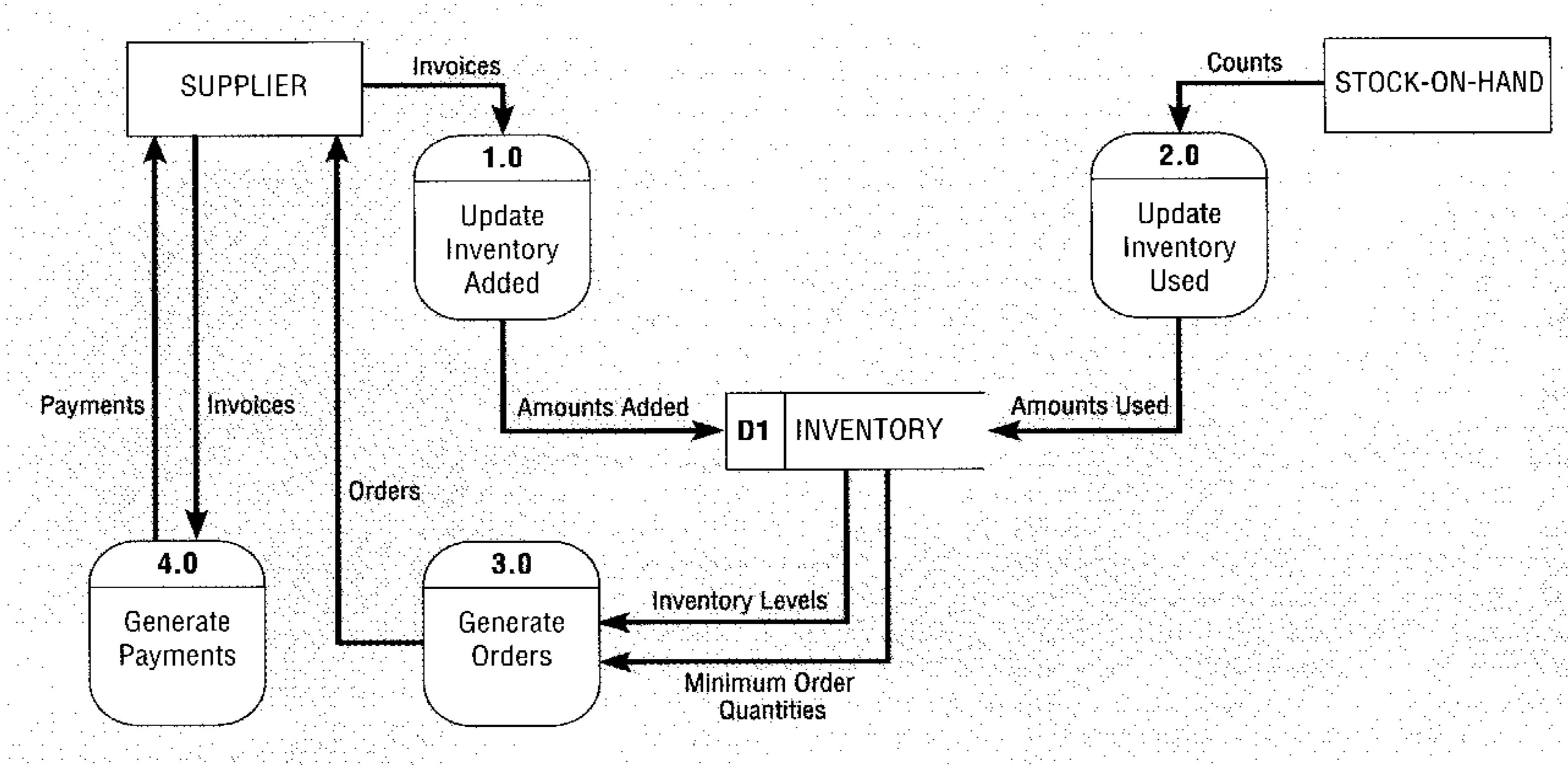


DFD types III

- Current logical DFD
  - Physical characteristics are removed
    - Names of people, departments, and other locations
    - Names of technological physical devices & facilities

DFD types IV

Figure 8-15
Level-0 data flow diagram for Hoosier Burger's current logical inventory control system

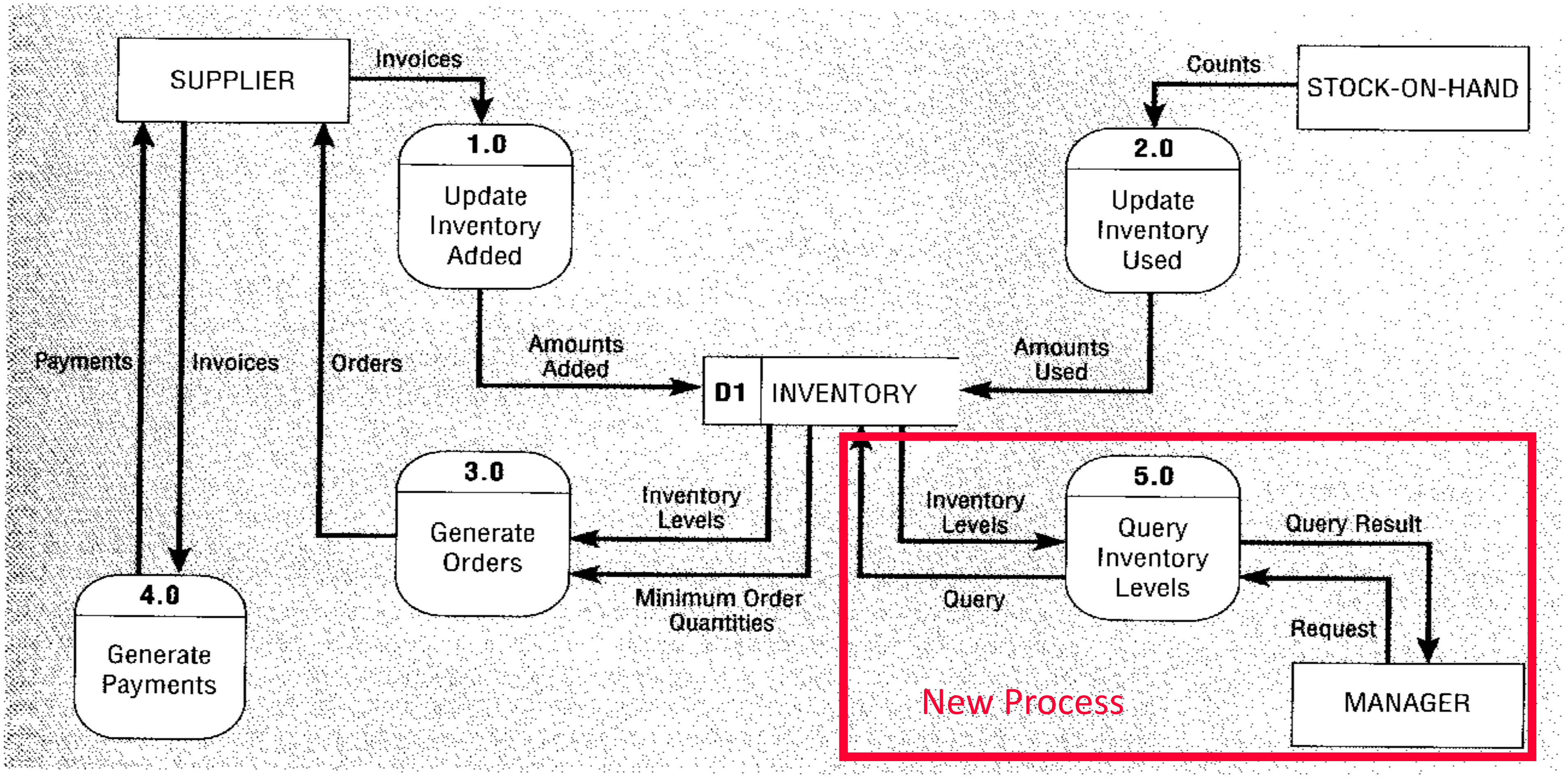


DFD types V

- New logical DFD
  - Derived from current logical DFD
  - Removed entities
  - Expanded and added entities
  - Flows and processes reorganized
  - Order modified
- May remain identical to current logical DFD

DFD types VI

Figure 8-16
Level-0 data flow diagram for Hoosier Burger's new logical inventory control system



DFD types VII

- New physical DFD
  - The physical implementation of the new logical DFD
  - Names and locations added
  - Technologies and devices identified
  - Identification of automated procedures

DFD guidelines & rules V

- Completeness: include and fully describe all necessary components of a system
- Consistency: Assure that all information at one level is also contained on the next/former level
- Iterative development process
- Timing
  - Cannot be represented by DFD
  - Will be represented by state-transition diagram
  - Assume system operates indefinitely
- Decide about the primitive (lowest level) diagrams