NBSIR 81-2240

Design of Information Systems Using Scenario-Driven Techniques

W. T. Hardgrave
S. B. Salazar
E. J. Beller, III

Data Management and
Programming Languages Division
Center for Programming Science and Technology
Institute for Computer Sciences and Technology
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

March 1981

NBSIR 81-2240

DESIGN OF INFORMATION SYSTEMS USING SCENARIO-DRIVEN TECHNIQUES

W. T. Hardgrave
S. B. Salazar
E. J. Beller, III

Data Management and
Programming Languages Division
Center for Programming Science and Technology
Institute for Computer Sciences and Technology
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

March 1981

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Design of Information Systems Using Scenario-Driven Techniques

W. Terry Hardgrave
Sandra B. Salazar
Edwin J. Beller III

6 January 1981

National Bureau of Standards
Institute for Computer Sciences and Technology
Washington, D.C.

This paper describes a technique for developing information systems using a scenario-driven design approach. The approach emphasizes client (that is, the user who is purchasing the system) participation in the design process. The first step is to develop a collection of "scenarios" which document the interaction between the computer and the human user. Using the scenarios, information-flow diagrams and database designs may be constructed. After the client has approved these documents, they can be used to establish disk capacity requirements and transaction rates, and finally to specify all hardware and software requirements. The primary advantage of this approach is that the scenarios provide a good indication of the ultimate usefulness and cost of the system. The client can review these documents and approve, modify, or reject the system design before any software is generated. This paper describes the scenarios, the information flow technique, and the database design approach using, as an example, a small business application.

TABLE OF CONTENTS

1. Introduction

Designing an interactive information system requires specialized techniques that are unnecessary in the development of other types of computer systems. Human factors issues are involved, since the user community typically does not consist of computer specialists. Interactions between humans and computers must be clearly defined. The database must be described independently of any particular programs. Standard flowchart practices must be modified to capture the data flow through an interactive system.

This paper describes a technique for developing a logical design for an interactive information system. The product of this design process is a document having these components:

* Collection of scenarios
* Information flow diagrams
* Data-base design

Step-by-step approaches to the development of the components, which are identified below, will be presented in subsequent sections.

The term "scenario" is defined for the purpose of this paper as a detailed documentation of the interaction between a computer system and the human user working at a terminal. A scenario is essentially a hard-copy of an interactive session. This level of detail is seldom available until after the system is implemented and capable of writing output to a terminal. One goal of the scenario-driven design approach is to capture this level of detail before implementation.

The collection of scenarios is a complete description of every screen format that the terminal operator may possibly encounter. The data appearing in the scenarios should be typical of data that may actually be used in the application. Thus, this collection of scenarios completely and "by example" describes the user interface to the information system. The scenarios are discussed further in Section 2.

The information flow diagrams describe the movement of data through the system. They show where data enters the system, where it exits, where it is stored, and where it is displayed for the human user. While standard flowcharts have had widespread usage, the application of flow diagrams to interactive systems requires a somewhat different

approach. A particular format and some stringent conventions are discussed in Section 3.

The database design is a complete description of the data that is to be held and permanently maintained by the system. One expects this data to have a long life span; certainly it will be longer than the execution time of any programs that manipulate it. Therefore, it is necessary to describe the data apart from any particular program or subroutine. The database design technique, discussed in section 4, is an application and extension of the "entity-relationship model" [CHEN76].

The scenario-driven technique is currently being designed as part of a project to automate the operation of a wholesale book dealer. The examples used throughout this paper are taken from that application.

There is one significant advantage of this method. There will be detailed communication between the system designer and the client (the user who has contracted to purchase the system) during the design stage, as the client will review the scenarios many times during their development. Thus, when the report on the logical design is available, the client is already familiar with the system and can be confident that it meets expected needs. There is no mystery and little chance that the client will receive a system substantially different than envisioned.

2. Scenarios

2.1 Definition

A scenario is a detailed documentation of the interaction between a computer system and the human user working at a terminal. Scenarios may be written to mimic the interaction on a line-by-line basis; a menu-by-menu basis, a screen-by-screen basis, or any other basis that can reasonably be put into the design document. For the book wholesaler system, scenarios are written on a screen-by-screen basis, that is, each complete computer-human interaction (such as "ordering", "receiving") is documented in terms of the screens sequentially viewed by the user. Each screen described here consists of a menu or a menu plus system prompts.

Most importantly, the scenario is a tool to facilitate communication between the client and the system designer. That is, the scenario is a very simple visual image that describes one part of system behavior. A comprehensive collection of scenarios can describe system behavior in enough detail to convince the client that the system will perform as envisioned.

There are some other benefits that come with the development of detailed scenarios:

* The scenarios may be included in the User's Manual as tutorial material.

* The scenarios may be incorporated into a test plan to determine system acceptance.

* The scenarios may be used as the basis for the contract between the client and the systems designer or software implementer.

The following subsections will discuss the development of screen-based scenarios and the menu format. Examples of menus from the wholesale book system are included.

2.2 Development

In a screen-based scenario design, all scenarios will have in common the top level screen, that is, the first screen that the user views. The master menu gives an overview of the aspects of the organization addressed by the

information system, thereby defining the scope of the system. The designer and the client must work on the master menu until an agreement is reached and the menu satisfies both parties. During this process, the system designer and the client must learn each other's terminology and develop a common basis for further work.

After some agreement is reached on the nature of the master menu, work may proceed on the screens at the next level of detail. There is no established format for the scenarios; the screens may describe menus, user commands, or any interaction that is acceptable to both the client and the designer. This process of defining progressively lower levels of the system continues until all possible interactions have been completely specified.

There are problems involved in the management of this documentation. An exhaustive enumeration of all possible scenarios will result in a large number of screens for even a small system. As menus at different levels are redesigned, it becomes tedious to maintain consistency and verify correctness of the interfaces.

The notation used in the scenarios is not a metalanguage. The data requested from the user or printed for the user should be typical data from actual situations. This avoids any formatting problems and similar misunderstandings between the client and the systems designer.

2.3 Menu Format

The menu format is illustrated in Figures 2-1 through 2-4. Each menu (and its corresponding screen) is numbered to indicate its level, starting with the master menu as Ø.Ø. A menu has four columns with the following headings:

* Number
* Option
* Next
* Page

Each entry in the menu represents a possible selection by the user. The "option" is the textual statement of the function that is displayed for the terminal operator. The "number" signifies the key stroke to select the corresponding option.

The "next" and "page" fields do not actually appear on the screen when the system is implemented. However, they are necessary for the writer and the reader of the logical design document in order to develop and follow the sequence of menu displays. The "next" field indicates which menu appears next if the corresponding number is selected. The "page" field gives the actual page number in the logical design document. Both of these fields are useful, albeit redundant. The page field is useful to the reader of the final design document; and the next field is useful during the development cycle, when menu numbers are available but page numbers are not.

2.4 Sample Screens and Menus

The sample screens on the following pages describe part of the "ordering" process for the book wholesaler system. The level Ø.Ø menu is the first menu that a terminal user would see after starting the system. "Ordering" is selected by typing "1" in response to the system's prompt "Select one:".

The "Next" field of the "ordering" entry indicates that the next menu to appear will be the 1.Ø menu. After it appears, as shown in Figure 2-2, the terminal user may choose to order for a customer or order something to be stored in the inventory. In this case, a "1" is typed to signify an order for a customer.

As indicated by the "Next" field of the first entry in the 1.Ø menu, the next menu to appear will be the 1.1 menu. In menu 1.1, the terminal user may choose to create a new order or to add to an existing order. In this case, the user chooses to add to an existing order by selecting "2". The system then prompts the terminal user for the "Customer Order Group Number", abbreviated COG#. The system prints the customer identifier and the shipping address.

According to the menu of the next screen, the 1.1.2 menu, the terminal user may select to search for the title based on one of several possible inputs. In this case, the terminal user selects a search on ISBN. The system prompts for the ISBN, the International Standard Book Number, which is the unique identifier for books. After the ISBN has been entered, the system adds the book to the order group and prints a message to that effect as shown. As indicated, the screen would refresh itself to allow the operator to order another title for this customer order.

Menu Ø.Ø

No.	Option	Next	Page
1	Ordering	1.Ø	2
2	Receiving	2.Ø	4Ø
3	Inquiry/Update	3.Ø	1Ø6
4	Picking List/Invoice/PO	4.Ø	147
5	Accounting functions	5.Ø	166
6	End of day	6.Ø	183
7	Return to Operating System	-	-

Select one: 1

Figure 2-1 Screen Ø.Ø

Menu 1.Ø

No.	Option	Next	Page
1	Order for customer	1.1	3
2	Order for inventory	1.2	26

Select one: 1

Figure 2-2 Screen 1.Ø

Menu 1.1

No.	Option	Next	Page
1	New order	1.1.1	4
2	Add to order	1.1.2	10

Select one: 2
Enter COG#: 53296

Customer Id. is: NLM

Shipping Address is: National Library of Medicine
9600 Rockville Pike
Bethesda, MD 20014

Figure 2-3 Screen 1.1

Menu 1.1.2

No.	Option	Next	Page
1	Search on alt. key	1.1.2.1	3Ø
2	Search on ISBN	1.1.2	1Ø
3	Search on series	1.1.2.3	35
4	Enter as new title	1.1.2.4	37
5	End order	Ø.Ø	1

Select one: 2
Enter ISBN: Ø-13-854547

An order for Structured System Design by Gane and Sarson has been added to COG# 53296 for National Library of Medicine.

Figure 2-4 Screen 1.1.2

3. Flow Diagrams

3.1 Characteristics

Flow charts have been used for many years to describe the flow of computer programs and data. Although the diagrams used here are similar to traditional flow charts, some stringent rules for their creation and use have been imposed. These restrictions, discussed below, make the diagrams easier to read and more useful as a management tool.

As depicted in Figure 3-1, the format for the flow diagram exhibits the following characteristics:

* All flow is from left to right.

* The triggering event is the first box on the left.

* The process ends with the rightmost box (usually labeled "end").

* The left margin is partitioned by "roles"; that is, the organizational (or external) entities are listed down the left margin. One role that is usually included is "database". In this way, processes that store data in or retrieve data from the database may be documented.

* The boxes have various shapes that are assigned meanings to fit the application. This example uses trapezoidal, square, and cylindrical symbols; the meanings are explained below.

The diagram shown in Figure 3-1 is short and fits on one page; most diagrams will require several pages.

There are several advantages in using this kind of flow diagram:

* The triggering events can easily be spotted by looking on the leftmost part of the diagrams.

* The involvement of a particular role can be isolated by looking along the appropriate horizontal strip.

* Time may be measured horizontally. Time intervals after triggering can be set up along the horizontal axis and charting symbols placed in the appropriate zones.

* Parallel charts may also be created. For example, in a manufacturing environment, a materials flow diagram can be set up parallel to the information flow diagram. The materials flow documents the flow of parts, subassemblies, etc.; the information flow documents the paperwork.

However, since a flow diagram should be generated for each possible flow sequence, there may be problems with the management of this volume of documentation.

3.2 Sample Flow Diagram

The sample flow diagram shown in Figure 3-1 depicts the flow for the part of the ordering process discussed previously. The trapezoidal boxes represent the display of a menu; the menu number is given inside the box. In cases where there is a square box underneath the trapezoid, the menu interaction requires input of a data-item by the terminal user. The name of the data-item is contained in the box. The freestanding square boxes represent processes that the system performs. If the process stores or retrieves data from the databases, this is signified by a line from the process to the appropriate database. Herein, the term "database" is used very loosely and could, in this case be used interchangeably with the term "file". The various databases are represented by the cylindrical symbols.

In Figure 3-1, the flow begins on the left when the customer prepares the order. The order then goes to the ordering department. The terminal operator calls the system into operation and the level Ø.Ø menu is displayed. This is signified by the trapezoid containing Ø.Ø. The terminal operator selects option 1; this is signified by the "1" above the trapezoid. Then menu 1.Ø is displayed. The terminal operator selects option 1; again this is signified by the "1" above the trapezoid. Then menu 1.1 is displayed; the terminal operator selects option 2. The box below the trapezoid indicates that the terminal operator must enter the "Customer Order Group Number", abbreviated COG#. As shown by the square box, the system retrieves the customer order group from the "Customer Order File", abbreviated CO. Also, the system gets the "Customer Identifier", abbreviated C#, from information contained in the COG. Finally, the system retrieves the "Shipping Address", abbreviated SA and displays it along with the customer number.

The next menu to be displayed is menu 1.1.2. The terminal operator chooses to search on ISBN. The system requests the ISBN and the operator enters it. The system then searches the title file to ensure that the ISBN is valid. If the ISBN is not valid, then other menus not shown here are invoked. Finally, the title is added to the COG and the process ends.

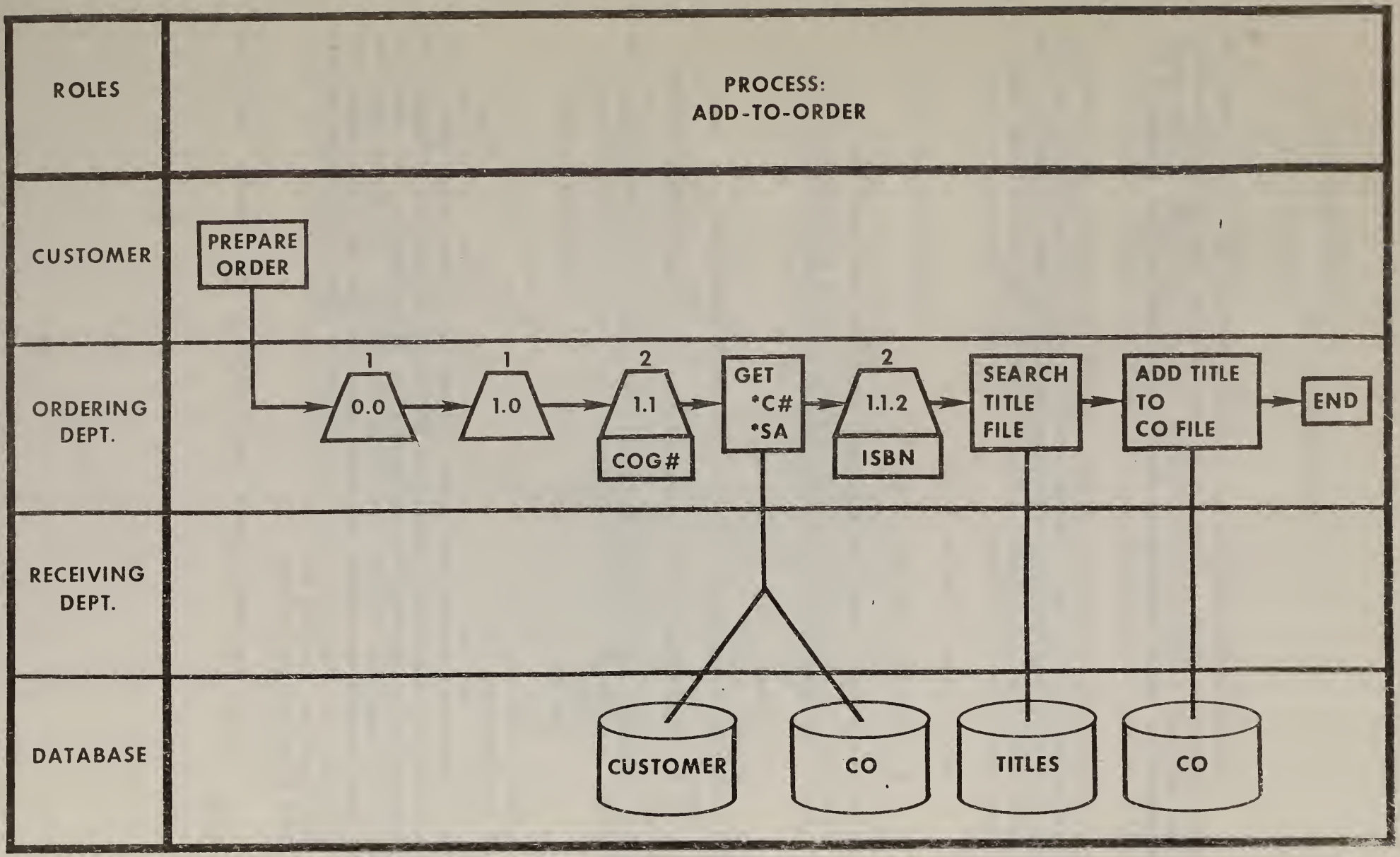


FIGURE 3-1: SAMPLE FLOW DIAGRAM

4. Data-base Design

The database design is, roughly speaking, the format of the various data files that are to be stored permanently (usually on a disk). The actual values in the files may change as different data enters and leaves the system. We will refine this concept as we proceed.

Our approach to database design consists of several steps as described below. While it is similar to the entity-relationship approach described by Chen [CHEN76], the technique is specified in more detail and may differ somewhat from his philosophy.

The process consists of:

* Enumerating entities
* Determining the key attribute of each entity
* Defining non-key attributes for each entity
* Enumerating relationships
* Determining keys for the relationships
* Defining non-key attributes for the relationships
* Determining which relationships may also have the dual role of entities
* Assigning dual keys to dual entity/relationships

4.1 Entities

First, the designer must enumerate the entities. An entity is a concept that the designer wants the information system to recognize and manipulate. For example, in an inventory system, a part would be an entity. For some applications, there are many possible choices for entities and decisions on tradeoffs may be required. The discussion herein will center on the Wholesale Books example rather than dealing with generalities.

In the wholesale book application, the basic entities are:

* Customers

* Publishers

* Titles

Sample record diagrams giving the most important attributes of each entity are shown in Figure 4-1. In the actual database, more attributes are necessary; some are omitted for this discussion to keep it manageable.

Each entity is uniquely identifiable by a single attribute called the key (denoted by "*" in the figures). For example, a person may be uniquely identifiable by his/her social security number or a part may be uniquely identifiable by the part number.

4.2 Relationships

Next, the designer must enumerate the relationships, that is, the associations among entities. The attributes of a relationship should describe the relationship, not the individual entities, and therefore must include the keys, but no other attributes, of the entities involved.

Figure 4-2 shows some typical relationships; again these are modified from the actual application for simplicity. However, they are quite similar to an early design before the complexity required by the client was introduced. The item-order relationship relates customers to titles. A customer (CID) orders a title (ISBN) in some quantity (QTY). The title/publisher relationship relates titles to publishers. A publisher (PID) publishes a title (ISBN).

Our example is somewhat more complex. The item-order relationship needs to be grouped to reflect which books appear on an incoming order. Also, the item orders need to be grouped to reflect which items are included on a single order to the publisher. Therefore, the item-order relationship needs to be referenced as an entity in other relationships. This is accomplished by adding an attribute, Customer Order Group (COG#) which becomes the key of the augmented item-order dual entity/relationship (as shown in Figure 4-3, with the key denoted by "**").

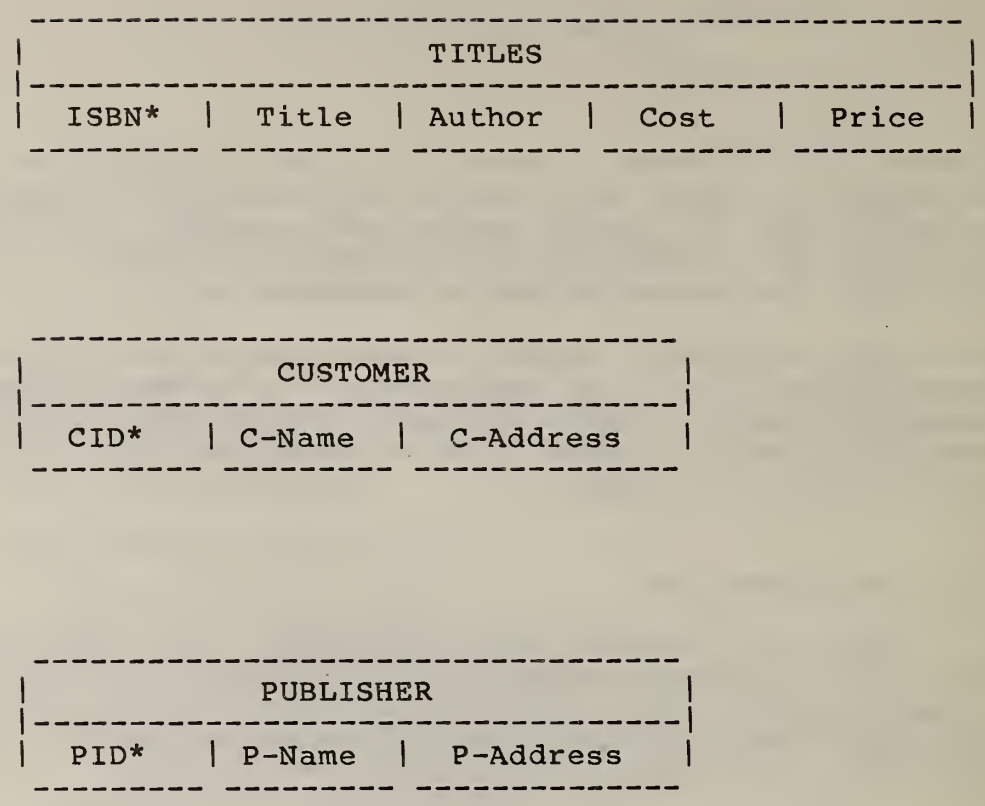
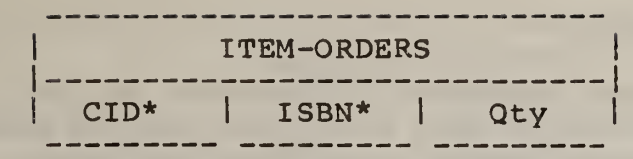


Figure 4-1 Entities



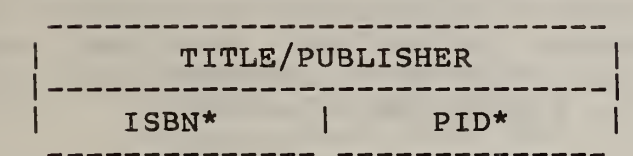


Figure 4-2 Relationships

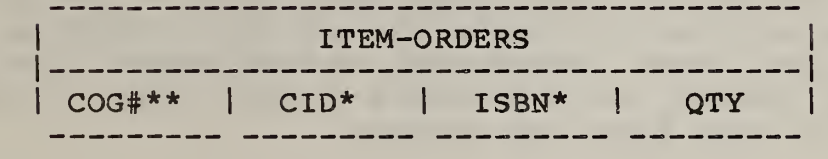


Figure 4-3 Dual Entity/Relationship

4.3 Data Dictionary

The purpose of the data dictionary is to provide a written description of each attribute that is referenced in the database (in both entities and relationships). The data dictionary can be used as a basis for communication among the various people working on the design of the system and for resolving the inevitable questions that arise during the design process concerning the purpose and meaning of various data-items. Whenever the database is reorganized, the data dictionary is updated to reflect the new organization. Thus, the data dictionary is an important tool for managing and controlling the system design process.

Although the creation and maintenance of a dictionary can be an expensive undertaking, it is possible to scale the complexity to fit the application. For example, the dictionary may contain many attributes for each data-item or as few as two; it may be managed manually or with a computerized system.

Because this approach is intended for a small organization with limited resources, the data dictionary described here is as simple as possible and may be managed manually at a small cost. As depicted in Figure 4-4, the dictionary is a table with two columns: data-item and narrative. Each entry in the table describes one data-item, or attribute, that is defined in the system. The column labeled "data-item" contains the names of the data-items, while the column labeled "narrative" contains descriptions of the data-items meaningful to the user. The usefulness of the data dictionary (as well as its complexity and cost) could be increased by adding more information to each entry. For example, a column "entity" could be added to keep track of the entities in which each data-item appears.

DATA DICTIONARY	
DATA-ITEM	NARRATIVE
ISBN	International Standard Book Number
Title	Title of book
Author	Author of book
Cost	Publisher's price of book
Price	Our normal price for this book
CID	Unique identifier for customer
C-Name	Name of Customer
C-Address	Address of Customer
PID	Unique identifier for Publisher
P-Name	Name of Publisher
P-Address	Address of Publisher
Qty	Quantity: Number of this ISBN ordered by this CID
COG#	Customer Order Group Number: Unique identifier for Purchase Order

Figure 4-4 Data Dictionary Table

5. Concluding Remarks

The information system design technique presented in this paper is based on a surprisingly simple but rarely-used idea-- that the systems analyst build a mock-up of the system for review by the client before implementation begins. The method of constructing the mock-up, described herein, is the development of a collection of "scenarios," the possible interactions between the computerized information system and the human user. The client and the system designer take part in a number of iterations of reviewing and revising the scenarios, while the information flow and the database design proceed in parallel. Approval of a collection of scenarios by the client is the first milestone in the development of the system.

6. References

CHEN76 Chen, Peter, "The Entity-Relationship Model -- Toward a Unified View of Data", ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-36.

GANE79 Gane, Chris and Sarson, Trish, Structured Systems Analysis: Tools and Techniques Prentice-Hall, Englewood Cliffs, NJ, 1979, 241 p.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 81-2240	2. Performing Organ. Report No.	3. Publication Date March 1981

4. TITLE AND SUBTITLE

Design of Information Systems Using Scenario-Driven Techniques

5. AUTHOR(S)

W. Terry Hardgrave, Sandra B. Salazar, Edwin J. Beller III

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This paper describes a technique for developing information systems using a scenario-driven design approach. The approach emphasizes client (that is, the user who is purchasing the system) participation in the design process. The first step is to develop a collection of "scenarios" which document the interaction between the computer and the human user. Using the scenarios, information-flow diagrams and database designs may be constructed. After the client has approved these documents, they can be used to establish disk capacity requirements and transaction rates, and finally to specify all hardware and software requirements. The primary advantage of this approach is that the scenarios provide a good indication of the ultimate usefulness and cost of the system. The client can review these documents and approve, modify, or reject the system design before any software is generated. This paper describes the scenarios, the information flow technique, and the database design approach using, as an example, a small business application.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Database design; data dictionary; design; flowchart; information flow; information systems; interactive systems; requirements; scenarios.

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES 21

15. Price $5.00