1976

Levels of Abstraction for Determining Data Requirements

Thomas I. M. Ho

Report Number:

76-195

LEVELS OF ABSTRACTION FOR DETERMINING DATA REQUIREMENTS

Thomas I. M. Ho
Purdue University
Computer Sciences Department
and
School of Management
West Lafayette, Indiana 47907

CSD-TR 195
May 1976

LEVELS OF ABSTRACTION FOR DETERMINING DATA REQUIREMENTS

Thomas I. M. Ho
Purdue University
Computer Sciences Department
and
School of Management
West Lafayette, Indiana 47907

Keywords: systems analysis, data base design

Categories: 2.42, 3.50

ABSTRACT

It is apparent that a systems analyst's original perception of an information problem is not sufficiently precise to enable an implementation that satisfies the exacting constraints of a computer system. Even if an analyst perceives the essential requirements of his problem, available high-level software facilities, e.g. data base management systems, do not clearly suggest how they might be applied to solve the problem of interest. Therefore, this paper outlines a systems analysis methodology for translating the organizational view of an information system into an intermediate form that spans the gap to the computerized implementation of that information system. Based upon a foundation of concepts that characterizes the role of a data base in an information system, the methodology emphasizes the determination of data requirements. Consistent with the approach of structured programming, the methodology employs levels of abstraction with step-wise decomposition and restricted data structures.

INTRODUCTION

A major part of the information systems development effort is expended in data base design. The common approach to this task revolves around the construction of the physical structure of the data base and the implementation of algorithms that manipulate the resulting data structure. To aid this process, a wide variety of techniques, e.g. data base management systems and file organization techniques, have been developed to enable the construction and manipulation of complex data structures. However, there is a notable absence of aids to the activity that must necessarily precede the more apparent task of physical data base design. The data base designer is sorely in need of a conceptual framework that establishes guidelines for inferring the characteristics of the data base that supports the information system.

A review of recent approaches to program design will reveal the substance of this assertion. The availability of high-level programming languages has only recently been supplemented by a strategy for the design of programmed implementations of algorithms. The conceptual framework afforded by structured programming establishes guidelines for translating algorithms into programs. By virtue of this framework, we are now able to more effectively perform and teach the intricate art of programming.

The conceptual framework that we seek is based upon a foundation of concepts. We shall briefly review these concepts before we outline the application of these concepts to the task of data base definition.

FOUNDATION CONCEPTS

Systems concepts.

The significance of system concepts to management information systems has been pointed out by Davis [1] and many other authors. A **system** may be defined as a set of interacting elements that operate together to accomplish an objective. Therefore, a system is composed of **subsystems** which may also be composed of other subsystems. The division of a system into its various subsystems is called **factoring** or **decomposition**.

The interconnections and interactions between subsystems require **interfaces** between subsystems to enable inter-subsystem communication. Communication is necessary to enable co-ordination of subsystems. To promote independence between interacting subsystems, and thereby reduce the need for co-ordination, systems may employ **decoupling** mechanisms. Decoupling mechanisms include:

1. Inventories, buffers, and waiting lines
2. Slack and flexible resources
3. Standards.

Finally, the basic model of a system consists of an input, a process, and an output as illustrated in Figure 1. In order to control operation of a system, a feedback and control loop is added to the basic model as illustrated in Figure 2. The feedback loop consists of a sensor that measures system output and that compares these measurements against some specified standard. If the sensor detects any variance from the standard, the loop invokes a control device to send an input to the process to bring its output into conformance with the specified standard.

Figure 1. Basic System Model

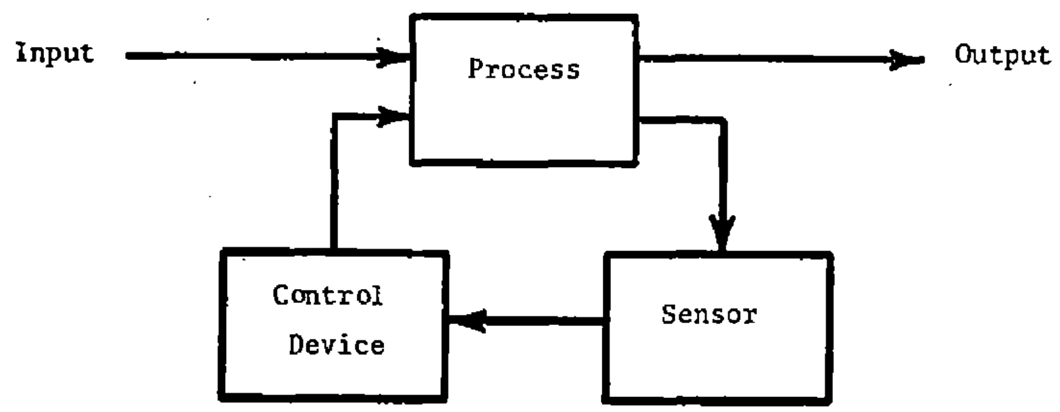

Figure 2. System with Feedback and Control Loop

Management information systems

A computer-based management information system includes diverse organization and computer systems that are linked by an information system [2]. On one hand, the organization is qualitative and unstructured. On the other hand, the computer is technical and rigid. An overview of the management information system in terms of its component systems is illustrated in Figure 3.

Organization system

The organization system is composed of functional subsystems that represent the various functions performed by the organization. Each of these subsystems performs actions or decisions in order to achieve designated goals. These actions or decisions can be performed only if the subsystem receives either the data that enables the action or the data that establishes the constraints and objectives of the decision. Co-ordination of these activities motivates interest in the attributes of selected persons, objects, and events in each subsystem.

It is apparent that the organization system consists of persons and objects of interest to organizational activities. In addition, events occur in conjunction with organizational activities. For example, in a manufacturing firm, orders are received, products are produced, and raw materials are purchased. The role of events is especially substantial in co-ordinating the functional subsystems of the organization.

Since the functional subsystems are parts of a single organization system, co-ordination of the subsystems is necessary in order to achieve organizational objectives. For example, the filling of customer orders requires co-ordination of the Sales, Inventory, Production, and Shipping subsystems. Co-ordination requires communication among the interacting subsystems as illustrated in Figure 4. This co-ordination is effected by the establishment of interfaces between interacting subsystems to communicate the occurrence of events in one subsystem to another subsystem that must perform some action in response to that event. For example, the Sales subsystem reports customer orders to the Inventory subsystem which reports either shipments to the Shipping subsystem or backorders to the Production subsystem depending on product availability.

The integration of the various functional subsystems is promoted by associations that exist among the various persons, objects, and events of interest. For example, there may be an association between a person and the events that he caused, between events and the object that they affect, or between an event and another event caused by the first event. In the first case, a Customer may be associated with an Order. In the second place, a Backorder may be associated with a Product. In the final case, an Order may be associated with a

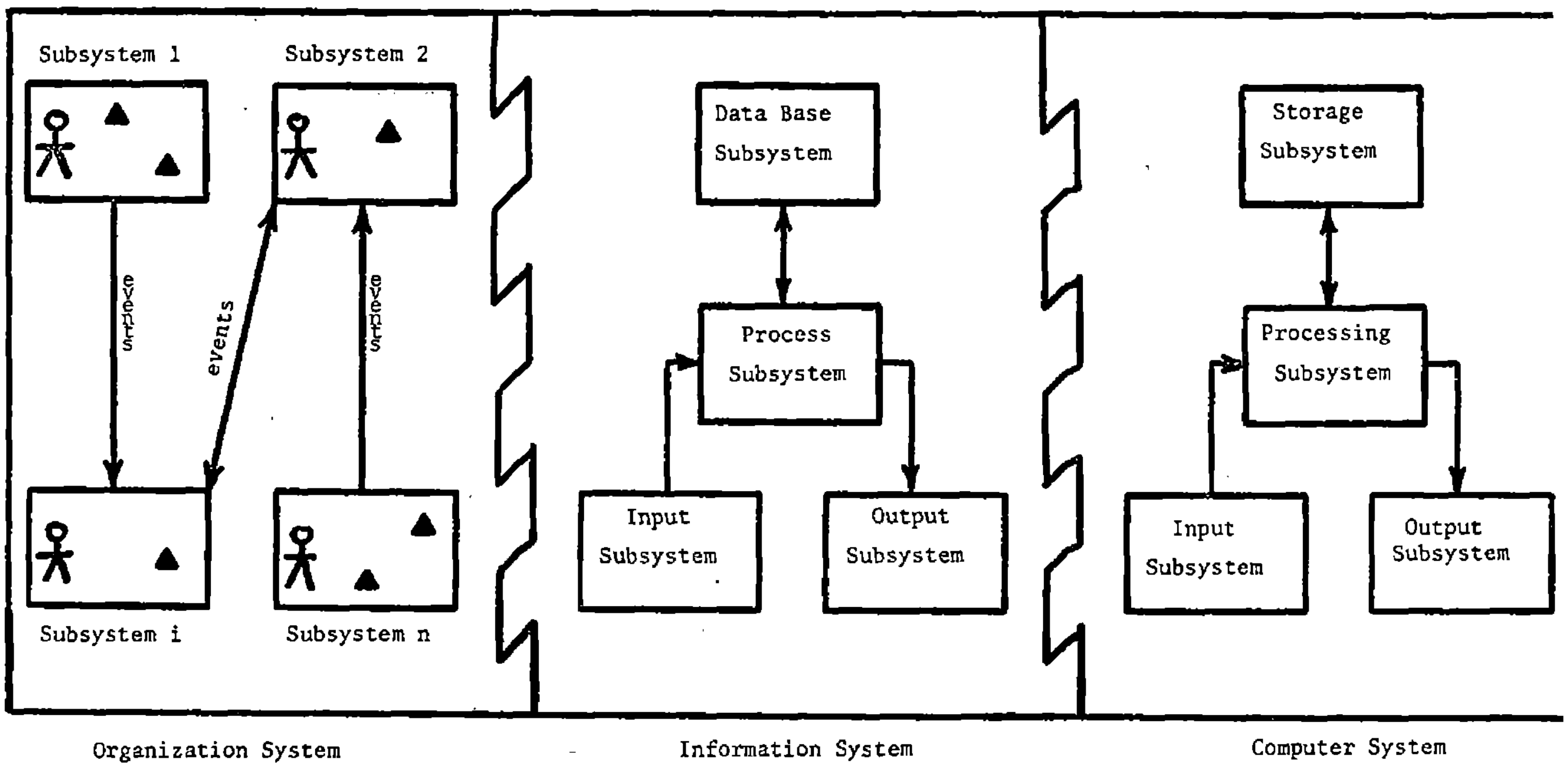

Figure 3. Management Information System

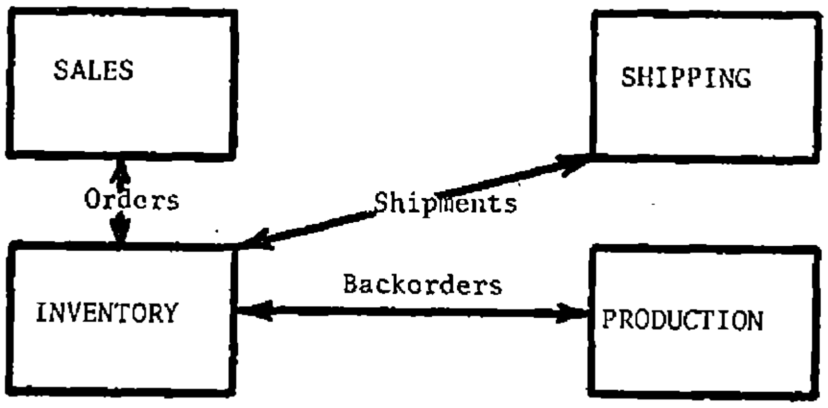

Figure 4. Subsystem Interaction

Backorder. Less apparent is the contribution of these associations to enabling feedback. For example, by virtue of the association of a Customer with an Order and of an Order with a Backorder, the subsequent fulfillment of a Backorder could be reported to the affected Customer. The interplay of persons, objects, events, and their associations among subsystems is illustrated in Figure 5.

Computer system

The computer system is composed of functional subsystems that represent the various functions performed by the computer. Each of these subsystems fulfills its designated role in the performance of computer workloads. A subsystem can fulfill its role only if it interacts with the other subsystems that support its function. In this way, the input subsystem receives the data to be processed, the storage subsystem stores the data before and after it is processed, the processing subsystem processes the data, and the output subsystem generates the data after processing. This interaction among subsystems also requires the establishment of interfaces between the subsystems to enable them to function in an integrated fashion to process the workload.

In a computer system, the mechanisms available for supporting organizational activities bear little resemblance to their counterparts in the organization system. Even with achieved high-level software facilities, there is little well-conceived strategy for implementation of computerized processing in support of organizational activities. For example, the data definition and data manipulation facilities of data base management systems do _not_ clearly suggest how a systems analyst might apply them to solving his problem. At least, the structured programming strategy guides the programmer toward a programming language representation of an algorithm. However, the systems analyst has little guidance in formulating the data structure and manipulation appropriate to his problem.

The gap between the organization and computer systems suggests the need for a conceptual bridge between these two systems. Such a bridge would guide and structure a systems analyst's activities while he formulates his approach to an organizational requirement. Such a bridge would _not_ remove the necessity for the analyst to be familiar with the application domain with which he is dealing. Instead, the conceptual link identifies the concepts common to all applications of management information systems in order to supplement specific knowledge of organization and computer systems.

Information system model

The conceptual link is an information system model that provides a standard that enables organization system concepts to be expressed

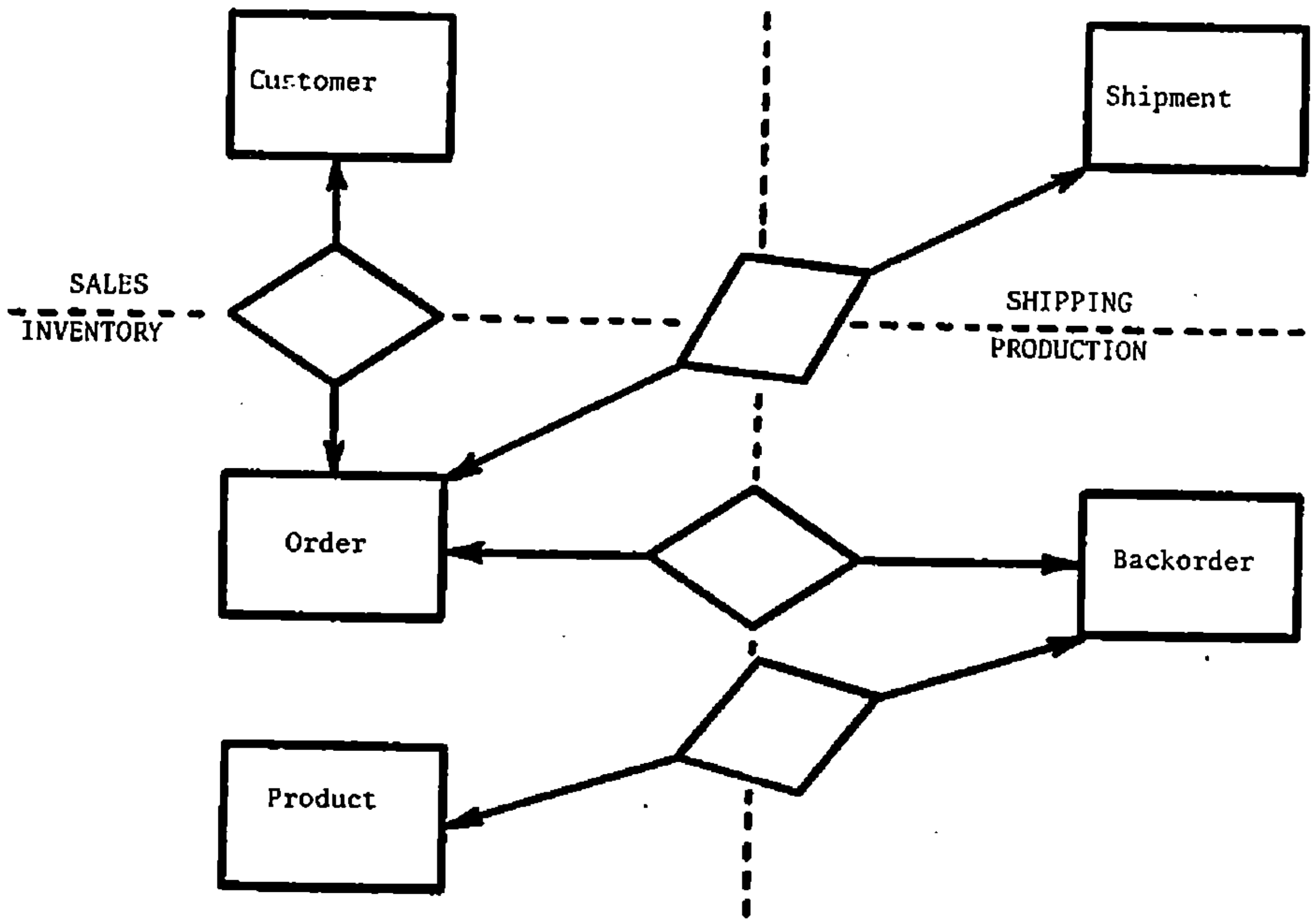

Figure 5. The Role of Persons, Objects, Events, and their Associations in Subsystem Interaction

in a conceptual framework that is also compatible with computer system concepts. The information system model is itself a system composed of interacting subsystems:

1. Input subsystem
2. Output subsystem
3. Data base subsystem
4. Process subsystem.

The correspondence to the various subsystems of the computer system is clear and this is no surprise. Correspondence to the elements of the organizational subsystem can be established. The elements of the output subsystem correspond to the actions and decisions performed by each functional subsystem. The elements of the process subsystem correspond to the procedures and models used to perform each action or decision. The elements of the input subsystem correspond to the data received from the environment by the elements of the process subsystem to generate the elements of the output subsystem. Finally, the elements of the data base subsystem correspond to the data received from another process by an element of the process subsystem to generate the elements of the output subsystem.

Data base subsystem

The data base subsystem serves as a decoupling mechanism between the input and output subsystems. The input subsystem gathers the data from the environment to be used to generate information to the environment through the output subsystem. However, the output subsystem does not necessarily generate information at the same time nor at the same rate as the input subsystem receives data. Therefore, the data base subsystem is an inventory of data resources. Furthermore, the output subsystem does not necessarily request information in a format that is identical with that of the data used to generate the desired information. Hence, the data base subsystem maintains a standard specification for data resources in order to decouple the incompatibilities between the input and output subsystems. The decoupling role of the data base subsystem in these respects motivates the residence of the data base subsystem in the storage subsystem of a computer system.

With respect to the organization system, the data base subsystem also functions as a decoupling mechanism. The various functional subsystems of an organization system are interacting subsystems that must communicate with one another to achieve the desired synergistic effect. Again, the data base subsystem serves as both an inventory and as a standard for the data resources that are generated by any functional subsystem and can be used by any other functional subsystem in pursuit of that subsystem's objectives. Similarly, the data base subsystem also decouples separate procedures and models within a single subsystem. However, it is the data base subsystem's role as a decoupling mechanism between functional subsystems that elevates it to its central role in an integrated information system.

In its role as a decoupling mechanism, the data base subsystem should therefore contain representations of the persons, objects, and events of interest to organizational activities. The elements of the data base subsystem that represent these persons, objects, and events are called entities. Furthermore, the data base subsystem should also contain representations of the relevant associations among the organizational persons, objects, and events. The elements of the data base subsystem that represent these associations are called relationships among the corresponding entities. The concepts of entity and relationship for data base definition have been proposed by both Teichroew [3] and Chen [4].

Process subsystem

In order to function effectively as a decoupling mechanism, the data base subsystem must maintain the data resources so that they accurately reflect the state of the organizational environment. Only in this way can the process subsystem use the data resources to effect the appropriate organizational decisions and actions. Therefore, the process subsystem functions as a data base control mechanism that insures that the data base subsystem conforms to the standard established by the environment. Then, the update function of the process subsystem senses the state of the data base for comparison with the state of the environment. If the data base is not in conformance, the update function invokes a control mechanism to modify the data base to conform to the state of the environment.

In the same way that the elements of the organization system motivate the data base subsystem of the information system, the functional subsystems motivate the elements of the process subsystem of the information system. The process subsystem consists of processes that perform functions in support of organizational activities. Each functional subsystem is factored into a collection of processes that perform the activities of the subsystem. This decomposition may motivate additional entities and relationships to promote inter-process communication. For example, decomposition of the Production subsystem into Scheduling and Requirements Generation processes reveals Component entities that are related to Part entities by virtue of the fact that components consist of parts. Furthermore, the Product entities themselves are related to the Component entities by virtue of the fact that products consist of components. The role of entities and relationships in process interaction is illustrated in Figure 6.

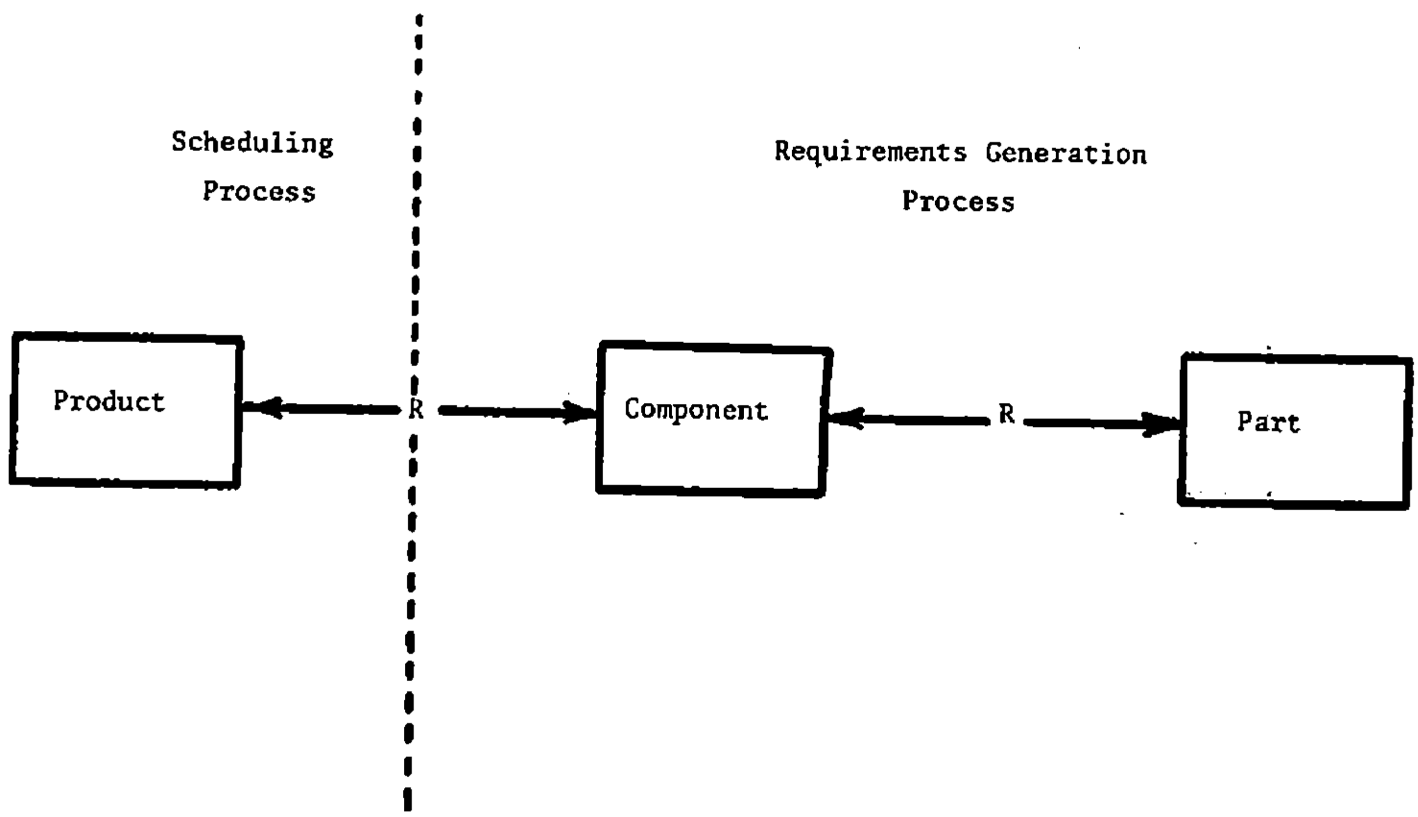

Figure 6. The Role of Entities and Relationships in Process Interaction

LEVELS OF ABSTRACTION

Our approach to systems analysis employs levels of abstraction for inferring data processing requirements from the organization system in preparation for implementation in the computer system. The data base designer is aided by levels of abstraction that enable him to cross from the organization to the computer system in incremental steps rather than in a single jump. The highest level uses concepts that relate to the organization system and the lowest level uses concepts that relate to the computer system while the middle level uses concepts that relate to the information system. This approach is consistent with the structured programming [5] approach to program design that employs step-wise decomposition and restricted control structures. For systems analysis, we employ step-wise decomposition for inferring processing requirements and restricted data structures for inferring data requirements. The relevant concepts at each level of abstraction and their correspondence to the organization, information, and computer systems are listed in Tables 1 and 2.

Table 1
Levels of Abstraction for Processing Requirements

Concept	System
1. Functional subsystem	Organization
2. Process	Information
3. Program module	Computer

Table 2
Levels of Abstraction for Data Requirements

Concept	System
1. Person, object, event, association	Organization
2. Entity, relationship	Information
3. Logical record, tree, network (Set, owner, member, parent, child)	Computer (Data base management system)

Summary

At the highest level, initial decomposition of the organization system reveals functional subsystems that represent the various functions performed by the organization. The activities of each subsystem motivate interest in selected persons, objects, events, and their relevant associations in order to co-ordinate subsystem interaction.

At the intermediate level, the initially identified persons, objects, and events motivate their representation as entities in the data base subsystem of the information system. Furthermore, the relevant associations among the various persons, objects, and events motivate their representation as relationships in the data base subsystem. Additionally, further decomposition of the functional subsystems of the organization system identifies processes in the process subsystem of the information system. This decomposition may motivate additional entities and relationships to promote inter-process communication.

At the lowest level, the data definition requirements specified by the identified entities and relationships and the data manipulation requirements specified by the identified processes form the basis for design of the logical program module and data structure to be implemented in the computer system. An algorithm for logical design of program modules from data manipulation requirements has been developed by Ho [6]. An algorithm for logical design of data structure from data definition requirements has been developed by Blosser [7].

REFERENCES

1. Davis, G. B. Management Information Systems: Conceptual Foundations, Structures, and Development, McGraw-Hill (1974).

2. Ho, T. I. M. Systems analysis perspectives. Proc. 14th Annual Conference on Computer Personnel Research (July 1976).

3. Teichroew, D. et al. An introduction to PSL/PSA. University of Michigan, Department of Industrial and Operations Engineering, ISDOS Working Paper No. 86 (March 1974).

4. Chen, P. P. S. The entity-relationship model: Toward a unified view of data. Trans. Database Systems 1, 1 (March 1976), 9-36.

5. Wirth, N. On the composition of well-structured programs. Computing Surveys 6,4 (December 1974), 247-259.

6. Ho, T. I. M. Toward a formal theory for the requirements statement, analysis, and design of information systems. Ph.D. dissertation, Purdue University (December 1974).

7. Blosser, P. A. An automatic system for application software generation and portability. Ph.D. dissertation, Purdue University (May 1976).