1975

A Teaching Aid to Simulate the Requirements Statement Process

Thomas I. M. Ho

Benn Konsynski

J. F. Nunamaker

Report Number:
75-142

A TEACHING AID TO SIMULATE THE REQUIREMENTS STATEMENT PROCESS*

Thomas I. M. Ho**
Benn Konsynski***
J. F. Nunamaker***

CSD-TR 142

*Presented at Sixth Annual Pittsburgh Conference on Modeling and Simulation, April 24-25, 1975
**Computer Sciences Department and School of Industrial Management, Purdue University, West Lafayette, IN 47907
***College of Business and Public Administration and Department of Computer Science, University of Arizona, Tucson, AZ 85721

A TEACHING AID TO SIMULATE THE REQUIREMENTS STATEMENT PROCESS

Thomas I. M. Ho[1], Benn Konsynski[2], and J. F. Nunamaker[2]

[1]Computer Sciences Department and
School of Industrial Management
Purdue University
West Lafayette, IN 47907

[2]College of Business and
Public Administration and
Department of Computer Science
University of Arizona
Tucson, AZ 85721

ABSTRACT

A model of a teaching aid designed to educate the student in the problems of data collection for an information system design effort is presented. A data base of case facts and a limited query facility are utilized to simulate some of the behavioral and technical problems which arise in attempting to capture the system requirements.

INFORMATION SYSTEMS EDUCATION

Information processing in business organizations is the most common application of computer-based information systems. These information systems employ the greatest number of computer professionals and are expected to require substantial numbers of professionals in the foreseeable future.

To meet this need, high-quality information systems education programs will be required. Recommendations of the ACM Curriculum Committee on Computer Education for Management[1 & 2] represent the combined wisdom of academic and industrial personnel engaged in the training and practice of information systems analysis and design.

SYSTEM DEVELOPMENT CYCLE

The conceptual framework established for the training and practice of systems analysis and design is the System Development Cycle (SDC). Defined as an evolutionary procedure for the development of information systems, the System Development Cycle consists of a sequence of well-defined phases for the determination of system requirements and for the design and implementation of a system fulfilling those requirements. Teichroew[3] divides the SDC into the following phases:

1. Perception of need.
2. Determination and statement of requirements.
3. System design.
4. System construction and testing.

SYSTEM ANALYSIS: REQUIREMENTS DETERMINATION AND STATEMENT

The relative location of system analysis in the SDC is indicative of the primary importance of system analysis in the development of information systems. The information system interfaces two dissimilar systems: the organizational system, consisting of the personnel and structural components of the organization whose information requirements are served by the information system, and the computerized system, consisting of the hardware and software components that implement the information system. Therefore, the objective of system analysis is the determination and statement of the information requirements of the organization in preparation for the design of the computerized system that satisfies the stated requirements.

Requirements Determination

The determination of information requirements is approached by analysis of the data currently used by the existing information system. This analysis involves study of the structure of the organizational system and of the flow of information among the components of the organizational system.

Requirements Statement

Concurrent with requirements determination, statement of requirements proceeds in top-down fashion leading to an evolutionary understanding of the system. A requirements statement consists of a complete description of the following characteristics of an information system:

1. Environment.
2. Data definition.
3. Processing and logic.
4. Timing.
5. Volume.

The description of the environment includes the composition and structure of the organizational system. Data definition includes description of the inputs, outputs, and files used by the organization. Processing and logic definition involves the data manipulation and decision rules that transform the inputs into the desired outputs and updated files. Finally, the description of timing and volume specifies the performance and workload requirements of the information system.

SYSTEM DESIGN

Upon the completion of a consistent statement of requirements, system development proceeds with the

design of program module and data base structure and the selection of hardware. Characterized as an iterative process, the design phase generates and evaluates alternative configurations that are feasible with respect to constraints imposed by the requirements. System design is completed with the selection of an alternative that is optimal with respect to an objective function defined by the requirements.

A COURSE IN SYSTEM ANALYSIS AND DESIGN

A course in system analysis and design encompasses all the concepts described herein. Using the framework of the System Development Cycle, the course teaches a methodology for the analysis and design of an information system that satisfies a well-defined set of requirements. Playing a central role in the development of an information system, the requirements are defined during system analysis for use during system design. Therefore, system requirements are emphasized throughout the course.

Unfortunately, the determination and statement of requirements are among the most difficult tasks of system development. Since there exists no model for the interface between the organizational system and the information system, there is no defined procedure for inferring the requirements from the information and decision-making needs of an organization. The matter is further complicated by the complexity and variety of organizations composed of diverse interacting functional areas. Therefore, a total systems approach is required to insure the integration of essential applications competing for the organization's limited resources.

A strategy exists for organized study of the existing information system applications in preparation for the determination and statement of requirements for a new information system. For example, techniques like the Study Organization Plan developed by IBM[4] provide a framework for description of an existing information system in terms of the organizational structure and its activities and of the computer operations, inputs, outputs, and files. More advanced approaches like the top-down strategy proposed by the Information Systems Design and Optimization System (ISDOS) Project[5] provide a framework for requirements determination and statement in an evolutionary manner.

With only general guidelines available, all approaches to system analysis rely upon a procedure that incrementally leads to a broader and deeper understanding of the system under scrutiny. Lacking a well-defined methodology, such a procedure is a slow, painstaking task that is best learned through actual experience.

In conjunction with a course in system analysis and design, actual experience is usually provided by a student project involving the analysis and design of an information system for an organization. This organization is either an actual operating company or an imaginary one described in a case study.

Although possessing the obvious quality of realism for student training, the environment of an actual functioning organization is hardly the place for beginners to gain much-needed experience. Aside from the disruptive effect that training activities would have upon the functioning of the organization, real problems lack the controlled behavior so essential to the lessening of distraction and to the reinforcement of ideas valuable for education.

In order to maintain control of problems from which students will learn concepts and principles, educators have produced case studies representing some compromise between the complexity of actual problems and the simplicity of trivial exercises. In general, educational experience with case studies has been satisfactory except for one notable exception.

Whether intended for the teaching of industrial management principles or information system analysis and design, case studies do not develop the student's ability to collect data that will enable him to assess a situation and to recognize apparent difficulties. Essential to the success of a system study, data collection gradually uncovers the layers of a problem. Complex situations cannot be perceived as a whole by the limited capacity of the human mind and its sensory receptors. Instead, one can only see part of the problem at a time and therefore, dealing with complex problems involves the partitioning of a problem into manageable subsets that can be attacked in organized fashion.

Therefore, information systems education can be enhanced by opportunities to deal with realistic situations. In academic surroundings, these opportunities can be made available with a teaching aid that simulates the environment encountered during the analysis and design of a complex information system.

PEDAGOGICAL AIDS FOR SYSTEM ANALYSIS TRAINING

Analysis and design of an information system rely upon an essential understanding of the system requirements. The analyst can only achieve this understanding through a meticulous data collection effort involving all aspects of the system under study. To provide the student analyst with relevant experience in performing a system study, a pedagogical tool can serve as the respondent to the student.

Consisting of a data base and a query facility, the tool simulates the dialogue that typically occurs between the analyst and the system user whose requirements are to be fulfilled. The data base contains the requirements of an information system for a hypothetical company described by Nunamaker and Konsynski[6]. The query facility enables the student to question the requirements data base. In this way, the student learns what questions must be asked in order to determine a complete and consistent set of user requirements.

First, the student queries the data base to gain insight into the overall requirements of the system:

1. Organizational structure of the environment.
2. Physical structure of the environment.
3. Document structure of the environment.

The organizational structure describes the departments composing the organization served by the information system. The physical structure describes the entities, e.g., employees and products, that exist in the environment. The document structure describes the inputs and outputs that flow among the departments and that describe the entities. Then, by querying the data base, the student infers the flow of documents and entities in the organizational structure.

Finally, the student queries the data base to extract the detailed requirements of the system. These requirements include detailed specifications for the inputs, outputs, and files to be handled by the system and for the processing and logic to be performed in order to transform the inputs and files into the desired outputs. Also, the requirements include the timing and volume of the various inputs, outputs, files, and processes that have been defined.

Requirements statement is facilitated by the use of a Requirements Statement Language (RSL), a high-level language for describing the requirements of an information system without necessitating statement of the procedures for implementing the requirements. An example of a primitive RSL is Accurately Defined Systems (ADS), developed by the National Cash Register Company[7]. An advanced RSL is the Problem Statement Language (PSL) developed by the ISDOS Project[8] at the University of Michigan with affiliates at Purdue University and the University of Arizona. Both these pedagogical tools are used for requirements statement by students enrolled in the systems analysis and design course.

Using an RSL, the student describes the system requirements according to the facts he learns by querying the requirements data base. The query facility is designed to provide only limited information in response to each query. This limited response compels the student to organize his query strategy in order to collect a complete set of requirements. An RSL promotes completeness by providing an organized framework for requirements statement that makes an omission apparent. In general, requirements statement promotes a discipline for determining what information is relevant and how the relevant requirements relate to one another.

IMPLEMENTATION OF A PEDAGOGICAL REQUIREMENTS DATA BASE AND QUERY FACILITY

The vehicle for implementation of the requirements data base and its query facility is the Problem Statement Analyzer for PSL (PSA/PSL) developed by the ISDOS Project[9]. PSA/PSL is a software package that maintains a data base of requirements that have been defined in PSL for subsequent logical analysis and reporting. For the pedagogical use described in this paper, the report capabilities of PSA/PSL are applicable.

Problem Statement Language

Requirements statement includes the identification and naming of various types of objects and of the relationships among those objects. In PSL, those objects which produce, store, or use the information from the system are called Real World Entities. The physical units which transport data between the Real World Entities and the information system are called Inputs and Outputs. A collection of information that is maintained within the information system is called an Entity. Structure is imposed upon the data requirements by the definition of Sets consisting of Inputs, Outputs, or Entities.

Elementary data definition is accomplished through the definition of an Element, the lowest level data object, or a Group, a collection of Elements and/or other Groups. Groups and Elements are themselves contained in Inputs, Outputs, and Entities.

Data processing is defined by the description of Processes which receive Inputs in order to update Sets and to generate Outputs. The dynamic behavior of the system is described by the specification of information flow among the various Real World Entities, Processes, and Sets and by the specification of the Conditions which trigger the Events that govern the information flow. Finally, the size of the information system is specified by defining System Parameters. See Figure 1.

Problem Statement Analyzer

PSA/PSL outputs report the requirements defined in PSL. Armed with a limited subset of the PSA/PSL command language, the student analyst may request a variety of reports relevant to his determination and statement of requirements.

A number of directory and dictionary reports are produced by PSA/PSL to summarize the data element names defined in the requirements. In particular, the NAME-GEN and NAME-LIST commands print a list of data names and their respective types. This list can then be used as an index or directory for various other PSA/PSL reports. The DICTIONARY command prints a report containing the description of each data name that has been defined. The DICTIONARY report explains the meaning of each data name and reinforces the concept of a data dictionary essential for co-ordination of the development of an information system.

The structural requirements are described in a number of PSA/PSL reports of the hierarchical relationships among the defined objects. The CONTENTS command lists the Entities, Inputs, or Outputs contained in a Set, and the Groups and Elements contained in an Entity, Input, Output, or another Group. This same information may be obtained in graphical form with the CONSISTS MATRIX command. Finally, the data names that have been specified as Entity identifiers are generated by the ENTITY-IDENTIFIER command.

Information flow is displayed in different formats by various PSA/PSL commands. Description of the

flow of Sets, Inputs, Outputs, Entities, Groups, and Elements among the Real World Entities and Processes can be requested in graphical, matrix, or narrative format. The PICTURE command displays information flow in graphical flow chart format. The DATA-PROCESS command produces the Data Process Interaction Matrix indicating which data objects are either inputs, outputs, or updates of each process and the Process Interaction Matrix indicating the precedence relationships among the processes. Finally, the PROCESS-INPUT-OUTPUT command summarizes the inputs, outputs, and updates of each process in narrative form.

Timing requirements are provided by several PSA/PSL dynamic analysis commands. The FREQUENCY command presents the frequency of occurrence of each Input, Output, or Process in a specified time interval. Other timing reports are expected to be useful as they become available.

Size and volume requirements can be obtained by specifying several planned PSA/PSL commands. It is expected that these commands will enable the student analyst to determine the size of Sets of Inputs, Outputs, or Entities and the volume of processing in terms of the frequency of Processes.

Experience with the selected subset of PSA/PSL commands will indicate any additions or deletions that may be necessary to provide a satisfactory query facility. Hence, a monitoring facility is planned for logging the queries requested by each student analyst. Such a profile will indicate the usefulness of each command in addition to enabling evaluation of each student's strategy and progress. The need for a command that is not available with PSA/PSL can be filled by using the report generator facility of PSA/PSL[10].

Further improvements include the introduction of incompleteness and inconsistency into the requirements data base in order to test the student analyst's ability to recognize these deficiencies during his system analysis. An advanced long-range enhancement is the introduction of a random factor that suppresses a portion of the output produced in response to a query. Such an enhancement will promote realism by demonstrating the forgetfulness, resistance, or incomplete knowledge that an analyst often encounters during system analysis.

CONCLUSION

It is expected that the application of pedagogical tools to information systems education will provide students with valuable experience in preparation for real-world design problems. In addition, information systems educators will be provided with a useful facility for monitoring student development and a progressive environment for educational experimentation.

REFERENCES

(1) Ashenhurst, R. L., ed. Curriculum recommendations for graduate professional programs in information systems. Comm. ACM 15, 5 (May, 1972), 363-398.

(2) Gouger, J. D., ed. Curriculum recommendations for undergraduate programs in information systems. Comm. ACM 16, 12 (December, 1973), 727-749.

(3) Teichroew, D., ed. Education related to the use of computers in organizations. Comm. ACM 14, 9 (September, 1971), 573-588.

(4) IBM Corporation. Study Organization Plan Documentation Techniques. Technical Report No. C20-8075-0 (1963).

(5) Berg, D. and Teichroew, D. An example of the use of PSL/II using top-down analysis. ISDOS Working Paper No. 74, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor (March, 1974).

(6) Nunamaker, J. F. and Konsynski, B. Company Z: an information systems design problem. College of Business and Public Administration, University of Arizona, Tucson (January 2, 1975).

(7) National Cash Register Company. Accurately Defined Systems. Dayton, Ohio (1967).

(8) Teichroew, D.; Hershey, E. A.; and Bastarache, M. An introduction to PSL/PSA. ISDOS Working Paper No. 86, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor (March, 1974).

(9) Teichroew, D. and Bastarache, M. PSA outputs. ISDOS Working Paper No. 99, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor (October, 1974).

(10) Dissen, R. L. Problem Statement Analyzer report generation facilities. ISDOS Working Paper No. 100, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor (October, 1974).

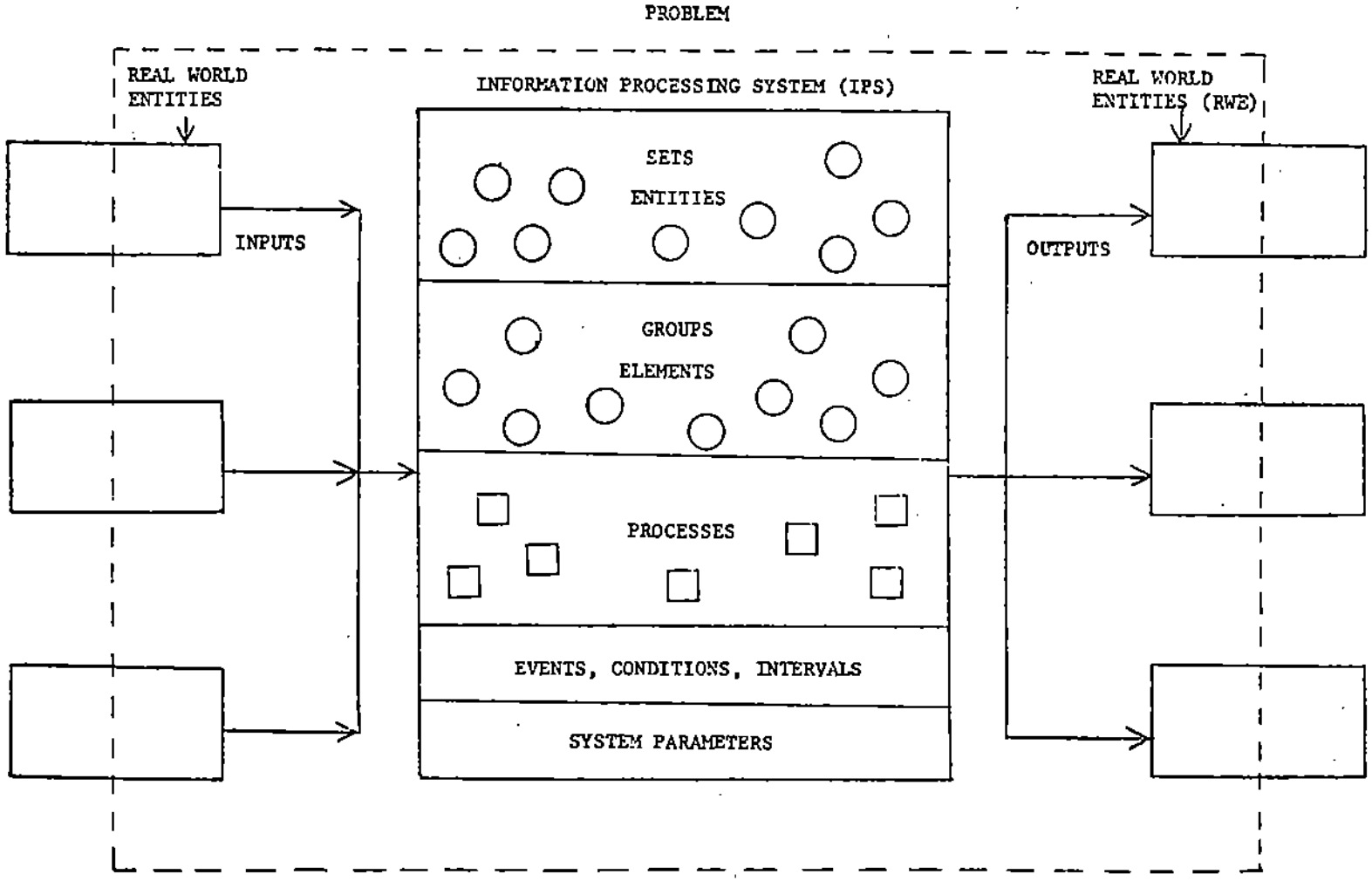

Figure 1 Model of the Target Systems being described in the Problem Statement Language showing the object being described.