Technical Progress Report

Summary

MASTER

The development work in the ISDOS Project at The University of Michigan continued during 1975, particularly in improvement in the formal language, Problem Statement Language (PSL), and a software system, Problem Statement Analyzer (PSA). The system as installed at the Michigan Computing Center, was made available to ERDA and has been used for demonstration and instruction. User manuals and training materials have been supplied and ERDA representatives have attended a PSL/PSA training session and Project Review meeting.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

INFORMATION PROCESSING SYSTEMS ANALYSIS AND DESIGN

by

Daniel Teichroew

Progress Report

for Period January 1, 1975 - September 30, 1975

ERDA Special Research Support Agreement

E(11-1)-2544

ISDOS Research Project
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48104
313/763-3469

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed or represents that its use would not infringe privately owned rights.

The ISDOS Project at The University of Michigan is developing a structured computer-aided methodology of stating, recording, and analyzing requirements for Information Processing Systems. The methodology includes a formal language, Problem Statement Language (PSL), and a software system, Problem Statement Analyzer (PSA). The development work continued during 1975, particularly in improvement in PSL/PSA. The system is now operational on a number of large computers and is used by a number of public and private organizations. The attached report (Information Processing Systems Analysis and Design, ISDOS Working Paper No. 128) describes the major features of the language and the system.

The system as installed at the Michigan Computing Center, was made available to ERDA and has been used for demonstration and instruction. ERDA plans to make more extensive use of the system during the remainder of 1975 and 1976. User manuals and training materials have been supplied and ERDA representatives have attended a PSL/PSA training session and Project Review meeting.

COMPUTER-AIDED, STRUCTURED DOCUMENTATION
OF INFORMATION PROCESSING SYSTEM REQUIREMENTS

by

Daniel Teichroew

ISDOS Working Paper No. 128

September, 1975

ISDOS Research Project
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48104
313/763-3469

This is a working paper and should not be quoted or reproduced in whole or in part without the written consent of the ISDOS Project. Comments are solicited and should be addressed to the ISDOS Project.

ABSTRACT

The ISDOS Project at The University of Michigan has developed a structured computer-aided methodology of stating, recording, and analyzing requirements for Information Processing Systems. The methodology includes a formal language, Problem Statement Language, and a software system, Problem Statement Analyzer. The system is operational on a number of large computers and is used by a number of public and private organizations. This report describes the major features of the language and the system.

1. Introduction

The ISDOS Research Project is being conducted by faculty and students in the Department of Industrial and Operations Engineering at The University of Michigan. The objective of the Project is to study the process by which Information Processing Systems are being built and operated. The term Information Processing System is used to mean a collection of hardware, hard software, programs, files and procedures which have been assembled to accomplish some requirements. Usually the basic requirements are to produce a number of outputs (reports, documents, messages, displays, etc.). Examples of such systems are various business data processing applications, information storage and retrieval systems, management information systems, command and control systems, etc. The systems may include batch, remote job entry, on-line, interactive, real-time, etc., facilities or some combination. They may utilize data base management systems and communication systems. Most large organizations have a number of such Information Processing Systems frequently sharing hardware and files and occasionally also software.

The long term object of the ISDOS Project is to develop methodologies for automating as much as possible the process of building systems. Shorter term objectives are (1) to improve the present (primarily manual) systems building process and (2) to prove the feasibility by providing computer-aided methods.

One fundamental concept of the Project is that the basic data in building a system, namely the description of the requirements, should be recorded, as early in the process as possible, in machine readable form. Thereafter, the building of the system

(construction of programs, files, scheduling of jobs, etc.) is to be accomplished with the aid of the computer itself using algorithms currently being developed or to be developed. These algorithms would analyze the requirements and aid in the construction of programs and files using operations research techniques to develop first feasible and then, eventually, optimal solutions in accordance with a stated performance criterion.

Computer-aided systems building will result in a number of benefits. Perhaps the most important is flexibility - a change in a requirement can more easily be incorporated into the system. Another major benefit will be the increase in productivity of systems analysts, designers and programmers since they will be concerned with the development and maintenance of the partially-automated system which will then be able to produce software to satisfy the user requirements. The computer, in effect, will be used to amplify the capability of analysts and designers. Even without implementation, the formalization of the design process forms the basis for education and research.

2. The ISDOS System

An overview of the ISDOS system as initially conceived in 1968 is given in [3]. This paper discusses the system building process and outlines the major components of a proposed "ideal" (completely automated) system.

The basic approach in the ISDOS Project is to focus on the system building process as an organizational activity similar to many other activities in that it depends very heavily on information flow and data processing. Furthermore, it is a process which is now primarily manual - the only automated part being the compilation of source

language statements into object code. The approach being followed in developing the ISDOS system is therefore very similar to that followed in automating other information systems:

- Study and describe the "present system."
- Improve the present system wherever possible.
- Propose a "new system" which makes use of the computer and operations research methodology whenever technically feasible and economically workable.
- Divide the proposed system into modules and develop a plan for phased development, testing, and installation of modules.
- Develop, test, and install the modules.

In the development of the proposed system the "data base" approach is being followed. The steps are:

1. Identify what information is, or should be, recorded.
2. Develop language for expressing this information.
3. Provide system for storing information in a computerized data base.
4. Provide capability for displaying with appropriate rearrangement, etc.
5. Provide capability for checking for consistency, completeness, etc.
6. Provide capability for analysis and evaluation.
7. Provide decision-making aids.

The goal is to eventually replace the current manual documentation methods in which the necessary information is stored in a computerized data base and is available for use by the various individuals who need access to it. The final system is shown schematically in Figure 1.

Project strategy has been to begin at the beginning. The initial module is concerned with the phases (Perception of Need and Logical System Design in Figure 1) of system building in which users

Figure 1 The Data Base in the Information System Life Cycle

requirements are determined, recorded, and analyzed and a "logical" system is designed. This requires the development of a language for stating requirements and software for storing the requirements in computer manipulatable form and for retrieval and analysis.

A number of manual methods for formalizing methods for collecting and recording requirements have been developed. A survey of these methods is given in [4]. The most widely known probably is ADS (Accurately Defined Systems). A similar method which included a computer program for analyzing requirements and aiding in the design of files and programs is TAG (Time Automated Grid) [2]. Neither these nor any of the other methods has been widely used; some of the reasons for this are discussed in the survey article and in [5].

The methodology and the software developed in the ISDOS Project have been designed to overcome the factors which have limited the use of earlier attempts. The Analyzer (Figure 2.) is a software package that performs two main functions:

1. It accepts statements describing an information processing system in a formal language called the Problem Statement Language (PSL), checks them and accumulates the information in a data base.
2. It produces a number of reports containing information for the data base.

The Analyzer is designed to operate in an interactive environment using the facilities of the host operating system. To facilitate the use in different host environments the Analyzer has been designed to be portable. The use of the system has been made as simple as possible and is supported by considerable user documentation.

Figure 2 The Analyzer as Part of an Information Processing System

The Problem Statement Language is briefly described in Section 3 and the Analyzer in Section 4. The use of the methodology is described in Section 5.

3. Problem Statement Language (PSL)

The requirements for a target system are expressed in an unambiguous, machine processable form. PSL has been designed to meet this goal by being able to accurately and completely express all relevant requirements for the logical design of the target system and by having a precise syntax and semantics. The complete collection of requirements for a target system written in PSL is termed a Problem Statement, in that it is a "problem" to be solved during the physical design phase.

A target system is described by listing its subsystems or components and giving their properties and relationships among the components. In PSL each component is called an "object." Each object is given a unique name and must be of one of the allowable types, which are:

ATTRIBUTE	MAILBOX
ATTRIBUTE-VALUE	MEMO
CONDITION	OUTPUT
ELEMENT	PROBLEM-DEFINER
ENTITY	PROCESS
EVENT	RELATION
GROUP	SECURITY
INPUT	SET
INTERFACE	SOURCE
INTERVAL	SUBSETTING-CRITERION
KEYWORD	SYSTEM-PARAMETER

For each object a number of properties can be stated, including the following:

ATTRIBUTES

KEYWORDS

SEE-MEMO

SYNONYMS

DESCRIPTION

SOURCE

SECURITY

Relationships can be stated among two or more objects. There are approximately fifty different relationships that may be used. Which relationships are allowed depends on the type of object involved. The relationships fall into the following classes:

- System Flow
- System Structure
- Data Structure
- Data Derivation
- System Size
- System Dynamics
- System Properties
- Project Management

PSL is a free format language. Statements may be entered either from a terminal or by cards, tape, etc. A number of facilities such as synonyms and abbreviations are available to facilitate terminal use.

4. Structure of The Analyzer (PSA)

A simplified view of the structure of the Analyzer is given in Figure 3.

The PSA software system consists of a set of interrelated modules. The software is intended to be as system independent as possible. In order to achieve this independence, the software, including the data base management system, is written almost entirely in FORTRAN IV and can be installed on any environment which has a FORTRAN IV compiler and sufficient memory.

The main subsystems are a Command Language Interface, Data Base Update Facility, Report Generator Facility and a Library Facility.

The Command Language Interface module interprets commands in the Command Language from the user and causes the execution of the appropriate module to handle that command. The command processing modules fall into two categories: data base update modules and report generation modules. The command processing modules interface with the data base via a data base management system which is part of the Library Facility. This allows a convenient representation of the information contained in the problem statement. A "name selection" module is used to extract a subset of the data base for use with the various report generation modules. The library facilities also perform certain peripheral functions, such as data base initialization, and dump and restore.

5. Use of the Problem Statement Language and Analyzer

The Analyzer data base contains, at all times, the complete, up-to-date version of the problem statement. The PSL statements

Figure 3 Internal Structure of the Analyzer

for the target system can be entered incrementally, and in any order. This allows the requirements for the target system to be entered in the data base, and become part of the Problem Statement as soon as they are known. Facilities are also provided for modification of requirements already stored in the data base during a previous session with the Analyzer. Modifications may be made at three levels: to the objects themselves, to the relations among objects, and to the textual information stored in the data base. Whenever reports are produced from the PSA data base, all information recorded to date, and any modifications which have been made, are taken into account.

The analysis performed by the Analyzer is of two types: analysis during data entry, and analysis during preparation of reports. The latter is based on the current state of the data base.

Analysis performed by the Analyzer, as information is added to the data base, is based on the assumption that the information is being added or modified as it becomes known. The Analyzer checks that the new information does not contradict the information already in the data base. No information is mandatory at this time based on the premise that it will be entered at a later date. There are also checks that the problem definer uses correct syntax, enforcing a consistency of usage, and avoiding the ambiguous definition associated with normal natural language requirements specifications.

Analysis performed by the Analyzer in preparation of reports is of two types: completeness, and summarization. Because the input processing allows incomplete specification of the problem statement at any stage, any potential or actual incompleteness in the Problem

Statement as represented by the data base, is noted in the reports where appropriate. Examples of this are PROCESSES which use data, but do not derive anything, or incomplete data structure definition. Summarization and comparison reports provide many useful tools for the problem definer to catch inconsistencies which the analyzer could not derive, and mismatches between user expectations and implementation. Summarization is also helpful in providing valuable inputs to the physical design process. Examples of this type of analysis range from providing pictures to the user showing all the information he will be required to supply to the target system, and all the information he will be required to supply to the target system, and all the information he will be able to obtain from the target system, to analysis of input-output redundancy and overlap.

The Analyzer is intended to be used in an interactive environment where the problem definer can request reports and provide input via on-line terminals. The Problem Definer controls the action of the Analyzer via command language. Commands fall into three categories: report commands, modifier commands, and control commands. There is a separate command for each possible report available from the Analyzer, and a separate command for each "type" of modification to the data base. There are facilities to allow the problem definer to obtain reports for only selected portions of the data base, (e.g., only those objects for which he is responsible) and facilities to suppress those parts of the reports which are not of interest at this time. During a single session with the Analyzer, a problem definer may make modifications to the data base, generate some analysis, make further modifications based on this analysis, and finally generate summary reports to be used in further work. When there are several problem definers working on the same Problem Statement, they may each interface with

the same data base. Although they may not update concurrently, each session will have available all information from all previous sessions with other problem definers.

6. Status

The documentation for the use of the Problem Statement Language and Analyzer includes the following:

PSL USER'S MANUAL, Teichroew, Daniel and Bastarache, Michel J.,
ISDOS Working Paper No. 98, Preliminary Draft, March 1975;

PROBLEM STATEMENT LANGUAGE VERSION 3.0 LANGUAGE REFERENCE
MANUAL, Hershey, Ernest Allen III; Teichroew, Daniel; Berg,
Douglas L. R.; Winters, Edward; Bastarache, Michel J., ISDOS
Working Paper No. 68, April 1974 ;

PSA OUTPUTS, Teichroew, Daniel and Bastarache, Michel J.,
ISDOS Working Paper No. 99, Preliminary Draft, October 1974 ;

PROBLEM STATEMENT ANALYZER COMMAND DESCRIPTIONS, Dissen, Rick;
Hershey, Ernest Allen III, and Bastarache, Michel J.,
ISDOS Working Paper No. 91, January 1975.

Version A2.0 of the Analyzer has been operational at the University of Michigan since July 1974. Previous versions of PSA have been operational since 1971. During this time the Analyzer has been used for several classroom projects and for the description of several proposed systems in user organizations. The Analyzer is also installed at several user organizations where it is either being tested, or phased into operation. It is operational in IBM 360/370 equipment under MTS at The University of Michigan, OS batch environment, TSO, and CP/CMS. It is also operational on UNIVAC 1100 series under EXEC 8, and CDC 6000

series equipment under MACE like systems and being installed on the Burroughs 6700. Further development is underway at The University of Michigan to improve and extend the system.

References

1. ADS

National Cash Register Company, "Accurately Defined Systems", 1967 ;

Lynch, H. J., "ADS: A Technique in System Documentation", DATABASE, Vol. 1, No. 1, Spring 1969, pp 6-18 .

2. TAG

Myers, D. H., "A Time Automated Technique for the Design of Information Systems", IBM Systems Research Institute New York, 1962, 50 pp;

IBM, "The Time Automated Grid System(TAG): Sales and Systems Guide", Publication No. Y20-0358-1 (No date approx 1968) 40 pp (Reprinted in J. F. Kelly Computerized Management Information Systems Macmillan 1970 pp. 367-400).

3. Teichroew, D., Sayani, H., "Automation of System Building", DATAMATION, Aug. 15, 1971, pp. 25-30.

4. Teichroew, D., "A Survey of Languages for Stating Requirements for Computer Based Information Systems", Proc. FJCC, 1972, pp 1203-1224.

5. Teichroew, D., "Improvements in the System Life Cycle", INFORMATION PROCESSING 74 -- North-Holland Publishing Company 1974.