HANDBOOK OF INDUSTRIAL ENGINEERING

Edited by
GAVRIEL SALVENDY
Purdue University

1807 1982

A Wiley-Interscience Publication
JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto • Singapore

Library of Congress Cataloging in Publication Data:

Main entry under title:

Handbook of industrial engineering.

"A Wiley-Interscience publication."
Includes index.
1. Industrial engineering—Handbooks, manuals, etc.
I. Salvendy, Gavriel, 1938–
T56.23.H36 658.5 81-23059
ISBN 0-471-05841-6 AACR2

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Virgil
Profe
Depar
Purdu
West

Thom
Assist
Hyg
The U
Ann

Jeffre
Assist
Depar
Orego
Corval

Guy J.
Mana
Indust
Intern
Chicag

Andre
Profes
Unive
Minn

Mosh
Profe
Scho
Purd
West

Fran
Priva
Ann

CONTENTS

14 OPTIMIZATION IN INDUSTRIAL ENGINEERING

CHAPTER 12.7

Office Automation

ANDREW D. BAILEY, JR.

University of Minnesota

JAMES GERLACH, R. PRESTON McAFEE, ANDREW B. WHINSTON

Purdue University

12.7.1 INTRODUCTION

Office automation is an evolving trend centered around the automation of office procedures and document manipulation by means of computer technology. The purpose of Section 12.7.2 is to acquaint the reader with the generally accepted concepts and capabilities of office information systems (OISs), the issues surrounding their development, a real-life prototype (Citibank's Management Work Stations), and some of the implications that office automation will have for business and society.

Section 12.7.3 presents two research OIS prototypes: Officetalk-Zero® and System for Computerization of Office Processes (SCOOP). Officetalk-Zero is an electronic aid to performing routine office tasks, similar in manner to the traditional way of performing the equivalent clerical tasks; SCOOP stresses the automation not only of office devices, but also of office procedures. The need for nontechnical office personnel to implement automated office procedures leads to a discussion of the office worker-machine interface in Section 12.7.4. Two potential languages for describing automated office procedures are presented: the Office Procedures by Example language and the Business Definition Language. Each language is examined according to design philosophy, objectives, prospective uses, and limitations.

Automated office procedures and devices are effective means by which the office worker can manipulate, store, and send electronic documents. Such capabilities should be adequate for supporting formal communications. However, informal communications, serving important organizational needs, may not be adequately accounted for by the document-oriented OIS. Thus Section 12.7.4 concludes with a discussion of formal and informal communications in an automated office setting.

Section 12.7.5 deals with the modeling of OISs in terms of formal office activities and formal lines of communication. A model for describing and analyzing information flows within offices is presented. The model lends itself to automatic analysis and transformations intended to automate, streamline, and reorganize the OIS model. The outlook for OIS simulation as an analytic tool and a personnel training tool follows. In addition to office efficiency and productivity, the office designer and manager must incorporate internal control mechanisms in the automated office. A computer-assisted internal control verification system, TICOM II, is introduced as an effective means of analyzing OIS models for violations of sound internal control. Finally, Section 12.7.6 treats the problem of preparing for the implementation of OISs. The chapter ends with some thoughts on the implications of OISs for future office activities.

This project was funded in part by a grant from the Peat, Marwick, Mitchell Foundation through its research opportunities in auditing program. The views expressed herein are those of the authors and do not necessarily reflect the views of the Peat, Marwick, Mitchell Foundation. The research was supported in part by Army Contract No. DA79C0154. We gratefully acknowledge the fellowship support made possible by a grant from IBM.

12.7.2 OFFICE INFORMATION SYSTEMS IN PERSPECTIVE

With the introduction of the computer in the business environment, business functions such as inventory control, accounts receivable, accounts payable, and payroll accounting have been successfully automated. The benefits have included faster and more accurate reporting, more effective storage and management of large amounts of data, and greater reliability and consistency of system performance. However, despite the merits of automation, traditional office work has remained relatively unchanged. The goal of the automated OIS is to facilitate office workers' performance of their routine duties by providing them a totally office-oriented system that exploits the processing speed, precision, and generally infallible memory of computing machines.

The OIS is differentiated from an information processing system (IPS), a word processor (WP), or a decision support system (DSS) by its emphasis on the integration of all of the components necessary to support office personnel. The IPS, WP, and DSS are either not oriented toward office functions or address only a limited set of office activities. The basic functions of office work include text editing, document preparation and filing, copying, simple numerical processing, communication, and information analysis and verification. With respect to office procedures, a WP and overlapping areas of an IPS and DSS are only isolated elements or subcomponents of an OIS. The primary advantage of an OIS over a system in which each of the basic office functions is automated independently is the reduction in the complexity of the interface between the office personnel and the machine. Unless this interface is kept simple and is easy to learn and use, an OIS may be of little value except to the office specialist.

A second benefit obtained from an OIS as compared to a WP, IPS, or DSS is the result of the office structure the OIS is expected to automate. An office organization is designed to accommodate office workers performing highly interactive, autonomous, and concurrent business tasks. Fixed programs are inflexible, and simplistic interactive systems are deficient in meeting the needs of the office worker as required by the dynamic office environment.

General OIS Capabilities

As mentioned previously, an OIS is an integrated computer system comprising facilities for performing office work. In particular, office automation aids office personnel in the preparation of documents, information management, and decision making.[1] The types of documents to be prepared, processed, and managed by an OIS are memos, letters, reports, and typical business forms such as requisition and purchase order forms. The documents, still visually depicted in their regular format, are stored electronically in an OIS. The methods of capturing the information provided to the office workers produce documents with the standard headings and lettering appropriate to each particular form.

The office worker interacts with the OIS through work stations. A work station is a programmable microcomputer equipped with office-oriented devices and software. The work station will enable the office worker to see and manipulate the electronic documents, communicate electronically with other work stations, access data base information, and stepwise direct the work station to perform designated tasks on a one-time basis or automatically. Such a one-time task might be preparing a memo incorporating only sections of a previous memo and a previously prepared summary report.

An example of a task performed automatically by the work station might be searching an inventory data base to determine whether items listed on any purchase requisition are in stock or on order or whether they need to be special ordered. The work station operator could then fill out a blank purchase order, using the information found on the requisition form and in the inventory data base. Additional accesses to other system information could be made as the situation warranted. The completed purchase order form could then be transferred to another work station determined by the nature of further processing, such as managerial approval of the purchase.

In addition to electronic documents and the traditional data types, OISs will support video data such as pictures and graphs. Graphic packages are currently available and can be integrated into the OIS. Speech recognition would be a very desirable feature of an OIS, since some managers are opposed to keyboard interaction with the work station and prefer dictation and recordings in communicating with office personnel. Currently, speech recognition systems are in a very primitive stage of development. An experimental speech understanding system, Hearsay-II, recognizes utterances in a 1000-word vocabulary, with a correct interpretation rate of approximately 90%.[2] Other alternatives to the keyboard are touch screens and light pens.

Communications between work stations can be instantaneously carried out by means of electronic mail or other electronic-type messages sent over telephone lines or by way of a satellite communication network. The electronic mail system will contain known addresses and other pertinent information such as telephone numbers. The sending and receiving of mail is handled automatically with little interaction by the work station operator. A "mailbox" is maintained for each work station, the contents of which may be surveyed upon request. Immediate notification of newly arrived mail is possible if desired, and an indicator light that shines when the mailbox is not empty is feasible.

The physical configuration of an OIS could be as simple as a single processing unit supporting several work stations or as complicated as a distributed network of large, interconnected computers, each of which supports a cluster of work stations. The degree to which each work station can operate independently of other work stations and network processing devices determines the degree to which the OIS is immune to hardware malfunctions and downtime. Each work station will have local data bases and data objects and indirect access to information local to another work station. The design of a particular OIS will determine which data bases are available to each work station at any given moment and how that information may be accessed and used.

Multiple tasks may be performed in parallel at individual work stations. This will occur most often when the office worker describes entire tasks or parts of tasks so that the procedures may be handled automatically by the work station. When the work station encounters an unanticipated discrepancy or situation, it will notify the office worker, describe the problem, and await further instructions.

In addition to the concurrent processing at individual work stations, parallel processing may occur when multiple workstations perform operations on a single transaction. For example, consider the purchase order that requires the independent approval of two managers. Once the purchase order is ready for the approvals, it would arrive simultaneously at the two managerial work stations. Each manager could then approve the purchase, put it into a "wait state" for future action, or block the purchase by denying approval. The main point is that the two approval processes do not necessarily have to be staged serially; instead they can be performed concurrently. If one manager should disapprove the purchase, the other could be notified, and the purchase order handled accordingly.

Summary

An OIS can be viewed as a distributed network of programmable (intelligent), office-oriented work stations capable of communication with other work stations. The bulk of the work supported by an OIS focuses upon preparing and managing electronic documents. These documents are sent to the work stations where work is performed. The route of each document is denoted by the chain of actions required to process the document. The tasks associated with document processing may be autonomous and interactively performed. In addition, other facilities may be integrated into an OIS to automate office procedures. The OIS supports the concurrent processing of tasks assigned to an individual work station and of independent tasks associated with the processing of documents for a particular transaction. The Citibank system discussed next is an early attempt at creating an automated OIS.

Citibank's Management Work Stations

In 1976, the 12 management work stations (MWSs) installed at Citibank were intended to alleviate office workers' burdensome problem of managing and processing the paperwork needed to handle customer transactions.[3] The MWS system serves as an interesting introduction to OIS prototypes for a number of reasons: first, it is in a real-life business environment; second, it was designed to solve a business problem and not simply treat its symptoms; and third, it integrated several business functions into a single system with the end user, that is, the office worker, in mind.

The MWS system eliminated many of the inefficiencies of the traditional paperwork system by providing (1) automated internal correspondence, (2) an electronic mail system, (3) memo processing, and (4) financial and other report processing. The MWS system has built-in capabilities for word processing, document creation, editing and filing, business planning, distributed calendar maintenance, forms development, cost-benefit analysis, financial reporting, hard copy production, and the transmission of documents and messages from one work station to another. Each management work station consists of a PDP-8A processor, two keyboard-display units, an impact printer, and floppy disk drives. The MWSs are connected by telephone lines and shared by a manager and a secretary.

The MWS functions were designed to be consistent with the manual paper-based counterparts, so as to minimize the adverse effects of the transition and thus increase the acceptability of the system. Citibank observed secretaries and managers trained to use the new MWS system in order to gauge the effect of its introduction. Citibank concluded that the secretaries have showed "more flexibility and adventurousness in using the system than managers have."[4]

Citibank officials consider the primary advantages of the automated office over traditional office methods to be its impact on control, security, expense, and productivity issues. Electronic files were found to be relatively secure, and electronic forms were shareable and safe from accidental loss. Speedy filing and retrieval of office information and the other capabilities of the MWS system improved worker productivity and customer service. Together these advantages seem to justify the costs.

However, the limitations of the MWS system are many. Reliance on basic security and control mechanisms may be overly simplistic. The electronic mail system, to be more effective, needs to be able to handle external documents and correspondence as well as those generated internally. The MWS capabilities need to be extended to encompass the full spectrum of business operations and to permit the user to "program" the MWS to perform some tasks automatically. Clearly shown, however, is that an automated office, even this rather basic model, is practical and beneficial and will help set the trend for the OIS of tomorrow.

OIS: An Evolving Trend

As demonstrated by the Citibank example, a progressive and experimental project for its time, OISs are still in the developmental stage. Much of the technology required to make an OIS a reality is available. The degree to which the unsolved technological and psychological issues are resolved will greatly determine the usefulness and acceptability of OISs.

The technological issues remaining to be answered span several disciplines and are interrelated. The computer scientists' developments in natural office languages, communications, and hardware suitable for the office will strongly influence the roles and responsibilities of office workers and the physical organization of the office. Accountants and auditors, concerned with the possibilities of electronic fraud, need to ensure that internal controls are implemented. This will require adequate hardware and software support as well as managerial control of sensitive information.

Research into the psychological impact of the OIS on office workers is required so that the office workers' capabilities and views support its successful implementation. Sociologists may be able to provide recommendations based on their research into man-machine relationships and their effects on the social structure. The prospect of large, interconnected OISs will undoubtedly send reverberations through Washington, initiating, for example, the further propagation of privacy legislation regulating their use. Managing and integrating these essential advancements, findings, and technologies into a total OIS are themselves complex problems without immediate solutions.

The current limitations of OIS implementations are vast, but certainly not insurmountable. As work continues, answers to current questions will be found, and OISs will evolve, incorporating each new finding and experience. Since most of the formative work is done in laboratories, examining state of the art prototypes is appropriate.

12.7.3 OIS PROTOTYPES

Two OIS prototypes are discussed in this section. One, Officetalk-Zero, comes from an industrial setting, whereas the other, SCOOP, was developed in an academic setting.

Officetalk-Zero

Officetalk-Zero is a prototype OIS that was designed and implemented by William Newman, Tim Mott, and employees of the Office Research Group at Xerox Palo Alto Research Center (PARC).[5] The Officetalk-Zero project began in 1976, and the prototype was in operation by June 1977. Its goals are consistent with those designed into Citibank's MWS system. Officetalk-Zero is an electronic aid to performing routine office tasks, similar in manner to the traditional way of performing the equivalent clerical tasks. The fundamental object in Officetalk-Zero is the electronic document. The Officetalk-Zero commands are for the management and preparation of documents and for their transfer between work stations.

A major portion of the Officetalk-Zero development effort was invested in constructing a man-machine interface that would enable the office worker to use all of the system's facilities simply and uniformly. These goals were achieved by developing a work station supporting desktop activities and through access to an integrated system providing the user with a large set of office functions.

Configuration of Officetalk-Zero

Officetalk-Zero is configured as a network of interconnected minicomputers, each supporting a single work station. The minicomputer employed is the Xerox Alto with 128K 16-bit words of main memory and a 2.5 megabyte disk for external storage. The system supports both personal and system data bases for the storage of electronic forms, electronic mail, and information concerning each authorized user of Officetalk-Zero. Communications between work stations are conducted by means of the electronic mail system and the transfer of electronic documents. The work station is an advanced CRT device that electronically simulates the office worker's desktop. The user manipulates forms on the CRT device by employing a pointing device and a keyboard.

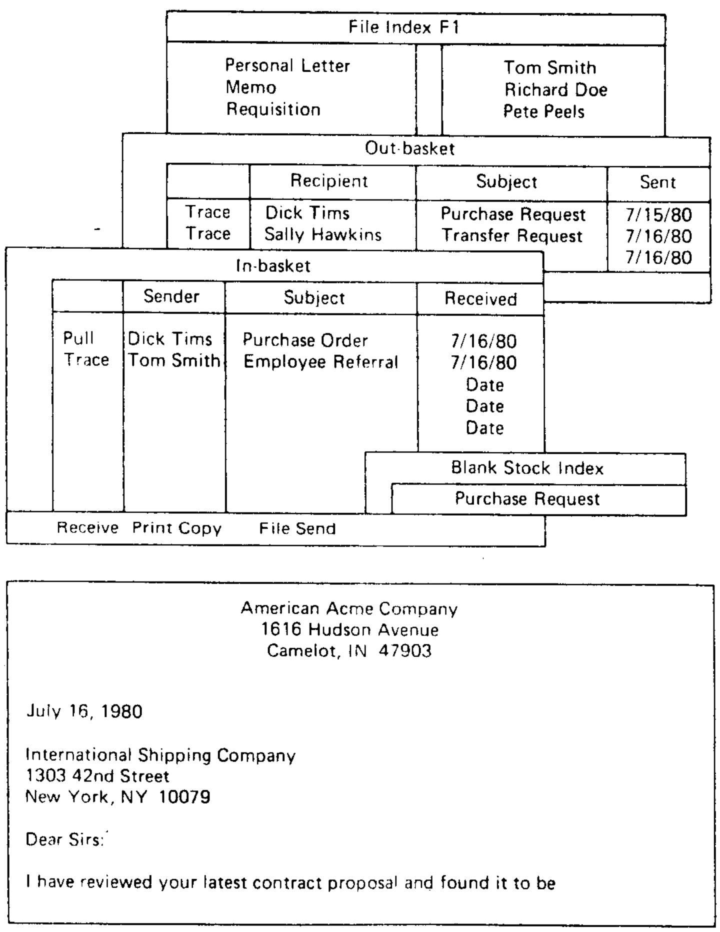
Work Station Description of Officetalk-Zero

The contents of the electronic desktop are represented by the display of forms on which work is currently being performed and four file indexes of forms available at the work station, for example, incoming mail, outgoing mail, retained forms, and blank forms, as shown in Exhibit 12.7.1. The system automatically updates indexes whenever the work station operator issues a command causing a form to be moved from one file to another. Each entry in the index consists of an action field specifying which Officetalk-Zero command may be used to process the form and pertinent descriptive information about the form.

Documents and the repository lists are displayed in rectangular windows on the CRT device. To start work on a form, the user indicates the form and the action to be performed by pointing the cursor to the action field of the appropriate entry of any index. If the specified action is to make a form available for processing, then the system will display the form in a new window. Each window includes a "menu" of Officetalk-Zero commands applicable to the form. Pointing the cursor to the desired function displayed on the menu causes the work station to enter the appropriate mode for interactive processing or prompts the work station to execute the selected function automatically. The user can enter data on a form by first pointing the cursor to the field that is to contain the data and then typing in the data on the keyboard. Data type checking is automatically performed by the system so that the entered information is in accordance with the forms specification; for example, a quantity field may contain only numerical data. A provided forms editor describes the standard letterings and format of the form and the style of each field on the form.

Opened windows on the CRT displaying forms or indexes may be adjusted by the office worker

Exhibit 12.7.1 Officetalk-Zero's Work Station Display



so that the form can be seen in its entirety or in part. A form may be moved around under the window for viewing various parts, or the window may be relocated on the screen. Overlapping windows will produce the same effect as overlapping pages on a desktop; the last window moved is wholly visible and hides those portions of the windows beneath it. The work station user controls the movement of a window and the form within a window by pointing a cursor to the appropriate area(s) of the screen. When finished with a document, the user may file or mail it, freeing the window and causing the document to disappear from the screen.

The work station possesses another interesting capability: It allows the user to afix signatures or freehand illustrations to a form. The freehand markings can later be removed without altering the form's original condition.

Capabilities and Limitations of Officetalk-Zero

Officetalk-Zero provides the office worker with the basic functions required to process documents effectively. These capabilities include a text editor, electronic mail with a trace function that enables the sender to monitor the location of a mailed document, data entry capabilities, and functions for copying and filing documents. The limitations of Officetalk-Zero appear not to be the result of shortcomings in the implementation of its existing capabilities, but rather the omission of other desirable capabilities. The nonprogramming environment of Officetalk-Zero does not allow the user the flexibility of defining personalized office procedures and functions to enhance the effectiveness of the work station.

System for Computerization of Office Processes

Zisman developed SCOOP at the University of Pennsylvania as part of his doctoral research. The system stresses the automation of office procedures as well as of office devices.[5,6] The heart of SCOOP is a model-driven monitor that traces the progress of office procedures and automatically executes portions of them at the appropriate times. By so doing, the computer and the work station operator interact in a joint undertaking. The system description for driving SCOOP's monitor comprises document definitions and office activities expressed in a nonprocedural language. The underlying formalism of the system description is the Petri net augmented with simple and compound predicates whose truth initiates the execution of corresponding actions. The predicates can be used to detect certain events such as the existence of a record, a value set by a transaction, or the passing of a set amount of time. Like Officetalk-Zero, SCOOP has a single, uniform interface for integrating the special-purpose systems into a unified system. The overall system supports electronic message passing, document manipulation, file services, and other functions. The creative idea of automatically monitoring office procedures to assist the office worker is extendable to monitoring office work for compliance with security and internal control specifications. This issue is explored in Section 12.7.5.

12.7.4 THE OIS USER INTERFACES

Earlier this chapter stated the essential properties of an OIS: It must be flexible and powerful enough to meet the user's needs, and yet simple enough that the office worker, a nonprogrammer, can use it effectively. Historically, interactive application systems were developed under a systems analyst's supervision and management. The systems analyst's task was to survey the user's applications to establish the necessary system capabilities and to design a system meeting those needs. The system design was a blueprint from which the computer technicians programmed the system. Thus the systems analyst served as an intermediary, bridging the communication gap between the user and the programmer. If failures in communication were kept to a "reasonable" level, the system design would indeed meet most, if not all, of the user's needs and would be implemented as specified.

However, often the net result of these weeks or months of implementation effort would be an inflexible system of parameterized programs that were invoked through a kind of menu selection that only temporarily satisfied the user's requirements. Changes in such systems were often difficult and costly. Yet changes in the way business is conducted are inevitable and will require such alterations in the existing systems.

A high-level office language that office workers can easily learn and use would substantially improve the potential for efficient and effective OIS development and maintenance. However, because of the complexities inherent in office work itself and in natural language development, the attainment of such a language will not be easy. Nevertheless, the current state of the art makes it reasonable to expect the development of high-level office languages enabling the user to set up and maintain a wide variety of office application systems, and yet requiring only minimal training.

The languages will combine WP and data base management facilities, enabling the user to define, maintain, and query data bases and to incorporate the selected and processed data base information into reports, documents, memos, and so on. Micro Data Base Systems is a good example of a sophisticated data base management system for the control of complex data structures. It is commercially available and operational on microcomputers.[7] Word processor and data base management capabilities will make it possible for office workers to develop software for common office applications such as those in a doctor's office.

Two proposed high-level languages for use in OISs are presented next, followed by a brief discussion of the controversial issues surrounding the problem of the user interface with automated OIS.

The Office Procedures by Example Language

The Office Procedures by Example (OBE) language is a nonprocedural language intended for use in automating office and business applications.[8] The result of a research project of IBM, OBE is an extended version of the IBM-marketed Query-by-Example data base management language. The OBE language makes it possible for office personnel to maintain data bases and program work stations to create tables, forms, and other documents with information already available in a data base. Various mathematical calculations are also possible. Office Procedures by Example also maintains facilities to save programs for reuse or automatic invocation whenever certain critical events occur within the operation. Communication in OBE is by means of an electronic mailing system.

The fundamental object in OBE is the two-dimensional form. The user specifies how the data base information is to be mapped onto the form by filling in the form with sample data. The sample form specifications constitute an automated procedure, which, when executed, produces a document. This document can be edited further or can be sent to another work station in the network.

Exhibit 12.7.2 exemplifies an automated procedure written in the OBE style. The purpose of the procedure is to prepare a list of new employees arranged by manager. The procedure is set up to run automatically once a month without user interaction. The reports are sent automatically to the corresponding managers through the electronic mailing system. The procedure specification consists of (1) references to two data bases, (2) query criteria, (3) a template for creating the desired form, (4) a trigger command, and (5) a send command.

Exhibit 12.7.2 OBE Procedure for Personnel Reporting to Management

APPLICANT	NAME	H/R/W	POSITION	DATE	DEPT
TR1 (MONTHLY)	N	H	P	DT	D

DEPARTMENT	DEPT	MANAGER
	D	M

CONDITIONS
$1/01/80 \leq DT \leq 12/31/80$

A

Dear M

This is the current listing of new employees who are scheduled to work for you this year, 1980.

Juney Bird

NAME	POSITION	DATE
N	P	DT

SEND (TR1) A TO M

The first data base of the procedure description, APPLICANT, contains the name, status (H = hired, R = rejected, W = hired and working), job position, hiring date (implies status is an H or W), and hiring department for each applicant referral. The DEPARTMENT data base contains the title and manager's name for each department. The underlined character strings, N, P, DT, D, and M, are example elements. The presence of each example element in the data base description and in the body of the sample form indicates a mapping of the data base information onto the form. The APPLICANT and DEPARTMENT data bases are linked in this procedure by the department title field, DEPT. The department title field is common to both data bases. The result of this cross-referencing is that M is a place holder for the manager's name for each department. The character H in the status field of the APPLICANT data base is a constant and states that only applicants with a status of H are to be accessed by the procedure. The condition box places a further constraint as to which applicants are to be reported. Specifically, those applicants who have a hiring date in 1980 will be a part of the report.

The automatic scheduling facilities of OBE maintain a list of procedures that are to be run routinely. It automatically schedules them for execution at the appropriate times. The trigger command, TR1 (MONTHLY), and the SEND (TR1) A to M command specify that each monthly report is to be sent automatically to the manager that the letter A addresses. Whenever this procedure is executed, reports similar to the ones shown in Exhibit 12.7.3 are created. The exhibit also presents the contents of the data bases used to create the forms.

The OBE language provides means for defining, updating, querying, and delegating user access rights to data bases. Data base integrity can be preserved through the use of integrity constraints. Potential integrity constraints include the editing of the update data for reasonableness of validity

Exhibit 12.7.3 Sample Input and Output for OBE Procedure for Personnel Reporting

APPLICANT	NAME	H/R/W	POSITION	DATE	DEPT
	Sandy Wilson	H	Secretary	03/20/80	ACCT
	Burt Williams	R	Accountant		
	Sue Fields	H	Accountant	01/01/81	ACCT
	Tom Smith	W	Mgr-Asst	01/15/80	ACCT
	Joe Peels	H	Mgr-Asst	09/01/80	PAYR
	Gerald Farrel	H	Accountant	07/15/80	ACCT

DEPARTMENT	DEPT	MANAGER
	ACCT	Kathleen Roberts
	ENGR	Joe Doe
	PAYR	Peter Davis

Dear Kathleen Roberts

This is the current listing of new employees who are scheduled to work for you this year, 1980.

Juney Bird

Name	Position	Date
Sandy Wilson	Secretary	03/20/80
Gerald Farrel	Accountant	07/15/80

Dear Peter Davis

This is the current listing of new employees who are scheduled to work for you this year, 1980.

Juney Bird

Name	Position	Date
Joe Peels	Mgr-Asst	09/01/80

of values and for interdata and data base consistency. The integrity constraints may be established such that they are unconditionally enforced or enforced for only particular data base operations. As noted in the preceding discussion, in addition to executing procedures automatically based on specified timing criteria, procedures and other OBE actions may also be executed automatically whenever specified events occur, for example, an overdrawn account. The conditions controlling the automatic execution of OBE actions and commands may be simple or compound.

The Business Definition Language

The Business Definition Language (BDL) developed by IBM at its Thomas J. Watson Research Center is a high-level language designed specifically for business data processing.[9] The view of business data processing manifested in BDL centers on the fundamental observation that document processing is the primary operation in business data processing. Documents are prepared and transmitted singly and in groups between departments and divisions as a means of communication. These same documents are later used to support recording and cross-referencing transactions. The numerical operations involved in the preparation of these documents are usually very low level.

An objective in designing BDL was to create a language office workers could use to write programs that would be structured, modularized, self-documenting, and easily maintainable. In pursuing these goals, one means employed by the designers was providing the BDL programmer with a language in which whole algorithms are written in meaningful problem solving terms. Even though BDL was designed for general business data processing, the design philosophy employed in its development is a blueprint suitable for the design of future OIS languages.

The basic objects in the BDL are documents, steps, paths, and files. Documents serve as input and output mediums for steps and programs. A step is a sequence of actions for the manipulation of documents, and a program is a sequence of steps. Each step action is either irreducible, that is, a primitive step, or a composite of more primitive steps. An irreducible step represents a standard, predetermined document transformation or a user-written routine describing the input and output behavior of the step. Composite steps are reducible to primitive steps. Each step corresponds to an organizational unit such as a department or work station. The nesting of the composite steps is intended to reflect the hierarchical structure of the organization. A path is the representation of the flow of documents between steps. Files are used for long-term storage of documents.

Major Components

The three major components of a BDL system description are the form definition component (FDC), the document flow component (DFC), and the document transformation component (DTC). The FDC enables the user to define templates for creation of blank documents. Each document's design is developed by drawing it on a graphic device. Fields within a document are described by the type of data the field is to contain and by its format. Groups of fields can be defined as single entities, and a sort facility can be used to order the entities. Provisions also exist for describing floating fields and for handling form overflow (the extraneous information spills onto a user-designed overflow form).

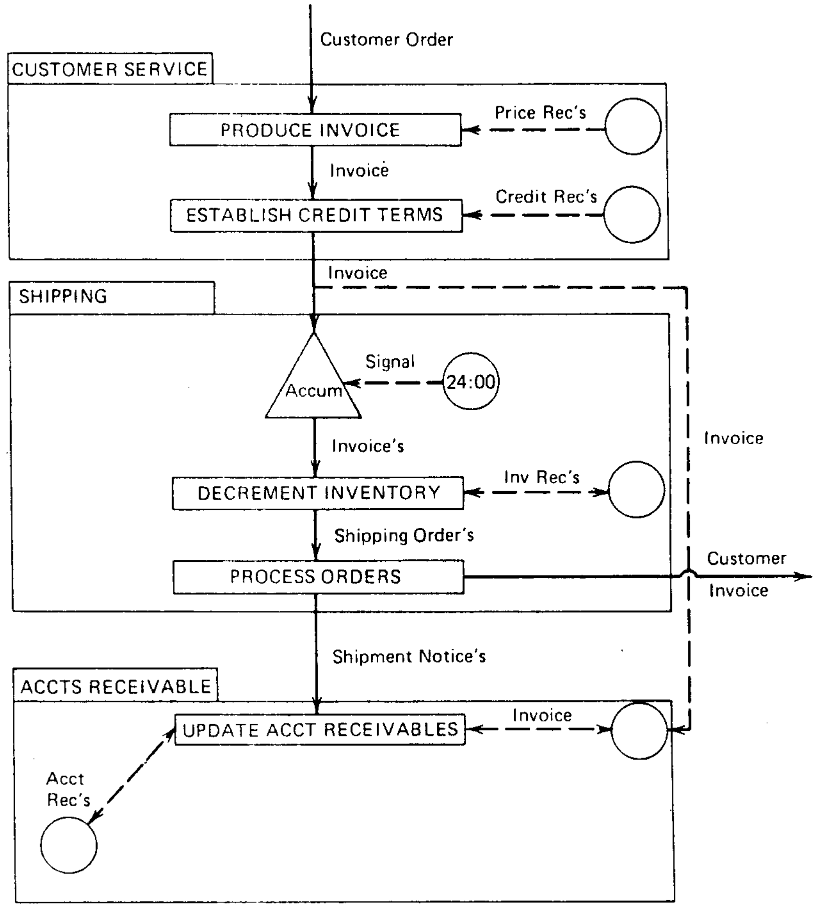
The DFC application system is graphically defined in terms of the organizational units involved in the processing and flow of documents. The basic diagram constructs are rectangles, circles, and arrows, used to represent steps, files, and paths, respectively. Forked data flows indicate that a copy of the document is to be sent along each path identified. Exhibit 12.7.4 presents a DFC representation of a sales processing system.

The sales processing system description has three major composite steps: customer service, shipping, and accounts receivable. The internal structure of the major composite steps is further defined in terms of more elementary steps and data flows. These constituent steps may themselves be composite steps which could be further refined in other DFC diagrams. Previously unspecified file accesses may be introduced in the more refined DFC diagrams. The system is fully described by the DFC when each stop is irreducible, or a primitive step.

The sales processing example is especially interesting in that it combines aspects of both OIS and data processing applications. The customer service step prepares the invoice and approves credit terms. The invoice preparation step is performed by a secretary at the work station. The secretary obtains a blank invoice and fills in the customer's name, address, product numbers, and quantities. A procedure written by the secretary then accesses the price file and enters the product unit price, the total price by product number, and the sales invoice total price. Finally, the secretary reviews the invoice, makes any necessary changes, and sends it to the appropriate work station for credit approval action.

The credit officer, having specified general approval policies in granting credit terms on routine orders, only affixes a specific approval to requests outside the general approval policy bounds. For these special cases, the work station is programmed to display the sales invoice in question along with the customer's credit history. In either case, the approved invoices are forwarded to shipping.

Exhibit 12.7.4 Sales Processing System



Shipping and accounts receivable are data processing systems with routines differing from the OIS procedures in that they need no autonomous procedures, but can be fixed algorithms requiring very little human interaction. The Accum function queues the invoices until midnight, at which time the function releases the day's entire group of invoices to the shipping data processing system. The shipping data processing system fills the orders according to a fixed priority scheme, updates inventory, and schedules shipping. In an automated warehouse, workers then supervise the packing of the orders and notify the system as each order is filled. The system then generates a shipment notice document destined for accounts receivable. Similarly, the accounts receivable data processing system accepts the shipment notices as input and updates the accounts receivable and invoice files.

The DTC of the system description is used to define the actual computation of each irreducible step in the DFC. Theoretically, any programming language or marriage of programming languages could be used in place of the DTC language. Thus, as in the previous example, an OIS language and a business data processing language could both be used to implement mixed systems, taking advantage of the office worker's programming skills and those of the application specialists. The very high level DTC language developed assumes a general structure for all steps and thus gives the programmer the advantage of a built-in control structure oriented toward business data processing.

In exchange for the resulting programming simplicity, we accept a loss in language flexibility. The programmer utilizes the resulting control structure in defining transformations of the input documents to create output documents. The computations appropriate to each field included on the output document are described using normal arithmetic operators, aggregate arithmetic operators, and logical and relational operators applied to available fields on the input and output documents. The DFC's run-time mechanism automatically readies the step for execution whenever documents are available on all its input paths, when prior system activity is complete to that step.

Office Workers and Machine Interface Issues

In any organization, including the business office, formal and informal communication lines exist between individuals and organizational units. Formal communications are designed to meet certain anticipated, critical information needs, with written documents usually serving as the medium. Preparation of the necessary forms is generally detailed as a part of the system description.

This type of communication closely corresponds to the document-oriented view of an OIS. The well-defined procedures lend themselves to being algorithmically specified. The documents can be structured and formatted and electronically or mechanically manipulated. Moreover, a mailing system is a natural and efficient means of transferring documents between the organizational units. Office information systems, properly complete with user-matched capabilities, are capable of efficiently supporting formal office communications.

The critical issue concerning formal communication in an OIS environment is the office worker–machine interface. As formal communications are machine representable, the question becomes whether clerical users can get the necessary support to perform office tasks with standard office capabilities such as WP and user-written routines.

Informal communications—conversations, gestures, jokes, and unwritten guidelines—serve an important organizational need often not adequately accounted for by the system. Dialogue is used by senior employees to provide on-the-job training for junior employees. Stimulating conversations about business problems, sharing ideas and knowledge, can give problem solvers new insights. Conversation can encourage personal involvement, concern, and a feeling of group membership. These forms of social contact are essential to maintaining an effective office organization and exemplify information communications that cannot be adequately supported by a purely document-oriented OIS. The integration of information communications into an OIS is a major concern in interfacing people and machines.

Office Worker–Machine Interface

The implications of a particular design philosophy and the nature of the resulting OIS interface are important to office personnel staffing and training. Clerical workers and managers cannot be expected to possess expert programming skills, and yet they need to control their own office procedures. An OIS will generally incorporate only specific problem-oriented functions and capabilities commonly needed in an office setting. For example, Odyssey is a knowledge-based system for aiding an individual user with business trip preparations.[10] Knowledge of trip preparation is built into Odyssey so that it can *intelligently assist* the user.

Specialized office assistance packages such as WP, DDS, and Odyssey can provide interactive procedures to aid the office worker in performing office tasks. These procedures enable the worker to perform effectively the particular task for which the system was designed. Users will require little training since most of the systems will have simple interfaces using basic problem-oriented commands or prompting techniques. However, they cannot cover the wide range of autonomous office applications currently required by business. Depending upon the nature of the task, office workers will be expected to specify and correct programs and to maintain their own automated office procedures.

This section addresses a few of the many issues concerning the design and implementation of OIS languages that will ease the office workers' burden in this respect. A number of these issues have been addressed by the computing community and correspond to questions regarding conventional programming languages: structured programming, concurrent languages, program and system documentation, data base languages, and the elimination of technicalities such as memory dumps from the programming environment.

Programming languages are defined by their syntactic and semantic rules. For a programming language to be properly interpreted by the computer, it must adhere to a set of rules that can be encoded as an algorithm. This need to structure and interpret the languages rigorously puts even limited natural language processing outside current programming technology. Clearly, the styles of conventional programming languages such as COBOL, FORTRAN, or Pascal are inappropriate for unsophisticated OIS users. For this reason, languages more appropriate for OIS have been described and are reviewed here.

An example of a special-purpose OIS language is OBE. Its simple, mimicking style makes OBE very easy for even unsophisticated clerical personnel to master. It is well suited to programming work stations called upon to prepare simple forms formatted according to the OBE standards. The automatic task initiation feature is also easy to use and very practical. The problem with special-purpose languages like OBE is that their application span is very narrow. Tasks coinciding with the abstract model on which the language was designed can be easily encoded; however, any deviation from the language's original problem formulation assumption will greatly increase the programming problem, if not make it impossible to express in the language.

At the other end of the language continuum are general-purpose languages. For a language to be general-purpose, it must consist of fundamental operators, data types, control structures, and data structures with which problem-oriented routines can be constructed. Translating a conceptualized algorithm into such a language is difficult; however, the more general-purpose the language, the wider the range of problems it can be used to solve. This benefit is obtained at the cost of increasing the conceptual difficulty in the programming task. Matching the right OIS language to the clerical user's or application specialist's needs and abilities is essential to effectively increasing productivity. A mismatch will reduce an OIS's effective use and could cause the system to fail.

Problem-oriented BDL is about midway between a general-purpose and a special-purpose language. Variations on BDL may provide an almost general system framework permitting the automatic interface of two or more languages. As the sales processing example showed, the DFC served as the system framework in depicting the irreducible steps and their relationships. Office-oriented steps could be programmed in the OIS language, and data processing steps could be programmed in a more appropriate language. If each step complied with the same document input and output concept, then the interface between the languages could be the document. On the surface this style of integrating languages is appealing because it permits the selection of the proper language for the task and the automatic interfacing of OIS with data processing systems.

The introduction of the *intelligent form* could help to lessen individual tasks by embedding controls within the form. The intelligent form would automatically perform complex edit and data consistency checks on itself and restrict the individual user's rights to view, update, or add information, depending upon the user and/or the form content. The intelligent form would schedule itself for work station processing according to specified priorities and current work station loads and could become persistent if neglected too long. Once the form is so described, office personnel would need be concerned only with primary document transformations. This would guarantee a measure of consistency in processing the document and would improve data security.

Another noteworthy issue includes the possibility of standardizing OIS languages to eliminate manufacturer dependency and problems involving constant, necessary training and retraining of personnel and to increase the portability of office procedures as well as data and procedure security.

Informal Communications

As previously mentioned, informal communications are necessary to performing group tasks in the business office. In a pure document-oriented environment, casual memos delivered by means of an electronic mail system cannot replace social contact and information exchange. An OIS that could support video and audio transmissions would heighten social interaction. Whether this would be adequate is an unanswered question. Office organization designs aimed at increasing social contact are a potential solution to the problem, but perhaps at the cost of some lost machine-based effectiveness. As hardware technology advances, work station portability will increase the prospect of the office worker's home becoming the office–a situation with far-reaching social and business implications.

As first-generation OISs are introduced into the business environment, there is likely to be little loss to the informal communication system. Implementation will probably not be complete, and initially clerical personnel will rely on each other while they learn how to use the system effectively. However, as OIS users become more proficient and OISs more powerful and complete, the task requirements for social contact will decrease.

12.7.5 OFFICE INFORMATION SYSTEM MODELING

Office information systems developed around the formal lines of communication and formal work station activities are subject to precise modeling. The advantages in modeling OISs are several. First, an OIS described in a formal manner can be checked for descriptive consistency and completeness. Second, a validated OIS model constitutes a standard and uniform documentation of the system. Third, the model can be used to further various types of analysis. For example, office efficiency specialists might use the OIS model to predict system performance, or auditors concerned with internal control issues would be able to analyze the control and processing structure. Fourth, managers, once convinced that the OIS model is secure, adequately controlled, and efficient as modeled, will require assurances that the implemented OIS complies with the model. Given the complexity of a full-scale office, it is infeasible to perform such analyses manually. What is needed is an OIS modeling process capturing these advantages as well as providing managers with the capability to restructure the organizational units and tasks under their authority as circumstances dictate.

Information Control Nets

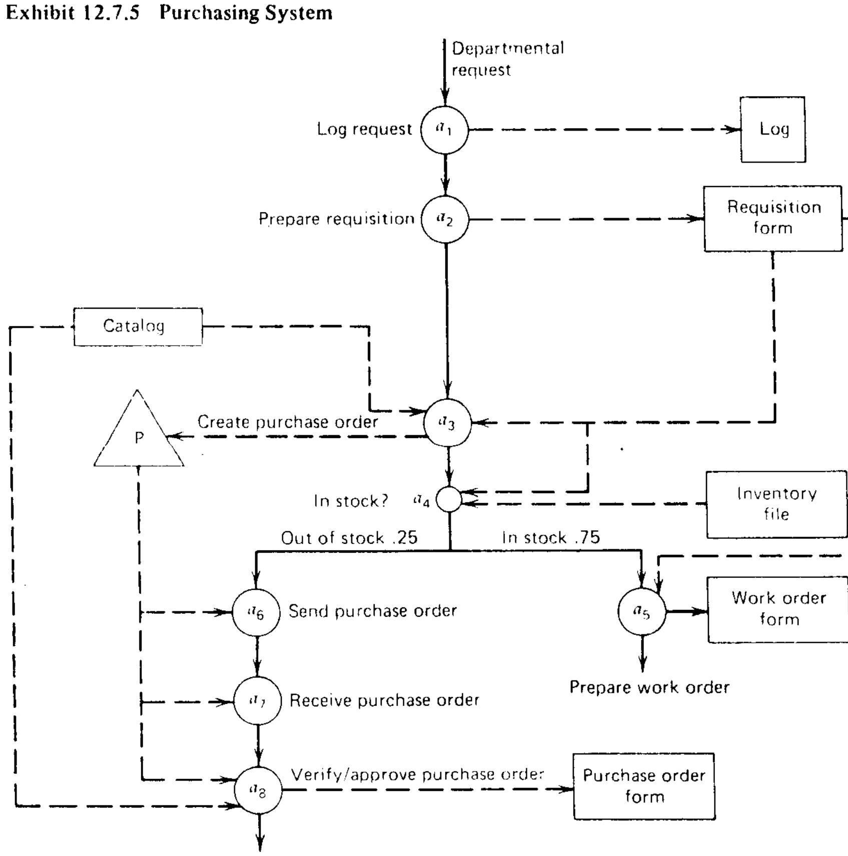
Information Control Nets (ICN) is a model developed at Xerox PARC to describe and analyze office information flows.[11] The ICN model defines an office as a set of uninterpreted *activities* that access and update information stored in *repositories*. Possible execution sequences of activities are defined by "precedence constraints." A precedence constraint is a simple rule stipulating an immediate successor activity for the parent activity. Upon completion of an activity, one of its immediate successor activities can proceed. As with most office data flow models, the execution of a subsequent activity need not immediately follow the termination of its predecessor. If the work station operator is busy or away from the work station, then the activity is temporarily suspended.

The office models presented here share a fundamental perspective on the processing environment: that offices are transaction oriented and that, even though many transactions are performed concurrently, the activities required to process each type of transaction can be adequately described without explicitly considering the concurrent processing of like transactions. The models are designed to depict all preconceived processing for a particular type of transaction. Specific knowledge of how each type of transaction is individually processed can be used to infer knowledge of the highly parallel office environment.

An ICN can be represented graphically or as mappings defined over a set of activities and a set of repositories. In ICN diagrams, rectangles represent repositories; triangles represent temporary repositories; labeled circles denote activities; small, hollow circles denote conditional branches (choice nodes); solid lines connecting activities denote precedence constraints; broken lines denote repository access and updates; and a small, solid circle denotes the start (end) of parallel activity sequences.

Exhibit 12.7.5 is an example ICN graph. In the exhibit, activity a_1 begins upon the arrival of a departmental request. Activity a_1 simply logs the request in a logbook. Upon completion of a_1, a_2 is initiated, resulting in a typed requisition, which is then stored in the requisition form repository. Activity a_3 prepares a purchase order form. Choice node a_4 follows a_3 and establishes whether activity a_6 or activity a_5 is to follow. The choice depends upon whether the requested items are in stock. The branch associated with the in-stock choice is assigned a probability of occurrence of .75. If the out-of-stock branch is taken, activities a_6, a_7, and a_8 designate that the purchase order is to be sent to another worker for verification and approval; otherwise, activity a_5 completes a work order form for inventory. Note that the activities are only vaguely defined. The designers

Exhibit 12.7.5 Purchasing System



of the ICN suggest that a vagueness is advantageous since the model can illustrate information flows without obscuring the basic flows with overcomplicated operational details.

Ellis and Morris have developed techniques to automate the analysis and optimization of ICN models.[12] The ICN models are intended to be optimized according to three kinds of office transformations: *automation*, *reorganization*, and *streamlining*. Automation transformations replace manual activities with corresponding automated activities. Reorganization transformations shift activities to enhance some particular property of the model, such as concurrent processing. Streamlining transformations are used to simplify the ICN by eliminating useless activities and reducing communications overhead.

Exhibit 12.7.6 is a result that could be obtained by automated optimization transformations to the purchasing system of Exhibit 12.7.5. Activities a_6 and a_7 on Exhibit 12.7.5 are automated on Exhibit 12.7.6 within activities a_3 and a_8. The OIS communications system is utilized to accomplish this end. Shifting the prepared requisition activity (a_2) with respect to the log request activity (a_1) permits parallel processing of activities a_4 and a_1, the in-stock check and log request, respectively. This latter transformation belongs to the reorganization category. The primary advantage of parallel processing is to reduce the elapsed time requirement to complete a transaction. Activity a_3 has been rolled forward through decision node a_4 since the catalog needs to be accessed only if the requested items are out of stock. Since activities a_3 and a_8 access the catalog, and activity a_8 is an immediate successor to activity a_3, the catalog information obtained by activity a_3 can be stored with the purchase order form in the temporary repository p for use at a_8. This transformation will reduce competition for access to the catalog and will make the catalog information readily available to activity a_8. These preceding transformations were designed to reduce file access overhead and are examples of streamlining transformations.

By augmenting the ICN diagram with branching probabilities and average service times, it is possible to calculate the expected service time to process a single transaction. Consider Exhibit 12.7.7, which is an example of such an augmented ICN diagram. (Files are omitted for simplicity.) The precedence arrows are labeled with the branching probabilities and the expected service times, s, of each specified activity. For example, the service time for activity a_3 is 1.5 min, and the probabilities of activity a_3 initiating activity a_4 or a_5 are .3 and .7, respectively. Thus the expected service time required to complete activities a_3 and a_4 or activities a_3 and a_5 is $1.5 + .3(5) + .7(10) = 10$ min. Similarly, we could calculate the expected service time of the entire graph.[11] Using matrix notation, the computation is performed as follows:

$$\text{expected service time of the graph} = W[\Sigma(V\pi^i)]$$

Exhibit 12.7.6 Purchasing System of Exhibit 12.7.5, With Automated Optimization Transformations

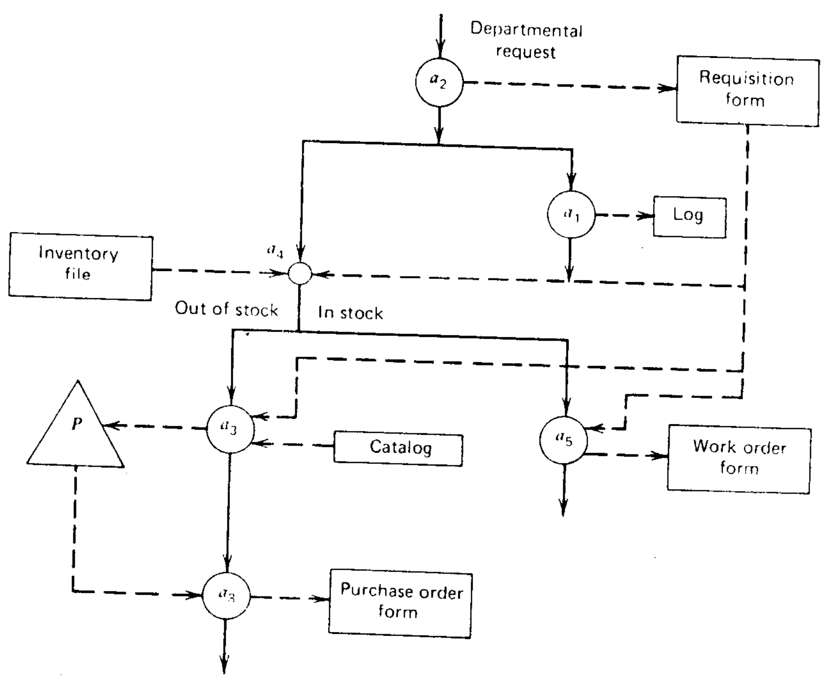
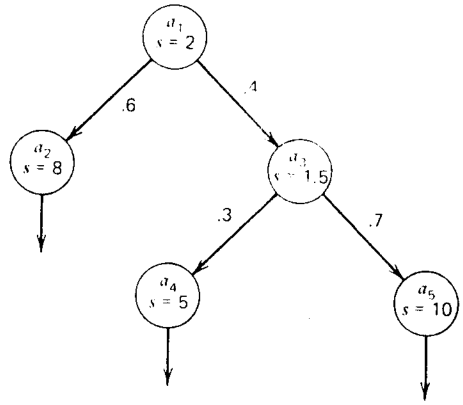


Exhibit 12.7.7. Augmented ICN Diagram and Expected Service Time Calculations



$$W = [2 \quad 8 \quad 1.5 \quad 5 \quad 10]$$

$$V = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\pi = \begin{bmatrix} 0 & .6 & .4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .3 & .7 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(V\pi^0) = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$(V\pi^1) = [0 \quad .6 \quad .4 \quad 0 \quad 0]$$

$$(V\pi^2) = [0 \quad 0 \quad 0 \quad .12 \quad .28]$$

$$(V\pi^3) = [0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$(\Sigma V\pi^i) = [1 \quad .6 \quad .4 \quad .12 \quad .28]$$

$$W[\Sigma(V\pi^i)] = [2 \quad 4.8 \quad .6 \quad .6 \quad 2.8] = 10.8$$

where $W_i = s_i$
V_i = probability of entering the graph at activity i
π_{ij} = probability of a_j directly following a_i

This type of analysis may be useful in analyzing office models' characteristics. It is possible to ascertain the effects of introducing parallelism into a model, to answer questions concerning individual and group work loads and processing times, and to study system throughput characteristics. Such elementary analysis is useful, but very limited. For example, consider an office where order transactions are processed on a priority basis. The transaction's priority may be based upon simple criteria, such as a customer's credit rating, or on more complex, multiple criteria, such as the purchase amount, the customer's purchasing history, and/or current inventory levels. Now, complicate the problem by adding arrival rates for seasonal or sporadic transactions. Calculating the expected elapsed time to process a particular order transaction becomes extremely difficult and its meaning questionable. Currently, simulation techniques are employed to analyze such models.

OIS Simulation

System simulation is an interdisciplinary technique widely used to assist in measuring system performance. The ability to simulate presumes the ability to model the system. Industrial engineers and operations researchers use simulation as a tool for analyzing such systems as machine networks. Accounting researchers use simulation techniques to study transaction processing systems and to measure the effectiveness of the error detection and correction facilities incorporated in the system design. System designers also rely heavily upon simulation in fine-tuning system configurations. Common to most simulation applications of this type is the need to measure a system's performance and its components using measurements such as utilization, work load, throughput, and response time.

Simulation models employ descriptions of system procedures, queuing strategies, and probability distributions for such attributes as arrival rates, service times, system resource needs, and routing probabilities. Since the automated office is a system similar to those currently being simulated, office designers can also employ simulation techniques as a design tool. Detection of bottlenecks and underutilized work station capabilities can give the office designer crucial insights into reorganizing office and work station tasks. The statistical results of two simulation models operating under the same work load provide comparative measurements for evaluating one against the other.

Work Station Simulators

Nutt and Ellis propose the use of an office environment simulator not only for observing the real-time performance of an OIS, but also as a personnel training tool.[13] The personnel training tool is a work station simulator that can give the office trainee hands-on, real-time experience. Flight and driving simulators have already proved the effectiveness of simulators for learning purposes. The advantages of training on a work station simulator are as follows:

1. *Events and situations can be manufactured to specifications.* Using a real OIS, the trainee would repeatedly perform the same ordinary tasks, spending very little time handling important exceptional situations. At a work station simulator, all preconceived events and situations would occur according to specified frequency rates.
2. *Measurement of office worker performance.* Measurement of office worker performance is an extension of advantage 1. By presenting office workers with identical work loads and collecting statistical information as the office worker manipulates the work station, measurements are obtained that are comparative between office workers, between individual sessions, and against prespecified standards.
3. *A practicing environment for developing automated office procedures.* The work station simulator would permit the definition and execution of office procedures by the operator of the work station. This would enable the trainee to gain valuable programming and debugging experience without incurring the costs associated with mismanaged informations.
4. *A controlled environment (laboratory) for studying human performance in relation to changes in the work station design or capabilities.*
5. *A systems development tool.* As work station operators develop and refine automated procedures, these procedures can be used to simulate unmanned work stations. In this way the procedures' correctness and completeness can be tested in a real OIS environment.

A facility for real-time interactive simulation of a distributed office system has been developed at Xerox PARC. The facility, Backtalk, comprises a network of nodes, each of which is a work station or a dedicated device such as a printer or a data base server. One or more of the work stations are operated by automatic procedures that model the would-be operator in a realistic fashion. The remaining unautomated work stations serve as simulators that interact with the entire system.

The realism of the simulation depends upon the automated work stations' ability to reply to the simulator work stations' specific demands. Since the construction of the reply is contingent upon the nature of the demand, it is impossible for automated procedures to reply realistically to information communication events such as memos. However, a certain degree of realism can be achieved by identifying some basic characteristics of the demand and answering with prestored replies or replies taken from a data base of historical interactions between real work station operators.

The Backtalk facility possesses many of the abilities and properties presented in the list of advantages above. In a single work station simulator mode, it enables the work station operator to program procedures; access data bases; manipulate forms; transfer them to other automated work stations, which in turn can generate new forms; transfer forms to other work stations, including the work station simulator; and destroy forms.

Theoretic Internal Control Model—TICOM II

An important aspect of any OIS is the auditability, accountability, and internal control protections of the system. The Foreign Corrupt Practices Act of 1977[14] requires firms' procedures to satisfy certain internal control criteria that lessen the possibility of misallocation of corporate assets. As a result, every OIS operation is subject to external legal criteria. Consequently, significant design criteria are imposed on any OIS in addition to the usual engineering and convenience constraints.

The legal restrictions on office procedures, whether automated or not, are not the only criteria an OIS must satisfy. Firm managers also will wish to verify that only authorized individuals access company assets for valid purposes. When document processing is executed by machine rather than manually, the difficulty in verifying internal control procedures becomes more complex, as is corroborated by increasing "computer theft." As corporations grow more complex, successfully auditing firms in the traditional fashion—with involved, laborious flowcharting and narrative descriptions of document and asset flows—becomes more difficult.

Therefore satisfying legal constraints and management desires will soon require more extensive computer-assisted auditing. There are two important advantages to automated auditing. First, the speed, precision, and cost of computer processing compare favorably with their human counterparts. Second, interfacing an automated auditing and internal control system with an OIS will result in a nearly continuous audit, considered to be ideal in auditing literature.[15] In this way the OIS may be modified to satisfy legal constraints and permit economical internal control verification.

TICOM II[16,17] is an example of a computer-assisted internal control verification system. In a sense it proceeds in much the same way a human auditor does in establishing that certain internal control conditions are satisfied. However, the analytic capabilities are automated and therefore can be interfaced effectively with an OIS. The result of this conjunction is continual proof that the OIS does not violate sound management principles of internal control.

Intuitively, when firm employees use the OIS to process documents, the result of this processing will be examined by TICOM II. The latter will then consider whether the specified processing implies a violation of sound internal control. For example, it is considered good practice for the purchase order sent to the receiving department to contain only quantity data. The receiving department must enter the actual quantity received from the vendor and not merely copy from the quantity listed on the purchase order when completing a receiving report. Thus, if the receiving department attempts to access quantity data, TICOM II will immediately flag and prohibit the access as a violation of internal control. This prohibition exemplifies the interaction between TICOM II and an OIS in thwarting internal control violations.

Mechanics of TICOM II Modeling

The TICOM II modeling procedure is a generalization and semiautomation of the flowchart and narrative procedures auditors use to describe and evaluate internal control systems. The modeling input is a description of the duties and tasks of individuals within the firm. Instead of the flowchart, however, this modeling is expressed in the TICOM II modeling language.[16] The procedure is similar to flowcharting these duties just as writing a flowchart for an algorithm is similar to writing a computer program. An example of the TICOM II modeling is given in Exhibit 12.7.8. It shows the relationship of the flowcharted boxes representing document processing and the commands of the TICOM II modeling language. A complete description of the language and an extended example can be found in Bailey et al.[16]

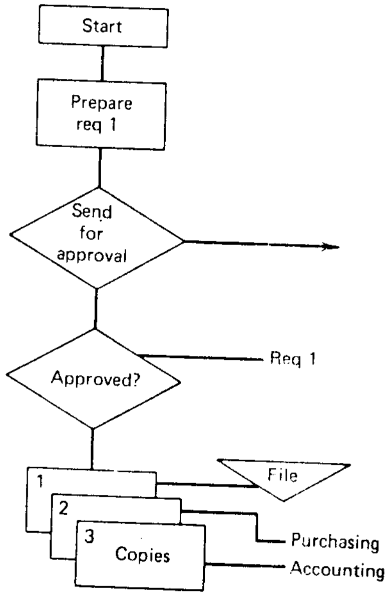
A unique aspect of the language is embedded in the WAIT expression illustrated in Exhibit 12.7.8. The modeling procedure currently used for internal control descriptions involves the simultaneous construction of a flowchart for the entire firm. In the TICOM II modeling procedure, individual tasks are described separately, and these are integrated into a complete model of the firm by an automated algorithm. The links between individual tasks are determined by the WAIT expressions. In Exhibit 12.7.8 we see clerk 1 waiting for requisition approval. This informs the computer that the TRANSFER of approval fits into clerk 1's duties. Thus, by matching TRANSFER to WAIT expressions, a model of the firm's internal control system can be recovered from the individual descriptions.

Once the firm model is pieced together, TICOM II then analyzes it for consistency and completeness. This processing phase involves checking various items. First, it establishes whether each command performs a valid task, whether the document processing involves possible actions. For example, if an item on a purchase order, such as "vendor," is altered, it is established that this item really exists on a purchase order. Second, it is determined that every WAIT expression has a corresponding TRANSFER that satisfies it. Finally, each document in the system is examined to see that it is well defined. Consistency and completeness checking is a large part of the auditor's task, and thus the machine algorithm easily assists the auditor with automatic model construction and the initial testing.

The major value of the TICOM II system, however, is its ability to solve automatically queries about the internal control model. This is executed by means of a query language. The query language is expressed in a predicate calculus format, with all predicates (e.g., purchasing (p) means p is in the purchasing department) and terms defined in the model context. The types of queries the system can answer include the following: (1) Can an individual or department access a set of documents and assets? (2) Can a member of a given set of commands follow (precede) another set of commands? and (3) Can a set of conditions characterized by the predicate calculus query language ever be true in the firm model? These queries respond to most internal control issues currently examined in the literature. The mechanics of query answering and the computational complexity are examined in Bailey et al.[18]

Although the modeling procedure, consistency and completeness checking, and query processing have been extensively examined, the interface to an OIS is still under investigation. Intuitively, the procedure is to preprogram the TICOM II system to answer internal control queries that concern auditors. Then, every time the OIS is faced with document processing, it will be established that

Exhibit 12.7.8 Example of TICOM II Modeling



Clerk 1

Get req 1 from clerk 1 – desk–drawer;

Assign source-dept, date, items/item-num
qtys of req 1

Transfer req 1 to manager

Wait for req 1
If req 1. approval in dept-super-appr then

Copy req giving req 2, req 3;
Put req 1 into requisition-file;

Transfer req 2 to purchasing. Agent
Transfer req 3 to accounting. Accountant

Else review;
End if;
End task;

End clerk 1.

this processing is consistent with the preprogrammed security issue. Thus every document processing request may be actively shown to satisfy accepted accounting practices.

Another important use of TICOM II involves the design of the OIS. Once an OIS is proposed, it may be modeled in the TICOM II modeling language. Relevant internal control issues may then be evaluated for the proposed OIS. If violations of accepted accounting principles are discovered, additional security precautions can be implemented until the OIS is deemed acceptable. In this way TICOM II functions not only as a security enforcer for an operational system, but also as a design tool for a proposed OIS.

Accounting principles form an important part of any operating management system. In an OIS, the links between electronic and manual document processing compound the difficulty of establishing satisfactory internal controls. Verifying that such controls exist is the purpose of TICOM II. In addition, TICOM II may be used as a design tool, by detecting failures in the internal control system. Finally, it may be interfaced with an operating OIS to provide continuous updating of internal control criteria satisfaction and a desirable, continuous audit.

12.7.6 INTRODUCING AN OIS INTO THE OFFICE SETTING

Implementation of an OIS is a much-discussed subject among business and office managers. Its particularly interesting problems are (1) the magnitude of the automation task; (2) the shortness of the implementation lead time; (3) the magnitude of the clerical training; (4) the acquisition of unavailable technicians for technical support; (5) the need to interface OIS applications with EDP applications; (6) the need to develop auditability criteria and techniques; and (7) the limited managerial, personnel, and technical experience in these areas. It would be folly for us to suggest step-by-step procedures for successful OIS implementation; these will be derived only through actual experience. However, careful planning should eliminate many possible pitfalls and lessen the consequences of inevitable, unforeseen obstacles.

One of the most positive steps an office manager could take would be to formalize job description specifications. A fundamental objective of an OIS is to permit the office worker to conduct business as before, but with the aid of electronic processing capabilities. Automating office work is not primarily intended to alter individual task assignments or responsibilities significantly, although this may result. It may be true that increased productivity brought on by automation will free office workers to perform more important tasks, but this will only add to their current roles.

Therefore accurate, precise, and complete job descriptions are an essential input to the automation cycle. It is strongly suggested that the job descriptions be represented in a uniform and formal manner so that the office worker and application specialist can automate directly from the procedural specifications.

A second desirable property of the job description specification would be its machine acceptability, using a language similar in style to a computer programming language. The TICOM II language may be particularly useful in this respect. Such a representation would provide not only an operational statement of the procedures, but also a systemwide control of data flow relationships among the tasks. Compilerlike programs could check the descriptions for inconsistencies, and models of the office could be mechanically constructed from the procedural descriptions. Using the computer, office analysts could then apply similar techniques to rigorously evaluate and measure the system's performance on efficiency and internal control scales. The validated office model and the system and job descriptions are the blueprints from which an OIS can be implemented.

The cost of such a project will be expensive, but cost beneficial, over the implementation cycle. It would seriously address and partially solve some of the aforementioned problems of OIS implementation, particularly, problems 1, 2, 3, 5, and 6. In addition, rigorous analysis of current office procedures could reap immediate gains from the discovery of inefficient and inadequate internal controls.

A managerial task force of representatives from all relevant disciplines should be formed to prepare specific long-term plans for OIS implementation. Suggested task force members include computer and EDP technicians, personnel and behavioral experts, accountants and auditors, departmental representatives, and office designers. Open communications between task forces to share ideas and identify problems and solutions would be beneficial. Throughout the planning and development stages, it must be remembered that an OIS is not in itself a labor saving device. It is by no means guaranteed to solve today's office problems automatically. The philosophy and design of the OIS must take into account the needs, abilities, and limitations of the organization and its individual members. However, if the organization is ready, the automated office will be implemented successfully and beneficially.

12.7.7 IMPLICATIONS OF OIS

As the world becomes increasingly interdependent, reliance on office-type communications, information processing, and knowledge storage and retrieval will grow. The advent of conglomerate firms has required business offices to handle information supplies and requests from all markets and all parts of the globe. This has strained traditionally labor-intensive office processing to the breaking point in some industries, for example, the stock exchanges. In banking and the stock market, electronic processing and office automation have proved necessary, simply to deal with the growing volume of transactions. Other industries have not yet been quite so strained.

The effects of office automation will be startling and pervasive. As computers direct more office procedures, office workers will increasingly find their job focusing on machine interactions. As for effects on the office job itself, office workers will be offered more direction from machines providing documents to be processed and comments on the processing. Consequently, the office worker will face a corresponding decrease in the need for self-reliance in job performance. The machine also will actually perform many of the simpler tasks now done by the office worker. However, as OIS concepts permit job requirements to be designed to fit the worker, job enrichment is a potential variable available in developing systems for maximum production.

Throughout this work, we have described and discussed OIS issues we feel will greatly affect the lives of office employees and the manner in which office business will be conducted. Conceptual knowledge of these OIS design issues should give the reader, the office manager, and the office researcher insights into the problems as well as the benefits of office automation. Although we could not report on all the aspects of OISs in this short space, we have surveyed the breadth of a new and exciting field.

REFERENCES

1. M. D. ZISMAN, "Office Automation: Revolution or Evolution," *Sloan Management Review*, Vol. 19, No. 3 (Spring 1978), pp. 1-16.
2. L. D. ERMAN, H. FREDERICK, V. R. LESSER, and D. R. REDDY, "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *ACM Computing Surveys*, Vol. 12, No. 2 (1980), pp. 213-253.
3. R. B. WHITE, "A Prototype for the Automated Office," *Datamation*, April 1977, pp. 83-90.
4. WHITE, "A Prototype for the Automated Office," p. 85.
5. C. A. ELLIS and G. J. NUTT, "Office Information Systems and Computer Science," *ACM Computing Surveys*, Vol. 12, No. 1 (March 1980), pp. 27-60.

6. M. D. ZISMAN, "Representation, Specification and Automation of Office Procedures," unpublished doctoral dissertation, Wharton School, University of Pennsylvania, Philadelphia, 1977.
7. MICRO DATA BASE SYSTEMS, INC., *User's Manual* (MDBS.DMS and MDBS.DDL), Author, Lafayette, IN, 1979.
8. M. M. ZLOOF, *A Language for Office and Business Automation*, IBM Research Report #35086, IBM, Cambridge, MA, January 1980.
9. M. HAMMER, W. G. HOWE, V. J. KRUSKAL, and I. WLADAWSKI, "A Very High Level Programming Language for Data Processing Applications," *Communications of ACM*, Vol. 20, No. 11 (November 1977), pp. 832–840.
10. R. E. FIKES, "Odyssey: A Knowledge-Based Personal Assistant," *Artificial Intelligence*, in press.
11. C. A. ELLIS, "Information Control Nets: A Mathematical Model of Office Information Flow," *ACM Proceedings of the Conference on Simulation, Modeling and Measurement of Computer Systems*, Boulder, CO, August 1979.
12. C. A. ELLIS and P. MORRIS, "Office Streamlining," IRIA, Rocquencourt, France, November 6–9, 1979.
13. G. J. NUTT and C. A. ELLIS, "Backtalk: An Office Environment Simulator," *Proceedings of the 1979 International Conference Communications*, Vol. 2, June 1979, pp. 22.3.1–22.3.5.
14. DELOITTE, HASKINS, and SELLS COMPANY, "Internal Accounting Control: An Overview of the DH&S Study and Evaluation Techniques," *DH&S*, 1979.
15. J. I. CASH, A. D. BAILEY, and A. B. WHINSTON, "A Survey of Techniques for Auditing EDP-Based Accounting Information Systems," *The Accounting Review*, October 1977, pp. 813–832.
16. A. D. BAILEY, J. H. GERLACH, R. P. MCAFEE, and A. B. WHINSTON, "TICOM II–The Internal Control Language," in A. R. Abdel-khalik, Ed., *Proceedings of the Florida Symposium on Internal Controls*, March 1981, in press.
17. A. D. BAILEY, J. H. GERLACH, R. P. MCAFEE, and A. B. WHINSTON, "Internal Controls in the Office of the Future," *IEEE Computer*, May 1981, pp. 59–70.
18. A. D. BAILEY, J. H. GERLACH, R. P. MCAFEE, and A. B. WHINSTON, "TICOM II: Theory," working paper, Krannert Graduate School of Management, Purdue Univeristy, West Lafayette, IN, 1980.

BIBLIOGRAPHY

KLING, R. W., "Social Analyses of Computing: Theoretical Perspective in Recent Empirical Research," *ACM Computing Surveys*, Vol. 12, No. 1 (March 1980), pp. 61–110.

WOHL, A. D., "A Review of Office Automation," *Datamation*, February 1980, pp. 117–119.

WOHL, A. D., "Replacing the Pad and the Pencil," *Datamation*, June 1980, pp. 169–176.