MAPPING OFFICE WORK TO OFFICE TECHNOLOGY

William C. Sasso

and

Sung K. Kim

February 1986

Center for Research on Information Systems
Computer Applications and Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #115

GBA #86-8

MAPPINGS BETWEEN OFFICE WORK AND OFFICE TECHNOLOGY

William C. Sasso
Sung K. Kim

Center for Research on Information Systems
Graduate School of Business Administration
New York University
90 Trinity Place
New York, NY 10006

ABSTRACT

While several procedures designed to facilitate office analysis have achieved success with respect to describing what happens in the office, they have contributed far less with respect to prescribing how computer-based technologies can support the office. Here we present TEMO (TEchnological Mapping of Office-work), a procedure which aids the analyst in determining the feasibility of supporting a given office task and suggests which specific software packages might improve performance of that task. In order to illustrate the procedure's application, we present a case in which TEMO is applied, in step-by-step fashion, in order to assess the feasibility of automating a simple set of tasks and to assist in the selection of an appropriate software package. Directions of continuing work in the procedure's extension, enhancement, and evaluation are also described.

1. INTRODUCTION

An office analysis is performed in order to understand office activities. This understanding of office activities, which is an initial and essential step toward office automation, has two main objectives: 1) to describe activities of offices and 2) to identify activities which may be susceptible to computer support. We use this term to denote the application of office technology to improve the performance of an organization's office activities, which may be accomplished through either their "automation" or their "assistance".

"Automation" implies the presence of

1. rules, both intelligible to and executable by the technology, which describe the processes to be performed;
2. objects, on which these processes are to be performed, the complete set of which is both available to the technology and manipulable by it; and
3. control information, enabling the technology to determine which processes should be performed, when they should be performed, and whether (or not) they have been successfully performed.

"Assistance", on the other hand, involves humans actively as controllers and processors. The technology performs processing at a logically subordinate level, "doing the dirty work" for the human, because its capabilities provide only a partial match for those required by the situation. Some of the rules for processing, objects to be processed, or information required to control the process are beyond the functions provided by the technology. Thus, we use the term support as an umbrella concept, including both automation and assistance. Identifying activities and evaluating their potential for computer-support via certain technologies is the more important objective of office analysis.

While several procedures designed to facilitate office analysis have achieved success with respect to description of office activities, they have contributed far less with respect to prescribing how computer-based technologies can support the office [Sasso et al. 86a]. For example, Sutherland and Sirbu [Sutherland and Sirbu 83], in an evaluation of MIT's Office Analysis Methodology, describe it as "... a useful approach to understanding offices ..." [p. 14] but note that its "... purely descriptive approach can make diagnosis difficult in all but the simplest cases." [p. 11]. These procedures' failure to facilitate the second objective may be ascribed to their unreasonable assumption that there exists a simple and direct map between a set of activities and available technologies [Higgins and Safayeni 84].

Here, we analyze more completely the relationships between office activities and information technologies. We delineate a procedure, TEchnological Mapping of Office-work (TEMO), which defines such relationships and provides "rules" with respect to prescriptions for possible technological support. The rules involve:

1. Determining whether the office task-operation is appropriate for computer support.
2. Suggesting the nature of technology appropriate for automation or assistance of certain office activities.
3. Suggesting how to reconfigure these activities, improving their level of technological support in order to enhance organizational performance.

We feel this research has important implications for theory as well as applications in practice. Theoretically, our model establishes a set of relationships between information-processing

© 1986 ACM 0-89791-210-1/86/1000-0043 75¢

activities (described in abstract terms) and concrete, existing technologies. We believe this is important in itself, and may also form a foundation for future research in office automation. With respect to application, we note that explicit, formal procedures are generally more completely and more consistently applied than are implicit and highly judgmental ones, such as those currently available. (See, for example, OADM's discussion of systems, causes, and opportunities ([Sutherland 83] pp. 22-23), or TAM's "Rules for Automation, Support, and Mechanization" ([Sasso 85] p. 50.)) Thus, we believe the use of TEMO will enhance the analyst's ability to identify potential technological support for office work.

After a brief discussion of the previous work on office analysis, we will present the TEMO procedure and illustrate its application using a case example.

2. LITERATURE REVIEW

Methodologies on office analysis may be broadly classified into two main categories, macro-level and micro-level, according to the unit of office work they study in order to represent activities that occur within the office. The main objective of macro-level analysis is to identify key functions or processes in an office [Panko 84, Harris and Brightman 85]. The main objective of micro-level office analysis is to describe office work in terms of a set of generic tasks or operations [Sasso et al. 86b, Sirbu et al. 83, Sutherland 83, Conrath et al. 81, Conrath et al. 83].

2.1. Macro-level Methodologies

Panko [Panko 84] applied the Strategic Approach to determine the support needs of individual offices. This technique first identifies goals and functions of the office by using the Critical Success Factors Method [Rockart 79] and/or the External Contribution Approach [Drucker 67]. Second, it attempts to find strategies to achieve the goals and objectives. Then, by suggesting major changes in the strategies used by an office, it attempts to generate greater external contributions. Finally, it develops and implements tools to support the new strategies. This approach does not provide any rules with respect to mapping office activities to technologies, but rather suggests some general principles. Specifically, it suggests using certain programming tools if the strategy selected is procedural; if the strategy is not procedural, tailoring available tools to meet the needs of individual office may be necessary.

The Critical Task Method (CTM) [Harris and Brightman 85] is another approach to needs assessment evaluation. Its driving assumption is that support requirements are best prescribed by identifying the "bottleneck cognitive tasks" in an office. However, in this approach, there is an inherent assumption that every task can be supported by computerized tools. Thus, CTM does not make any specific assertions on whether a particular task is susceptible to automation or assistance. We feel an explicit mapping between tasks and technology is essential in determining the feasibility of computer support for the overall function or process.

In summary, the macro-level office analysis methodologies are complete in the sense that their outputs usually include not only descriptions of office activities, but also suggestions for possible technological support. However, their suggestions are mainly based upon descriptions of key functions or processes, removed from the larger context of information processing in the office. This isolation of bottleneck activities from their original context can easily result in the provision of enhanced technology in their support, possibly resulting in little more than a shift in the location of the bottleneck. This may lead, in effect, to sub-optimal solutions, where the suggested technological support may sometimes be appropriate to the task itself, but inappropriate to the task considered in context.

2.2. Micro-level Methodologies

The basic objective of the Office Activity Methodology (OAM) project at the University of Waterloo is to develop suitable methodologies for specifying the support needs of office systems and evaluating their impact on users [Conrath et al. 81, Conrath et al. 83]. The latest emphasis in the project is on providing insights into the work of white collar workers. Waterloo's OAM classifies information-processing activities of white collar workers into two major categories: communication tasks and processing tasks. Their procedures, however, are purely descriptive. They do not provide guidelines with respect to mapping office activities into office systems, but make the general suggestion that the feasibility of office systems applications is greatly dependent upon the degree of structure and complexity of managerial activities.

The Office Automation Group at MIT has developed a family of office analysis methodologies. The goal of their Office Analysis Methodology (OAM) [Sirbu et al. 83] is to increase understanding of office work and to specify this knowledge procedurally in order to support the process of developing computer-based office systems. The application of OAM, however, is limited by its lack of prescriptive guidelines; its focus is purely descriptive [Sutherland and Sirbu 83].

In order to overcome this limitation, OAM was extended to the Office Analysis and Diagnosis Methodology (OADM) [Sutherland 83], which includes techniques to identify operations where improvement is necessary or feasible. These techniques use new concepts (symptoms, causes, and opportunities) as a basis for the collection of additional types of data. Using these techniques, OADM makes the identification of problems and opportunities explicit, and facilitates the cost justification process, in order to help determine the feasibility of support. However, although OADM collects more data, it includes no specific guidelines for determining the relationship between the collected data and the feasibility of computer support. This process remains an essentially judgmental one. Moreover, once the determination of feasibility has been made, OADM fails to suggest any mapping between its abstract process descriptions and the office technologies currently available.

Sasso [Sasso 85] developed the Task Analysis Methodology (TAM). TAM considers office work as sets of information-handling activities and describes it using four basic concepts: information-objects, tasks, conditions, and structures of tasks. Once these descriptions have been completed, TAM proceeds to the analytical stage. The goal in this stage is to identify opportunities for improvement of an office's task configuration. These opportunities vary, from automation to assistance to mechanization, depending upon the nature of the task. Based on the information-processing operation it performs, each task is classified as algorithmic, discretionary, or complex [Sasso et al. 86b]. This approach uses insights from cognitive psychology to identify operations which humans do poorly in order to identify opportunities for applying information technology to enhance office performance. These rules help, to some degree, determine the general feasibility of office systems support. However, they are incomplete in that no mapping from an office description to currently available office technology is presented.

In summary, the micro-level office analysis methodologies focus on descriptions of office activities. Indeed, these methodologies can describe structured and semi-structured office work more rigorously than the macro-level approaches, although the latter are probably more useful for the analysis of unstructured work. However, both levels have neglected, to a large degree, the issue of mapping from an office description to suitable office technologies.

3. MAPPINGS BETWEEN WORK AND TECHNOLOGY

The TEMO procedure primarily supports the analysis phase of office analysis. The version of TEMO we present here uses as input the office descriptions produced by the Task Analysis Methodology (TAM) [Sasso 85]. First, to enhance the reader's understanding of our approach, we will describe briefly the descriptive phase of TAM office analysis. TAM considers office work as sets of information-handling tasks. By identifying information objects and the operations performed on these objects, we can describe a task structure, i.e., a set of operations performed on a particular information object (TAM-1). In TAM-2, the existing configuration of tasks is streamlined, resulting in a more efficient office process at the current level of technological support. The information-processing operations are then classified as algorithmic, discretionary, or complex, in order to suggest their potential for computer-based support in TAM-3. Table 1 summarizes these mappings from specific activities to activity classes. An example of a simple Task Structure Diagram, after being processed through TAM stages 1, 2, and 3, is shown in Figure 1.

Table 1.

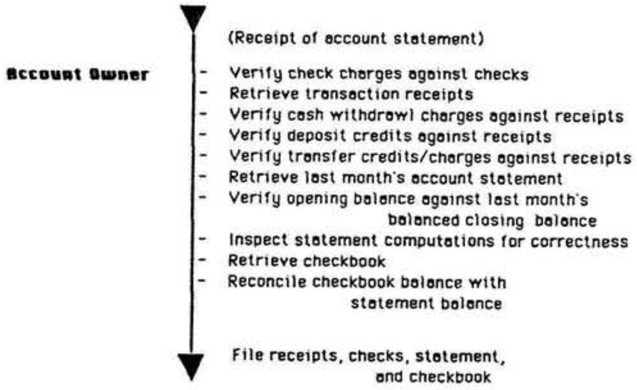
The TAM Information-Processing Hierarchy

TAM Activity-Class	General Description	TAM task-operations
Physical	Transformation of Medium	Destruction Preparation Transfer
	Communication to new location	Sending
Procedural	Algorithmic (1 to 1)*	Calculation, Coding Sorting, Inspection
	Disjunctive (1 to many)	Distribution Retrieval Selection Separation
	Correlational(many to 1)	Filing Assembly Merger Verification
Discretionary	Judgmental (1 to 1)	Acquisition Review Determination
	Analysis (1 to many)	Reconcilation
	Synthesis (many to 1)	Synthesize
Complex	Negotiation Creation	Negotiate Create

* denoting number of inputs to number of outputs

Figure 1.

Example Task Structure Diagram


Account Owner
(Receipt of account statement)
- Verify check charges against checks
- Retrieve transaction receipts
- Verify cash withdrawl charges against receipts
- Verify deposit credits against receipts
- Verify transfer credits/charges against receipts
- Retrieve last month's account statement
- Verify opening balance against last month's balanced closing balance
- Inspect statement computations for correctness
- Retrieve checkbook
- Reconcile checkbook balance with statement balance
File receipts, checks, statement, and checkbook

Task Structure for Balancing a Checkbook with a Bank-generated Account Statement

3.1. Overview of TEMO Procedure

At present, information technology is embodied in a wide variety of commercially available software products. Each product has unique functionalities and requirements. Products that have major functionalities in common can be grouped into sets, for which we use the term "tools". For example, terms such as "spreadsheet software", "electronic mail systems", and "statistical packages" denote tools. TEMO-1 maps such tools into classes of technologies with respect to their information handling facilities. TEMO-2 maps the TAM activity classes onto these classes of technologies in order to estimate the ease with which a particular activity class can be automated or assisted. TEMO-3 determines which task-operations a given software package can perform or support. TEMO-4 uses specific descriptions of both task-operations and the information-object to identify the particular capabilities required to automate or assist a given task structure. Thus, while TEMO-2 gives us a preliminary estimate of the feasibility of automating or assisting a particular task structure, TEMO-3 and TEMO-4 generate a more concrete, detailed set of functional specifications enabling us to select a software package which can support the task structure. Like the micro-level procedures whose outputs form its inputs, TEMO's applicability is limited to structured and semi-structured office work. The entire TAM/TEMO analytical process is shown in Figure 2.

3.2. Step TEMO-1

TEMO-1 maps existing tools into classes of technology. Here the assumption is made that any operation-class, to be supported, requires input, output, and storage/access technology. Though the input and output requirements are obvious, the storage/access requirement may need further explanation. It derives from the empirical observation that practically any office activity can be interrupted, suggesting the need to store work-in-process for any type of operation.

Thus, technology is classified into two broad schemes: essential and functional. The essential technologies include the input, output, and storage/access technologies. The functional technologies, on the other hand, can be said to support certain abstract task-operations as delineated by TAM. Our classification is as follows:

1. Essential Technologies

   a. Input: Convert information from human-intelligible to machine-intelligible form (e.g., Card Reader, Keyboard, Light Pen)

   b. Output: Produce information in the form of new physical objects (e.g., Printer, Plotter, Voice-output device)

   c. Storage/Access: Store information systematically so that it can be efficiently accessed later (e.g., Disk Drive, Tape Drive)

2. Functional Technologies

   a. Transfer: Send information objects from one physical location to another (e.g., Electronic Mail, Voice Mail)

   b. Data Processing: Process pre-specified information as defined by simple algorithms (e.g., Accounting Packages)

   c. Text Processing: Perform algorithmic operations on written information (e.g., Document Preparation System i.e., "Mail-Merge")

   d. Text Manipulation: Handle generating or editing written information (e.g., Word Processing System, Editors)

   e. Data Manipulation: Analyze data for better understanding about one or more information objects (e.g., Spreadsheet Software, Linear Programming Systems, Statistical Software)

   f. Decision-making: Generate and/or execute decisions using data from information objects (e.g., Expert Systems, Intelligent Decision Support Systems)

TEMO-1 (and its "descendant", TEMO-3) are the technology-oriented aspects of the overall procedure. For this reason, their application is needed only when the technology has advanced, e.g., when a distinctly new type of software becomes commercially available. TEMO-2 and TEMO-4, on the other hand, are used to analyze the organization's information-processing systems. Hence, they are applied in each TEMO analysis.

3.3. Step TEMO-2

The goal of this step is to generate a preliminary estimate of the feasibility of supporting a given task structure through the application of information technology. In order to do this, we map the technology classes produced by TEMO-1 onto the categories of office activities produced in TAM-3. Table 2 depicts this high-level mapping from office work to office systems.

FIGURE 2.

THE TEMO PROCEDURE

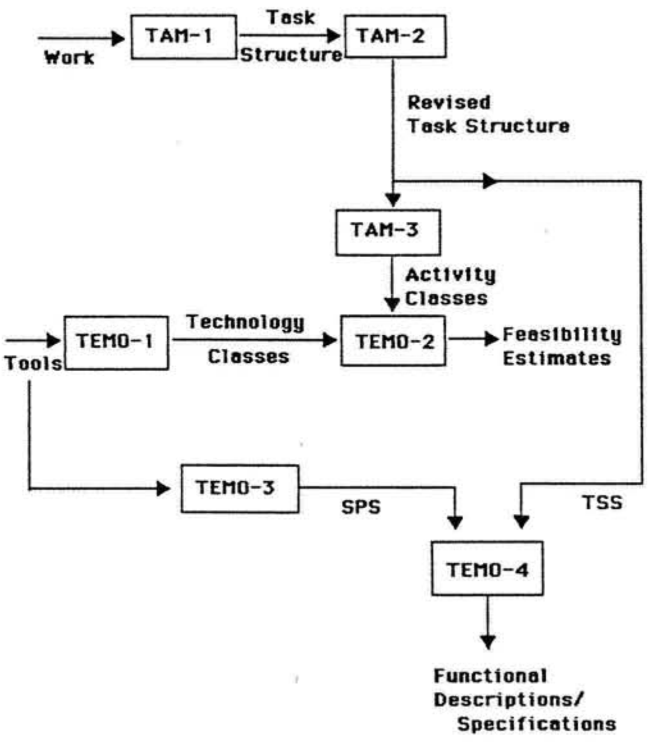


Table 2 relates technologies to classes of office activity. Because the essential technologies of input, output, and storage are necessary (but insufficient) for support of any activity, they are shown as "required" in support every row. Beyond this, the table uses activity class information (from the TAM office description) to identify levels of technology which can automate or assist each activity class. For a given task structure, one would identify the most heavily judgmental activity class present, and infer the least advanced (and generally least expensive) level of technology appropriate for its automation or assistance. For example, where discretionary activities are present, assistance may be provided via intermediate technologies, but automation will require the application of decision-making technology (and even then cannot be guaranteed). Alternatively, we may decide to "ignore" the discretionary activities, leaving them to be accomplished via human intervention, which enables us to employ a less costly technology (such as data processing) to support the remainder of the task structure.

The table would be utilized as follows. Suppose we are investigating potential support for a task structure which includes two communication operations, three transformation operations, a correlation operation, and four algorithmic processing operations. To automate this task structure in toto, i.e., in an integrated rather than a piecemeal fashion, we need to identify a software system with transfer and processing capabilities, in addition to the individual input, output, and storage/access capabilities. Since these capabilities can automate, rather than merely assist, all the operations present in our task structure, we see that potential for its complete automation exists, in this case regardless of whether it processes data or text.

Had our hypothetical task structure included even a single judgmental or analytical task, then its complete automation would become more difficult, for it could be accomplished only via the application of advanced, complex technologies in the decision-making class, such as an expert system. Moreover, had it included a single negotiation or creation operation, we would be forced to

Table 2.

Interactions between Activity Classes and Techniques

<table>
<tr><th rowspan="3">ACTIVITY CLASSES</th><th colspan="7">TECHNOLOGIES</th></tr>
<tr><th colspan="4">BASIC</th><th colspan="2">INTERMEDIATE</th><th>ADVANCED</th></tr>
<tr><th>Essential</th><th>Transfer</th><th>Data Processing</th><th>Text Processing</th><th>Data Manipulation</th><th>Text Manipulation</th><th>Decision Making</th></tr>
<tr><td>Transformation</td><td>Req/Auto</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Communication</td><td>Req</td><td>Support</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Algorithmic</td><td rowspan="4">Req</td><td></td><td colspan="5" rowspan="3">Automate</td></tr>
<tr><td>Disjunction</td><td></td></tr>
<tr><td>Correlation</td><td></td></tr>
<tr><td>Discretionary</td><td></td><td></td><td></td><td colspan="2">Assist</td><td>Support</td></tr>
<tr><td>Negotiation</td><td rowspan="2">Req</td><td>Assist</td><td></td><td></td><td colspan="3">Assist</td></tr>
<tr><td>Creation</td><td></td><td></td><td></td><td colspan="3">Assist</td></tr>
</table>

rule out automation in toto, regardless of the price we might be willing to pay. Although support of the task structure's other operations might do much to improve its efficiency and accuracy, the presence of judgmental operations, such as negotiation or creation, implies that human intervention is required for the task structure's successful completion.

If the TEMO-2 results suggest that our task structure is a viable candidate for support, we continue the process in a more specific fashion, by performing the third and fourth steps in the TEMO procedure.

3.4. Step TEMO-3

The goal of TEMO's third step is to specify the capabilities of existing software packages in terms of the TAM operations which they can assist or automate. To this end we employ a concept called the Software Package Set (SPS) which is defined as the set of all TAM operations which a given software package is capable of automating. The SPS is complemented by the Extended Software Package Set (ESPS), which includes the SPS and, in addition, any TAM operations for which the package can provide some type of assistance.

The SPS and ESPS are formulated in a straightforward, if rather tedious, fashion by comparing each package's capabilities with the TAM task-operations (See [Sasso 85]). A few examples may illustrate these concepts. For example, a standard line editor may automate filing, retrieval, destruction, separation, assembly, and merger, while assisting transfer and review. The first seven of these operations would form its SPS, while the latter two would be added to form its ESPS. Alternatively, a spreadsheet might automate filing, retrieval, destruction, calculation, inspection, assembly, and verification, while assisting transfer and review. Another spreadsheet might differ from the first in that verification and assembly operations are assisted, that is, verification and assembly require human intervention for their successful execution.

Through the application of TEMO-2 and a general awareness of the capabilities of different software tools, the analyst should be able to identify a number of candidate packages for support of the particular task structure in question. These are then subjected to TEMO-3 evaluation, and their SPS's and ESPS's are identified.

In closing our discussion of TEMO-3, we note that the large (and ever-increasing) number of packages currently available on the market makes this task an impossibly arduous one if it is to be carried out in an exhaustive fashion. We feel, however, that a number of heuristics may be employed by the analyst to make completion of this step both feasible and worthwhile. By limiting the field of software considered to that which (a) operates on the hardware and systems software available, (b) is priced within a certain price range, and (c) has received overall performance ratings (e.g., ease of use, reliability, etc) above a certain criterion, the analyst can reduce the field of packages evaluated to a manageable size. As TEMO's application becomes widespread, it is possible that evaluations along these lines could be conducted on a regular basis and published, facilitating the application of TEMO very significantly.

3.5. Step TEMO-4

This step uses specific descriptions of both the task-operations involved and the information-object to identify the specific capabilities required to support a given task structure. Thus, while TEMO-2 gives us a preliminary estimate of the feasibility of supporting a particular task structure, TEMO-3 and TEMO-4 give us a more concrete, detailed set of capability specifications for software packages which could accomplish the task structure's support.

Fundamentally, both task structures in organizations and software packages existing in the marketplace can be conceptualized as sets of processing capabilities or requirements. Where a task structure set (TSS) and a software package set (SPS) match very closely, the package is a viable means for supporting the task structure.

In other words, a package includes a set of functions, implicitly capable of automating a set of task-operations and assisting an additional set of task-operations. Thus, a package is a resource (or perhaps more accurately, a set of resources) capable of automating certain task-operations and of assisting others. A task structure, on the other hand, is a set of task-operations which the organization needs, wants, or finds it advisable to accomplish. The greater the intersection of a package's capabilities (in terms of the task-operations it can support) with the elements of a task structure (in terms of the task-operations it must perform), the more feasible is support of the task structure via the particular package.

This intersection may be expressed quantitatively, in the form

$$I_{a,b} = \frac{\sum_{i=1}^{n_b} k_i}{n_a}$$

where

a identifies a particular task structure,
b identifies a particular software package whose use in support of task structure a has been proposed,
$n_a = 1, 2, 3, \ldots$ and represents the number of operation-types present in task structure a,
$n_b = 1, 2, 3, \ldots$ and represents the number of operation-types supported by package b, and
$k_i = 1$ if task-operation i of b is included in a, 0 otherwise.

In some cases it may prove illuminating to enhance this measure by adding a weighting factor which reflects the number of times the operation occurs within the task structure. This weighted intersection value would be formulated as $W_{a,b}$ where

$$W_{a,b} = \frac{\sum_{i=1}^{n_b} f_i}{N_a}$$

where

$f_i = 0, 1, 2, \ldots$ and indicates the number of occurrences of operation i in task structure a, and all other variables are defined as above
$N_a = 1, 2, \ldots$ and indicates the total number of tasks present in task structure a,

Alternatively, this intersection may be depicted graphically using such familiar techniques as Venn Diagrams. Both this quantitative measure and graphic representation allow extension to consider the ESPS as well as the SPS, and facilitate comparison of different software packages with respect to a given task structure.

The advantage of considering the set of task-operations defined by the task structure rather than the individual task-operations themselves is that it reduces the likelihood of sub-optimization. To further reduce this likelihood, we can perform a "sensitivity analysis" of sorts, by extending our task structure's set of task-operations to include any additional task-operations present in the task structure sets (TSS's) of the focal task structure's precedent and subsequent task structures. TAM's task structure concept is based on the assumption that organizations "break" large-scale information-processing functions into smaller pieces of work, e.g., Accounts Receivable might be "broken" into Customer Billing, Payment Processing, and Collection. The task structure corresponds to this latter level of work activity. Thus, if we were evaluating packages with respect to support for the Payment Processing task structure, our "Extended TSS" would also include activities present in the Customer Billing and Collections task structures. The intersection of the "Extended TSS" so formed with the SPS's/ESPS's of the various packages can be used to rank them in the manner described above.

We now present an example illustrating the application of TEMO's four stages to determine potential technological support for a set of office activities described in a TAM task structure.

4. ILLUSTRATION OF TEMO'S APPLICATION

In this section, we will apply the TEMO procedure to generate a mapping from a simple example of information-processing work to a software package suitable for its support. The description we will use is that presented in Figure 1 above, which depicts a particular (hypothetical) set of activities used to balance a checkbook with a checking account statement. This description is a direct input to the TEMO-2 process; it is used in conjunction with Tables 1 and 2 to generate an estimate of the feasibility of supporting the task structure in question.

But what happened to TEMO-1? As noted in section 3.1 above, TEMO-1 need not always be applied, because the results of its application, as shown in Table 2, are fairly stable. In fact, only when a new class of tool emerges (such as the spreadsheet or the voice mail system) is the application of TEMO-1 necessary. In that case we will update Table 2, and again "shelve" TEMO-1 until another new class of tool emerges.

To apply TEMO-2, we examine our task structure description to determine which activity classes, shown in Table 1, it includes. In the example, we observe a large proportion of correlation activities, such as verification and filing. The algorithmic activity of inspection is present, as is the disjunction activity of retrieval. Finally, we note the presence of a judgmental operation, the reconciliation of checkbook and account statement closing balances. Thus, in using Table 2, our goal becomes location of the simplest technology which can automate the correlation, algorithmic, and disjunction activity-classes, and assist judgmental activity.

From the table, we see that, because the majority of the task-operations involved can be automated via basic or intermediate level tools, computer-based support for the task structure clearly

Figure 3.

SPS's and ESPS's for Hypothetical Spreadsheets

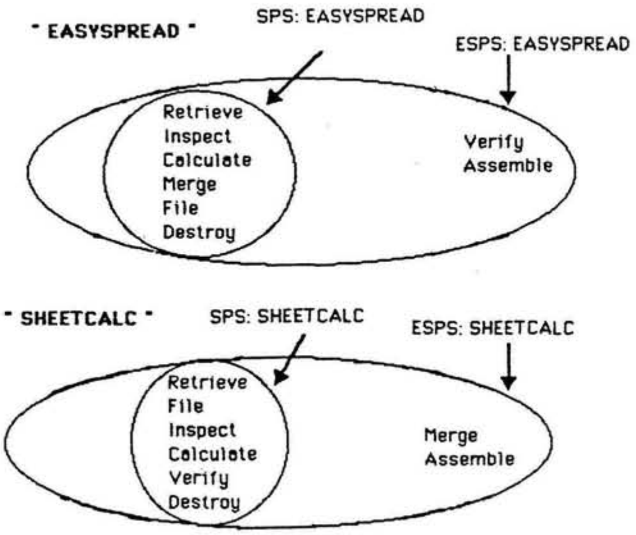
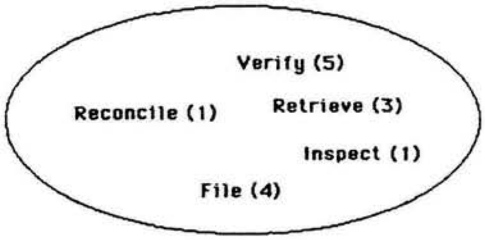


Figure 4.

Example Task Structure Set



(for Task Structure depicted in Figure 1)

deserves further study. However, the presence of the judgmental "reconciliation" operation implies that the complete automation of the process is possible only through use of advanced tools such as expert systems, which is not a practical option in this context. We also see from the table which technologies may be appropriate to support the checkbook balancing task structure. Because this task structure is a data-oriented one, the data processing and data manipulation technologies appear most appropriate. Thus, the rough results we achieve through application of TEMO-2 are positive, encouraging us to continue the process through TEMO-3 and TEMO-4.

In TEMO-3 we evaluate the specific capabilities of particular packages, and produce software package sets (SPS's) and extended software package sets (ESPS's) depicting those specific capabilities. For the purposes of this example, we will restrict our TEMO-3 analysis to two hypothetical spreadsheets. Suppose that, by reviewing the documentation of the two packages, talking with their developers' personnel, and reading published reviews, we determine their capabilities to be as follows. "Easyspread" can automate filing, retrieval, destruction, calculation, inspection, and merge operations; and can assist verification and assembly operations. "Sheetcalc", on the other hand, can automate filing, retrieval, inspection, destruction, calculation, and verification, while it assists merger and assembly. Figure 3 shows the software package sets and extended software package sets for Easyspread and Sheetcalc.

Our first objective in TEMO-4 is to determine the Task Structure Set (TSS) for our task structure under study. Given the Task Structure Diagram in Figure 1, this is a straightforward process, and the Task Structure Set depicted in Figure 4 is derived.

Once we have derived the Task Structure Set for the particular task structure under investigation, we can compare it with the Software Package Sets and Extended Software Package Sets for the particular products being considered as support alternatives Using the methods described in section 3.4 above, we derive the intersection values in Table 3.

The presence of the reconciliation task, which cannot be automated without application of advanced technologies such as expert systems, implies that complete automation of the task structure is not feasible. This is shown in our table by the fact that no values of 1.00 are present. A value of 1.00 indicates that the TSS is a subset of the SPS (or ESPS), implying that the package has all capabilities required to automate (or in the case of an ESPS, automate or assist) the entire task structure.

Both the base values and the weighted values presented in Table 3 suggest that Sheetcalc, by virtue of its automated verification capabilities, is a more appropriate package for supporting the checkbook balancing process than is Easyspread. This is apparent when we consider their simple TSS/SPS intersection values, .8 and .6, respectively. The distinction becomes even more apparent when we study the weighted intersection values, which consider the frequency of a particular operation's occurrence in the task structure. The disparity in these intersection values becomes even more dramatic, with Sheetcalc's .93 overwhelming Easyspread's .57.

Alternatively, we can represent this information pictorially by combining the software sets shown in Figure 3 with the Task Structure Set shown in Figure 4. Figure 5 depicts this combination of the previous figures.

In this simple example we have illustrated the application of the TEMO procedure to assist in the selection of a particular software product as suitable for the support of an information-processing procedure. The example, while admittedly over-simplified, depicts the basic sequence of activities an analyst would perform in a more complex "real-world" situation. We believe that the procedure can be applied to application on a much larger scale with little, if any, major change, so long as the analysis focuses on the evaluation of commercially available software packages.

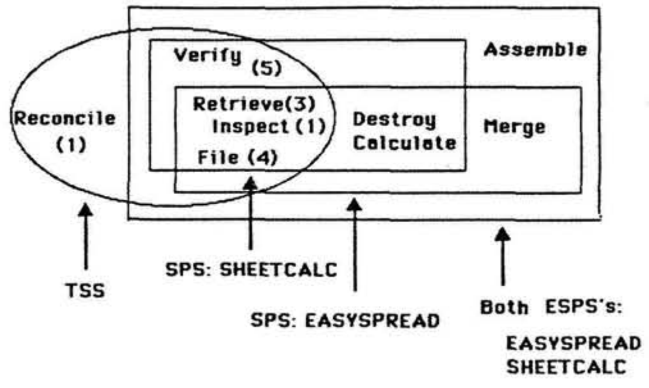
Table 3.

SPS, ESPS, and TSS Interaction Values

	Base Values	Weighted Values
EASYSPREAD		
TSS/SPS	.6	.57
TSS/ESPS	.8	.93
SHEETCALC		
TSS/SPS	.8	.93
TSS/ESPS	.8	.93

Figure 5.

TSS/SPS/ESPS Interactions



5. DISCUSSION AND SUMMARY

5.1. Directions for Research

Though TEMO provides insights into the relationship between office activities and available technologies, its presentation here has several limitations. First of all, the validity of the TEMO procedure has not been tested with empirical data. In other words, we do not know whether its recommendations result in better software selection decisions than those generated using other formal procedures, common sense approaches, or even those produced randomly. Secondly, the reliability of TAM descriptions, the inputs to the TEMO procedure, has not been established. That is, it is possible that different analysts using TAM might produce different task structure descriptions, especially with respect to coding task-operations. Obviously, the TEMO package selection is quite sensitive to the TAM task-operation codings. Thus, at this moment, we do not know how good our TEMO mappings are. Moreover, the TEMO procedure is (intentionally) incomplete in that software functionality is its only evaluation criterion. Furthermore, the evaluation is measured in binary values, e.g., whether the package can assist or automate an operation (or not). Finally, each application of TEMO (especially TEMO-3) will involve a great deal of effort until a publicly available body of SPS's and ESPS's has been built up. Without the ability to evaluate software selection decisions independently and objectively, we must consider the possibility that TEMO's application requires an excessive amount of effort.

Several of these problems form intriguing topics for future research. These include, for example, the evaluation of TEMO recommendations and the reliability of TAM office descriptions. Also, our future efforts will extend the model presented here, taking into account additional dimensions of software and its application context. For example, the inclusion of task cost data as a weighting factor in the computation of TSS/SPS intersection values is a straightforward extension of the procedure. We believe this future research can facilitate more precise and complete mappings from organizational needs to the successful application of information technology.

5.2. Summary

This paper has presented an extension of existing office analysis procedures. This extension is embodied in the TEMO procedure, mapping office work onto information technologies. The main theme of this procedure is to establish more complete, concrete relationships between a set of activities and technologies suitable for their support. As an input, the TEMO procedure uses a specific task structure that is produced by TAM (See [Sasso 85]) and described in terms of a set of operations performed on a particular information object. Operations within the task structure are mapped into an abstract set of classes of technology, in order to provide an estimate of the likelihood that the particular activity can be supported. If the task structure is determined to be a promising candidate for support, the second half of the TEMO procedure is applied. This produces a more concrete, detailed set of capability specifications, identifying software packages appropriate for the task structure's support.

In practice, this two-phase analysis enables the analyst not only to determine the susceptibility of the office activities to computer support, but also to identify commercially available software suitable for that support. Theoretically, our approach extends existing office analysis approaches by adding capabilities for prescribing the application of specific technologies. Finally, our model may provide a foundation for future research in the evaluation of office automation, by providing a basis for integrating organizational needs for information-processing support with commercially available software.

ACKNOWLEDGMENTS

We would like to thank Margrethe Olson, of New York University, for having drawn to our attention this basic problem, i.e., the lack of mappings between work and technology in previous office analysis work, as well as her detailed and thought-provoking comments on an earlier draft of this paper. We are equally grateful to Ted Stohr, also of NYU, for his encouragement and valuable suggestions concerning our initial approaches to developing such a mapping. We appreciate also the helpful comments of the three anonymous OIS'86 reviewers.

References

[Conrath et al. 81] Conrath, D., C. Higgins, C. Thachenkary, and W. Wright. The Electronic Office and Organizational Behavior-Measuring Office Activities. *Computer Networks* 1:401-410, 1981.

[Conrath et al. 83] Conrath, D., C. Higgins, R. Irving, and C. Thachenkary. Determining the Need for Office Automation: Methods and Results. *Office:Technology and People* 1:275-294, 1983.

[Drucker 67] Drucker, P. *The Effective Executive.* Harper & Row, 1967.

[Harris and Brightman 85] Harris, Sidney and Harvey Brightman. Design Implications of a Task-Driven Approach to Unstructured Cognitive Tasks in Office Work. *ACM Transactions on Office Information Systems* 3(3), 1985.

[Higgins and Safayeni 84] Higgins, C. and F. Safayeni. A Critical Appraisal of Task Taxonomies As a Tool for Studying Office Activities. *ACM Transactions on Office Information Systems* 2(4), Oct., 1984.

[Panko 84] Panko, Raymond. 38 Offices: Analyzing Needs in Individual Offices. *ACM Transactions on Office Information Systems* 2(3), July, 1984.

[Rockart 79] Rockart, J. Chief Executives Define Their Own Data Needs. *Harvard Business Review* 57(2), March-April, 1979.

[Sasso 85] Sasso, William C. *A Comparison of Two Potential Bases for Office Analysis: Function and Organizational Unit.* PhD thesis, The University of Michigan, 1985.

[Sasso et al. 86a] Sasso, W., J. Olson, and A. Merten. *Procedures for Office Analysis: A Critical Review.* Technical Report 119, New York University, Graduate School of Business Administration, CRIS, April, 1986.

[Sasso et al. 86b] Sasso, W., J. Olson, and A. Merten. The Practice of Office Analysis: Objectives, Obstacles, and Opportunities. 1986. in preparation.

[Sirbu et al. 83] Sirbu, M., S. Schoichet, J. Kunin, M. Hammer, J. Sutherland, and C. Zarner. *Office Analysis: Methodology and Case Studies.* Technical Report LCS/TR-289, Massachusetts Institute of Technology, 1983.

[Sutherland 83] Sutherland, Juliet. *An Office Analysis and Diagnosis Methodology.* Technical Report LCS/TR-290, Massachusetts Institute of Technology, 1983.

[Sutherland and Sirbu 83] Sutherland, Juliet and Marvin Sirbu. *Evaluation of an Office Analysis Technology.* Technical Report LCS/TM-239, MIT, March, 1983.