A FRAMEWORK FOR ADDRESSING END USER TRAINING NEEDS

Jane M. Mackay
Texas Christian University

Charles W. Lamb, Jr.
Texas Christian University

Although end user computing (EUC) training has been widely recognized as an important topic, very little published research has focused on identifying the limitations of current training protocol and/or developing innovative approaches to enhancing the productivity of computer users. This paper examines the efficacy of the Functional Job Analysis classification for designing EUC training programs.

The growth of end user computing has been one of the most significant information management developments of the 1980s (Benson, 1983). End user computing (EUC), the use and/or development of information systems by the principal users of the systems’ outputs or by their staffs (Wetherbe & Leitheiser, 1985), has diffused rapidly over the past two decades. This diffusion has generated substantial interest within the management information systems (MIS) discipline (Doll & Torkzadeh, 1989; Magal, Carr, & Watson, 1988; Rivard & Huff, 1988).

It has also created an unprecedented need for end user training (Magal et al., 1988). Studies by Mackay (1987), Hughes (1988), and Mykytyn (1988) support the proposition that software products are not easy to learn or use. They maintain that ease of use is highly dependent upon effective end user training.

This paper further develops the proposition that end users need training to use software efficiently and effectively. Specifically, the purpose of this paper is to apply a conceptual job analysis framework called Functional Job Analysis (FJA) to EUC training. FJA, based on a philosophy of hierarchical training from simple to complex, provides an approach which might improve end user training.

The paper is divided into the following five sections: (1) a review of the EUC training literature; (2) an overview of functional job analysis; (3) a description of the steps involved in using FJA to develop EUC; (4) implications for
EUC training and (5) defining the research agenda.

A Review of EUC Training Literature

Most of the past research conducted on EUC training can be characterized as exploratory or descriptive (Ein-Dor & Segev, 1988; Nelson & Cheney, 1987; Rivard & Huff, 1988). Sein, Bostrom and Olfman (1987) reported that little research has been conducted in the area even though the contribution of effective training towards the success of EUC has been acknowledged by both practitioners and researchers in MIS.

One important series of experiments focused on the effects of different instructional formats on learning (Carroll, Mack, Lewis, Grischkowsky, & Robertson, 1985; Carroll, Smith-Kerker, Ford, & Mazur-Rimet, 1988). They reasoned that trainees tend to be overwhelmed by the numerous features available in an application software package and that novice users should only be presented with the full system after they have gained confidence in their ability to use the software (Carroll & Carrithers, 1984). They used the term “training wheels” to refer to the idea of restricting the number of computer software features available to beginners.

Another experimental approach to avoid inundating trainees with information was reported by Carroll et al. (1985). The researchers used a set of instructions designed to encourage “active” learning by the trainees. These software instructions were “guided exploration” cards which consisted of a goal statement, hints, check-points, and remedies for that card’s topic. Thus, users were presented with the minimal instructions necessary to complete the task.

These guided exploration cards evolved into what Carroll et al. (1988) called the “Minimal Manual.” The manual is based on the principle that novice users of application systems need to use a tool that they believe will help them do their own work. The manual specifically trains users in error recognition and correction. Users are expected to approach the learning task with considerable understanding of task relevant concepts and to be motivated to use the tool. The rationale for this approach is that training should make it easier for users to apply the knowledge they already possess in performing job-related tasks (Carroll et al., 1988).

Although the Minimal Manual was successful in increasing productivity compared to self-instruction manuals, neither method utilized a strategy of training users sequentially from the least complex function to the most complex function. An alternative method of training proposed in this paper entails adopting a hierarchical complexity approach of first training users to master simple commands before attempting the more complex commands (Fine & Wiley, 1971). Users who attempt complex procedures without having learned the simple commands may be less efficient in accomplishing their tasks.

The importance of organizing EUC training from the least complex to the most complex tasks has been documented by Mackay (1987). She concluded that novices should be trained to use simple functions before they attempt to learn more complex functions. The rationale is that an understanding of more complex functions is often dependent upon understanding simple functions that are used in a series of complex key strokes.

Mackay and Lamb (1989) also concluded that EUC training based upon a hierarchical
complexity framework makes sense. They proposed a research agenda comparing productivity rates of users trained in a hierarchical format versus others trained using non-hierarchical formats. One method of establishing such a hierarchy is FJA.

An Overview of Functional Job Analysis

The United States Training and Employment Service (USTES) developed FJA for several purposes including developing job summaries, job descriptions, employee specifications, and measuring worker activity levels (McCormick, 1979). This system is widely used in the public sector, and used to a lesser extent in the private sector of the economy (McCormick, 1976; Milkovich & Newman, 1987). FJA may be the strongest single influence on job analysis practice in the United States (Milkovich & Newman, 1987).

Figure 1 shows the FJA functions associated with data, people, and things. A combination of the highest functions which workers perform in relation to data, people, and things expresses the total level of complexity of the job-worker situation (U.S. Department of Labor, 1972). The USTES has used these worker functions as a basis for describing over thirty thousand job titles in the Dictionary of Occupational Titles (U.S. Dept. of Labor, 1977).

Figure 2 provides a description of the six data function scales listed in Figure 1. Data are described as:

...information, knowledge, and conceptions related to data, people, or things resulting from observation, investigation, interpretation, visualization, and mental creation. Data are intangible and include numbers, words, symbols, ideas, concepts, and oral verbalization (U.S. Department of Labor, 1972, p.73).

Fine and Wiley (1971) have noted that involvement with data is inherent in even the simplest job instruction. Data are always present in a task even though the major emphasis might be dealing with things and/or people. Where things are primarily involved, data tend to show up as specifications. Where people are primarily involved, data tend to show up as information about objective events or conditions, information and/or feelings.

Steps Involved in Using FJA to Develop...
LEVEL  DEFINITION

0  SYNTHESIZING
Takes off in new directions on the basis of personal intuitions, feelings, and ideas (with or without regard for tradition
experience or existing parameters) to conceive new approaches to, or statements of problems and the development of
system, operational, or aesthetic solutions or “resolutions” of them, typically outside of existing theoretical, stylistic,
or organizational context.

1  COORDINATING
Decides time, place, and sequence of operations of a process, system, or organization, and/or the need for
revision of goals, policies (boundary conditions), or procedures on the basis of analysis of data and of
performance review of pertinent objectives and requirements. Includes overseeing and/or executing deci-
sions and/or reporting on events.

2  ANALYZING
Examines and evaluates data (about things, data, or people) with reference to the criteria, standards, and/or requirements
of a particular discipline, art, technique, or craft to determine interaction effects (consequences) and to consider
alternatives.

3  COMPILING
Gathers, collates, or classifies information about data, people, or things, following a schema or system but using
discretion in application.

4  COMPUTING
Performs arithmetic operations and makes reports and/or carries out a prescribed action in relation to them.

5  COPYING
Transcribes, enters, and/or posts data, following a schema or plan to assemble or make things and using a variety of
work aids.

6  COMPARING
Selects, sorts, or arranges data, people, or things, judging whether their readily observable functional, structural, or
compositional characteristics are similar to or different from prescribed standards.

Upjohn Institute for Employment Research: 32-33.

Figure 2: Data Functions Descriptions

Figure 3: Task Evaluation Process

<table>
<thead>
<tr>
<th>STEP 1</th>
<th>Determine Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP 2</td>
<td>Divide Task into Subunits</td>
</tr>
<tr>
<td>STEP 3</td>
<td>Rank Subunits Based on Complexity (Low=6; High = 0)</td>
</tr>
<tr>
<td>STEP 4</td>
<td>Assign Task Rank of Most Complex Subunit</td>
</tr>
<tr>
<td>STEP 5</td>
<td>Determine Training Requirements for Task and Subunit</td>
</tr>
</tbody>
</table>
Figure 3 shows the steps involved in developing a training program based on FJA. In order to put into practice the principle of teaching less complex procedures before teaching more complex procedures, a particular task must first be defined; preparing a departmental budget, for example.

Second, the subunits which comprise the task must be defined. The subunits of budget development include entering budget categories, entering and formatting appropriate amounts and determining budget totals. Each of these subunits may be further subdivided if necessary.

Following the breakdown of the task into subunits, the subunits must be ranked according to complexity from most to least complex. In this example, determining amount totals is the most complex, and would be assigned level 4, computing. Entering the budget amounts in the appropriate format and entering the budget categories are both copying, level 5.

Then the overall task must be assigned the highest complexity ranking associated with a subunit of the task. For budget creation, this is the calculation of totals. Based on the FJA hierarchy, the rank of the task would be level 4, computing, equal to the rank of the most complex subunit.

The final step is to determine the training requirements for the task. This includes (1) conceptualizing the layout of the budget information, (2) entering the appropriate budget categories and amounts, (3) formatting amounts, and (4) entering formulas to calculate the appropriate totals. The training program would cover, in order, design, entry and calculation. It is, of course, assumed that appropriate error recovery techniques would be included with each command taught.

Figure 4 shows how business software commands are related to the various FJA data levels for a variety of software applications. FJA data levels are hierarchical which means that if a user is performing a compiling function, then he or she must know how to perform computing, copying, and comparing functions.

As Figure 4 shows, the lowest FJA level - comparing - entails activities which occur prior to all other functions.

---

**Figure 4: Examples of Software Functions Associated with FJA Data Levels**

<table>
<thead>
<tr>
<th>FJA Data Level</th>
<th>Word Processing</th>
<th>Spreadsheets</th>
<th>Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Synthesizing</td>
<td>Integrating graphics into text</td>
<td>Using conditional if statements</td>
<td>Relating fields of different files</td>
</tr>
<tr>
<td>1 Coordinating</td>
<td>Creating form letters to use with merge</td>
<td>Writing macros</td>
<td>Programming, creating standard input forms</td>
</tr>
<tr>
<td>2 Analyzing</td>
<td>Editing texts</td>
<td>Executing selects, what if tables</td>
<td>Performing selects from databases</td>
</tr>
<tr>
<td>3 Compiling</td>
<td>Executing mail merge</td>
<td>Moving, copying or sorting</td>
<td>Creating reports in arranged format (sorted)</td>
</tr>
<tr>
<td>4 Computing</td>
<td>Printing documents</td>
<td>Performing arithmetic functions</td>
<td>Creating reports without data arranged</td>
</tr>
<tr>
<td>5 Copying</td>
<td>Organizing and entering text</td>
<td>Entering data into the spreadsheet cells</td>
<td>Entering data into the database fields</td>
</tr>
<tr>
<td>6 Comparing</td>
<td>Determining type of document to be produced</td>
<td>Making visual comparisons of data without the use of software commands</td>
<td>Determining which data are entered into which fields</td>
</tr>
</tbody>
</table>

*FJA data levels are arranged from the most complex to the least complex.
to software utilization. In each case the user must
make some comparison of precomputerized data.
Users of spreadsheet software, for example, can
make a visual comparison of data on the com-
puter screen without any manipulation. At the
comparing level, users do not manipulate soft-
ware, but make precomputer decisions about
procedures to be followed.

Synthesizing is at the top of the hierarchy
and is the most complex FJA data level. In
spreadsheet software, using conditional ‘if state-
ments’ to perform syntheses of manipulated data
falls into this level. Conceptualization and execu-
tion of this procedure is difficult. To synthesize
tasks, users need to understand the lower-level
tasks that result in synthesizing. To synthesize,
users should know how to execute the functions
in the levels of coordinating, analyzing, compili-
ing, computing, copying, and comparing.

In developing the training program for
the budget task, the appropriate software would
be a spreadsheet. Using Lotus 1-2-3 as an ex-
ample, a training course can be devised that takes
the user from the least complex software com-
mands to the most complex commands needed to
complete the task. Figure 5 illustrates the se-
quence in which these commands should be
taught to the user. The order is from design to

Figure 5: Example of FJA Training Applied to Budget Task

<table>
<thead>
<tr>
<th>FJA Data Level*</th>
<th>Spreadsheet Definition</th>
<th>Software Commands*</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Computing</td>
<td>Performing arithmetic functions</td>
<td>Enter formulas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter total functions (@SUM)</td>
</tr>
<tr>
<td>5 Copying</td>
<td>Entering data into the spreadsheet</td>
<td>Enter labels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Position labels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format values</td>
</tr>
<tr>
<td>6 Comparing</td>
<td>Making visual comparisons of data without the use of software commands</td>
<td>Visually examine budget data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and develop layout</td>
</tr>
</tbody>
</table>

*FJA data levels are arranged from the most complex to the least complex
+Software commands are arranged within each level from least complex to most complex

entry to calculation.

Thus, how users perform tasks using soft-
ware will depend upon the complexity of the
functional area task and the level of expertise that
the user has reached with the software. The user
will not always employ different software with
the same level of expertise, since tasks using
different software applications may differ in
complexity. Training should be linked to the
complexity of the task the user is expected to
accomplish.

Implications for EUC Training

Dodge and Lindholm (1987) note that
there are probably 50 ways to train a user. How-
ever, most of them are inefficient and/or ineffec-
tive (Mackay, 1987; Hughes, 1986; Mykytyn,
1988; Sein et al., 1987). The reason for this is that
they focus on the software rather than on the task.

The FJA hierarchy of functions suggests
an entirely different approach to EUC training
than is commonly practiced. The FJA approach
suggests that the focus should be on the task the
dum user is to perform. Within the task domain,
training proceeds from the least to the most
complex subunit of the task and the software
commands necessary to accomplish each sub-
If a task only requires the worker to perform lower-level functions such as comparing and copying, the FJA framework suggests that no need exists to train them in computing, analyzing, or other higher level functions. Therefore, the level of required training is dependent upon the accurate identification of all processes necessary to complete the task. This task analysis may require more time using the FJA framework than was previously allocated. However, the time savings will be realized in the training programs and in the end user’s effectiveness.

FJA also provides a useful framework for developing data to be used in achievement and performance tests focusing on job content. The task data translate directly to information on the questions of “how to” and “know how” knowledge or skills, especially in performance on machines (Fine, 1986). It is an excellent means for developing performance standards in relation to job content. The paradigm, “To do this task, to these standards, the worker needs this training” establishes the task relevance of the standards and the degree to which they must be achieved for satisfactory performance (Fine, 1986).

FJA produces communicable information about productivity. Since each task ends with a result and the results must add up to required outputs, management can track the efficiency and effectiveness with which jobs are designed to achieve management’s goals and objectives. Through the use of the FJA framework, managers may realize that certain tasks should be grouped together, and, therefore, job descriptions could be streamlined to reflect more similar tasks. This would allow users to attain a higher level of expertise in fewer areas rather than dividing their efforts over numerous areas.

The validity and usefulness of the FJA framework for EUC training need to be investigated. Several questions that need to be addressed are identified in the following section.

**Defining the Research Agenda**

The FJA classification has been extensively tested in a wide range of applications. Its utility as a framework for EUC training has not, however, been examined. Research questions which need attention include:

- Will users trained based upon the FJA framework express different levels of satisfaction with their training than users not trained with the FJA framework?
- Will users trained within the FJA framework learn faster or slower than those not trained with the FJA framework?
- Will the retention of users be higher, lower, or the same in an FJA training framework as those trained in a non-FJA framework?
- Will user productivity levels be affected by the training framework employed?
- Will the command of the software and its capabilities be affected by the type of training employed?
- Will FJA enhance users’ building of mental models?
- Will users’ information processing strategies be more effective if they are trained within an FJA framework?
- Will the simple-to-complex-functions concept facilitate their problem solving processes?
- Will the process of simple-to-complex-functions training frustrate users?

These questions are illustrative of the research issues that need to be addressed. With the increase in EUC, and the concomitant increase in training budgets, the importance of effective training for the software use has be-
come more imperative.

References


Jane M. Mackay is currently an assistant professor of management information systems at Texas Christian University in Fort Worth, Texas. She has published in the Journal of Health Care Marketing, Interface, and Government Publications Review. Her research interests are in decision support systems, the impact of technology on decision making, behavioral implications of technology and end user training.

Charles W. Lamb, Jr., is the M.J. Neeley Professor of Marketing in the Neeley School of Business at Texas Christian University. His primary teaching, research, service, and consulting interests are in the areas of marketing management and strategy and marketing for nonprofit organizations. He has contributed over 50 articles to academic and professional journals, including *Journal of Marketing Research, Journal of Business Research, Journal of Academy of Marketing Science, and the Journal of Health Care Marketing*. 
Related Content

Work Ethic Differences Between Traditional and Telework Employees
www.irma-international.org/article/work-ethic-differences-between-traditional/55760/

Exploring the Effects of Hardware Performance, Application Design and Cognitive Demands on User Productivity and Perceptions
www.irma-international.org/chapter/exploring-effects-hardware-performance-application/4467/

Adaptation Engineering in Adaptive Concept-Based System
www.irma-international.org/chapter/adaptation-engineering-adaptive-concept-based/18263/

Collaborative Knowledge Management in the Call Center
www.irma-international.org/chapter/collaborative-knowledge-management-call-center/38087/

Computational Engineering in the Cloud: Benefits and Challenges
www.irma-international.org/article/computational-engineering-cloud/58546/