Human–Computer Interaction – INTERACT '90
D. Diaper et al. (Editors)
Elsevier Science Publishers B.V. (North-Holland)

Integrated Office Software Benchmarks: A Case Study

James R. Lewis
Suzanne C. Henry
Robert L. Mack

User Interface Institute IBM T. J. Watson Research Center, Hawthorne P.O. Box 704, Yorktown Heights, NY 10598

In this paper we present a case study of a benchmark evaluation of integrated office systems. The case study includes developing scenarios, benchmark measures, and quantitative and qualitative analysis of user performance and user problems. We studied two systems, one loosely integrated windowing environment and one more tightly integrated (with respect to consistent graphical interface style). Multivariate analyses showed that significant differences were attributable to performance/analytical variables and to patterns of error impact classifications, but not to subjective ratings. Somewhat surprisingly, users experienced serious problems with the seemingly more integrated (consistent) system largely because of a handful of serious problems. This was taken as evidence that improvement of the poorer performing system should be based primarily on an analysis of errors. Some examples are presented to indicate the potential diagnostic value of analyzing problems and the development of testable behavioral objectives from benchmark measures.

1.0 INTRODUCTION

Evaluating computer systems using realistic scenarios is a common practice in the software industry. Formal quantitative benchmark evaluations, in particular, are useful in a usability engineering context to help set measurable usability targets to guide iterative design of developing systems (see Gould, 1988; Whiteside, Bennett & Holtzblatt, 1988). The key characteristics of benchmarking are measuring user performance across a set of systems or techniques of interest, using common scenarios, measures and test procedures so that meaningful comparisons can be made of these measured objects (Williges, Williges & Elkerton, 1987). Surprisingly few such benchmark studies have been published however, particularly for software applications reflecting the current generation of integrated applications and graphical direct manipulation interface styles. Roberts and Moran (1983) published an early benchmarking case study involving text editing systems and tasks, and relatively expert users. Roberts and Moran conclude that as a whole, the evaluation methodology provided an objective, multidimensional picture of the functional and usability characteristics of text editors. In a more recent case study, Whiteside, Jones, Levy and Wixon (1985) compared seven systems contrasting command-, menu- and direct manipulation interface styles for relatively novice users performing a complex file manipulation task. Their main conclusion was that graphical direct manipulation interfaces had numerous problems of their own, different from, but comparable in impact to problems commonly associated with command and menu-based interface styles.

In this paper we present a case study of comparative benchmark evaluation in the domain of integrated office software, and non-expert users for whom problems may be a more salient experience than through-put. Our method is similar to that described by Roberts and Moran (1983), but we focus on on the rationale for the specific scenarios we selected and the results of a benchmark evaluation aimed at answering three questions:

- What are the obtained values of our usability measurements?
- How good (reliable and valid) are these measurements?
- How can this information be used to improve a system?

In order to set reasonable behavioral objectives, a limited number of human-system usability characteristics should be measured. The reliability and validity of these measurements necessarily influence the degree of confidence placed in the objectives derived from the benchmark data. We are especially interested in the possible diagnostic value of user feedback obtained in the benchmark evaluation. Such

diagnostic information is often associated with more informal qualitative evaluation methods (Gould, 1988; Whiteside, Bennett & Holtzblatt, 1988; Lewis, 1982), in contrast to formal quantitative benchmarking. Both methods are needed in a development context, but we believe there is considerably more diagnostic information in benchmark evaluations than is often appreciated.

2.0 THE OFFICE SCENARIOS: CONTENT AND RATIONALE

Table 1 summarizes ten scenarios we developed to evaluate different implementations of integrated office software packages, along with key subtasks comprising the scenarios, and the total steps in each subtask and scenario. As we write, we know of no single software package for the office which offers integrated text editing, mail, calendar, and decision support (e.g., database, spreadsheet, chart applications) for the PC hardware and system environments we are interested in. It is possible, however, to integrate individual applications to some extent using windowing platforms that enable data transfer between applications, using cut, copy and paste functions.

These scenarios are motivated by several sources, including internal marketing expertise, and published field studies and analyses of integrated software (Nielsen, Mack, Bergendorff, and Grischkowsky, 1986; see also Mack & Nielsen, 1987). Based on this work content analysis and discussions with relevant development personnel involved in office systems we initially developed a large number of tasks. Potential tasks were defined by crossing all possible objects with all possible actions, similar to the procedure described by Roberts and Moran (1983), but at a somewhat higher level. A subset of these tasks was then organized into ten scenarios. These scenarios were designed to fulfill the following goals:

- Broad coverage of the types of applications used in an office setting, such as text editing, mail, calendar, and decision support. Since office work consists of a complex set of tasks, scenarios intended to sample this task set are necessarily broad in scope.
- Some scenarios which would be accomplished most efficiently by using techniques for data transfer between applications or presenting multiple applications in windows.
- A set of scenarios which could be performed in one day by most users. This reduces the likelihood of participants dropping out of the study and allows the study to be completed in a reasonably short time.
- Scenarios which required a minimum of typing to reduce performance variability attributable to typing skill.
- Scenario tasks which are written at a high enough level for use across systems. The tasks are specific with respect to text and objects such as files, but step-by-step procedures are not specified.

Table 1. Scenario Descriptions.

- Mail 1 (M1): Open, reply to, and delete a note.
- Mail 2 (M2): Open a note, forward with reply, save and print the note.
- Calendar 1 (C1): Create a calendar entry and print today's appointments.
- Calendar 2 (C2): Open a note, open specified calendar entries, compare the note and calendar entry information, delete a calendar entry. (*)
- Address 1 (A1): Create, change, and delete address entries.
- File Management 1 (F1): Rename a file, copy a file, and delete a file.
- Editor 1 (E1): Create and save a short document.
- Editor 2 (E2): Locate and edit a document, open a note, copy text from the note into the document, save, mail and print the final version of the document. (*)
- Decision Support 1 (D1): Create a small spreadsheet, open a document, copy the spreadsheet into the document, save and print the document, save the spreadsheet. (*)
- Decision Support 2 (D2): Locate information in a calendar entry, revise a spreadsheet title using the calendar information, create a pie chart from the spreadsheet, print and save the chart, save the spreadsheet. (*)

* indicates a scenario which tests integration by requiring data transfer between applications or window manipulation.

3.0 APPROACH TO THE BENCHMARK EVALUATION

3.1 Participants

Thirty employees of temporary help agencies participated in the study, with two groups of 15 hired for evaluations in two locations. Each group of 15 con-

sisted of three groups (five participants per group), with the following characteristics:

- Clerical/secretarial with no experience using a mouse.
- Business professional with no experience using a mouse.
- Business professional with at least three months experience using a mouse with a computer system.

All participants had at least three months experience using some type of computer system. They had no programming training or experience, and had no (or very limited) knowledge of the DOS operating system.

3.2 Measures

The measures collected for each scenario are described below.

- Performance
  - Time on Task: The time to complete a scenario successfully.
  - Completion Rate: The percentage of participants completing a scenario successfully. The participant may have experienced problems (as defined below), but must have completed the scenario without assistance and with correct outputs in order for the completion to be considered successful.
  - Error Free Rate: The percentage of participants completing a scenario without any problems.
- Analytical
  - Step Counts: A step is really a subtask, and not individual physical actions. For example, *opening the file pull-down menu* is a step, but not *moving the pointer to the action bar option "file"*, followed by *pressing mouse button down*. Steps generally were chunks of relatively routinized and generic actions (select, open, drag) applied to diverse objects. Steps also tended to be subtasks for which substantive mistakes were possible, i.e., a mismatch between what the user might want to do, and how the user tried to it with the system (see Norman, 1982; Carroll & Mack, 1984).
- Opinion:
  - Satisfaction with (1) the ease of scenario completion, (2) the amount of time required, and (3) the support information (help, messages, and documentation).
  - The frequency with which these types of tasks were done in the real work environment.

  For these opinion measures, participants were asked to complete a short questionnaire at the end of each scenario. The satisfaction items used 7-point scales. The frequency item had four points corresponding to Daily, Weekly, Monthly, and Never.
- Problem Analysis: Specific user problems were also recorded. Any deviation from the optimum sequence of actions required to complete the scenario was considered to be indicative of a problem. These problems were classified in terms of impact on scenario completion and frequency (number of users experiencing the problem). Four impact levels were defined:
  1. Scenario failure or irretrievable data loss. A scenario could be failed if the participant required assistance to complete the scenario, or if the participant believed the scenario to be properly completed, but the output of the scenario was incorrect (excepting minor typographical errors).
  2. Considerable recovery effort. The recovery effort was defined as considerable if a participant worked on recovery for more than one minute or repeated the error within a scenario.
  3. Minor recovery effort. The recovery effort was defined as minor if the error only occurred once within a scenario and required less than a minute for recovery.
  4. Inefficiency. A problem was considered to be an inefficiency if it did not fall within any of the impact classifications above.

3.3 Systems and Environment

Two office systems were put together by installing a word processor, a mail application, a calendar application, and a spreadsheet on two different platforms which allowed a certain amount of integration among the applications. Both platforms allowed participants to cut, copy, and paste data between applications and to present data from several applications simultaneously in windows. System I was more tightly integrated than System II (see Mack & Nielsen, 1987) with respect to consistency of graphical interface style across applications. In System II e.g., the address and calendar applications were host-based menu- and text-based applications quite different from other component applications in the software environment.

3.4 Procedure

Introduction Participants began with a brief tour of the lab, a description of the study's purpose and events of the day, and completed a background questionnaire. Participants who used System I be-

gan by working with the interactive tutorial which was provided. Those who used System II were given a brief demo about how to move, point and select with a mouse, how to open the icons for each product, and how to minimize and maximize windows.

Task Scenario Activity Following the system familiarization, participants worked on Scenario M1. Following this, the ten scenarios (including a different version of Scenario M1) were presented in mixed orders. The instructions emphasized that we were especially interested in what happened when people first began to use a new system and assured the participants that we did not expect perfect performance. The instructions also focused on working "at your own pace" and using supporting documentation "whenever you like". If it became clear that a participant had failed to complete a scenario successfully, he or she was helped to finish, and then began the next. At the end of the day, participants were debriefed.

4.0 RESULTS AND DISCUSSION

4.1 What are the obtained values of our usability measurements?

Table 2 shows a subset of the usability measurements gathered in this study. The table is organized by scenario and dependent measure.

Table 2. Subset of Scenario Data (System I/System II).

Scenario	Comp. Rate (%)	Med. TOT (min)	Step Count	Impact 1 Errors	Impact 2 Errors
M1A	33/80	*/10	15/13	21/4	36/10
C1	57/80	10/15	14/13	12/2	19/16
A1	64/93	7/15	28/32	8/1	10/17
D1	36/47	*/*	43/62	20/13	15/30

* If the Completion Rate is less than 50%, then the Median Time-on-Task cannot be calculated.

Statistical analyses can be used at various levels for various reasons. Univariate analyses such as t-tests may be used to determine for which variables and scenarios a behavioral objective has been exceeded beyond a statistical criterion such as alpha < .05. Analyses can be conducted at the participant or scenario level. It is reasonable to consider the scenario as a unit of analysis because scenarios, like participants, are sampled from a larger population and are expected to exhibit individual differences. At this level of analysis, it is possible to include analytical information such as step counts as well as measures calculated from the participant sample (e.g., completion rates, median subjective ratings).

We experimented with multivariate statistical techniques at the scenario level. Multivariate techniques are of value when dependent variables are expected to be correlated, or when one is concerned with patterns of dependent variables. For example, we conducted three discriminant analyses (Cliff, 1987; SAS, 1979) to help discover if some variable sets of a priori interest were useful in discriminating the systems. The first set included the performance/analytical variables of completion rate, error-free rate, and step count, and was significant ($F(3,18) = 4.98$, $p = .01$). The next analysis used the error counts by impact level, and was significant ($F(4,17) = 9.87$, $p = .0003$). The discriminant analysis using the three subjective ratings was not significant ($F(3,18) = .27$, $p = .84$).

The three discriminant analyses indicated how one might begin an exploration of the data. The most significant discriminant function was obtained by examining the error counts by impact rating. System I had significantly more high impact problems than System II, that is, problems which could not be resolved without help, and which may have led to loss of data. The analysis using completion rates, error-free rates, and step counts was also significant. Participants using this system completed significantly fewer tasks without assistance for System I compared System II (about 50 % vs 69%, respectively). The systems did not differ in frequency of tasks users were able to complete with no help and no problems, a low frequency outcome for both systems in any case (about 20% and 17 % for System I and II respectively). The subjective ratings, however, provided a poor discrimination between systems. These patterns of results indicated that, in this case, we should focus on user problems when providing design guidance.

4.2 How good (reliable and valid) are these data?

One of the goals of quantitative benchmarking is to develop a cumulative and reliable database of benchmark assessments for systems over time and investigators, in the face of known sources of variation in human performance (e.g., individual differences, see Egan, 1988). To establish this goal requires (1) reliable data measurements, (2) standard data collection and analysis methods (applicable across time and development groups), and (3) valid measurements with respect to providing representative and diagnostic information about user's experience in the real workplace. We discuss each issue in turn.

4.2.1 Reliable benchmark measures

The reliabilities of the benchmark measures were estimated by creating subsamples of the data for each

system. Subsamples consisted of two representatives from each participant group for a total of six participants per subsample. Dependent measures were calculated for each subsample by scenario, and correlations were calculated by system for each measure of interest (median time-on-task and satisfaction ratings, completion and error-free rates). Of the eleven correlations computed, only one was clearly nonsignificant (System II Time Rating, $r = .29$). The others ranged from .5 to .8. This is encouraging since there were only six participants in each subsample, consistent with many industrial benchmark evaluations. Although some reliabilities fell slightly below the recommended values of .7 to .8 (see Landauer, 1988) most of the coefficients were fairly high considering the small number and heterogeneous quality of the participants in the subsamples. These measurements should be adequate for most system evaluative purposes, since they are more experimental and exploratory in nature than they are psychometric and normative (see Nunnally, 1978).

4.2.2 Reliable Data Collection and Analysis Methods

In each study, the data were collected by a different set of observers in a different location. We do not have any quantitative assessment of the inter-rater and inter-location reliability. It may be that the values we have reported represent the upper limit for reliabilities since they are based on measures collected in the same location by the same set of observers. The lead observers for each study were in constant communication to ensure that consistent methods were followed, both for procedure and judgements. At the conclusion of the study, all recorded errors were reviewed by the lead observers to ensure that all disagreements were resolved before the summary measures were derived.

4.2.3 Validity of Scenarios and Measurements

The ultimate validity of our laboratory benchmarks is in predicting the success of our product in the marketplace. We know of no actual statistical interpretation of validity assessed from the laboratory (or even field) and the marketplace. Rather, we relied on the content validity of our scenarios, the usability attributes we develop, the way we measure those attributes, and the design guidance these measurements provide. It is possible in some cases to obtain converging evidence for these conclusions. For example, our confidence in the validity of the scenarios is based not only on earlier field work and collective judgment of office system experts, but also on judgments we elicited from participants in this study about the frequency with which they performed tasks corresponding to the scenarios. For this judgement, averaged across participants by scenario, the correlation between the systems was .92 ($p < .001$), indicating that the estimates were very reliable. This result is consistent with our belief that these scenarios are representative of real office work.

4.3 How can this information be used to improve a system?

4.3.1 Testable Behavioral Objectives

Although it is not presented in the form usually associated with behavioral objectives, the data presented in Table 2 can be considered a matrix of testable behavioral objectives. The data in the upper left cell imply that the objective for a system under development should be that, under the same measurement conditions described in our Methods section, the successful completion rate for Scenario M1A should exceed 80%. In the same way, the median Time-on-Task should be no greater than 7.0 minutes for Scenario A1. It may not be realistic to expect the system under development to exceed the better competitive system for every measure and every scenario, but these targets enable the developer to understand those tasks and measures for which the system under development is failing to be competitive. With this knowledge, the developer can make more appropriate engineering decisions and tradeoffs than would otherwise be possible.

4.3.2 Qualitative Diagnostic Information

On the quantitative side, our classification of user problems by severity was similar to alternative methods of characterizing the impact of problems (see Good, Spine, Whiteside and Peter, 1986), but did not involve estimating time spent recovering from errors based on video data. Error impact turned out to be an important factor discriminating the systems we evaluated. In particular, high impact errors are precisely those which a developer is likely to want to solve to achieve usability objectives.

On the qualitative side, we are interested in diagnosing the possible causes of user problems and possible design solutions. Diagnostic information can be obtained from other kinds of evaluation, often qualitative, informal and exploratory such as thinking aloud, methods which may preclude obtaining reliable quantitative performance assessments (see, e.g., Nielsen, 1989; Landauer, 1988; Lewis, 1982). However, we believe that substantial qualitative, and diagnostic information is available from benchmarking studies if the investigator records participants' errors and comments.

Several psychologically grounded frameworks are useful for interpreting problems users experience in various human-machine and human-computer domains (examples include Arnold and Roe, 1987; Norman, 1982; Rasmussen, 1988; Reason, 1988; Lewis & Norman, 1986; Carroll & Mack, 1984). We

have found that design-relevant interpretations of problems often do not involve deep psychological analysis but seem to point to violation of basic, commonplace guidelines associated with usable interfaces. Many of the high severity problems in Table 2 resulted from problems with lack of feedback, lack of consistency in operation, unintuitive modes and lack of visibility in system states. These observations apply to both systems, and in the case of System I were somewhat surprising because of the seeming surface consistency and intuitiveness of the interface style. We illustrate these general observations by discussing some serious problems users experienced.

Lack of feedback: Many problems observed simply involved lack of feedback about outcomes or states of the system relevant to successfully accomplishing a task. Copying and pasting between applications was a problem for both systems and was relatively severe in impact and frequency (47% of the participants had trouble using the functions on both systems). Users did not always specify a to-be-cut object first, or did not recognize when selection has somehow failed. The result in both cases is that either nothing was pasted or some prior material was incorrectly pasted. There was no feedback when "cut" or "copy" was executed without specifying an object. These problems suggest providing more feedback about what has been copied (prior to pasting) and/or to somehow reinforce the need to select to-be-copied material before selecting the copy action. Note that solving these problems would improve the System I completion rate for the spreadsheet scenario D1 from 36 % to 50 % (assuming the solutions created no new problems).

Consistency of operation: Another basic guideline is to implement functions consistent with users' tasks and expectations. The cut/copy/paste problem may also be an example of inconsistency with users' expectations. Users who selected copy or paste actions before selecting relevant to-be-copied information seemed to expect an action-object style of interaction, at least in this instance. Another example involved calendar use (Scenario C1) for System I compared to System II. The quantitative data in Table 3 shows that for Scenario C1, users' completion rate was 57% for System I and 73% for System II. The most serious error for System I was that 40% of the participants had great difficulty simply printing daily appointments. The proximate cause of the difficulty was that users misinterpreted an instruction in the user manual. The deeper cause, however, may be that the calendar application for System I did not implement that function in a way consistent with its other print options. Instead, users had to print the screen for that subwindow, a procedure that is actually part of the operating system, not the application, and covered in documentation, but not referred to within the application. Given development resources one would obviously recommend implementing this print option consistent with other available print options. The correction of this problem would improve the completion rate for this scenario to 80% (assuming the solution did not create additional problems).

These analyses are highly interpretative and based on user actions rather than immediate comments (e.g., thinking aloud). Two aspects of this analysis should be clear. Interpreting user problems in a design-relevant way is highly contextual in that it often depends on the details of the application, interface implementation and user task. Also, more than one interpretation is possible, and even useful: e.g., more than one guideline or psychological generality may provide a useful interpretation.

4.4 Conclusions

The benchmarking methodology described here provides a reliable and standard framework. In this paper we described a set of integrated office scenarios, their rationale, and their use. The scenarios were useful in developing both quantitative and qualitative descriptions of two integrated office systems. Users experienced many serious problems with both systems. We were somewhat surprised to find that the system that was seemingly more tightly integrated (with respect to data transfer and consistency of interface implementation) did not provide clear usability advantage for users in terms of our benchmark measures and our broader set of integrated office tasks. Equally surprising, these problems seem to involve failure to observe basic user interface design guidelines as well as a deeper failure in some cases to match user's expectations or intuitions about how tasks can be accomplished using computers. These problems could be analyzed both quantitatively and qualitatively to help diagnose system differences and provide clues to possible system improvements.

5.0 REFERENCES

Arnold, B. and Roe, R. (1987). User errors in human-computer interaction. In M. Frese, E. Ulich, and W. Dzida (Eds.), Psychological Issues of Human Computer Interaction in the Work Place. Amsterdam: North-Holland.

Carroll, J. and Mack, R. (1984). Learning to use word processors: By doing, by thinking and by knowing. In J. Thomas and M. Schneider (Eds.) Human factors in computer systems. (13-52), Norwood, N.J.: Ablex Publishing.

Cliff, N. (1987). Analyzing multivariate data. San Diego: Harcourt-Brace-Jovanovich.

Egan, D. (1988). Individual differences in human-computer interaction. In M. Helander (Ed.) Handbook of human-computer interaction. North-Holland: Elsevier Science Publishers.

Good, M., Spine, T., Whiteside, J., and Peter, G. (1986). User-derived impact analysis as a tool for usability engineering. In Proc. CHI'86 Human Factors in Computer Systems. (Boston, April 13-17), 265-283.

Gould, J.D. (1988). How to design usable systems. In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York: North-Holland Press.

Landauer, T.K. (1988). Research methods in human-computer interaction. In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York: North-Holland Press.

Lewis, C. (1982). Using the "thinking aloud" method in cognitive interface design. Research Report RC 9265, IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, N.Y..

Lewis, C. and Norman, D. (1986). Designing for error. In D. Norman and S. Draper (Eds.) User-centered system design: New perspectives on human-computer interaction. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Nielsen, J. (1989). Usability engineering at a discount. In Proc. Third International Conference on Human-Computer Interaction, (18-22). Boston, MA.

Nielsen, J., Mack, R.L., Bergendorff, K.H., and Grischkowsky, N.L. (1986). Integrated software usage in the professional work environment: Evidence from questionnaires and interviews. In CHI'86 Proceedings, (162-167). New York, NY: ACM.

Mack, R. and Nielsen, J. (1987). Software integration in the professional work environment: Observations on requirements, usage and interface issues. Research Report RC 12677, IBM T.J. Waston Research Center, P.O.Box 704 Yorktown Heights, NY.

Norman, D.A. (1982). Steps toward a cognitive engineering: design rules based on analyses of human error. In Human Factors in Computer Systems, (378-382). Gaithersburg, MD: ACM.

Nunnally, J.C. (1978). Psychometric theory. New York: McGraw-Hill.

Rasmussen, J. (1988). Human error mechanisms in complex work environments. Reliability Engineering and System Safety, 22, 155-167.

Reason, J. (1988). Modelling the basic error tendencies of human operators. Reliability Engineering and System Safety, 22, 137-153.

Roberts, T.L. and Moran, T.P. (1983). The evaluation of text editors: Methodology and empirical results. Communications of the ACM, 26, 265-283.

SAS Institute. (1979). SAS User's Guide. Cary, NC: SAS Institute, Inc.

Whiteside, J., Jones, S., Levy, P. and Wixon, D. (1985). User performance with command, menu and iconic interfaces. In Proc. CHI '85 Human Factors in Computing Systems (185-191). San Francisco, CA: ACM.

Whiteside, J., Bennett, J., and Holtzblatt, K. (1988). Usability engineering: our experience and evolution. In M. Helander (Ed.), Handbook of Human-Computer Interaction. New York: North-Holland Press.

Williges, R., Williges, B., Elkerton, J. (1987). Software interface design. Handbook of human factors. G. Salvendy (Ed.), New York: J. Wiley & Sons.