Quantitative measures for evaluating human-computer interfaces

Quantitative Measures for Evaluating Human-Computer Interfaces.

M. Rauterberg

Work and Organizational Psychology Unit, Swiss Federal Institute of Technology (ETH)
Nelkenstr. 11, CH-8092 Zurich, Switzerland

Abstract

There currently are 4 different views on human computer interaction in measuring interactive qualities: (1) the interaction-oriented view, (2) the user-oriented view, (3) the product-oriented view and (4) the formal view. Two different possibilities of measurement within the product-oriented view are introduced in this paper. Different types of user interfaces can be described and differentiated by the concept of "interaction points". Regarding to the interactive semantic of "functional interaction points" (FIPs), 4 different types of FIPs must be discriminated. Based on the concept of FIPs, the dimensions "[visual] feedback" and "interactive directness" can be quantified.

1. INTRODUCTION

One of the main problems of standards (ISO, DIN, etc.) in the context of software ergonomics is that they cannot be measured in a quantitative way. Four different views on human computer interaction to measure interactive qualities currently exists (see also [11]; [1]:651).

(1) The *interaction-oriented view (iov)*: usability is measured in terms of how the user interacts with the product ("usability testing"). This view is the most common one. All kinds of usability testings with "real" users are subsumed in this category [6].

(2) The *user-oriented view (uov)*: usability is measured in terms of the mental effort and attitude of the user ("questionnaires" and "interviews").

(3) The *product-oriented view (pov)*: usability is measured in terms of the ergonomic attributes of the product (quantitative measures). All heuristic evaluations [4] carried out by ergonomic experts investigating a concrete software product fall in this category, too.

(4) The *formal view (fv)*: usability is formalized and simulated in terms of mental models (formal concepts). Karat [5] describes formal methods in the context of "theory-based" evaluation.

The interactive qualities of user interfaces currently are quantified in the context of iov and uov, but these both approaches are time consuming and more or less expensive. Usability testing is constrained to the investigated task solving processes and the selected users, too.

2. A QUANTITATIVE DESCRIPTION LANGUAGE FOR INTERFACES

It is necessary to define measures of usability for the product-oriented view, a concept of descriptive terms, which can be counted. The granularity of the descriptive terms must be on a medium level – not to specific (e.g. "push button", "menu option", etc.) and not to general (e.g. "transparent", "flexible", etc.). A level, at which it is possible to describe the different types of user interfaces ("batch", "command", "menu", "desktop") in a uniform and precise way, and at the same time a level, which is powerful enough and easy to apply, is required.

The interaction space (IS) consists of two different interlaced spaces: the object space (OS), and the function space (FS). OS encloses all perceptible represented objects (RO) and all hidden objects (HO), which users can grasp and bring into the actual dialog context The same situation is valid for FS: we have to distinguish between perceptible represented functions (RF) and hidden functions (HF). A concrete dialogue context (DC) contains a subset of $\{OS \cup FS\}$.

An interactive system can be distinguished in a dialog and an application manager [3]. Belonging to this differentiation we distinguish between two types of objetcs and two types of functions: dialog object (DO, e.g. "window") and application object (AO, e.g. "textdocument"), and dialog function (DF, e.g. "open window") and application function (AF, e.g. "insert section mark"). Each function has a functional interaction point (FIP): AF –> AFIP, DF –> DFIP. RF is the set of all implemented representations of FIPs. A perceptible AFIP is called a RAFIP, a perceptible DFIP is called a RDFIP (see Fig. 1 and 2). These perceptible structures can have visible, audible and/or tactile representations. RO is the set of all implemented representations of DOs (e.g. "button", "icon", "window", etc.) and AOs (e.g. "text document", "grafic", "data base", etc.). A perceptible AO is called a RAO, a perceptible DO is called a RDO. An AFIP changes the state of an AO, and a DFIP changes the state of a DO. All DFIPs are more or less "interactive overhead". DFIPs are only suitable to handle one of the most constrained interactive ressource, namely the *screen space*.

The complete set of all description terms is defined as follows:

$IS := OS \cup FS$	[interaction space]
$DC \in IS$	[dialog context]
$OS := RO \cup HO$	[object space]
$FS := RF \cup HF$	[function space]
$RO := RDO \cup RAO$	[(perceptible) represention of objects]
$HO := DO \cup AO$	[hidden objects]
$RF := RDFIP \cup RAFIP$	[(perceptible) representions of functions]
$HF := DFIP \cup AFIP$	[hidden functions]
$RDFIP := \{(df,rf) \in DFIP \times RF: rf = \delta(df)\}$	[(perceptible) represented DFIP]
$RAFIP := \{(af,rf) \in AFIP \times RF: rf = \alpha(af)\}$	[(perceptible) represented AFIP]
δ := mapping function of a $df \in DFIP$ to an appropriate $rf \in RF$.	
α := mapping function of an $af \in AFIP$ to an appropriate $rf \in RF$.	
$RDO := \{(do,ro) \in DO \times RO: ro = \mu(do)\}$	[(perceptible) represented DO]
$RAO := \{(ao,ro) \in AO \times RO: ro = \nu(ao)\}$	[(perceptible) represented AO]
μ := mapping function of a dialog object $do \in DO$ to an appropriate $ro \in RO$.	
ν := mapping function of an application object $ao \in AO$ to an appropriate $ro \in RO$.	

The intersection of RF and RO is sometimes not empty: $RF \cap RO \neq \emptyset$. In the context of graphical interfaces icons are elements of this intersection e.g. RDFIP "copy" ≡ RDO "clipboard", RAFIP "delete" ≡ RAO "trash". The "interaction point (IAP)" introduced by Denert [2] is not differentiated enough to appropriately describe graphical user interfaces; an IAP is more or less the same as the "actual dialog context (DC)" discussed in this paper (see Fig. 1-5).

If both mapping functions δ and α are of the type 1:m(any), then the user interface is a command interface (see Fig. 2) where the command interface has only one $rf \in RF$, the "command prompt" (e.g. the RF in Fig. 1). If both mapping functions δ and α are of the type 1:1, then the user interface is a menu or direct manipulative interface where each $f \in FS$ is related to a perceptible structure RF on the i/o interface. One important difference between a menu and a direct manipulative interface is the "interactive directness". A user interface is 100% interactively direct, if the user has fully access in the actual dialog context to all AFIPs [7]. Good interface design is characterized by optimizing the multitude of DFIPs (e.g. "flatten" the menu tree [8]) and by allocating an appropriate RDFIP to the remaining DFIPs.

In the context of an actual dialog state the user must know what he or she can do next. To support the user in this way, different kinds of representational structures for functions (RF, e.g. "menues", "icons") have been developed (see Fig. 3). If each functional interaction point (FIP) has its own representational interaction point (RF), then the user has 100% feedback (FFB) of all available functions. To estimate the amount of "feedback" of an interface a ratio is calculated: "number of RFs" (#RF = #RDFIP + #RAFIP) divided by the "number of HFs"

(#HF = #DFIP + #AFIP) per dialog context. This ratio quantifies the average "amount of feedback" of the function space (FB). [D = number of all different dialog contexts]:

$$FB = 1/D \sum_{d=1}^{D} (\#RF_d / \#HF_d) * 100\% \qquad \textbf{[(visual) feedback]}$$

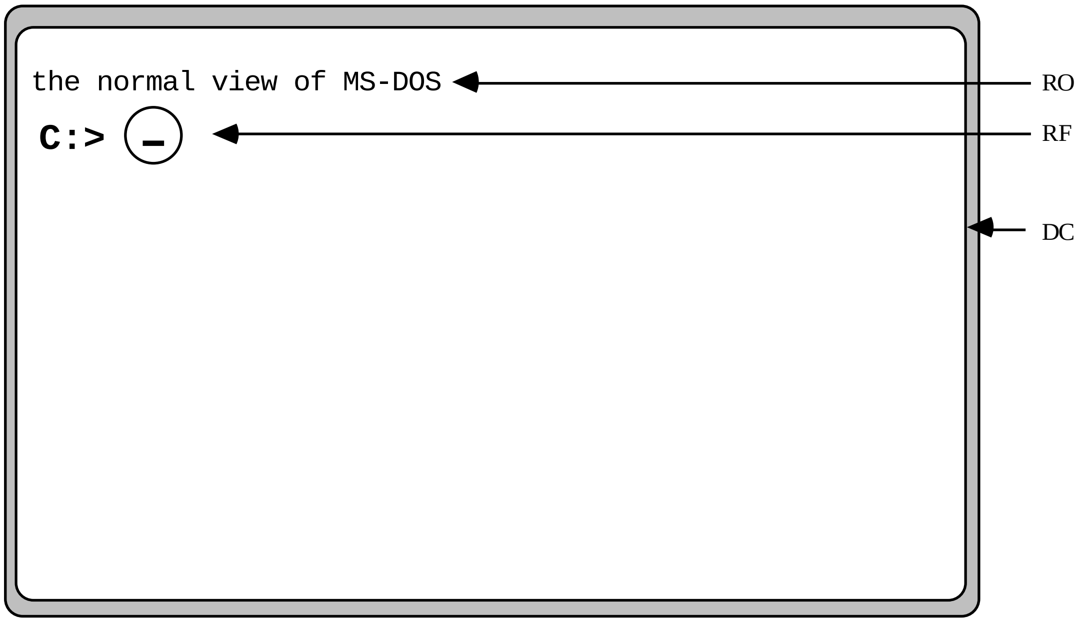

Figure 1. An actual dialog context (DC) of the operating system MsDOS with the representation space of the interactive object (RO = RDO ∪ RAO: "text output") and the representation space of the interactive functions (RF = RDF ∪ RAF: "command entry point" marked by a circle).

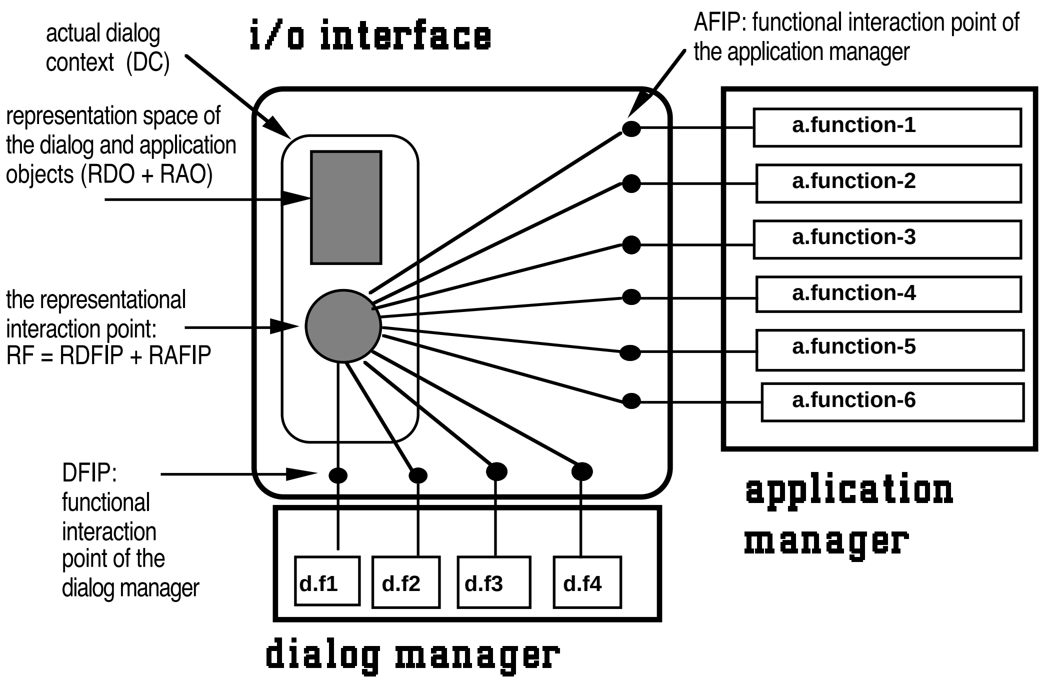

Figure 2. A schematic presentation of a fictive i/o interface, a dialog and an application manager of an interactive system with a *command interface (CUI)*.

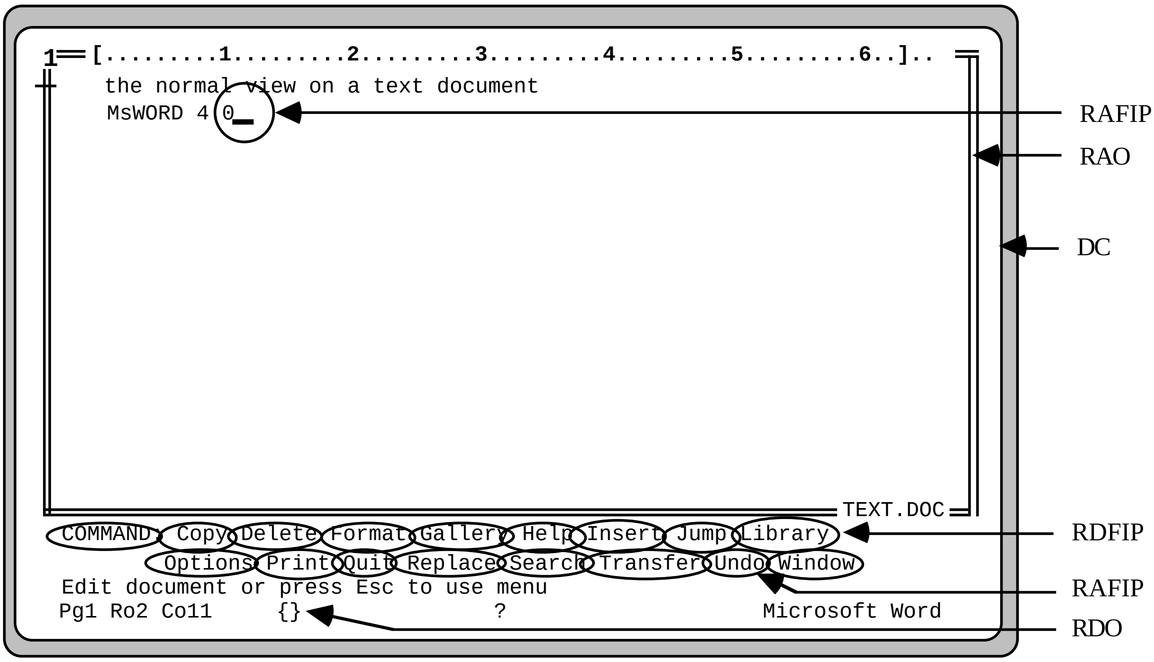

Figure 3. An actual dialog context (DC) of the textprocessing program MS-Word with the representation space of the interactive object (RAO: "text document"; RDO: "clipboard"), and the representation space (RF: marked by circles) of the interactive functions (RAFIP: "text entry point", "undo"; RDFIP: menu options).

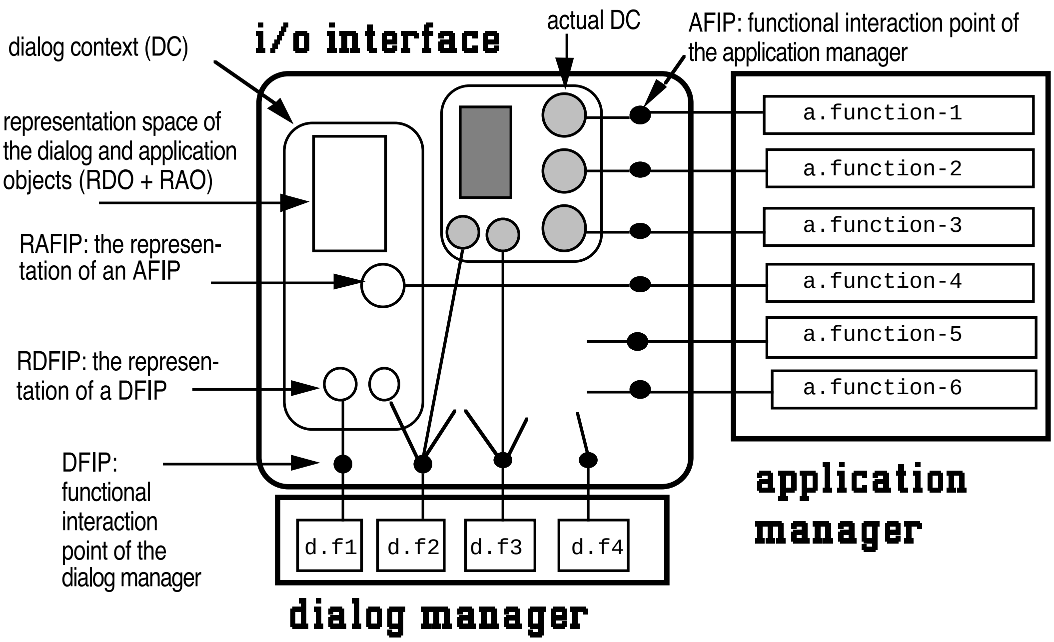

Figure 4. A schematic presentation of a fictive i/o interface, a dialog and an application manager with a two level *menu tree (CUI)*.

The physical limitations of the i/o-interface (screen size) is one reason, not to present all available functional interaction points (FIPs) with a specific representation (RF) on the screen. So, the user has to navigate through menu structures (= activating DFIPs) to come down to a DC with the desired AFIP (see Fig. 4). The average length of all possible sequences of dialog operations (PATH) from the top level dialog context down to DCs with the desired AFIP can be used as a good quantitative measure of "interactive directness" (ID): the reciprocal value of the average path length (= number of dialog steps). An interface with the maximum ID of 100% has only one DC with path lengths of 1 dialog step. [P = number of all different dialog PATHs]:

$$ID = \left\{ 1/P \sum_{p=1}^{P} \mathrm{lng}(PATHp) \right\}^{-1} * 100\% \qquad \textbf{[interactive directness]}$$

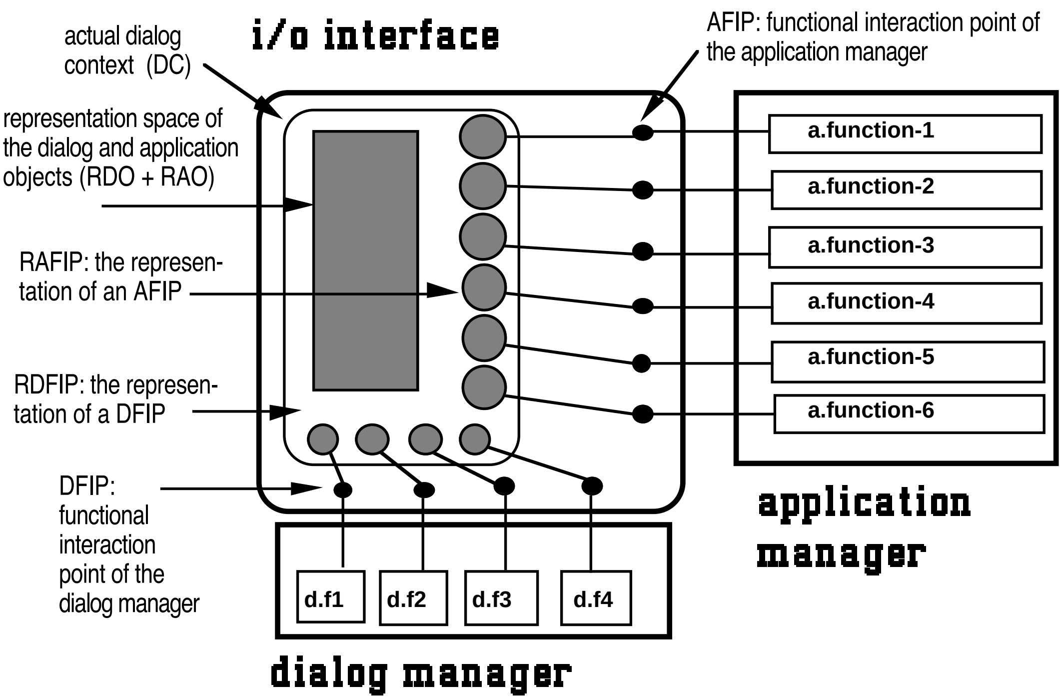

Figure 5. A schematic presentation of a fictive i/o interface, a dialog and an application manager of an interactive system with a *graphical user interface* (GUI, e.g. desktop interface).

3. A CLASSIFICATION OF INTERFACE TYPES

Using the two quantitative measures "feedback" and "interactive directness" it is possible to classify the most common interface types: batch, command, menu, desktop (see Tab. 1).

Table 1. A classification schema of user interfaces.

		[visual] feedback low	high
interactive directness	low	[batch]	menu (CUI)
	high	command (CUI)	desktop (GUI)

The command language interface is characterized by high interactive directness, but this interface type has a very low amount of visual feedback. Only graphical interfaces (GUIs) can support the user with sufficient visual feedback and high interactive directness (c.f. [10] and [12]).

4. CONCLUSION

Standards and norms need product oriented operationalization of interface features. To attain this goal, a description language for interface structures which is general enough to classify the different interface types and detailed enough to allow quantification is required. The description language which is introduced in this paper meets these both conditions. The function space (FS) can be distinguished in (1) functional and representational interaction points, and (2) dialog and application specific interaction points. The degree of visualization and interactive directness can be described and measured based on these interaction points.

Using the two quantitative measures for "feedback" and "interactive directness" in measuring the interactive quality of user interfaces it is possible to classify the most common types: batch, command, menu, desktop. The command interface is characterized by high interactive directness, but has a very low amount of visual feedback. Only graphical interfaces (GUIs) can support the user with sufficient interactive directness and high visibility. The presented approach to quantify usability attributes and the interactive quality of user interfaces is a first step in the right direction. The next step is a more detailed analysis of the relevant characteristics and validation of these characteristics in further empirical investigations. Standardized criteria need to be developed to test user interfaces for conformity with standards.

In addition to the measures of "feedback" and "interactive directness" two other quantitative measures have been defined and validated: "flexibility of the dialog interface" and "flexibility of the application interface" [10] The empirical validation of all 4 measures was carried out with two different interfaces ("menu" and "desktop") of a database system (described in [9]).

REFERENCES:

[1] Bevan, N., Kirakowski, J. & Maissel, J (1991) What is Usability? in: Human Aspects in Computing: Design and Use of Interactive Systems with Terminals; (Bullinger, H-J.; ed.); Amsterdam: Elsevier; 651-655.

[2] Denert, E. (1977) Specification and design of dialogue systems with state diagrams. in: International Computing Symposium 1977; (Morlet, E. & Ribbens, D.; eds.); Amsterdam: North-Holland; 417-424.

[3] Edmonds, E. & Hagiwara, N. (1990) An experiment in interactive architectures. In: Human-Computer Interaction - INTERACT '90. (Diaper, D. et al.; eds.) Amsterdam: Elsevier; 601-606.

[4] Jeffries, R. & Desurvire, H. (1992) Usability testing vs. heuristic evaluation: was there a contest? *SIGCHI Bulletin* 24(4), 39-41.

[5] Karat, J. (1988) Software Evaluation Methodologies. in: Handbook of Human-Computer Interaction; (Helander, M.; ed.); Amsterdam: Elsevier; 891-903.

[6] Kirakowski, J. & Corbett, M. (1990) Effective Methodology for the Study of HCI. in: Human Factors in Information Technology, vol. 5; (Bullinger, H. & Polson, P.; eds.); Amsterdam: North-Holland.

[7] Laverson, A., Norman, K. & Shneiderman, B. (1987) An evaluation of jump-ahead technique in menu selection. *Behaviour and Information Technology* 6(2), 97-108.

[8] Paap, K. & Roske-Hofstrand, R. (1988) Design of menus. in: Handbook of Human-Computer Interaction; (Helander, M.; ed.); Amsterdam: North-Holland; 205-235.

[9] Rauterberg, M. (1992) An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. *Behaviour and Information Technology* 11(4), 227-236.

[10] Rauterberg, M. (1993) A product oriented approach to quantify usability attributes and the interactive quality of user interfaces. in: Work with Display Units WWDU'92; (Luczak, H., Cakir, A. & Cakir, G.; eds.); Amsterdam: Elsevier; (in press).

[11] Rengger, R. (1991) Indicators of usability based on performance. in: Human Aspects in Computing: Design and Use of Interactive Systems with Terminals; (Bullinger, H-J.; ed.); Amsterdam: Elsevier; 656-660.

[12] Ulich, E., Rauterberg, M., Moll, T., Greutmann, T. & Strohm, O. (1991): Task orientation and User-Oriented Dialog Design. *International Journal of Human-Computer Interaction* 3(2), 117-144.

Advances in Human Factors/Ergonomics, 19A

Human-Computer Interaction:

Applications and Case Studies

Proceedings of the Fifth International Conference on Human-Computer Interaction, (HCI International'93), Orlando, Florida, August 8-13, 1993
Volume 1

Edited by
Michael J. Smith
University of Wisconsin, Madison, WI 53706, USA

and

Gavriel Salvendy
Purdue University, West Lafayette, IN 47907, USA

ELSEVIER SCIENCE

Amsterdam – London – New York – Tokyo 1993