

キーストローク・レベル・モデルによる ドキュメント電子化プロセスの検討

大野邦夫 金子哲也

ohno@inse.co.jp kaneko@inse.co.jp

INSエンジニアリング株式会社

〒141 東京都品川区西五反田4-31-18

キーストローク・レベル・モデルは、Xerox PARCで開発された計算機への入力操作に関する評価モデルである。このモデルは、ユーザインタフェースを評価するために用いられているが、操作時間を対象とすることから、稼働工数の算定に用いることができると思われる。ここではこのモデルを用いて、イメージ、文字、表、図形を構成要素とする文書の電子化に要する時間を算出する方法を述べ、その簡単な適用例を紹介する。従来、この種の作業の評価は、実測による以外に方法が無く、その場合も、操作者のスキルや疲労などにより影響され、客観性に乏しかった。その点、本手法は客観的なデータとなり得るため、費用対効果の検討のようなトレードオフを生じる場合の検討に於いて有効な手段になる。

A Study on the Process of Making Digital Documents Based on the Keystroke Level Model

Kunio OHNO Tetsuya KANEKO

INS Engineering Corporation

4-31-18, Nishi-Gotanda, Shinagawa-ku, Tokyo, 141 Japan

Keystroke Level Model, which was developed by Xerox PARC, is a generalized model for evaluating computer-human interfaces. The model can be applied to evaluate the work load for the development of digital document contents, because it estimates the duration of certain tasks. This paper describes an evaluation example of the digital document which is composed of text, tables, graphics, and/or image information. The result proves the effectiveness of the model especially in the case of the cost vs. effect trade-off situations.

1.まえがき

ここでは既存文書を電子化するための稼働の評価法について検討する。文書の電子化の課題は技術的な課題であると同時に極めて社会的な課題である。要するに電子化が、コンピュータで処理し、電子的に管理したり通信したりすることが可能であることを意味するだけでなく、図1に示すように、文書の作成、承認、配布といった組織体としての情報管理の手法を変革し、必然的にそれを運用する組織自体の変革をも促すからである。[1]

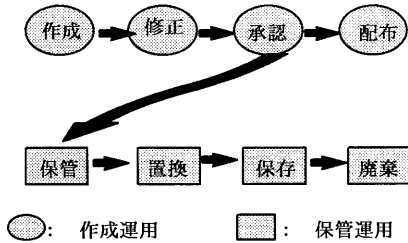


図1 標準的な文書のライフサイクル

従って文書の電子化には、決まりきった普遍的な方法が存在するわけではなく、情報管理手法に応じた多様な方法から選択せざるを得ない。その選択の基準としては、組織体の使命、規模、人員構成、関連組織での情報管理手法、既存の文書の扱いなど様々な要因が存在する。

ところで、文書の電子化を考える際に、重要な観点として、文書の構造的な扱いが挙げられる。紙の文書の場合、情報の管理は、ページや行といったレイアウト構造的な視点から把握され管理される。それに対し、電子化文書は、テキスト、図表といった文書の構成要素を章、節、項といったツリー状の論理構造に基づいて収容する容器として扱われる。[2]

従って、文書の電子化にあたり、電子化される文書の論理構造を決めることが重要である。文書の論理構造を記述する標準的な言語の規格としては、ISOにより定められたSGMLがあるが、必ずしも普及してはいない。現実には、既存のワープロやDTPシステムによる文書形式が幅広く用いられており、SGMLで統一して標準化し運用するためには若干の日時を必要としそうである。

一方、最近の計算機の世界における標準化は、事実上の標準（デファクト・スタンダード）であることが重要である。その観点から考えると、SGMLは実力不足である。むしろSGMLの簡易なDTDから出発したHTMLが、ブラウザ兼ネットワーク上の標準文書としての地位を獲得している。特に最近のインターネット、イントラネット・プ

ームはHTMLの普及がトリガーとなっており、HTMLの果たしている役割は極めて大きいと言える。

しかしながら、現状のHTMLは、その貧弱なレイアウト機能を考えると、従来の紙の文書を全面的に置き換えてしまうものではない。また、SGMLのような文書要素を管理する汎用的な枠組みとしては不適切である。HTMLはあくまでも「ネットワークを通じた情報配布」すなわち配信用の枠組みであると考えられる。

筆者らは、文書要素を管理する枠組みとして、Interleaf5およびInterleaf6を用いている。その理由は、以下の通りである。

- (1) DTDが定義されている構造であれば、SGMLツールキットを用いてSGMLに変換可能である。
- (2) JStar、Word、一太郎など、主要な文書ファイルからの変換フィルターが用意されている。
- (3) 文書構造や文書要素の変換に際して手を加えたい場合には、Interleaf Lispによりカスタマイズすることが可能である。
- (4) 配布用のシステムとして、WorldViewが提供されており、Interleaf文書をCD-ROM媒体により大量配布することが極めて容易である。

以上の方式で、電子化文書の管理システムをいくつか構築しその一部は、既に報告している。[1][2][3][4][5]

以上の経験を通じて言えることは、文書の電子化にはトレード・オフが存在するという点である。入力経費を節約したために再利用が不可能で、却ってコストが増大したり、その逆に、単にイメージで管理すればよい情報を構造化文書として再入力するようなことを行ったりする。本報告は、以上のような場合に問題となる、文書を電子化するためのコストの評価法について検討するものである。

2.キーストローク・レベル・モデル

2.1 モデルの概要

キーストロークレベルモデル (KLM)は、Xerox PARCの Card等によって提唱された計算機に対する、入力操作評価モデルである[6]。このモデルは、従来は、ユーザインタフェースを評価するためのモデルとして用いられ、実測値との良好な一致が得られているが[7][8][9]、操作時間を評価の対象とすることから、工数の算定に用いることも可能である。以下このモデルについて簡単に紹介する。

このモデルは、計算機との会話的な操作を、実行時間により評価するものである。ワープロやパソコン、ワークス

ーションなどを操作する場合は、必ず操作するための目的が存在する。例えば、紙で書かれた手書きの原稿をワープロファイルとして入力するといったことである。この場合、以下のようなステップで作業を実施するであろう。

- (1) ワープロのメニューから新規文書作成を選択し、
- (2) 必要なら、書式を設定し、
- (3) 文書を入力する
- (4) 入力し終わったら、ファイル名を入力しハードディスクに格納する

これら一連の操作に要する時間を予測する場合、以上の個別操作の時間の総和を取ればよい。さらに個別操作の時間を算定するためには、個別操作におけるさらに詳細な操作手順を求めることになる。

例えば、メニューから新規文書作成を選択するという操作を取り上げると

- (1) マウスによりメニューバー上の「ファイル」メニューにカーソルを移動させる。
- (2) マウスボタンを押下してプルダウンメニューを出現させる。
- (3) さらにマウスボタンを押下したままカーソルを移動させる。
- (4) プルダウンメニュー上の「新規」を選択する。
- (5) マウスボタンをリリースする。

といった、一連の操作を実行する。さらに、文書を入力する場合であれば、

- (1) 入力する場所にカーソルを移動させる。
- (2) 一連の文章を頭に入れる。
- (3) その文字列をキーボードから逐次入力する。

わけである。

以上のように、計算機との会話的な操作は、最終的には、キー入力やマウス操作のようなプリミティブな操作のレベルに還元される。このレベルをCard等はキーストロークレベルと呼んだ。

2.2 モデルの内容

Card等によると、キーストロークレベルは以下の6種類のカテゴリに分類される。

- K (Keystroking) : キー入力時間
P (Pointing) : ポインティングデバイス (マウス) の指示に要する時間
H (Homing) : ホームポジションに戻るための時間
D (Drawing) : ポインティングデバイスによる描画時間
M (Mental Parameter) : 次の操作を思考判断する

ための時間

R (System Response) : システムの応答時間

キーストロークレベルモデルは、操作者が意図する一まとまりのタスクにおける実行時間を予測する。このタスクのレベルに於ては、操作者はタスク実現のためのキーストロークレベルの操作手順を意識的な思考を伴わずに実行する。このレベルのタスクをユニットタスクと呼ぶ。

ユニットタスクの実行時間を T_e とすると、

$$T_e = \sum T_k + \sum T_p + \sum T_h + \sum T_d + \sum T_m + \sum T_r$$

となる。ここで、 T_k , T_p , T_h , T_d , T_m , T_r は、Keystroking, Pointing, Homing, Drawing, Mental operation, System responseに要する時間である。ユーザが、エキスパートの場合、

$$T_m = 1.35 \text{ sec}$$

$$T_k = 0.2 \text{ sec}$$

$$T_h = 0.4 \text{ sec}$$

$$T_d = 1.5 \text{ sec}$$

$$T_p = 1.1 \text{ sec}$$

で与えられる。 T_d , T_p の値は、カーソルが移動する距離により異なるが、大雑把な平均値を用いている。なお、以上の値はエキスパートの場合であり、ユーザのレベルに応じてこの値は異なる。

3. 既存文書の電子化の方法

以上に基つき、既存文書を、電子化文書システムに統合する手法と、それに費やす作業時間について検討する。しかしながら、記録文書の本質として、全ての情報を電子化して管理することは非現実的なので、紙の文書をも管理対象として含む、電子化文書システムとならざるを得ない。

3.1 電子化の分類

既存文書が紙の場合には、次のような手法が考えられるであろう。

- A. 今後も紙のまま管理する。
- B. 全てをスキャナで読み込みイメージ情報として管理する。
- C. テキストはコード化し、表、図形をスキャナで読み込む。
- D. テキストと表をコード化し、図形をスキャナで読み込む。
- E. 全てをコード化する。

個々の場合について、以下に具体的に検討する。

A. 紙のまま管理する。

電子化文書管理システムにリンクさせて、既存のファイリングシステム等を統合する手法である。

紙のまま管理する文書の例としては、

- (1) 紙でないと法的に効果がない文書
- (2) 外部からの書状等の紙の文書
- (3) 既存の紙の文書のうち、再利用頻度が少ないもの。

などが挙げられる。既存の紙の文書が多い場合は、ファイリングシステムを適用するのが最適であろう。しかし電子化システムを導入すれば、紙の文書は大幅に減らすことが可能になるはずであり、その場合は、責任者、担当者を決めてキングファイル等に格納する手法が現実的であろう。

B. 全てをイメージで管理する。

全てをスキャナで読み込み、イメージデータとして管理する手法である。この場合は、紙データを、ゼロックスコピーを取るのと同様の要領で電子化できる。電子化されたデータは、一般には大容量の光ディスクファイルなどに格納されてきたが、最近ではハードディスクの価格が低下したためハードディスクで保管される場合が多い。

1ページのイメージを入力するためには、以下のような操作を行う。

- (1) スキャナに紙データをセットする。
- (2) スキャナを事前走査させる。
- (3) 事前走査画面で記録範囲を指定する。
- (4) 濃度・コントラストを調整する
- (5) 情報を格納するファイル名を指定する。
- (6) マウスで実行のボタンをクリックする。

以上の処理において、キーストロークよりは、紙データを正確にセットしたり、スキャナが走査するために要する時間の方が支配的である。経験的には一回の入力のために2-4分を要する。

この方式の欠点は、電子化したにもかかわらず、構造を持たない点にある。その結果、電子化文書の最大のメリットである、文字列による検索を用いることができない。そのため、文書の検索のために、作成者、作成月日等の文書の属性を付随させ、検索キーワードを登録させ、それを用いて、格納された文書を検索することになる。

しかしキーワード付けは繁雑であり、しかも、登録者の個人差により、過去の文書を効率的に検索することは、かなり困難である。格納文書が膨大になると、この手法のみで管理するのは困難である。

C. テキストはコード化し、表、図形などはスキャナで読み込む。

この手法は、文字列検索を用いることが可能なため、電子化文書システムとしては、かなり効果的である。しかも、図形や表については、スキャナで読み込むために、データの切り出しと貼り付け以外の作業は不要となる。

従って、文書全体に較べ図形や表の割合が少ない既存文書の電子化方式としては、効果的と思われる。全体の文書をスキャナで読み取り、それをOCRを用いて文字列を抽出してコード化し、新規文書に挿入する。残りの図、表については、イメージとして切り出し、それを、上記文書の適切な場所に貼付ければよいわけである。

最近では、OCR機能がソフトウェア化され、安価に提供されるようになったので、以上の作業のためのシステムコストも小さくなった。

問題点としては、スキャナデータが、テキストに比較すると膨大なメモリーを必要とする点にある。そのため、主記憶、ファイル記憶などが大容量になる。また通信時間もかかる。また、OCRは、英文の場合にはかなり高い精度で変換してくれるが、日本語の場合は、精度が不十分である。(例えば、+と十、一とー、=と二、tと七、3とろ、とくく、等は、英文の場合に比べ、区別が難しくなる。)従って、変換誤りのチェックにかなりの稼働を費やすことになる。そのため、場合によっては、OCRで入力し、チェックするよりは、入力し直す方がコスト的に安くなることもある。

D. テキストと表をコード化し、図形などをスキャナで読み込む。

仕様書など表が多い文書の場合、表のデータをコード化しておく必要がある。この場合は、表データもコード化の対象となる。表データの場合、文字部分の切り出しが難しくなるため、OCRは一般に使えなくなる。(使えたとしても、チェックを含めるとコスト的に合わない。)そのため、大抵の場合は手入力に頼らざるを得ない。

E. 手入力ですべてをコード化する。

文書内容を全て手入力し直す場合である。手間とコストがかかり、一見無駄なようでも、再利用性が高い場合には、最も効率的になる。

以上の各種プロセスをフローとして表現したものを、図2にまとめる。図において、四角い箱はデータを示し、楕円形は個別のプロセスを示す。上記のC,D,Eについては、C1とC2、D1とD2、E1とE2といった添字がついているが、1の場合がOCR入力、2の場合が手入力を意味する。

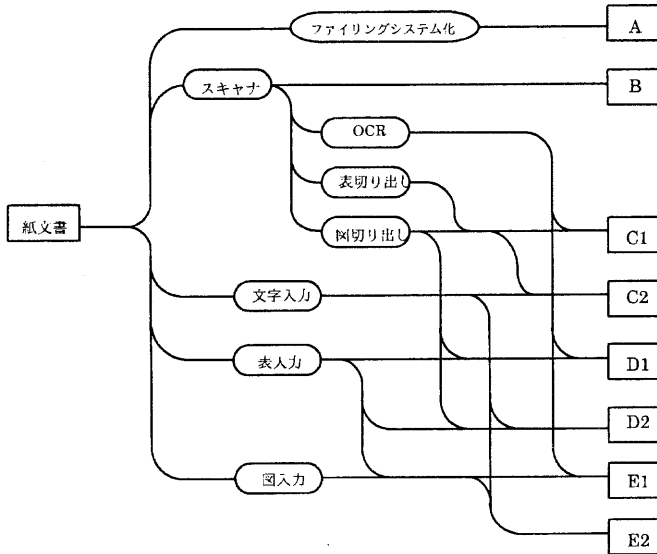


図2 紙ベースのシステムを電子化する方法の分類

3.2 所要時間

テキスト、表、図形についての入力操作にキーストローク・レベル・モデルを適用し操作時間を見積もる。このモデルは、Interleaf6での入力を対象としているが、他のワープロ、DTPでも大きな差異は無い。

3.2.1 テキスト

A4版1ページを基準に考える。10ポイント程度の文字であれば、一行40文字50行位になるので、全部で2000文字程度になる。その入力に要する時間の算定を試みる。例えば「10ポイント程度の文字であれば、一行40文字50行位になるので、全部で2000文字程度になる。」という一文の入力について考える。ローマ字で入力する場合、

$$T_m + 9 * T_k[10\text{pointto}] + T_m + 3 * T_k[(\text{ctr-i})] + T_m + 9 * T_k[\text{teidono(変換)}] + T_m + 12 * T_k[\text{mojideareba(変換)}] + T_m + 2 * T_k[(\text{ret})] + T_m + 9 * T_k[\text{ichigyoku(変換)}] + 7 * T_k[40\text{moji(変換)}] + 7 * T_k[50\text{gyou(変換)}] + T_m + 16 * T_k[\text{kuraininarunode(変換)}] + T_m + 2 * T_k[(\text{ret})] + T_m + 10 * T_k[\text{zennbude(変換)}] + T_m + 20 * T_k[2000\text{mojiteidoninaru(変換)}] + 2 * T_k[(\text{ret})]$$

となる。なお、Tkの後ろの括弧内の内容は、押下されるキーを示す。ctr-iは「コントロール」キーと「i」の並押下を、(ret)は「リターン」キーの押下を示す。以上を総合すると、

$$T_e = \sum T_k + \sum T_m = 106 * T_k + 19 * T_m \approx 34 \text{ sec}$$

30文字を入力するために34秒かかることになる。2000文字だと約2000秒、すなわち33分程度になるが、改行等のため実際は2000文字はなく、より少ない時間で済むであろう。以上から、A4の原稿1ページ作成に要する時間は30分程度、1時間に2ページ程度であろう。

この値は、コンピュータになれた一般の技術者であれば経験的に妥当な数字である。しかし入力スキルの高い人は、はるかに速く入力可能である。さらに変換辞書をカスタマイズしておけば、より速い入力が可能である。なお、ローマ字変換・かな漢字変換の方式により、入力速度の差異が生じることは既に検討している[8]。

3.2.2 表

表の場合は、テキストの場合に比べ、表としてのレイアウト設計の操作、作表操作などが含まれるので、テキストの場合に比べ作成のための操作時間が増大する。

表の中身がテキストのみの場合は、テキストに関する1文字当たりの入力時間は前項と大差ないであろう。表のための線や枠の入力が追加されるのでその分が増大する。線一本書くための操作を、Interleaf6の場合を例にして算出すると以下ようになる。

$$T_m + T_p(\text{Menu-bar:作成}) + T_r1(\text{Pulldown-Menu}) + T_m + T_p(\text{Menu:直線}) + T_k(\text{Mouth-Click}) + T_r2 + T_m + T_p(\text{start}) + T_d + T_r3 + T_p(\text{adjust})$$

具体的に時間を求めると、

$$\begin{aligned}
Te &= \Sigma Tk + \Sigma Tp + \Sigma Td + \Sigma Tm + \Sigma Tr \\
&= Tk + 4 * Tp + Td + 3 * Tm + Tr1 + Tr2 + Tr3 \\
&= 0.2 + 4.4 + 1.5 + 4.05 + Tr1 + Tr2 + Tr3 \\
&= 10.15 + Tr1 + Tr2 + Tr3
\end{aligned}$$

となり、システム応答時間を除き約10秒であることが分かる。

次に枠の記述のための時間を算出する。

$$\begin{aligned}
&Tm + Tp(\text{Menu-bar:作成}) + Tr1(\text{Pull-down-Menu}) + \\
&Tm + Tp(\text{Menu:四角形}) + Tk(\text{Mouth-Click}) + Tr2 + \\
&Tm + Tp(\text{start}) + Td(\text{対角線上}) + Tr3 + Tm + Tp(\text{adjust:縦}) + \\
&Tm + Tp(\text{adjust:横})
\end{aligned}$$

具体的に時間を求めると、

$$\begin{aligned}
Te &= \Sigma Tk + \Sigma Tp + \Sigma Td + \Sigma Tm + \Sigma Tr \\
&= Tk + 5 * Tp + Td + 5 * Tm + Tr1 + Tr2 + Tr3 \\
&= 0.2 + 5.5 + 1.5 + 6.75 + Tr1 + Tr2 + Tr3 \\
&= 13.95 + Tr1 + Tr2 + Tr3
\end{aligned}$$

となりシステム応答時間を除き約14秒となる。従って表を作成する場合は、線と枠を作成する時間を以上のデータから算出し、テキスト入力時間を加えればよい。

なお、類似の表が同一形式で繰り返されるような場合は、表の枠はコピーするだけで良いので、最初の表を除き、後はテキスト入力と変わらなくなる。コピー・ペーストに要する時間を算出すると、

$$\begin{aligned}
&Tm + Tp + Tk(\text{Select-Object}) + Tk(\text{Pop-up-menu}) + \\
&Tp(\text{Select-Copy}) + Tk(\text{Release}) + Tr1 + Tm + \\
&Tp(\text{New-location}) + Tk(\text{Pop-up-menu}) + Tp(\text{Select-Paste}) + \\
&Tk(\text{Release}) + Tr2 + Tm + Tp(\text{adjust})
\end{aligned}$$

$$\begin{aligned}
Te &= \Sigma Tk + \Sigma Tp + \Sigma Tm + \Sigma Tr \\
&= 5 * Tk + 5 * Tp + 3 * Tm + Tr1 + Tr2 \\
&= 1.0 + 5.5 + 1.5 + 4.05 + Tr1 + Tr2 \\
&= 12.05 + Tr1 + Tr2
\end{aligned}$$

従って、既に出てきている表の枠組みをコピーするためには、約12秒程度の時間で済む。なお、この操作は、表の枠組みだけでなく、任意の対象（上記ではSelect-Objectと記した）について成り立つことに注意してほしい。

3.2.3 図形

線図形の場合は、図形を、直線、四角形、円、楕円等に分解し、それらの全てを描画する時間の総和が一つの手がかりになる。実際にはそれらの図形に、説明用の文字列が加わり、さらに矢印や修飾（破線・太線などの線の種類や四角形、円等の塗りつぶし）などが加わる。因みに、図2の場合について評価してみよう。

(1) 線図形の描画

まず図2の線図形の構成要素（四角形：9、直線：25、楕円：8、円弧：29から構成されている。）について検討する。

直線と四角形については、前項で検討しているので、楕円と円弧について算出する。楕円の場合は、

$$\begin{aligned}
&Tm + Tp(\text{Menu-bar:作成}) + Tr1(\text{Pull-down-Menu}) + \\
&Tm + Tp(\text{Menu:円}) + Tk(\text{Mouth-Click}) + Tr2 + Tm + \\
&Tp(\text{start}) + Td(\text{対角線上}) + Tr3 + Tm + Tp(\text{adjust:縦}) + \\
&Tm + Tp(\text{adjust:横})
\end{aligned}$$

となり、具体的に時間を求めると、

$$\begin{aligned}
Te &= \Sigma Tk + \Sigma Tp + \Sigma Td + \Sigma Tm + \Sigma Tr \\
&= Tk + 5 * Tp + Td + 5 * Tm + Tr1 + Tr2 + Tr3 \\
&= 0.2 + 5.5 + 1.5 + 6.75 + Tr1 + Tr2 + Tr3 \\
&= 13.95 + Tr1 + Tr2 + Tr3
\end{aligned}$$

となり、四角形の場合と同様にシステム応答時間を除き約14秒となる。一方、円弧の場合は、

$$\begin{aligned}
&Tm + Tp(\text{Menu-bar:作成}) + Tr1(\text{Pull-down-Menu}) + \\
&Tm + Tp(\text{Menu:弧}) + Tp(\text{Menu:右回り}) + Tk(\text{Mouth-Click}) + \\
&Tr2 + Tm + Tp(\text{start}) + Td(\text{対角線上}) + \\
&Tr3 + Tm + Tp(\text{adjust:縦}) + Tm + Tp(\text{adjust:横})
\end{aligned}$$

となり、具体的に時間を求めると、

$$\begin{aligned}
Te &= \Sigma Tk + \Sigma Tp + \Sigma Td + \Sigma Tm + \Sigma Tr \\
&= Tk + 6 * Tp + Td + 5 * Tm + Tr1 + Tr2 + Tr3 \\
&= 0.2 + 6.6 + 1.5 + 6.75 + Tr1 + Tr2 + Tr3 \\
&= 15.05 + Tr1 + Tr2 + Tr3
\end{aligned}$$

となる。以上の図形（四角形：9、直線：25、楕円：8、円弧：29）を描画するために要した時間を総合すると、

$$\begin{aligned}
Te &= \Sigma Te(\text{四角形}) + \Sigma Te(\text{直線}) + \\
&\Sigma Te(\text{楕円}) + \Sigma Te(\text{円弧}) \\
&= 9 * 13.95 + 25 * 10.15 + 8 * 13.95 + 29 * 15.05 \\
&= 125.55 + 253.75 + 111.6 + 436.45 \\
&= 927.35 [\text{sec}]
\end{aligned}$$

従って約15分半を要することになる。

しかし、実際に描画する場合は以上のような方法は取らない。最初の図形は創製するが、それ以降は、コピー・ペーストするであろう。そのような方法で描画する場合について時間を算出すると

$$\begin{aligned}
Te &= Te(\text{四角形}) + \Sigma Te(\text{直線}) + \\
&Te(\text{楕円}) + 3 * Te(\text{円弧}) + \Sigma Te(\text{複写})
\end{aligned}$$

$$= 13.95+25*10.15+13.95+3*15.05+41*12.05=806.9[\text{sec}]$$

となり、13分半程度ですみ、2分程度時間を節約できる。

(2) 説明用の文字列の入力

説明用の文字が17個所にありそのための入力の時間を算出する。一個所について求めそれを17倍する。テキスト入力、図形入力の画面ではなくテキスト入力画面を用いて行い、それをカットし、フレーム画面にペーストする。

$$\begin{aligned} & Tm+n*Tk(n\text{ストロークの文字を入力})+Tm+Tp \\ & (\text{head})+Tk(\text{Draw-through})+Td(\text{end})+Tm+Tk \\ & (\text{Pop-up-menu})+Td(\text{Select-cut})+Tm+Tp \\ & (\text{Frame})+Tk+Tk(\text{Double-click})+Tr1+Tm+ \\ & Tk(\text{Pop-up-menu})+Td(\text{Select-paste})+Tr2+Tp+ \\ & Tk(\text{Select-object})+Tm+Tp(\text{adjust:縦})+Tm+ \\ & Tp(\text{adjust:横}) \end{aligned}$$

となり、具体的に時間を求めると、

$$\begin{aligned} T_e &= \sum Tk + \sum Tp + \sum Td + \sum Tm + \sum Tr \\ &= (N+4)*Tk+5*Tp+3*Td+7*Tm+Tr1+Tr2 \\ &= 0.2*n+0.8+5.5+4.5+9.45+Tr1+Tr2 \\ &= 20.25+0.2*n+Tr1+Tr2 \end{aligned}$$

nは、最大でも20ストローク（“ファイリングシステム化”）であり、その時間は4秒に過ぎない。従って、一個所の文字入力には、約20-25秒を要することになる。

平均として23秒とすると、全体では約18分を必要とする。このように、Interleaf6の場合、図形への文字入力には意外に時間を要するのである。

4.適用事例

4.1 検討条件

以上、キーストロークレベルモデルによる、文字、表、図形、イメージの入力時間の算出法を説明した。既存の紙の情報を電子化するに当たり、以上のような手法で、入力に要する稼働を見積もることが可能である。

実際に、組織が保有する紙の資料をInterleaf5の形式に変換するニーズはかなり多く、その見積もりを作成する場面でキーストローク・レベル・モデルを適用したことがある。その場合、全体では53ページの文書であったがその内訳は以下の通りであった。

(i) テキストのみ：26ページ（1ページ2000文字）

(ii) 表のみ：10ページ（表作成・20分程度；表の中は2000文字近くのテキスト）

(iii) テキスト+表：8ページ（表作成・10分程度；表以外のテキストは1000文字程度。表の中も1000文字程度のテキスト）

(iv) テキスト+図：6ページ（図作成・1.5時間程度。図以外のテキストは1000文字程度。）

(v) 図のみ：3ページ（図作成・3時間程度。）

4.2 モデルの適用

以上の文書を図2のB～E方法について、作業時間を算出した。以上の(i)～(v)の各々において、1ページを作成するために要する時間を、 $T(i) \sim T(v)$ とすると、全体の時間は

$$T=26T(i)+10T(ii)+8T(iii)+6T(iv)+3T(v)$$

となる。以下具体的に検討する。

4.2.1 全てをイメージ入力（図2のB）

スキャナ入力作業を2分/ページとすると、全体では106分を要する。しかし検索のための目次・索引の作成に同程度の時間を要する。

4.2.2 テキスト入力、図・表をイメージ入力（図2のC2）

$$T(i)=30(\text{min});2000\text{文字入力}$$

$$T(ii)=2(\text{min});\text{スキャナ入力}$$

$$T(iii)=15+2;1000\text{文字入力+スキャナ入力}$$

$$T(iv)=15+2;1000\text{文字入力+スキャナ入力}$$

$$T(v)=2(\text{min});\text{スキャナ入力}$$

$$T=26T(i)+10T(ii)+8T(iii)+6T(iv)+3T(v)=1044$$

4.2.3 テキスト・表を入力、図をイメージ入力（図2のD2）

$$T(i)=30(\text{min});2000\text{文字入力}$$

$$T(ii)=30+20;2000\text{文字入力+表作成}$$

$$T(iii)=30+10;2000\text{文字入力+表作成}$$

$$T(iv)=15+2;1000\text{文字入力+スキャナ入力}$$

$$T(v)=2(\text{min});\text{スキャナ入力}$$

$$T=26T(i)+10T(ii)+8T(iii)+6T(iv)+3T(v)=1708$$

4.2.4 テキスト・表・図を入力（図2のE2）

$$T(i)=30(\text{min});2000\text{文字入力}$$

T(ii)=30+20;2000文字入力+表作成

T(iii)=30+10;2000文字入力+表作成

T(iv)=15+90;1000文字入力+図入力

T(v)=180(min);図入力

$T=26T(i)+10T(ii)+8T(iii)+6T(iv)+3T(v)=2770$

以上から図2の、B、C2、D2、E2の各プロセスによって、電子化文書とするために要する時間は、各々、106分、1044分、1708分、2770分と算出された。認識精度の高いOCRによりテキスト入力が自動化されるとC2、D2、E2の値はさらに減少するであろう。(図2の、C1、D1、E1のプロセス)

5. 結果と考察

以上の結果について考察する。イメージのみの場合の稼働が極めて少なく、テキストのみ手入力、テキストと表を手入力、テキストと図表を手入力するに従い稼働が増大するという以上の結果は極めて常識的である。しかし、常識の範囲ではそれを定量的に把握するには至らない。キーストローク・レベル・モデルは、「この作業は大変そう」とか「この仕事は楽そう」とか言った常識レベルでの感覚を定量化してくれる点に大きな特長がある。

このモデルを用いて、具体的に成果が得られた事例としては、日本語ワープロから Interleaf への変換ツールの使用の適否についての検討が挙げられる。図に関する変換ツールの機能が不完全なために、図だけはゼロから書き直す方がよいのではないかという意見があったが、このモデルを用いて評価した結果、最悪の場合でも、書き直す場合の半分以下の時間で出来上がるという結果が得られた。さらに、後日判明したことであるが、このことは、Interleafの入力の専門家の意見とも一致していた。

このモデルは、従来ヒューマンインタフェースの評価に用いられてきたが、ヒューマンインタフェース自体、定量化に馴染むものではなく、インタフェースの良さと時間データが必ずしも合致しなかった。そのため、このモデルは習熟者用に限定された使われかたをしてきた。その点に関しては、このモデルは、本研究事例のような稼働の定量的な評価に向いていると思われる。

6. あとがき

以上、キーストローク・レベル・モデルを用いて、文書の電子化に要する稼働(時間)を算出する方法を述べ、その簡単な適用例を紹介した。従来、この種の作業の評価は、経験者の勘に頼る以外、実測により比較する以外に方法が無かった。また実測とは言っても、操作者のスキルや疲労具合などにより影響され、客観的なデータとはなり得なかった。

その点本手法は、客観的なデータとなるため、費用対効果の検討の場面のような、トレードオフを生じる場合の比較に於いては極めて有効と考えられる。但し、モデル化のための分析は、かなり煩雑なため、より簡易なモデル化手法が検討されても良いであろう。

文献

- [1] 中島, 大野; "ISO9000シリーズ用の電子化文書管理システム", 情報処理学会, テクニカルコミュニケーション研究グループ報告1994-9-14)
- [2] 大野, 五十嵐, 齊藤; "アクティブドキュメントによるマルチメディアの統合- 会津大学マルチメディアセンタのグループウェア", 情報処理学会, テクニカルコミュニケーション研究グループ報告, (1995-9-13)
- [3] 増田竜太, 大野邦夫; "保守支援システムにおける電子化ドキュメントの効果", 情報処理学会, テクニカルコミュニケーション研究グループ報告Vol. 93-TCG-5-13, (1994-3)
- [4] 大野邦夫, 森卓郎; "グループテクノロジーズ: ドキュメントをベースにしたグループ・マルチメディア / 会津大学マルチメディアセンタのグループウェアについて", NTT技術ジャーナル, Vol.17, No.12, pp.77-79, (1995)
- [5] 大野邦夫; "R&Dレポート: 公衆電話機保守支援システムの改良 / RESPONSE-2とAdaptiveRESPONSE", NTT技術ジャーナル, Vol.6, No.8, pp.75-79, (1994)
- [6] Card, S.K. et al; "The Keystroke-Level Model for User Performance Time with Interactive Systems," Comm. ACM, Vol. 23, No.7 pp 396-410 (1980)
- [7] 大野邦夫; "キーストロークレベルモデルのプログラム開発環境への適用例", 情報処理学会シンポジウム講演資料「計算機システムのヒューマンインタフェース - モデル・評価・展望」, pp.25-34, (1988.4)
- [8] 大野邦夫; "ローマ字入力方式のモードに関する検討", 情報処理学会 日本語処理 3-1 (1985)
- [9] 大野, 杉山; "テキストエディタにおけるマウスの適用性に関する一検討", 昭和60年度電子通信学会情報システム部門全国大会講演論文集 (1985)