

FlyingForm: 電子ペンを利用した紙フォーム記入の先読みシステム

丸山 泰史[†] 五十嵐 健夫^{*}

[†]マッキンゼー ^{*}東京大学大学院 情報理工学系研究科

ペンストロークを電子的に読み取るデバイスを利用し、リアルタイムに記入内容をオペレータに送ることによって、依頼者の待ち時間を短縮する手法を提案する。この手法では、オペレータは依頼者の記入完了を待たずに、送られてきた記入内容を順次処理していく。これにより、依頼者は定型フォームに用件を記入するという従来のスタイルそのまま、短い待ち時間の恩恵を享受できる。我々はこの手法について、実際の手続きを対象としてシステムを試作し、ユーザスタディを行った。この結果、提案手法の有効性ととも、実用システム開発の際に有用な示唆を得ることができた。

FlyingForm: Prefetching Handwritten Entries on Paper Forms By Using Digital Pens

Yasushi MARUYAMA[†] Takeo IGARASHI[†]

[†]McKinsey & Compnay ^{*}The University of Tokyo

We propose a method for transactions that use paper application forms. This method decreases the waiting time of clients by sending their handwriting to the operator in real time with devices that recognize pen strokes on the paper form. Operators process the entries of clients without waiting for the completion. Thus, clients can receive services at shorter times only by filling the paper form as usual. We have developed a prototype system, and carried out a user study. The results have revealed the effectiveness of our method and some interesting findings.

1. はじめに

今日、様々な窓口手続きがあり、多くの人々がこのような手続きに関係している。様々な産業のサービス水準の進歩にともない、窓口手続きにおけるよりよいサービスが求められている。窓口手続きのサービスの質は接客態度や手続きのしやすさなど多くの要素から判断される。このなかでもっとも重要な要素はスピードである。本研究では、窓口手続きの効率改善、特にスピードと顧客の待ち時間の改善を目的とする。

本研究では、紙の定型フォームを利用する窓口手続きについて着目する。定型フォームは非常に直感的でありほとんどすべての人が使うことができる。さらに、物理的な紙媒体を管理の基本としている場合には、紙であることが本質的である。将来的には、完全に電子的な情報管理に移行するかもしれないが、今後少なくとも数十年間は紙の定型フォームも使用されるものと考えられる。

2. FlyingForm

現在多くのビジネス処理が、経費削減や効率化のため、情報システムに統一されている。政府や地方自治体もまたこの傾向に従っていて、これにより組織だけでなく国民もメリットをもたらされている。例えば埼玉県はインターネットを介し電子手続きを行っている。これにより、人々は家にいながらいつでも様々なサー

ビスを受けることができる。

しかしながらこれらのシステムは必ずしも全員が使っているわけではない。多くの高齢者はコンピュータを使うことに慣れていないので、このシステムを使えない。加えて、コンピュータシステムはなんらかの理由で定型フォームでの手続きを要求されたとき有効に満たすことができない。例えば本人が手書きで書いた用紙のコピーの保管が必要な場合だ。また実際の紙の用紙はとても自然なのでほとんどの人がそれを使うことができる。したがって我々は手書きの定型フォームが将来残ると信じている。

しかしながら定型フォームは情報システムと合わないという大きな問題がある。この問題を解決するために、ペンデバイスと実際の紙を用いた日立のデジタルペンのような解決策が提案されてきた。デジタルペンは普通のインクペンであり、ホストコンピュータにペンストロークを電子的に読み取ってもらうものである。最近、リーズナブルな価格のデジタルペンが市場に多くでてきている。これらのデバイスはコンピュータにペンストロークを電子的に読み取ってもらうことができる。そして実際のインクの記録が残る。

ペンデバイスと定型フォームのメリットはフォームに書かれたストロークデータを簡単に電子的に保存することができる。これはビジネスの効率をよくする。従来の方法を用いるとオフィサーはデータを記録するために手書きの文字をスキャンしたり入力したりしな

なければならない。ペンデバイスはストロークをコンピュータに送ることでこの問題を解決する。また、ペンデバイスはストロークの認識機能としてしばしば使われてきた。ストロークが条件を満たしたとき、コンピュータはそれを認識し、この機能をオペレータの代わりに用いた。注意しなければならないのはこのメリットは実際の紙のメリットを失っていないということである。

現在コンピュータ処理のペンデバイスと定型フォームのシステムは実際には使用されていない。たとえ我々がこのようなシステムを使ったとしてもオペレータの仕事量は変わらない。すなわちオペレータは顧客のフォーム記入の完了を待ちフォームの処理を始める。たとえコンピュータが用紙に書かれたストロークを自動的に認識し処理することができたとしても、オペレータは認識結果のチェックや要求されたカードの印刷などの仕事を行わなければならない。したがって顧客はオペレータが処理し終わるまで待たなければならない。

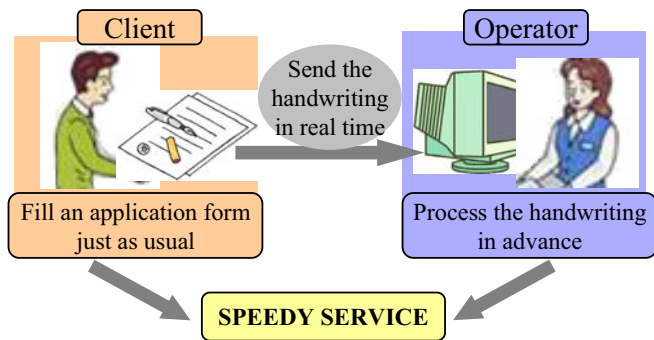


図 1 : FlyingForm の概要

この待ち時間の問題について、もしも依頼者の記入完了を待たずに記入内容を処理することができれば、オペレータはすばやく依頼者にサービスを提供できるのではないかと考えた。もちろんオペレータが処理中にストロークを観察することはむずかしかった。しかしながら我々はペンデバイスを用いてオペレータがそうすることができるシステムを開発することが実現可能である。依頼者はシステムを用いて用紙の記入完了までの短い時間でサービスを受けることができる。図 1 はアイデアの概要を示している。

本稿ではこのアイデアに基づいて設計された FlyingForm システムを紹介する。このシステムは地方自治体の窓口業務のためのものである。図 2 は FlyingForm によるすばやいサービスの仕組みを示している。左図は定型フォームを使った従来の方法の流れである。まず、我々はペンデバイスや自動ストローク認識といった技術を使うことによりオペレータの処理時間を減らすことができると考えた。この流れは中央の図に示している。そしてその効果は図 2 では電子化による改善として示されている。また依頼者の記入完了を待たずに用紙の処理をすることにより待ち時間を減らすことができる。この効果は図 2 では FlyingForm による改善とされている。後者の改善は本稿のコントリビューションとしてもっとも重要である。図 2 をみてわかるように、オペレータは依頼者が窓口に来るまで処理を完了することができない。例えば

オペレータが依頼者の ID を確認し、それを書き終えたあと住民票を渡さなければならない。そのため依頼者は FlyingForm を用いても少し待たなければならない。それにもかかわらず依頼者はほとんど待っていないという印象を受ける。なぜなら残りの作業がオペレータと依頼者とのやりとりを含んでいるからである。そしてこのようなやりとりは依頼者の待っている間のストレスを取り除く。この観点から FlyingForm は特に依頼者の暇な待ち時間を減らすというシステムとして考えられる。

ここで我々は FlyingForm の多重タスク処理の形態について注目する。図 2 において FlyingForm システムのオペレータは流れの最初半分で複数のフォームを扱っている。これは我々が FlyingForm 手法において克服しなければならない問題である。図 2 の左の 2 つの手法はオペレータは目の前にいる人の処理さえすればよく、特別な考慮は必要ない。FlyingForm 手法はオペレータは時々複数の依頼者を同時に対応しなければならない。というのも処理する対象はカウンターにいる人ではなく、記入している人だからである。実際我々はオペレータが 1 人の依頼者だけを処理することによりこの問題を解決できる。しかしこれは依頼者を前もって処理できないので FlyingForm のメリットは減少する。

我々はオペレータはコンピュータの支援により多重タスク処理ができると考えている。なぜならそれぞれの処理量はストローク認識機能などにより減っているからである。さらに、我々は多重タスク処理の研究結果を利用することができる。例えば Multicket システムの結論はオペレータが複数の顧客と対応するときはスケジューリング支援が非常に効果的であり、もしもオペレータの仕事の流れにおいて多くの状況切り替えがあったとしても resume 支援機能が有効であったということだ。我々はこれらの結果が本システムにも適応できると考えている。

最後に本システムのアプリケーション分野は主に定型フォームをつかう窓口業務である。例えば我々は本システムを地方自治体での発行業務やコンピュータ入場券の発券業務、銀行での多くのサービスなどに利用できる。

2.1. 関連研究

手書きで実際の紙に書いたものをコンピュータ伝える研究は多くなされている。なぜなら「ペンと紙」は非常に直感的なのでほとんどすべての人が使うことができるからである。よってもしも我々がペンと紙をスムーズに伝えることができれば非常に有効性がある。Wellner は紙を用いた電子手段をつなげて紙に書いた手書きの文字をカメラでキャプチャすることにより電子的に使用している提案をしている [7]。Johnson は紙を介し XAX システムの電子言語のための技術を示している [5]。荒井氏らによる PaperLink [1] は電子的な特徴を用いた紙の書類を提案しているシステムである。このシステムは、簡単なコンピュータビジョン技術を用いて、紙に記載されている記号をカメラつきのペンで読み取り電子的な情報と結びつけることができる。Heiner 氏らはパーソナルコンピュータのホスト情報と

手書き用紙をシンクロナイズさせた Paper PDA を紹介した[4]. 山下氏らはコンピュータが紙上の情報を処理する手法が提案されている, QRコードに基づく出力結果を論文に記述している[8]. Guimbretiere はコンピュータ画面か紙のどちらかを操作することができる電子書類, PADDs を紹介している[3]. 紙は教室やオフィスでノートとして広く使われている. メンバーの手書き入力をデジタル化することによってクラスやグループをもっと対話的で効果的なものにしようと試みている研究がいくつかある. Davis 氏などはグループのメンバーにそれぞれ他の人のノートを簡単にアクセスできるようにしたノートシェアリングシステムを開発した[2]. 三浦氏などはリアルタイムでノートをシェアできるコンピュータを媒介にしたクラスルームコラボレーションシステムを開発した[6]. そのシステムは紙に書かれた学生の手書き文字をキャプチャしそれを使って生徒の協調を高めるものである.

今まで窓口業務でデジタルペンをつかった研究はなされていなかった. 我々の研究はすばやくサービスを提供するために複数の依頼人の手書き入力のリアルタイム処理に注目しているという点で新しい.

多重タスク処理についての研究については Multicket の論文[9]を参照されたい.

2.2. FlyingForm System

我々は行政サービスのためのプロトタイプシステム, FlyingForm システムを開発した. このシステムは我々の考えに必要な基本機能である.

2.2.1. Target&Flow

FlyingForm の対象は住民票を受け取る行政組織における発行業務である. 従来の方法では, この処理の工程はすべて直列に行われてきた. まず, 依頼者は近くに立っている公務員の助けをかりながら定型フォームに記入する. 記入が完了したら, 依頼者は窓口に行ってフォームと ID をオペレータに手渡す. そしてオペレータは ID を確認し, 定型フォームの内容をコンピュータに入力する. 最後に住民票を印刷し, 依頼者に手渡す.

FlyingForm システムでは処理の流れが変わり, いくつかの処理は並列に行われる(図 3). 依頼者が定型フォームに記入している時にペンデバイスがストロークをオペレータのコンピュータに送る. そしてシステムは手書きストロークを処理し, 手書き認識を行い, コンピュータフォームに自動的に記入される. オペレータは依頼者の記入完了を待たずに順次処理していく. この点での作業はディスプレイで定型フォームのストロークをみながら認識結果を正確に完全なものにできる. オペレータはこのシステムの支援により処理を行いながら別の依頼者の処理を行う. この場合依頼者が窓口に来るまえにフォームの処理を終えることはできない. オペレータはできるだけ早くそれをしなければならない. なぜならこれは望ましい状況ではない. 我々は多重タスク処理の点からオペレータを支援することに注目した.

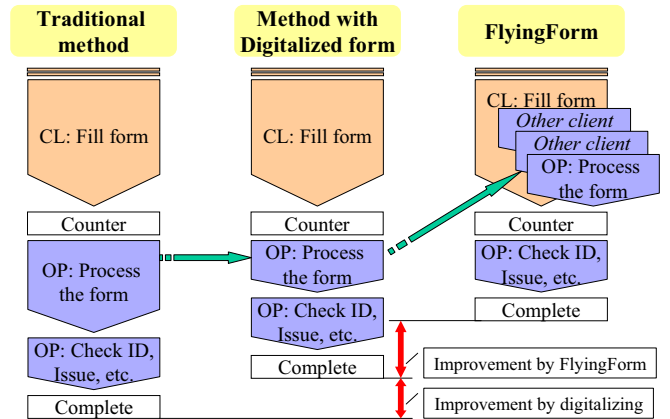


図 2: 従来の手法, 電子フォーム手法, FlyingForm 手法のワークフロー

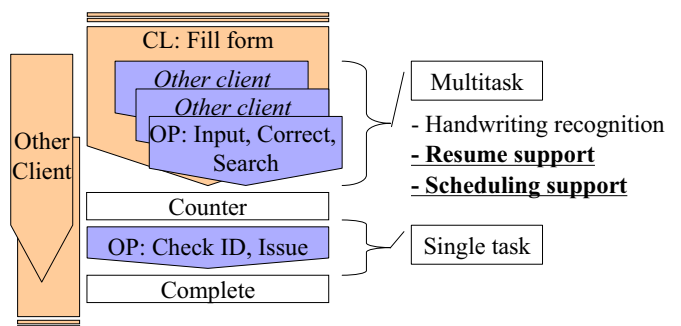


図 3: FlyingForm のワークフローの詳細

2.2.2. Architecture

図 4 は FlyingForm システムの構造を示している. 依頼者のシステムとオペレータのシステムはネットワークでつながっている. それぞれの依頼者のシステムはペンと紙のデバイスと手書きをキャプチャしオペレータのシステムに送るコンピュータから構成されている. オペレータのシステムはコンピュータ, ラージディスプレイ, プリンタで構成されている. オペレータはディスプレイ上の定型フォームを処理し, プリンタにより住民票を発行する.

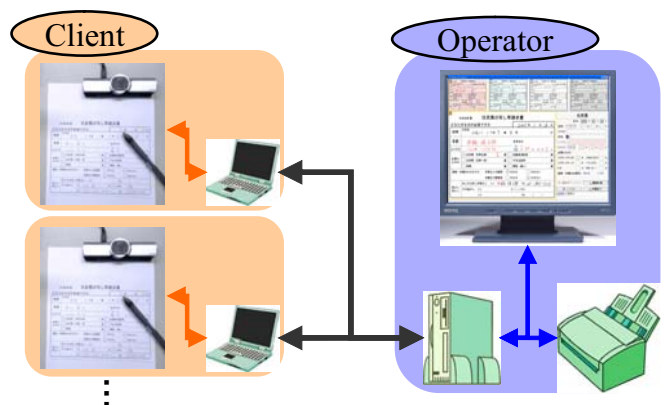


図 4: FlyingForm の構造

2.2.3. Clients'UI

依頼者のUIの基本は簡単であることだ。依頼者がすべきことはただ行動すること。この要求を満たすために我々は依頼のUIデバイスとして Dimensionline による NAVInote(図 5)を使った。これは定型フォームの手書きストロークをキャプチャーしてそれをマウスの動きとしてリアルタイムにホストコンピュータに送るクリップのようなデバイスである。

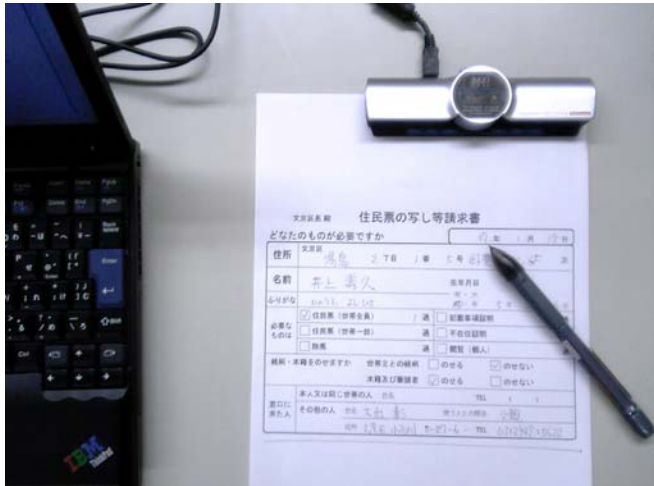


図 5 : 定型フォーム NAVInote

2.2.4. Operator'sUI

UIのスクリーンショットは図 6 で示している。

そのシステムは3つのパネルにわかれているひとつのウィンドウから構成されている。下2つのパネルは現在の依頼者を処理するためのものである。paper パネルは現在の依頼者の定型フォームを映し出している。システムはストロークを認識し search パネル上の空欄に記入していく。paper パネルと search パネルそれぞれ対応していて、paper パネルのストロークと search パネルの欄を対応させる。この対応は以下の2つの機能に使用されている。他の依頼者のフォームは thumbnail パネル(ウィンドウ上方)に表示されている。オペレータは thumbnail パネルのフォームをクリックしてほかの依頼者に切り替えることができる。

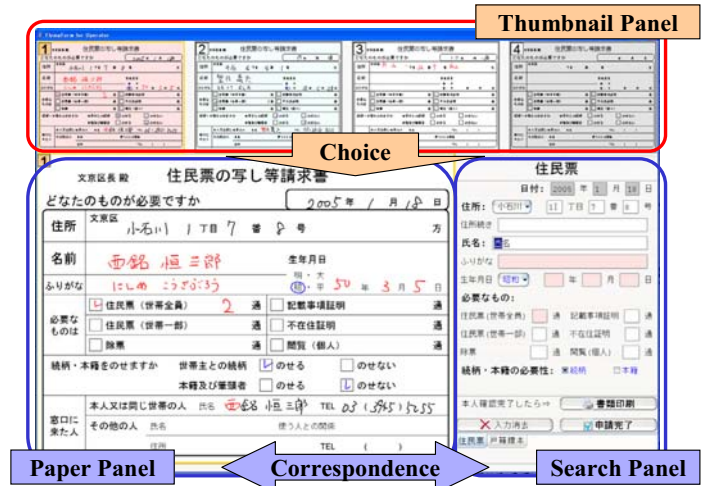


図 6 : FlyingForm システムのオペレータ UI

中断した作業の再開支援 : 色による強調

Multicket の研究[9]はオペレータが複数の依頼者に対応するとき resuming 処理におけるオペレータへの支援の仕組みの必要性を示した。FlyingForm 手法ではオペレータは時々他の依頼者に切り替えるがその後のフォームに戻らなければならない。その場合オペレータ自身が前のフォームや要求された仕事を覚えておかななければならない。中断した処理を記録しそれを適切な方法で表示する resume-support の仕組みによりオペレータは支援されると考えられる。

表示された定型フォームのストロークの色と search パネルの空欄の色を使って resume 支援の仕組みを実現できる。我々は以下の二つの状態に分類する。(1)原本(2)オペレータによる処理。システムはストロークと空欄の状態にしたがって色をかえる。この色で強調することによりオペレータが残りに処理すべきストロークや項目を区別することができる。そして resume から取り除く。図 7 はストロークと項目の色の変化を含んだオペレータのスクリーンショットを示している。

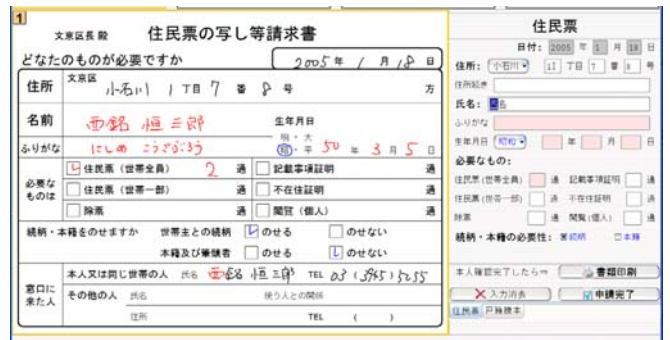


図 7 : FlyingForm の Resume support function. ??と対応する欄は赤色。

図 8 は色の推移を表している。例えば依頼者がストロークを手書き入力しているとき、オペレータが

search パネルに対応している項目を取り扱うまでそのストロークの色は赤である。項目の背景色はオペレータがそれを取り扱い他の項目に移るまでは赤である。ストロークと項目の色が変わるタイミングにはひとつ違いがある。オペレータがある項目と取り扱ったとき、背景色がすでに変わっているにもかかわらずストロークの色は変わらない。というのもオペレータが取り扱っている項目を強調する必要はないがストロークは強調する必要がある。なぜならストロークは明白には集中できず、もし強調されなかったらオペレータは間違えるかもしれない。

色による強調は依頼者が書いていたものを変えて修正するときにも有効である。これらの場合もし強調されていなかったらオペレータは見落とししたり間違えたりするかもしれない。加えて項目とストロークの色の対応はオペレータが項目とストロークの対応を理解するのに有効である。ここで述べた効果は以下のユーザスタディで調査した。

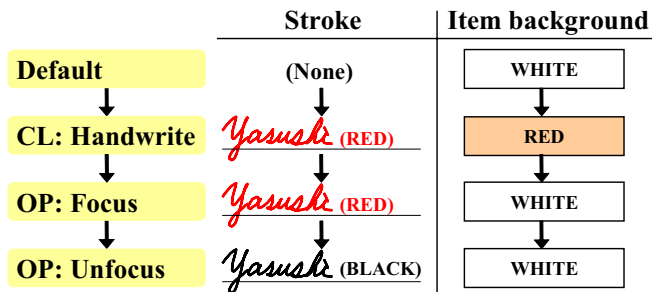


図 8：ストロークと項目の色処理フロー

スケジューリング支援機能

Multicket プロジェクトの結果に基づき、我々はまたこのシステムにスケジューリング支援の仕組みを付け加えた。FlyingForm システムに我々は二つの方法を採用した。

ひとつは次に処理するフォームを決めるための支援である。そしてそれはオペレータの情報処理量を増やすことを目的としている。他の依頼者の情報をアンビエントな方法で示し、オペレータに処理する要求を決めさせるようにシステムをデザインした。図 9 はこの機能を示している。

手段としてそれぞれの依頼者に対応している thumbnail パネルのフォームを使用する。ストロークがオペレータに送られたとき、それらも thumbnail パネルに表示され、ストロークの色が resume 支援機能と同じ方法で変わる。オペレータは適切な時に現在のフォームを 1 時中断した後フォームをみて、処理が必要な別のフォームに切り替える。

二つ目は依頼者が記入を終えて窓口に向かっていることをオペレータに教える支援である。急な処理が必要なときにオペレータの作業を中断するシステムを設計した。我々は依頼者の状態を以下の 3 つに分類した。(A) フォームに記入している (B) フォームへの記入が終了しそうである (C) 窓口に向かっている。緊急レベルは要求により増えていく。B と C の場合システムは

オペレータの作業を中断させる。システムは依頼者の状態を空白領域と NAVInote に装着されてるセンサーから送られてくる情報から判断する。もっと詳しく述べると、依頼者がフォームの後半部分を記入しているときは B、NAVInote から定型フォームをはずしたときは C と判断する。

我々は中断信号を送るために thumbnail パネルのフォームに背景色を採用する(図 9)。状態 B の場合フォームの背景色は黄色としオペレータに注意を促す。状態 C の場合フォームを点滅させることによりオペレータにすぐにフォームを切り替えるよう促す。この点滅はオペレータがフォームをクリックしたときにとまることに注意したい。

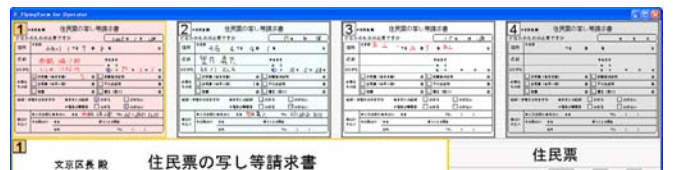


図 9：FlyingForm のスケジューリング機能

2.3. ユーザテスト

2.3.1. 目的

FlyingForm の可能性と有効性を調査し、デザインの改善点を見出すために我々は FlyingForm システムと従来の方法を以下の問題点に着目して比較するために予備実験を行った。

- (Q1)混んでいるときとそうでないときに二つの手法ではどれぐらい待ち時間に差があるのか。
- (Q2)resume 支援機能はどのぐらいうまく働くのか。
- (Q3)スケジューリング支援機能はどのぐらいうまく働くのか。
- (Q4)多重システム処理によりもたらされる総費用はどれぐらいなのか。

2.3.2. 被験者

大学から選んだ 23 歳から 30 歳(平均年齢 25.7 歳)の男性 3 人を被験者とした。すべての被験者がボランティアである。時間は全体で約 150 分である。

2.3.3. 装置

オペレータは 20 インチの Windows XP が OS の Xeon コンピュータ、1600×1200 ピクセルを使った。後ほど述べるレーザープリンタはオペレータのシステムにつながっている。依頼者は解像度が 1024×768 の Mobile Pentium note PC につながっている NAVInote を使用した。FlyingForm システム全体は Java アプリケーションで開発を行った。

依頼者は普通のオフィス机に座り、定型フォームに記入する。オペレータは依頼者から 5 メートル離れた机に座る。プリンタはオペレータがすぐに住民票を取

れるようにオペレータのシステムのすぐ近くに設置した。依頼者はオペレータの机に行くのに約5秒かかる。

2.3.4. 手順

FlyingForm と従来の方法を混雑時とそうでないときの両方比較するため、我々は以下の3つの状態を計画した。

T3 : 3人の依頼者、従来の方法

F3 : 3人の依頼者、FlyingForm 手法

F1 : 1人の依頼者、FlyingForm 手法

3つの方法のうちT3とF3は窓口が混んでいるときの2つの手法の比較である。これらの場合は両方依頼者は間をあげずに処理をはじめ。この状態は込み合っている場合を表している。F1の状態は前の依頼者が記入を完了するまでは依頼者は処理をはじめない。言い換えると1人の依頼者だけが好きなときに処理を行う。この状態は混んでいないときを表している。

それぞれの実験で1人の被験者がオペレータの役を行い残り手伝ってくれる人1人が依頼者の役をする。被験者が全員が1度オペレータの役をする。それぞれの実験はすべての依頼者が2つのフォームを書き終えるまで続けられる。すなわちオペレータは1回の実験で6つのフォームを扱うことになる。各実験の前にオペレータは好きなだけシステムを使って練習することができる。

タスクは行政の公共機関における住民票発行業務である。依頼者の行動の流れは以下の通りである。まず、実験の前に依頼者は印刷された指示カードとIDを与えられる。実験が始まった後に依頼者は名前や電話番号、生年月日といった個人情報、必要なカードのタイプや枚数といった他の情報を示した最初の指示を読む。そして依頼者は指示に基づき定型フォームに記入していく。記入が完了したら依頼者はオペレータの机まで歩いていきフォームとIDを手渡す。オペレータが依頼者に住民票を手渡したときひとつの作業は終了し、依頼者は状況に応じて次の作業にとりかかる。

オペレータの仕事は依頼者からフォームを受け取りできるだけ効率的に要求されたカードを発行することである。依頼者が窓口に来たときオペレータはIDをチェックし、カードを印刷して手渡さなければならない。オペレータはこの作業を終了するまで続けていく。

2.3.5. 条件

実験では、被験者はすべての条件での顧客役を交代で行った。なぜなら実験を通し依頼者がフォームに記入するスピードは変わらないと考えたからだ。オペレータ役は1人の被験者が連続して行った。1人の被験者に要求された状態は以下の通りである。被験者1-F3F1T3, 被験者2-F1T3F3, 被験者3-T3F3F1。

2.3.6. 結果

処理時間の比較

図10はそれぞれの状態での依頼者の待ち時間を示している。この待ち時間は実験の記録から計算されたものである。T3, T3', F1, F3の意味は以下の通りである。

- ・ T3 : T3の状態では依頼者が記入を完了してから住民票を受け取るまでの時間

- ・ T3' : T3の状態ではオペレータが依頼者と対応し始めてから依頼者が住民票を受け取るまでの時間。これはT3から並んでいる時間("a"図10)を引いて計算した時間と同等である。

- ・ F1, F3 : T3の状態では依頼者が記入を完了してから住民票を受け取るまでの時間

T3'とF1を比較すると、定型フォームの電子処理とFlyingFormシステムは依頼者の待ち時間を減らしていることがわかる("b"図10)。F1とF3を比較すると("c"図10)、FlyingFormでの依頼者の待ち時間は混んでいるときには増えていることがわかる。これはオペレータは複数の依頼者に対応する時は非常に忙しく、依頼者が来る前にいつでもフォームの処理を終えることができないからである。

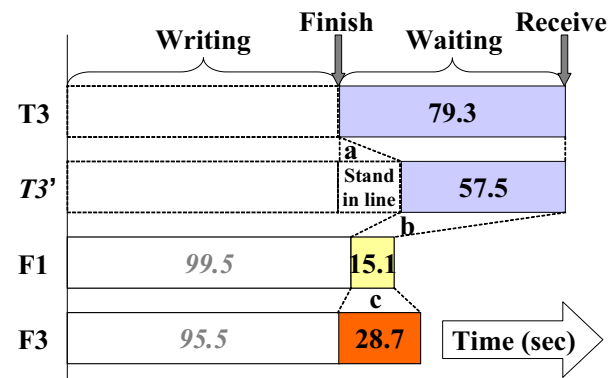


図10 : それぞれの状況での依頼者の待ち時間

図11はオペレータがそれぞれの処理にかかった時間を示している。我々はこのグラフからオペレータの情報処理量を評価できる。F1の場合、T3よりF1のほうが長い理由はオペレータが1人の依頼者だけに対応しているからである。最も重要なのはT3とF3の違いである。F3はT3よりも明らかに小さい。これは、多重タスク処理によるオーバーヘッドにもかかわらず、定型フォームの電子処理とFlyingFormシステムを用いることによりオペレータの情報処理量を改善することができたことを意味している。

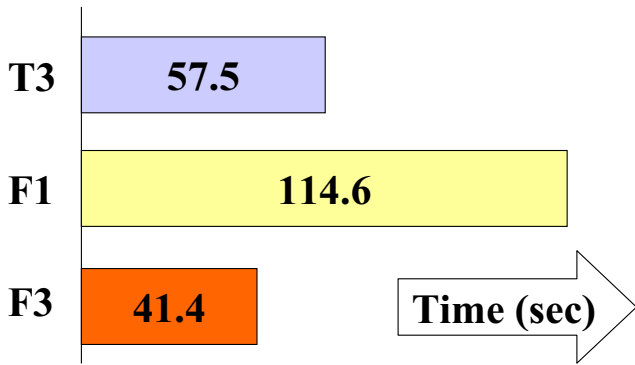


図 11: オペレータがそれぞれの処理で費やした処理時間

顧客役の被験者からのフィードバック

実験の後、我々は被験者にインタビューを行った。依頼者は FlyingForm の場合待ち時間がほとんど無かったとコメントした。もっとも重要な発見は、彼らは FlyingForm と従来の方法に有用性の違いを感じなかったということだ。NAVInote は FlyingForm を用いるときにアプリケーションとしてついてはいたがまったく問題とならなかった。

オペレータ役の被験者からのフィードバック

全体的な印象としてオペレータは FlyingForm システムは有効であると述べた。スクリーン上のフォームのほうが実際の定型フォームより処理が簡単ということに興味深い発見である。なぜならオペレータの目の動きが減ったからである。

色の強調はうまく機能した。色の強調は自然で、特に混んでる状態で、オペレータが前のフォームに戻らなければならないとき有効であったとの意見があった。混んでいないときは、オペレータは必ずしも有効というわけではないと答えた。しかしそのときでさえ色の強調は彼らを困らせるものではなかった。しかしながら処理の後にストロークが赤のままなのは少しいららしたと述べるオペレータもいた。

一方、スケジューリング支援機能は非常に有効なものではなかった。これは強調が十分でなかったというよりもオペレータはいま扱っているフォームの処理中には thumbnail パネルに注意を払うことができなかったからである。我々はこの問題を実際に使用する場合には解決できると考えている。なぜならそれぞれのフォームの仕事はこの場合より少なく、他のフォームを観察する時間があるからである。

多重タスク処理については、2人のオペレータが、3人の依頼者に対応するのはむずかしく、急いでやらなければならないと感じたといった。1人のオペレータはそれは刺激的な仕事だと言った。我々はこの問題をストローク認識機能などのさらに効果的なコンピュータの支援により仕事量を減らすことにより解決できると考えている。オペレータはまた他の依頼者に切り替えるとき thumbnail パネルのフォームをクリックしなければならないのは少し面倒だとも言った。

将来の改善点の提案として paper パネルと search パネルの対応の重要性を示した。例えばもしも paper パネルと search パネルがつながっていて対応する項目が近くに表示されていたら、オペレータは対応が非常に容易にわかり作業がもっと効率的になると言った。他の例としては、我々は search パネルと paper パネルのレイアウトを非常に似せて作ることができる。これはまたオペレータの作業を助けるものである。

2.4. 議論

実験の結果、FlyingForm システムによって依頼者の待ち時間を改善することが示せた。特に混んでいない状況では、依頼者は待つ必要が無かった。混んでいる時には、長く待たなければならなかったが、従来の手法より待ち時間は短くなった。ストロークの電子化、主に自動認識機能の効果と FlyingForm システムの効果両方から改善点が得られた。この結果から我々は完全に効果をわけて考えることはできないが、前者の効果のほうが比較的小さいと考えられる。なぜなら、このシステムは簡単な認識システムだけを用いて、手書き文字は認識することができないからである。したがって FlyingForm のコンセプトはシステムが十分な認識機能をもっていなくても有効性があると確認することができる。認識機能に関して言えば、我々はそれ自身の効果を次のユーザテストで調査したい。

上方で述べたように電子化と FlyingForm システム両方から改善があった。ここで我々はさまざまな状況における FlyingForm システムの有用性を議論したい。言い換えると前処理なしにデジタルペンシステムにおける FlyingForm のメリットを調査したい。簡潔に言うとそれらの違いは FlyingForm におけるオペレータが依頼者が窓口に来る前にフォームの処理をするということである。その有用性は以下の通りである。

- ・ Ta : 前もってすることができる仕事の時間 : その仕事は認識結果のチェック、情報検索、カード印刷などから構成される。これらの仕事は FlyingForm における多重タスク処理でなされており、この時間は待ち時間とは異なるものである。すなわち我々はこの時間により依頼者の待ち時間を減らすことができる。したがってもしこの時間が長ければ、効果は高いと考えられる。この時間は賢い自動認識機能などで減らせるので、これらの仕事から時間が減らせると確信している。

- ・ Tb : 残りの仕事の時間 : その仕事は依頼者の ID の確認、要求されたカードの手渡し、あいさつなどである。これらの仕事は依頼者がフォームを記入完了するまですることができない。これらの仕事はオペレータを多重タスク処理において混乱させる。これらの仕事はより良いサービスのためには重要なものなので減らすことは難しいということは注意をしたい。

- ・ Tf : 依頼者がフォームに記入する時間 : この時間は処理に直接影響する。なぜならもしもこの時間が短すぎるとオペレータは前もって手書きの処理を終えることができない。重要なのはこの時間と Ta との割合である。もし Tf が Ta と比べて小さい場合はオペレータは前もって仕事をおえることができない。

・C：混んでいるとき：これは間接的に影響する。もし窓口が混んでいるときはオペレータは多くの依頼者に同時に対応しなければならない。そしていつでも前もって仕事を終わることができない。これはまた多重タスク処理のオーバーヘッドも引き起こす。したがってたくさん混みあっていると効率をさげることになる。

FlyingForm の効果はこれらの要因によるものであり、そしてそれはさまざまな状況で変化する。future work としてこれらの要因を、仮説をたて FlyingForm の有用性を計算し、実際の状況で調査したいと考えている。

FlyingForm の改善のために我々は連続の処理モードに切り替える機能を考える。このモードは前処理なしにデジタルペンについている普通のシステムである。たくさんの依頼者が窓口に来たときオーバーヘッドの急激な増加によりオペレータは混乱するかもしれない。この場合 FlyingForm モードを連続処理モードに切り替えることが有効かもしれない。そしてオペレータは混雑が収まるまで依頼者を連続で処理できる。我々はこれを結局は全体の処理量を増やすだろうと考えている。

2.5. まとめ

我々はオペレータが依頼者が定型フォームにかいた手書き記入をリアルタイムで処理しすばやいサービスを提供できる窓口業務のためのシステムを提案した。地方自治体における発行業務を例とした FlyingForm というプロトタイプシステムを使って予備実験を行った。結果として、依頼者の待ち時間を減らし、オペレータの処理量を増やし、FlyingForm の有用性を示すことができた。スケジューリング支援機能は必ずしも有効ではなかったが、resume サポート機能とともに用いることで実際の業務では有効だと考えられる。

文 献

- [1] Arai, T., Aust, D. and Hudson, S. E. PaperLink: A technique for hyperlinking from real paper to electronic content. *In Proc of CHI'97*, 327-334, 1997
- [2] Davis, R.C., Landay, J. A., Chen, V., Huang, J., Lee, R. B., Li, F., Lin, J., Morrey, C. B., Schleimer, B., Price, M. N. and Schilit, B. N. NotePals: Lightweight Note Sharing by the Group, for the Group. *In Proc. of CHI'99*, 338-345, 1999
- [3] Guimbretière, F. Paper augmented digital documents. *In Proc. of CHI'04*, 51-60, 2004.
- [4] Heiner, J. M., Hudson, S. E. and Tanaka, K. Linking and messaging from real paper in the paper PDA. *In Proc of UIST'99*, 179-186, 1999.
- [5] Johnson, W., Jellinek, H., Klotz, L., Rao, R. and Card, S. Bridging the paper and electronic worlds: The paper user interface. *In Proc. of CHI'93*, 507-512, 1993
- [6] Miura, M., Kunifuji, S., Shizuki, B. and Tanaka, J. Augmented classroom: A paper-centric approach for collaborative learning system. *In proc. of UCS2004*, 57-64, 2004
- [7] Wellner, P. Interacting with paper on the DigitalDesk. *Comm. of the ACM*, 36(7):86-97, 1999
- [8] Yamashita, D., Takagi, H. and Hagiya, M. "Intelligent paper": A proposal of paper interface embedded in paper. *In proc. of WISS'99*, 1999.
- [9] 丸山泰史, 五十嵐健夫, Multicket: 液晶タブレットを利用した複数人同時対応インタフェース, 情報処理学会研究会報告, HI-118, 2006年5月19日.