

紙に埋め込まれたプログラムによる 紙インタフェース

山下 大輔[†], 高木 啓伸[‡], 萩谷 昌己[†]

[†] 東京大学 大学院 理学系研究科

[‡] 日本アイ・ビー・エム

紙の上の作業を計算機で支援し、より“賢い”紙を実現するために、紙の上の情報を処理し、結果を紙の上に出力するためのプログラムを、2次元バーコードであるQR(Quick Response)コードに符号化し、紙自身に埋め込む方法を提案する。このことで、その紙さえあれば、紙の上の作業を続行することができる。また、大量にコピーして配布したり、普通の紙と同じ管理や扱いを適用することができる。さらに、このような紙を計算機の環境に依存することなく利用できるように、紙に埋め込むプログラムを記述するためのAPIを提供する。今回の実装においては、Javaの上にAPIを構築している。

Paper Interface by Programs Embedded in Paper

Daisuke Yamashita[†], Hironobu Takagi[‡], Masami Hagiya[†]

[†] Graduate School of Science, University of Tokyo

[‡] IBM Japan

For realizing more “intelligent” paper by helping paper work with computer, we proposed the realization that programs for processing the information on paper and outputting the result on paper are encoded to QR (Quick Response) code which is a kind of two-dimensional barcode, and are embedded in the paper itself. With this, a user can continue the paper work only with paper. Such paper can be copied and distributed numerously, and can be applied to the same way of managing and using normal paper. Moreover, for using such paper without depending on the environment of computer, we provide the API (Application Program Interface) for describing programs embedded in paper. In this paper, we build the API on Java.

1 はじめに

WWWに代表される電子メディアが広く使われるようになった今日でも、紙はさまざまな場面で利用されている。例えば、書籍や雑誌、新聞、チラシやポスター、葉書、名刺などがそうである。これはなぜであろうか。

理由は幾つかあげられる。まず、電子メディアが実現していない、紙の利点が未だ数多くあることがある。例えば、情報の一覧性、可搬性、注釈（メモ）のし易さ、解像度の高さ、などが紙の持つ重要な利点である。電子メディアは、ディスプレイなどの制約により、これらの点で未だ紙に及ばない。また、紙は廉価であり、大量に複製して配布することも容易である。電池切れの心配もない。

さて、一般的に、計算機による紙の利用は

- プリンタによる“出力”
- スキャナによる“入力”

に限られている。しかし、紙の上の文字や図形を記憶されているデータとみなせば、紙は“記憶”媒体としても利用されていることになる。さらに紙は、人間による“計算”媒体としても用いられている。すなわち紙は、現在でも“情報処理”の媒体として有効に働いているわけである。

このような理由から、紙インターフェース [1] という試みが生まれ、いくつかの研究がなされてきている。紙インターフェースとは、紙と計算機をうまく融合して、紙の利点を活かしながら、より“賢い”紙を実現し、利用していくための手法である。また、これは“記憶”媒体としての紙の機能や、紙の上での“計算”を計算機で支援するための試みであるともいえる。

これまで研究されてきた紙インターフェースのほとんどのシステムでは、紙の上のID（バーコードや図形）と計算機内部のデータがなんらかの形で対応しており、それを利用して、関連するデータを引き出したり、操作していた。つまり、紙の上の情報は、計算機内部の情報とのリンクとして利用されており、システムを利用して紙の上で作業を行なう際の計算機への依存度が高かった。そこで我々は、紙の上でできる作業はなるべく紙のみでできるようにしようと考えた。そのようにすることで、計算機への依存度を減らし、紙の持つ可搬性などの利点をさらに活かすようなシステムができると考えた。

そのために本研究では、紙を使ってする作業に関して計算機が必要とする情報を、紙の上にすべて貼ってしまう、ということ提案する。こうすれば、その紙は、実際に計算機が何らかの処理を行うときのみ計算機（あるいは、その処理を行う処理系）に通せば良く、それ以外の場面ではまったく普通の紙として扱うことができる。大量にコピーして配布するようなこともできる。

紙を使ってする作業に関する情報としては、以下のようなものが考えられる。

- 手書きシンボル（文字や図形）を認識するための補助情報

- 紙の上の情報を処理するための手続き
- 手続き終了後、結果を紙に出力するための補助情報

このような情報を紙に埋め込むための手段として、我々は、2次元バーコードの一種であるQR(Quick Response)コード [10]を用いる。



図 1: QR コードの例

QRコードは、1セルで1ビットを表す、マトリックス式（縦横モザイク状）のバーコードであり、最大で約3kバイトの容量を持つ。また、誤り訂正、形状特定のための切りだしシンボルの配置、などの特徴を持っており、このような情報を格納するのに適していると考えられる。

さらに、紙の上の情報の処理を計算機でどのように記述するかであるが、我々は、紙の上の情報（文字や図形）に関する記述を容易にできるように、そのためのAPI(Application Program Interface)を設計し、それを用いて処理の手続きを記述できるようにする。このAPIは、紙の上の手書きシンボルの認識のための機能、および、紙の上に結果を出力するための機能を備えている。文字や図形の認識については、現在の技術で出来るものを考える。本論文では、認識技術は主眼としない。また、紙の上から取得された情報を用いた計算の部分については、既存のプログラミング言語を利用する。

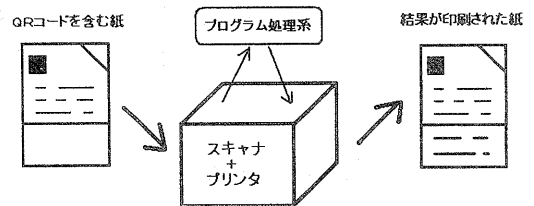


図 2: システムの概要

本研究のシステム構成は図2のようになる。ユーザは、計算あるいは処理に必要な情報を紙の上に記述し、それを、スキャナとプリンタが一体化したデバイスに通す。するとそのデバイスは、QRコードを認識してプログラムを復号化し、紙の上

の情報を認識して、QRコードに格納されていたプログラムを実行し、結果を紙の上に印刷する。

2 プロトタイプ

ここでは、実際の簡単な応用例を述べ、その応用例に必要なと思われる機能を実現していくことを通して、最小限のAPI-setを構築する。なお、今回のプロトタイプの実装に関しては、Javaを使って各機能を実現している。

2.1 “テストの採点”アプリケーション

応用例としては、“テストの採点”を考える。ここでのテストは、問題は○×形式であり、テストの紙面は図3のようなものである。

図 3: テストサンプル。左上がQRコード

ここでは、名前欄（実際には学生番号欄）と回答欄にそれぞれ特定の領域が割り当てられ、テストの回答者は、決められた回答欄にそれぞれ○、×で問題に解答する形を取るものとする。また、採点結果は同じ紙の結果欄に出力される。

このようなテスト採点のプログラムを作ろうと思ったときに、紙の上の情報を利用するために必要な最小限の機能としては、次のようなものがあげられる。

- 名前欄、回答欄の場所の特定の機能
- 読みとった領域に対して何らかの文字認識を施すための機能
- 結果欄の場所を特定し、採点の結果を出力するための機能

2.2 API

このプロトタイプにおいては、上で述べた必要な機能からも分かるように、回答欄などの閉じた領域が基本的な処理の単位となっている。後でいくつかの応用例を述べるが、それらの場合でも、“閉領域”を基本的な単位とした処理が多くなっている。このことから今回のプロトタイプの実装には、Javaを利用して、次のようなAPIを構築した。

- `searchClosedArea(int num)`
QRコードを原点とした座標系から右方向に閉領域をサーチし、`num`番目に見つかった領域を、`ClosedArea`クラス¹のオブジェクトとして返す。ただし、ある程度の大きさの閉領域のみを対象とする。（これは、文字の中に小さな閉領域があると認識されることを防ぐためである。）
- `f(ClosedArea arr, ...)`（ただし`f`は`ClosedArea`オブジェクトを引数にとる関数）
`ClosedArea`オブジェクトで指定されている閉領域の画像に対してなんらかの処理を行なう。ここでは、`f`として、
 - `judgeMarubatsu`
そこに書かれているのが○であるか×であるかを判定する
 - `ocrOfNumber`
文字認識（ここではintegerのみを扱う）を行なう
 - `printStringOnArea`
指定された閉領域の中に文字列を書き込む

の3つを利用する。

現在、閉領域サーチの実現は、紙をスキャンした画像を2値のビットマップ画像と見て行なっている。画像をスキャンしていき、ある点が見つかったら、その点を端点とする線分をた

¹`ClosedArea`クラスは、閉領域の形（四角など）と閉領域の位置情報（四角ならば、左上と右下の座標）を変数として持つ抽象データ型である。このクラスのオブジェクトが表している画像を利用したい時には、その情報を基に画像（ビットマップ画像）として取り出すことにする。

どっていく。線分をたどって行って元の端点にたどりつければ、その線分は閉領域を構成していると判断している。上で述べたように、小さい領域は、例えば漢字の「口」のような字が構成している閉領域である可能性があるため、切り捨てている。この単純な方法だと、画像全体をビットマップとみなしてスキャンしていくので、処理時間は画像の大きさに比例してかかる。この部分の実装は Java のソースにして約 500 行程度となっている。

これらの API を用いて、簡単なテストの採点プログラムを作成したのが以下である。実装は OS が WindowsNT と FreeBSD を併用し、CPU は Celeron300AMhz、Java の処理系は JDK1.1 を使用した。プログラムのサイズは約 1 k バイトであり、十分 QR コードに格納できる大きさであった。今回のプロトタイプでは、QR コードに、ソースコードをそのまま格納しているが、処理の時間、コンパイルでの問題発生の危険性等を考えるとバイトコードを格納する方が良いと思われる、そのようにする予定である。

テスト採点のプログラム、

```
public class MarkTest extends RecognOnPaper {
    public int point; // 得点
    public int namelen = 5; // 名前 (学生番号) 欄の長さ
    public int num = 5; // 問題数

    // 各問題の回答とその得点を配列に格納しておく
    public final int trueans[] = {0, 1, 0, 1, 0};
    public final int anspoint[] = {20, 20, 20, 20, 20};

    public static void main (String argv[]) {
        int i;
        int name[] = new name[namelen];
        int ans[] = new ans[num];

        // 学生番号を欄から読みとる
        for (i = 0; i < namelen; i++) {
            name[i] = ocrOfNumber (searchClosedArea (i));
            namelen++;
        }

        for (i = 0; i < num; i++) {
            // ○か×かを回答欄から読みとる
            ans[i] = judgeMarubatsu (searchClosedArea (i + namelen));

            if (trueans[i] == ans[i])
                .point += anspoint[i]; // 加点
        }

        // 点数に応じたコメント
        if (point < 40) String str = "しっかり勉強しましょう";
        else if (point < 80) String str = "もう少し努力が必要です";
        else String str = "よくできました";

        // コメントとスコアを、対応する欄に書き込む
        int newnum = num + namelen + 1;
        printStringOnArea (searchClosedArea (newnum), str);
        printStringOnArea (searchClosedArea (newnum + 1), point);
    }
}
```

3 応用例

この研究の簡単な応用例をいくつかあげる。

● スケジュール帳

QR コードが貼られたメモページにメモを書き、同じページの日付け欄にチェックして、それをスキャナに通す。そしてスケジュール帳の対応するページをスキャナに通すと、スケジュール帳にメモが反映される。

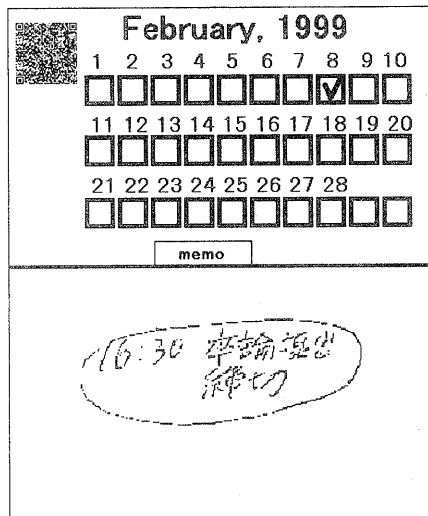


図 4: スケジュール帳のサンプル

● 住所録

印刷された住所録を更新するための手続きを QR コードとして貼っておく。例えば、二重線で住所録のある項目を削除することができる。また、計算機のない場所、例えば外出中に名刺などをもったら、その紙の“新規住所登録欄”に名刺を貼って、紙をスキャナ+プリンタのデバイスに通すと、住所録に新しい項目を追加できる。この他にもいろいろな手続きを登録できるだろう。

● 占い

占いプログラムを QR コードとして紙に貼っておき、人間はその紙の入力欄に自分の個人情報を書き、スキャナに通す。すると、占いの結果が出力される。世の中にはいろいろな占いがあるが、各々の占いプログラムを QR コードに入れておいて、本として綴じておけば、好きなものをピックアップして利用することができる。

4 関連研究

他の紙インタフェースの研究について述べる。

PaperWWW (脇田ら [5]) は、URL を圧縮したものを QR コードで表し、Web ページを印刷する時にリンク位置に書き込むことで、紙を Web ページのように扱えるようにするシステムである。

PaperLink (荒井ら [6]) は、VideoPen という、蛍光ペンと小型カメラが一体化したデバイスを用いて、紙の上の任意の場所にハイパーリンクを貼ることのできるシステムである。蛍光ペンで塗られた領域をカメラで切りだし、その場所にハイパーリンクを貼れる他、カメラで文字を認識してアプリケーションの引数とすることもできる。

Ultra Magic Key (白田ら [8]) は、ノートの端に印刷された“鍵”の向きでページを識別し、ノートと一体になっているディスプレイの上に設置された小型カメラで、指先を認識して、その場所に対応した情報がディスプレイに映し出されるというシステムである。

EnhancedDesk (小林ら [7]) は、上方から机の上の紙面をカメラで捉え、紙面の DataMatrix と呼ばれる 2 次元バーコードを認識し、対応する情報をプロジェクタで紙面に重ね合わせることで、紙面に動的な情報を与えることのできるシステムである。また、それらの電子的な情報を指で操作することも可能である。

XAX (Walter Johnson ら [4]) は、紙の文書を計算機に戻すためのシステムである。カバーシートと呼ばれる、文書の形式やその種類に関するチェックシートを用いて紙の文書を機械的に識別し、その情報に基づいて文書を読みとり、文書を計算機に再び戻すことができる。カバーシート自体の情報は計算機内に格納されており、DataGlyph という特殊なバーコードでカバーシート自体の ID を取得する。

5 議論

本研究のアプローチについて議論する。

1 章で述べた通り、関連研究のほとんどが、紙の上の情報を ID として、計算機内部のデータを引き出し、何らかの作業を行なう、というシステムである。そのようなアプローチでは、

- 計算機の操作を紙の上での物理的なインタラクションによって行なえるため、より直観的でわかりやすい操作が行なえる
- 計算機の中の動的な情報や音声情報などを紙に与えることができる

などの利点がある。一方で、

- 計算機への依存度が高く、紙の可搬性などの利点を十分活かしているとはいえない
- システムが大がかりになりがちである

などの欠点がある。

関連研究の中で、XAX が、本研究の考え方にもっとも近いといえる。XAX においては、情報はほとんど紙の上で存在している。しかし、カバーシートに記述されている文書のフォームはあらかじめ決められたものであり、各々のカバーシートの内容、つまり、紙の上の文書のフォーム情報は計算機の中に格納されている。また、紙の上の作業を指向したシステムではなく、あくまで計算機への入力に紙を使用する、という立場である。

本研究では、紙の上の情報をういた計算の処理自体を紙に直接埋め込んでしまうことで、計算機への依存度を減らし、紙の利点をなるべく活かすようにした。本研究のアプローチでは、以下のような利点がある。

- システムの利用者は、紙の上に必要な情報を書いて、それをスキャナとプリンタが一体化したデバイスに通すだけで良く、扱いが容易である
- 計算機への依存度を減らしたことで、紙の上の作業の大部分を紙のみで行なうことができる
- 結果が必要な時以外は普通のメモ用紙やスケジュール帳として、今までのいわゆる“紙”と同じ扱いができる
- 紙自体に全ての情報が含まれているので、紙のコピーとプログラムのコピーが“物理的”に対応している
- 紙はフロッピーディスクのメディアに比べ廉価であり、大量にコピーして配布することも容易である
- 紙の上の情報を処理するための基本的な機能をシステム側で提供することによって、実際に紙を用いて行なう作業のための手続きを容易に構築することが可能となる

本研究の問題点としては、

- QR コードの容量に限界があるため、大きいプログラムを入れることができない (複数並べる・圧縮するなどの解決策が必要)
- 計算機の能力を限られた範囲でしか利用できないため、動画や音声などは扱えず、インタラクティブ性に欠ける
- QR コードが人間に可読でないので、どのような処理が行なわれるかをあらかじめ知っておくか、あるいは紙の上に行なわれる処理についての説明がなければならない
- 上の項目とは逆に、もし QR コードを読むことができ、その手続きの詳細が分かってしまったら、例えばテストの採点のようなアプリケーションは成立しなくなる (暗号化などの対策が必要)

などが挙げられる。

以上の考察から、本研究の用途としては、次のような特徴を持つ作業が向いていると考えられる。

- 計算機とのインタラクションをほとんど必要とせず、紙だけで情報が閉じている

- 文書の形式（フォーマット）のバリエーションが豊富にとれる
- 計算機を余り利用しないユーザが普段紙の上で行なっている
- 計算機のアプリケーションとしても存在しているが、現実世界では普通は紙を利用して行なっている

考えられる応用例の中でも挙げているが、たとえば、占いやパズルといったものは、さまざまな種類のものがある。しかしフォーマット情報と占いやパズルのプログラムをQRコードに入れて紙に貼っておけるので、例えばたくさんの種類を本にまとめて配布すると、それを利用する人は好きなページを抜き出して占いやパズルを実行することができる。このような紙は、計算機をほとんど意識することなく利用することができる。

また、スケジュール帳などは、計算機や携帯端末上でもスケジュールとして存在しているが、それらが利用できない時、例えば急な電話や外出先などでは、QRコードにスケジュール帳更新のためのプログラムが入っている紙の上に手書きで情報を加えておけば、あとで計算機に更新を反映できる。また、紙のスケジュール帳の更新にも、そのような紙が利用できる。

6 まとめ

紙の上での作業を計算機で支援し、より“賢い”紙を実現するアプローチとして、紙の上に、計算機が行なう手続きを貼ってしまうという方法を提案した。さらに、そのような手続きを容易に記述するためのAPIの提供について、簡単な例をもとにどのような機能を提供すればよいかをみた。

現在の状況としては、スキャナに紙を認識させるとQRコードに格納されているプログラムが起動し、結果がプリンタから出力される、という一連の仕事をひとまとめに実装するまでに至っておらず、各仕事はバラバラとなっている。これらの仕事を完全に実装し、システムを利用していきながら、新たな応用例を見い出していきたい。

また、今回の閉領域を基本単位としたAPI構築のアプローチに関しても、それがいろいろな応用例に対してうまく利用できるか、を考える必要があると思われる。現在はJavaでの実装であるが、PostScriptのような独自言語を構築し、計算機の環境に依存せず、自由に紙の上の情報処理の手続きを記述できるようにすることも考えていきたい。

参考文献

- [1] 増井俊之: 「ペーパーインターフェース」, UNIX MAGAZINE 1998.10, pp.154-160, アスキー, 1998.
- [2] Pierre Wellner: Interacting with Paper on the DigitalDesk, In *Communication of the ACM*, Vol.36, No.7, pp.87-96, 1993.
- [3] Jun Rekimoto: Pick and Drop: A Direct Manipulation Technique for Multi-Display Environments, In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'97)*, pp.31-39, ACM Press, 1997.
- [4] Walter Johnson, Herbert Jellinek, Leigh Klotz Jr., Ramana Rao and Stuart Card: Bridging the paper and electronic worlds: The paper user interface, In *Proceedings of the ACM INTERCHI'93 Conference on Human Factors in Computing Systems (CHI'93)*, pp.507-512, Addison-Wesley, 1993.
- [5] 脇田敏裕, 長屋隆之, 寺嶋立太: 2次元コードを用いたWWWと紙メディアとの融合の試み, 情報処理学会ヒューマンインタフェース研究会研究報告 98-HI-76, Vol.98, No.1, pp.1-6, 1998.
- [6] Toshifumi Arai, Dietmar Aust, Scott E. Hudson: PaperLink: A Technique for Hyperlinking from Real Paper to Electronic Content, In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'97)*, pp.327-334, Addison-Wesley, 1997.
- [7] 小林元樹, 小池英樹: 電子情報の表示と操作を実現する机型実世界指向インタフェース「EnhancedDesk」, インタラクティブシステムとソフトウェアV (*WISS'97*), pp.167-174, 近代科学社, 1997.
- [8] Hiroshi Usuda, and Mitsuhiro Miyazaki: The multimedia interface using “paper”: Ultra Magic Key, In *Proceedings of Asia Pacific Computer Human Interaction 1998 (APCHI'98)*, pp.393-397, IEEE Computer Society Press, 1998.
- [9] 椎尾一郎, 美馬義亮: IconSticker: 実世界に取り出した紙アイコン, インタラクティブシステムとソフトウェアVI (*WISS'98*), pp.105-114, 近代科学社, 1998.
- [10] QRコードまめ知識, <http://www.denso.co.jp/EAP/mame/qrcode/index.qr.html>.