

TPO モバイルギア 『ぱび』 と情報キオスク

TPO mobile gear "PUBi" and information KIOSK

山本吉伸
Yoshinov YAMAMOTO

独立行政法人産業技術総合研究所サイバーアシスト研究センター(〒135-0064 東京都江東区青海 2-41-6 E-mail: yoshinov.yamamoto@aist.go.jp)

ABSTRACT. Today, many feel that the notion, "(to receive the service) any time any where" is the keyword of the ideal information environment; thus, high-tech cellular phones are believed by many to be the ultimate form of the informational service device. With the analysis that a cellular phone will act like credit cards and admission tickets, an image of the future, which all the informational services are provided through a single high-tech cellular phone, seems to be very persuasive. We, however, believe that the world of such is not the ideal world. In this paper, I describe our idea about "networked display in front of a shop". We developed a demonstrative system of the public board that is able to change what it shows on the display from at the town corner to information KIOSK. This system suggests that the mobile service of "only now, only here" is more valuable than "any time any where"

1. 背景

今日、もっとも注目されている情報環境のキーワードは「モバイル」と「ユビキタス」である。携帯電話インフラはこの二つのキーワードに対してもっとも有利な地位を占めているが、ホットスポットと呼ばれる第二のモバイルインフラが徐々に注目されつつある(1)。

このような状況にあつて、我々は従来のモバイル情報環境に次のような不満を感じている。

1. せっかく外出中だというのに、室内にいるときと同じように Web ブラウズをすることが未来の夢の生活か？
2. せっかく他者が存在する公共空間に居ながら、そこで提供されるサービスは個々人とインターネットの関係で閉じてしまっている。第三者とのインタラクションが一切発生しないのはもったいない話ではないか。

我々が本プロジェクトで中心的存在と位置付ける「パブボード」は、店舗の看板のように Welcome メッセージなどを表示するディスプレイをもったホットスポットの一種である。パブボードは、ぱび(PUBi)と呼ばれる小型の手持ちデバイスに反応する。PUBiを持ったユーザはパブボードを自分の PDA のディスプレイのように利用することができる。パブボードを利用することで、従来のモバイルサービスの延長線上では実現し得ない重要且つ興味深いサービスを提供できるようになる。

2. 目的

平成 13 年度末踏ソフトウェア創造事業「TPO モバイルギア『ぱび』と情報キオスク」プロジェクトの目的は、サービスコンセプトの議論とビジネスプランの構築、およびコンセプトに基づくデモシステムの実装である。

¹ ホットスポット: 空港や店舗等にアンテナ局を設置し、無線 LAN によってユーザがインターネットに接続できるようにするもの[1]。

3. モバイルサービスのニーズ

我々は、モバイル環境にはモバイル環境にふさわしい情報サービスがあり、それはデスクトップ環境で受けられる情報サービスとは異なると考えている。

CPU やメモリ等の半導体技術の向上は情報産業全般に大きな影響を与えるが、そのなかでも特にモバイル情報環境の進展に強く影響を与えている。いまや、PDA や携帯電話等のモバイルデバイスの機能は二世代之前のデスクトップ PC に迫る勢いである。電子メールや Web ブラウジングはもちろんのこと、Windows 上のビジネスアプリケーションソフトとして有名な Power Point や Excel とデータ互換性のあるソフトウェアを実行できるものも登場している。しかし、いったいどれほどたくさんの人が、外出中に Power Point や Excel を操作できることを嬉しいと思うだろうか。



Figure 1 パブボードのある風景

我々は「ユーザのモバイルリテラシー(モバイル環境を使いこなす技能)は進化する」と予測する。現在はまだモバイル環境で情報サービスを受けることができるようになってから日が浅い。そのためリテラシーはほとんど発達しておらず、ただ外でなんでもできるということを喜んでいるような状況にある。しかしながら、やがてリテラシーが進化すれば、外でやるべき作業とデスクに戻って行うべき作業を上手に切り分けることが

できるようになると予測する。このような時代には「外でやるべき作業」のニーズに答えることがモバイル情報サービスに求められるはずである。

では、外でやるべき作業とはなにか。

近未来のモバイル情報サービスのキーワードは「いまだからこそ／ここだからこそ」である。平成 12 年度未踏ソフトウェア創造事業「リソース共有メカニズム・パブボード向け基本ソフトウェアの実証研究」成果報告書で我々は、ユーザのいる場所に依じた(特定の空間(場所)に依存した)情報サービスこそ注目すべきであることを指摘した。同事業「携帯電話版 SpaceTag の試作: モバイル仮想環境への第一歩」成果報告書でも同様の主旨が述べられている。

しかし本稿では、このコンセプトを半歩進めたい。「いまだからこそ、ここだからこそ」のコンセプトだけでは、モバイル情報環境の重要性を際立たせるには不十分であると考えからである。つまり、Web を閲覧して回る(いわゆるネットサーフィン)で発見する情報の価値も、そのサイトが時限の開設であったり偶然に出会ったりするものであるならば「いまだからこそ、ここだからこそ」の価値があることには代わりなく、その意味からはモバイル情報環境の目指すべき方向として掲げるには不明瞭といえる。

(1) 生活圏の情報収集

我々が掲げる「外でこそ行うべき作業」の第一は、「生活圏の中の情報収集」である。

自宅の PC や携帯電話機から世界中の情報にアクセスすることを可能にしているインターネットの可能性は小さくないが、自分の興味関心に合う情報を上手に探すという作業は(少なくとも現時点では)多くの人にとって簡単なこととはいえない。地球の裏側の情報は時として貴重かもしれないが、多くの人にとっては地球の裏側よりも自分の生活範囲に関わる身近な情報こそが重要な情報であると考えられる。それに関わらず、検索サイトなどを利用して自分の生活圏の情報だけを検索することは一般的には困難な作業である。我々は、生活圏の情報の収集こそが外出中に行うべき作業と考える。

(2) 体験の共有

どのような理由で(人々は)外出するのかを考える。外出の目的は「購買(あるいは将来の購買)」や「(映画や美術館などの)視聴」、「(他者との)会合」など、さまざまあるが、これまでにほとんどモバイル情報サービスでサポートすることが想定されていない目的の外出がある。それが「体験の共有」を目的とする外出である。

シリコンバレー近辺では、各ダウントウンで頻繁に"Wine and Art Festival"が開催される。Festival の当日はダウントウンが歩行者天国になり、小さなブースが建ち並ぶ。各ブースでは民芸品を売る者、自作の CD を売る者、木の玩具を売る者、さまざまである。会社がブースを出すこともある。ブースを出している人はほぼ常連で、近隣のダウントウンで Festival がある時には同じ店をなんども見かけることになる。したがって、二〜三回足を運べば概ねあとは何度行っても同じことである。ところが、festival はいつも活況である。朝市のように、買い物か目的であればこれはなんら不思議なことではない。しかし Wine and Art Festival では、ほとんどの客が window shopping を楽しむことを主な目的としているように見え、商品購入を目的として来ている人の方が少ないように思われる。これが「体験の共有を目的とした外出」の例であると考えられる。

共通のスポーツが好きだという人同士ならばそのスポーツのために時間を使うだろう。映画が好きなら映画館にいったりビデオをみたりして過ごすであろう。音楽が好きなら人同士なら音楽を聞いて過ごすかも知れない。だが、とりたてて共通の趣味がない二人(あるいは二人以上)で共通の時間消費を考えなければならないときには、メンバーの最大公約数的な選択(全員が妥協できる選択)が行なわれるはずである。Wine and Art Festival のような「無料イベント」は、「夫婦＋子ども(=家族)」というグループには特に選択されやすいと想像される。「買物 (windows shopping を含む)」もまた選択されやすいイベント(時間共有のテーマ)である。日本でも米国でも、休日のショッピングセンターは活況である。必ずしも買いたいものがあるわけではない。しかしショッピングセンターにいけばとりあえず時間は潰れるレストランはあるし子どもを遊ばせることもできるし、なにか面白いものを見つけられるかも知れない。家族で「とりあえずショッピングセンターにでも出かけるか…」はとてもよい「時間を共有する企画」といえる。



Figure 2 キック

人間は社会を構成する生物であるが故に、他者とのコミュニケーションは自然な欲求の一つであると考えられる。体験を共有することで、相手とのコミュニケーションの話題を生成することが容易になる。

このことから、(特に目的のない多くの外出者に) 楽しい共有体験を演出(あるいは次回の外出の目的を演出)したり、なんらかの共有体験をデータとして残したりといったサービスが、モバイル環境の重要なニーズであると考察する。

4. サービスとビジネスイメージ

ここまでの議論を踏まえて我々が実装したサービスのうち、ユーザのメリットごとに代表的なものだけを述べる。

(1) PUBi ユーザからのメリット

a. キック

パブボード上に表示された情報のうち、気になったものの URL を自宅のメールアドレス宛に送信させる機能である (Figure.2)。「鉛筆を出してメモする程でも無い情報だが、それなりに気にはなっている(メモしなかったことを後で後悔することも往々にしてある)」ような情報に関しては、情報にアクセスする手間の閾値を下げさせることができる。これは『インターネットで配信されている情報のうち、ユーザの「(本人が実際に外向く、という意味での)行動範囲の中での」情報をフィルタにかけている』という、これまでのインターネット環境では実現できていない、生活圏内での情報フィルタリングを実現している。

b. 空間ブックマーク

新しく見かけた店舗に興味を持つことがある。「新しくこん

なところに骨董屋さんがあるな」とか「おいしそうなレストランだけど、いまは食事時間じゃないから今度来よう」といった発見は、しかし、容易に忘れ去られてしまう。空間ブックマークは、店舗の情報を自宅にメールとして届けるサービスである。ユーザは店舗の前で PUBi 上のボタンを押すだけでよい。このサービスが重要なのは、新しく発見した店について友人と情報交換が可能になるので話題ができるということと、もう一つはまとまった空間ブックマークのリストは有償無償で他人に提供できるということである。たとえば有名タレントのお勧めの店一覧を購入することも可能になるし、観光地のお勧めコースをやり取りすることも期待できる。

c. スナップショット

従来のカメラは、持ち主が一番スナップに写りにくい、という欠点を持っていた。家族などで出かけた際の写真に(カメラ係の)お父さんがいつも写っていないというような事例はよく見られる。この問題はパブボードと PUBi の組み合わせによって実現できるスナップショットサービスによって解決される。

ユーザは、観光地や遊園地の入口で PUBi を受け取る (Figure.3 中の A)。カメラは園内のスナップスポットに適宜配置されており、その近辺に近づくと PUBi はチャイムを鳴してスナップスポットであることを知らせる。ユーザは、好きなタイミングでどんどんシャッターを切ることができる。ときどき、システムの判断でシャッターチャンスに自動的にシャッターを切ることもありえる(B)。出口で PUBi を返却し、サムネイルを印刷した紙と、URL とパスワードを受け取る(URL とパスワードはメールでも発送される)(C)。自宅に帰ってから、その URL にアクセスすると、300万画素(写真クオリティ)のデジタル映像がダウンロードできる。ダウンロードした場合のみ課金される。なお、議論のためにダウンロードサイトのサンプルも試作された[2]。

d. 空間リマインダ

空間リマインダは、任意の空間に到着したときに要件を思い出させてくれるサービスである。ポストのそばのパブボードでは「郵便ポストがあります」などのメッセージを表示することができる。ユーザの手持ちデバイスのアラームを鳴すことも可能であるが、アラームは個人の作業に割り込みをかける手段であり、頻繁に鳴るようなことも想定されるので好ましくない場合も多い。パブボード上の表示であれば、ユーザに割り込みをかけることはないという利点がある。

空間リマインダの応用は広い。(なんらかの手段で) 買い物リストをサーバ上に登録しておけば、パブボード設置店では買い忘れを防ぐことができる。パブボードは PC で実現できるので、ユーザのオフィスの PC 上にパブボード機能を用意しておけば、「オフィスにいったらこれを思い出したい」という要

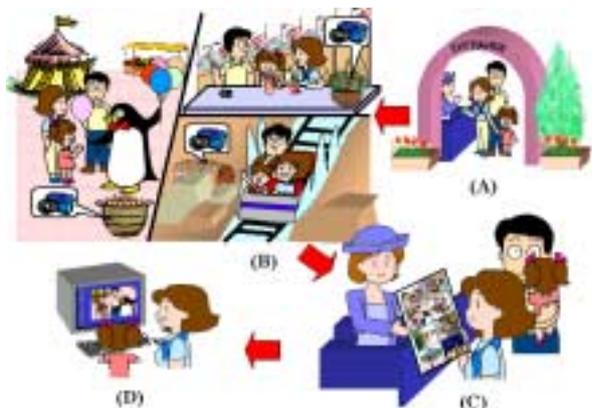


Figure 3 スナップショットサービス

件を登録しておくことができるようになる。さらに、場所/時間の条件で思い出させるだけでなく、さまざまなコンテキストによって情報を想起させるサービスへの発展が期待される。

(2) パブボードオーナーからのメリット

a. 顧客とのインタラクション創出

顧客がパブボードに近づくと自動的に「このパブボードの近くに来た」という情報を取得、記録することができる。さらにユーザからの反応 (PUBi のボタンを押したかどうかなど) を取得することができる。この基本メカニズムは、単なるディスプレイでの情報提示と本質的に異なる点の一つである。また、初めて来た人、二度目に来た人、常連さん(またはメンバー)といった条件で提示内容を変えることが可能であるだけでなく、顧客自身が表示するコンテンツを切り替えたりすることができるようになる。これらの特徴により、単なるディスプレイでは通り過ぎていた顧客も足を止めることが増加すると期待できる。ただし、コンテンツは各店舗で用意しなければならない。また、PUBi が十分に普及していない段階では CRM のような性能までは期待できない。

b. 社員証

導入した店の店員/社員に PUBi を配布することでタイムカードのように利用することができる。また、入店した顧客や天候その他の状況に合わせて店員がパブボード上の情報を切り替えて使うシステムが可能である。誕生日の客がいるときに誕生日用のコンテンツをすばやく表示できるレストランなどが容易に実現できる。

(3) 非 PUBi ユーザのメリット

a. クラシファイド

売ります/買いますの記事や、猫もらってくださいといった個人の広報を投稿することができる。我々のシステムでは、個人のクラシファイド広告に「もぎり」をつけることができる (Figure.4)。もぎりの表示によって、広告をみたユーザとのインタラクションを促進させることが期待される。ただし、これらのクラシファイド広告を表示するかどうかはパブボードオーナーに権限がある。

5. 実装



Figure 4 モギリ(下部)

(1) 全体構成

本システムは、PUBi、パブボード、サーバから構成されている (Figure.5)。パブボードのハードウェアは、機能的には PC と同じである。パブボードは PUBi と無線で接続する。本試作では無線手段として bluetooth を採用した。パブボードはインターネット上のサーバと接続されており、ユーザの認証等はすべてサーバで行う。サーバは各パブボードへの情報配信をはじめ、すべてのデータの管理を行う。

(2) PUBi

PUBi は、Handsprings 社の visor prism に Widcomm 社の Bluetooth device である BlueConnect を接続したハードウェア

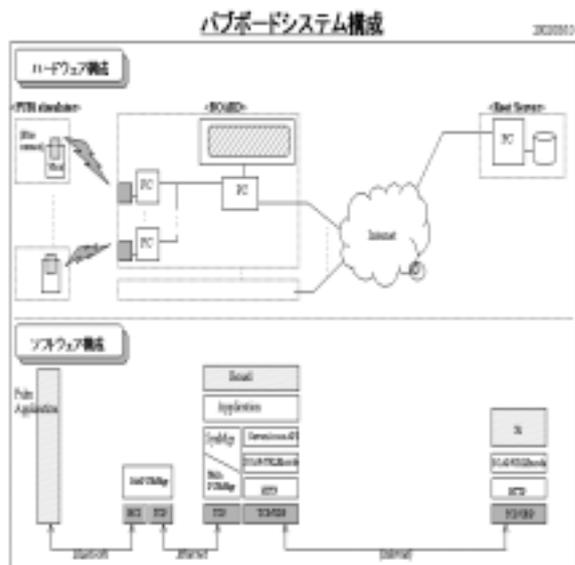


Figure 5 システム全体構成

アに専用のソフトウェアを搭載してエミュレートした、PUBi のヒューマンインタフェースは、ゲームパッドのようなボタンだけである。ディスプレイはない。常にパブボード側からの呼び出し信号の待ち受け状態であり、パブボードからの信号を受信したときには速やかに反応を返す。

(3) PUBi マネージャ

パブボードの PC 上で動作するモジュールであり、PUBi の発見と接続、各 PUBi とのセッションの管理を主な任務とする。PUBi マネージャはさらに、SubPUBiMgr と MainPUBiMgr に分かれる。これは、現在使用している bluetooth ドライバが1つの Bluetooth デバイスとしかシリアル接続出来ないという制限があることによる。接続拠点である SubPUBiMgr を複数の PC で実現して、中心となる PC (パブボード) に MainPUBiMgr を配して複数の PUBi との接続を可能する構成になっている。

MainPUBiMgr は SysMgr モジュールと接続されている。SysMgr は、PUBi のアクションにより生成されるプロセスとパブボードやサーバなどのシステムプロセス間の通信を実現するためのメッセージキューを提供するもので、その内部に個々の PUBi 情報管理する情報管理機能を持つ。

発見された PUBi には、PUBiMgr により一度はシリアル接続され(この接続をフック接続と呼ぶ)、SysMgr の管理するパブボードの情報が PUBiMgr をとおして PUBi 側に通知される。また、同時にパブボード特定番号を SubPUBiMgr に送り返す。パブボード特定番号とは、PUBi 上のあらかじめ定められたカーソルキー操作列に対応した数字である。たとえば1は↑、2は→、3は←、4は↓、5は↑↑、5は↑→のように決められている。パブボードの付近に来た PUBi は、パブボード側から bluetooth のアドレスを受け取り、パブボード特定番号と関連付ける。これによって、たとえば↑カーソルを一つ押すとどの bluetooth のことを指したいのか、ということ特定することができる。

最初のフック動作は、SubPUBiMgr が PUBi を発見して(その通知を MainPUBiMgr が受信し)、それに応じた MainPUBiMgr が接続を SubPUBiMgr に指示することにより行

われる。MainPUBiMgr はこの時、すでにフックしている PUBi であるかどうかを検査し、フック済みである場合は SubPUBiMgr に接続を指示しない。これにより同じ PUBi に対して何度も最初のフックをする事を防いでいる。また、SubPUBiMgr だけの判断でフックを行うと、複数の SubPUBiMgr が同時に同じ PUBi にフック動作を行う危険性があり、この回避もこの処理により行われている。

実際のフック接続は SubPUBiMgr と PUBi の間で、Bluetooth のシリアル接続として行われフック動作が終了すると直ちに切断する。接続したままにしない理由は、SubPUBiMgr の Bluetooth が一つの PUBi に占有されることが無いようにするためである。

フック動作の中には発見されたデバイスが本当に PUBi であるかの検出もフック接続に含まれる。具体的にはシリアル通信で SubPUBiMgr が「PUBi?」と言う文字列を PUBi に送信したときに、その応答として PUBiID が得られるかどうかで行っている。

(4) パブボード

このモジュールの主な機能は次の三つである。

- SysMgr やユーザアプリケーションからの指示に基づき画面表示を行う。
- 広告データを表示する。
- サーバに対してバナーメッセージを請求し、それを表示する。

Figure.6 はパブボードのインタフェース構成を示している。

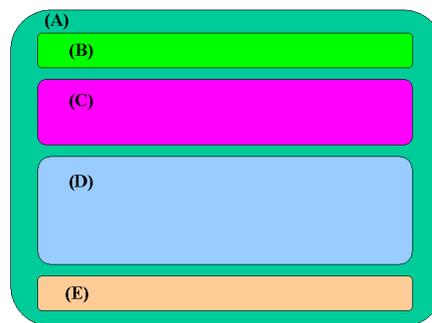


Figure 6 画面構成

領域(A):パブボードの背景領域である。画面(描画可能な最大面積)に対して背景画像が置かれる。領域(B)~(E)の横幅は、すべて 192dot の整数倍で構成されている。同様に縦幅は、96dot の整数倍によって構成されている。

領域(B):この領域の用途は二つある。ひとつはシステムからのメッセージを表示するためである。デバッグ作業を行う上で、この領域に文字列を表示できるようになっている。もうひとつの用途は「待合室」である。パブボードに近づいている PUBi のアイドル状態が続くとき、ユーザのアイコンはこの領域に移動する。この領域に入りきらないアイコンはバックグラウンドに隠れる。現在の実装では、システムからのメッセージを表示する領域として最低 192dot 分を残して、残りをアイコン表示のために使うことができるようになっている。

領域(C):この領域の用途は二つある。ひとつはモグリなどのポストイット風広告画面を置くためのものである。ポストイット風広告は、一定時間ごとに右方向から左方向にシフトして、新しい広告が右端から登場する。これによって同一の時間帯に複数のポストイット風広告を閲覧することができる。もうひとつの用途は、PUBiがパブボードに近づいたときにデフォルトのユーザエリアとして利用するためのものである。その位置は乱数で決められ、一定時間アイドル状態になるとユーザエリアは開放されてアイコンは領域(B)に移動する。なお、PUBiが近づいたときにすでにこの領域が他のPUBiのために割り当てられていた場合には、アイコンは最初から領域(B)に表示される。

領域(D):この領域は、パブボードのオーナーの所有である。したがって、その表示内容はオーナーが(Web インターフェースを通じて)設定できる。ユーザが領域(C)に割り当てられるデフォルトの面積を超えて多くの領域を必要とするときには、この領域が一定時間だけ割り当てられる。オーナーの所有領域をユーザが利用するとき、その面積と時間に応じて課金の対象となりうる。この課金はオーナーが行うものである。

領域(E):この領域は、15秒ごとに新しいバナーメッセージと入れ替わる。基本的にはテキスト情報だけを表示する目的で設置されており、Webを通じてメッセージを投稿することができる。領域(C)に表示されるポストイット風広告よりも手軽に作成/投稿できる。比較的短い期間だけ提示したい情報に適している。

Figure.7は、試作されたパブボードの画面を示している。画面下部バナーとポストイット風広告は時間と共に切り替わる。PUBiを持った利用者がボードに近づくと、ポストイット風広告が表示されている列に並んで、あらかじめ利用者が選択していたアイコンが出現する(S1)。接続後、PUBiからボード指定を行うための矢印(この矢印がパブボード特定番号に基づくものである)が、アイコン上に表示される(S2)。接続している利用者のアイコンはサイズを小さく表示させることができる。空いた隙間に(利用者が)思い出したい情報などをスクロール表示させることができる(S3)。アイコンは暫く経つと待合室に移動する(画面上部)(S4)。複数のユーザが同時に利用している状況である(S5)。待合室に4人いる。自分のPDAのようにローカルマップを起動して表示しているユーザがいる様子を示している。

(5) ユーザアプリケーション

ここでは代表的なアプリケーションであると考えられる「キック」と「スナップショット」がどのように記述されているかを説明する。キックの場合、ユーザの入力を受け付けた後は、その入力に対応した画面表示をパブボードに依頼し、サーバに情報を届けるだけで終了する。サーバからの戻り値等はない。サーバからの戻り値を不要とすることでサーバと通信路の負荷を軽減させているが、その反面エラー発生時には対応できない。しかし可能な限り「メッセージパッシング」的に作業することで、コネクションレス型の通信デバイス(例えば赤外線など)の利用が可能なようにデザインするほうが望ましいと考える。Figure.8はパブボード上で動作するプログラムである。

```
sprintf(buf,"exp: %s 3 ",apph.authcode);
```

はパブボードに対するコマンド列を生成している。

```
mito_sendSimpleMessage(&apph,apph.board_queid,buf);
```

は、生成したコマンド列をパブボードに対して発送している。

```
mito_sendServer(&apph, "kick");
```

は、別スレッドを生成してサーバに対してキック操作の行われた時間とユーザ認証コード(いずれも APPH に含まれてい



(S1)



(S2)



(S3)



(S4)



(S5)

Figure 7 パブボードの状態

```

#include "mitolib.h"
int main(int argc, char **argv) {
    APPH apph;
    char buf[MAX_MSG];

    mito_setAppHandle(&apph, argc, argv);

    /* send board "exp: auth_code 3 */
    if (apph.board_queid != -1) {
        sprintf(buf, "exp: %s 3 ", apph.authcode);

        mito_sendSimpleMessage(&apph, apph.board_queid, buf);
        mito_sendServer(&apph, "kick");
    }
    mito_deleteAppHandle(&apph, 0);
    return 0;
}

```

Figure. 8 キックアプリケーション

る)を送している。これにより"kick"で指定されたアプリケーションがサーバ側で起動されることになる。キック情報は、一度DBに格納され、ユーザが指定したタイミングでメールとしてキックしたユーザに届けられることになる。この時刻にこのパブボードではなんの情報を提示していたか、はサーバ側で時刻情報とパブボードID情報から再構成する。

一方、Figure.9はスナップショットサービスの場合のアプリケーションプログラムである。スナップショットサービスの場合、ユーザの入力を受け付けた後は、その入力に対応した画面表示をパブボードに依頼し、写真撮影を行うプログラムを別プロセスとして起動(execl)する。今回の実装ではネットワークに直接接続できるカメラ(RICOH RDC-i700)を利用したが、今後他の機種への対応を考え、撮影後のサーバへの映像収録は別プロセスとして起動するsnap_gコマンドの中で用意している。

(6) サーバアクセスモジュール

サーバアクセスモジュールは、パブボードなどのクライアントアプリケーションがサーバにアクセスする際に利用するライブラリである。クライアントはRPCを利用してサーバへアクセスするが、各クライアントがそれぞれ多くのプロトコルを実装するのは効率的ではない。従ってプロトコル層の部分をカプセル化し、サーバ側のロジック層のAPIをそのままクライアントアプリケーションに見せ、クライアントアプリケーションが容

```

#include "mitolib.h"
int main(int argc, char **argv) {
    APPH apph;
    char buf[MAX_MSG];
    char line[MAX_MSG];

    mito_setAppHandle(&apph, argc, argv);

    /* send board "exp: auth_code 10 */
    if (apph.board_queid != -1) {
        sprintf(buf, "exp: %s 10", apph.authcode);

        mito_sendSimpleMessage(&apph, apph.board_queid, buf);
    }

    strcpy(line, apph.authcode);
    mito_deleteAppHandle(&apph, 0);

    execl("snap_g", "snap", line, NULL);
    return 0;
}

```

Figure. 9 スナップショットアプリケーション

易にサーバへアクセスできるようにする必要がある。本モジュールはそのために作成された。

(7) サーバ

サーバは、顧客情報、パブボード情報などすべての情報を管理している。DBおよび、それらへのクライアントからのアクセス方法を提供するAPIが実装されている。

サーバは、データベース層、ロジック層、プロトコル層、管理アプリケーションに分かれる。

データベースのAPIはデータベース層として規定される。しかしサーバに接続するクライアントは、直接データベースを操作するわけではなく、ロジック層で用意されているAPIを介してデータベースにアクセスする。これにより、データベース操作のコードが各クライアントアプリケーションに分散することなく、ロジックをサーバ側で一括管理することが可能になる。ロジック層では、データベースにアクセスし、必要な情報を取り出したり更新したりするためのロジックが記述されている。プログラムはJava言語で記述されており、データベースへのアクセスはJDBCを用いている。

アプリケーションがサーバ1台で動作する分には、ロジック層のAPIを直接呼び出せばよいので大きな問題は発生しないが、アプリケーションがクライアントとサーバに分散している場合、クライアントアプリケーションがサーバのロジック層のAPIを呼び出すための仕組みが必要になる。この仕組みのことを一般的にRPC(Remote Procedure Call)と呼ぶ、サーバでRPCを実現するための仕組みがプロトコル層である。プロトコル層は、更にプレゼンテーション層、セッション層、トランスポート層、に分かれる。(ただし、OSIの7階層とはあまり関連はない)

クライアントとサーバ間の通信で実際に送受信されるデータの内容を定めたものがプレゼンテーション層である。サーバではSOAP-RPC形式とURLEncode形式の2種類に対応している。

プレゼンテーション層で定められた形式のデータを、どのような手続きで送受信するかを定めたものがセッション層である。サーバでは、HTTPを使用してデータを送受信する。もちろん、その他の通信プロトコル(SMTPなど)を使用してもよい。データは、別にソケットを使用して直接送受信してもよいが、そのような実装をせず、HTTPのような標準的な通信プロトコルを使用して送信する。その理由は、標準的なプロトコルを使用すれば、既存のWebサーバを使用することができるし、様々なライブラリが用意されているため、ソケットを意識しないプログラミングが可能であるからである。また、トランスポート層との分離も容易になり、SSLを使用した通信も容易に実装が可能である。

プレゼンテーション層で扱うデータを、セッション層で使用する通信プロトコルに載せて、どのようなネットワークプロトコルを使用して送受信するかを決めるのがトランスポート層である。サーバ側ではトランスポート層としてTCPを用意する。また、将来的にはUDPおよび、SSLを使用したTCPもサポートしていく予定である。

6. 参加企業及び機関

本プロジェクトは(株)アルゴグラフィの技術協力によって実施された。また、この場を借りて関係スタッフ全員、特に岸、川道、花岡、湯浅、白井、井坂、前川、飯島、岩谷の各氏および竹内PMに感謝の意を表したい。

7. 参考文献

- [1] http://www.zdnet.co.jp/mobile/0201/07/n_keyword.html
- [2] <http://www.carc.aist.go.jp/~yoshinov/2002/projects/pb/snap/top.html>