

異種メディア連携システムの携帯電話端末への実装

清水 章央 太田 慎司 山崎 升一 大橋 正良

{a_shimizu, ota, sh-yamazaki, ohashi}@kddilabs.jp

株式会社 KDDI 研究所

紙媒体と携帯電話端末を連携させ、紙媒体上のユーザ動作から携帯電話端末の情報処理機能を動作させる紙ベースの機能ポータルシステム、異種メディア連携システムに関する検討を進めている。本稿では、異種メディア連携システムを携帯電話端末上へ実装し、評価を行った結果を報告する。

Implementation of Paper based Cross-Media Cooperation System for Cellular Phones

Akihisa Shimizu, Shinji Ota, Shoichi Yamazaki and Masayoshi Ohashi

KDDI R&D Laboratories Inc.

We report an implementation and performance evaluation of Paper-based Cross-media Cooperation system, in which paper materials are used for portals of digital contents. Evaluation results are reported using a mobile terminal with Bluetooth interface

1. はじめに

昨今、紙面上に、URL やメールアドレスを印刷することで、紙媒体の利用者に情報システムやデジタルコンテンツへのアクセスを促す方法が一般的になりつつある。さらに、紙媒体に印刷されたバーコードを携帯電話のカメラで読み取り、バーコードから取得されたデータに対応する情報処理を実行するシステム[1]も実用化されている。

しかし、上記方式では、紙面上への URL やバーコードの印刷が伴い、紙面上に配置できる数が制限されること、印刷される URL やバーコードが紙面上のデザインを損ねてしまう場合があること、また、一旦印刷してしまうと、その後 URL やバーコードの追加・削除ができないことなどの制約がある。

異種メディア連携システム[2]は上記問題を解決することを目的としたシステムである。我々は、

- ・ 紙面上の制約なしに複数の機能を配置することができる。
- ・ 紙媒体のデザインに影響を与えない。

・ 印刷後の紙媒体にも機能の付加ができる。というコンセプトに基づき、これまでパーソナルコンピュータ上に構築した携帯電話エミュレーション環境上での評価を行ってきた。

本稿では、試作システムを Bluetooth を搭載した実携帯電話端末上へ移植し、その特性を評価した結果を報告する。

2. 関連研究

紙媒体と情報システムの連携では、これまでに PaperLink[3]や Audio-Notebook[4]が提案されている。PaperLink では、ペンにカメラを付けて、紙媒体の文字列を取得し、OCR を利用して電子メディアとの連携をとることを特徴とするシステムである。Audio-Notebook では、ボイスノートとバーコード付きのメモパッドを利用してメモとボイスノートを関連付けるシステムである。

また、紙自体にドットパターンを印刷し、その特殊なドットパターンをカメラ付きペンで読み取ることで、紙上のペンの動きを把握する

Anoto[5]がある。Anoto では、ペンの動きを把握するだけでなく、紙上の特定のドットパターンを読み取ることで PC 上の機能を実行することが可能である。

これらのシステムと比較し、本システムは紙面上の任意の領域に、複数の異なる携帯電話端末の情報処理機能 (e.g., 情報の表示, 保存, 転送等) を結び付けられるという、自由度の高いシステムとなっている。

3. 異種メディア連携システム

異種メディア連携システムでは、情報処理機能の実行・制御に、紙媒体上で行われる自然動作 (e.g., 指す, なぞる, 囲む) を利用する。1つのオブジェクトに対応付けられた複数の情報処理機能を区別して動作させる場合には、情報処理機能毎に異なるユーザ動作 (e.g., 丸囲み, 下線を引く, 特定の記号を描く等) を定義する。例えば、特定の単語を丸囲みすることで最新の情報が提示され、下線を引く動作で単語の意味が提示される等を想定できる。

図1に異種メディア連携システムの基本構成を示す。紙媒体提供者は、提供する紙媒体に関して、情報処理機能を埋め込む紙媒体上の座標、情報処理機能を識別するユーザ動作、ならびに実行する情報処理機能の対応を定義した操作データの作成を行う。紙媒体の利用者は、その携帯端末に操作データをダウンロードする。紙媒体と操作データの対応は、紙媒体固有の識別情報 (e.g., 印刷された URL, 添付された無線タグの ID, 本の ISBN 番号とページ数等) を利用する。

紙媒体利用者が紙面上である動作 (ユーザ動作) を行うことにより携帯端末上で特定の情報処理機能が実行される。具体的には、携帯端末に接続された電子ペンから報告される動作の座標から動作の種類とその中心座標が特定される。これらの情報をキーとして操作データから対応する情報処理機能が決定され、実行される。

このように、紙面上で行われるユーザ操作と実行する情報処理機能に対応付けることで、情報処理機能へのポインタ情報を紙面上に印刷する必要がなく、紙媒体へのデザインの影響を最小限に留めることが可能である。また、ユーザ操作と情報処理機能の対応付けを電子ファイル化し、必要に応じてサーバから取得して利用可能とすることで、印刷後の紙媒体に対しても情報処理機能の追加や変更が可能となる。また、ユーザ動作の種類を多様化し、紙面上での実行位置の推定精度を高められれば、紙面上の制約なしに情報処理機能を自由に配置できる。

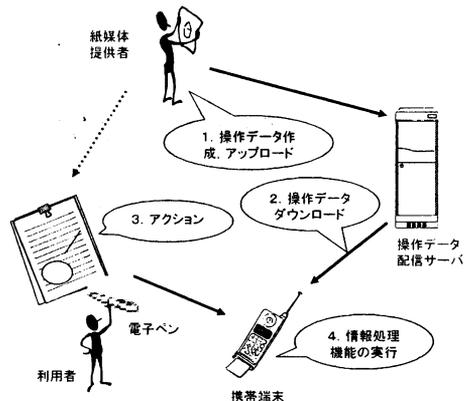


図1：基本システム構成

4. 実装システム

4.1 試作システム構成

図2に試作システムの構成を示す。利用者の携帯端末として、Bluetooth を内蔵した携帯電話試作端末を利用している。紙媒体には無線タグが添付されており、紙媒体の識別情報が埋め込まれている。携帯電話端末は Bluetooth 機能を介して、無線タグリーダと接続可能となるよう構成されており、無線タグリーダへ制御コマンドを送信することで、無線タグからデータを読み取ることが可能である。

無線タグには、その無線タグが添付された紙媒体に関する操作データのポインタ情報 (e.g., URL) が格納されており、このポインタ情報に対応する操作データを配信サーバから取得することが可能となっている。無線タグシステムは、OMRON 社製 V-700[6] シリーズを利用している。V-700 シリーズの無線タグリーダのインタフェースはシリアル通信のため、Bluetooth シリアル Dongle[7] を用いて、携帯電話端末との接続を行う。

操作データを取得後、携帯電話端末と電子ペンの中継する中継用 PC の間で Bluetooth 通信の確立が行われる。ユーザの動作を取得するための、電子ペンは In-Motion 社製 E-Pen[8] を利用している。E-Pen からは毎秒 200 回座標データの点が送信される。E-Pen 受信機のインタフェースが USB であり、現状提供されているドライバが Windows のみであることから、中継用 PC を利用して、携帯電話端末と E-Pen システムの連携を実現している。中継用 PC は E-Pen 受信機から E-Pen の動作の座標を取得し、確立された Bluetooth 通信を利用して、携帯電話端末へ送信する。携帯電話端末は動作の座標から操作データ内の該当する情報処理機能を実行する。

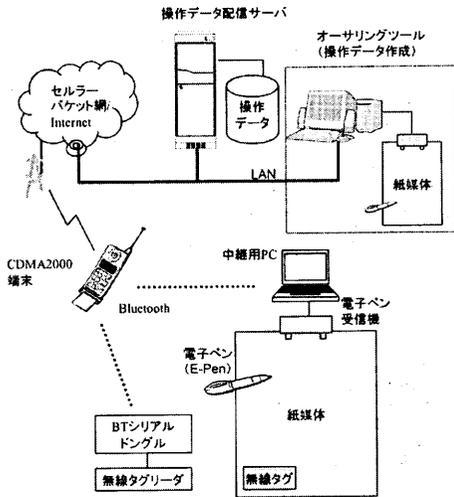
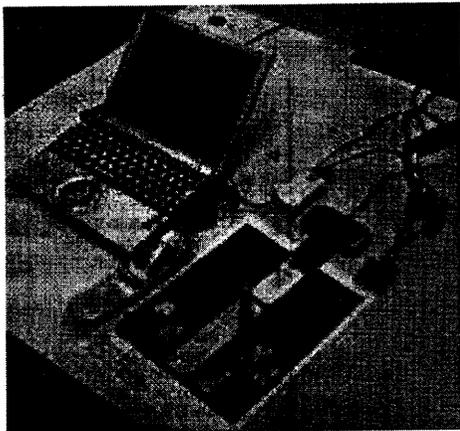


図2：試作システム構成

電子ペン受信機から送信されるデータは、通常 x 座標、y 座標のセットの座標情報が 0.5 ミリ秒間隔で送られてくる。しかしながら、現状の携帯電話端末の実装では毎秒 200 回のデータをリアルタイムで受信するとオーバーヘッドが発生してしまい、所望のパフォーマンスが得られない。このため、電子ペンのペンダウンからペンアップまでを 1 つのデータとして送信するように中継用 PC で処理を行っている。また、1 度のユーザー動作における点数は 200 点程度であり、データ量に換算すると 500byte 程であった。



参考：実装写真

4. 2 試作携帯電話端末ソフトウェア構成

図3に Bluetooth 内蔵携帯電話のソフトウェア構成を示す。

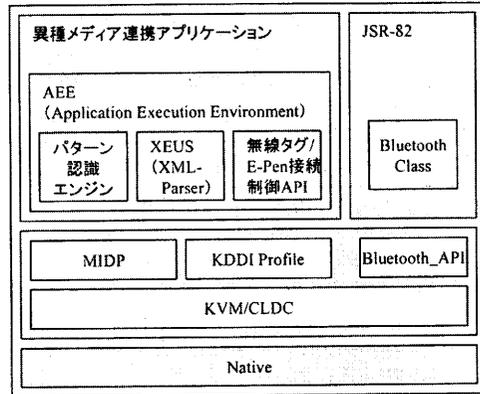


図3：試作機ソフトウェア構成

試作携帯電話端末には、JSR-82[9]に準拠した Bluetooth の API が Java 実行環境上に実装されている。

パターン認識エンジンは、階層型ニューラルネットワークを利用して実現されている。携帯電話端末で利用可能なメモリサイズや処理パフォーマンスを考慮したチューンアップを実施した。この結果、PC エミュレーション環境上では 7 種類 (“○”, “□”, “-”, “~”, “/”, “/”, “?”) 実装していたユーザパターンを、今回の実装では、4 種類 (“○”, “/”, “-”, “?”) に削減して実装している。

XML パーサは操作データのパーサを行うために必要である。パーサとして KDDI 研究所で開発した XEUS[10]を採用した。本パーサは、XML 文書のスキーマを利用した圧縮に加え、エンコード時にパーサ処理も行うため、伝送量と受信側の処理負荷を低減できる。デコード時には、エンコード時に使用したテーブルデータのパーサが初期処理として必要になるが、他の XML パーサを使用した場合のエンコード処理とパーサ処理と比べ格段に速度の向上が見られる。

無線タグ/E-Pen 接続制御 API は、その下位で Bluetooth_API をコールし、無線タグリーダならびに中継用 PC 間でシリアルポートを確立し、それぞれからデータを取得する。

これら、パターン認識エンジン、XML パーサ、無線タグ/E-Pen 接続 I/F はクラスモジュールとして分離しており、異種メディア連携アプリケーションの製作で汎用的に利用することができるよう構成されている。

無線タグ内の格納情報を図4に示す。前述の通り、V-700 シリーズの無線タグシステムのインタフェースは、シリアル通信(9600bps)のため、格納情報を読み取るためにかかる時間は、制御コマ

ンドの通信を含めても 1~2 秒と考えられる。

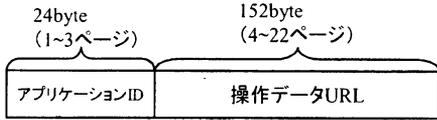


図 4：無線タグ格納情報

操作データの例を図 5 に示す。操作データは XML 形式で記述されており、前述の XEUS を利用して、エンコード、パース処理を行った後配信サーバへのアップロードを行う。今回利用したテスト用の操作データのサイズ（エンコード、パース後）は約 383byte であり、ダウンロードに要する時間は通信確立時間を考慮しなければ、1 秒以内と考えられる。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<mccs>
<field>
<area WIDTH="9000" HEIGHT="13000" NAME="default"
Y="0" X="0" MODE="over">
<area MODE="over" WIDTH="2800" HEIGHT="1500"
NAME="Area1" X="3200" Y="1000">
<circle ALT="ムービー"
VALUE="http://www.kddilabs.jp/tmp.amc"
FUNCTION="video" />
<line ALT="説明"
VALUE="http://www.kddilabs.jp/tmp.txt"
FUNCTION="text" />
<slantline ALT="Webサイトへ"
VALUE="http://www.kddilabs.jp/mccs/index.html"
FUNCTION="urlto" />
<question ALT="ヘルプ" VALUE="/help.txt"
FUNCTION="text" />
</area>
</area>
</field>
</mccs>
```

図 5：操作データ例

5. 評価結果

5. 1 処理時間測定結果

図 6 に異種メディア連携システムの一連の動作のフローを示す。本フローは 4 つのブロックに分かれており、このブロックの単位は、ユーザ操作（e.g., キー入力、E-Pen 入力）の発生のタイミングを表している。

表 1 に、上記フローに基づいた各処理時間を計測した結果を示す。数値は 5 回試験を行った結果の平均である。5 回の結果の中で、処理時間の大幅なばらつきは見られなかった。

今回の実装では、可能な限りユーザ操作時の応答性を高めることを主眼とした。このため、初期化処理が大きくなっているが、ユーザ動作読み取

り処理に時間がかかるものの、体感で気にならない応答性が実現できていると思われる。

結果より、Bluetooth の初期化（getLocalDevice()実行）、通信路確立（無線タグリーダーへの接続、E-Pen への接続）、データ取得処理（無線タグ情報の取得、E-Pen 情報の取得）での処理時間が大きいことがわかる。また、XEUS 関連の処理時間（XEUS テーブルデータのパース時間、操作データのパース処理）も大きな割合を占めている。

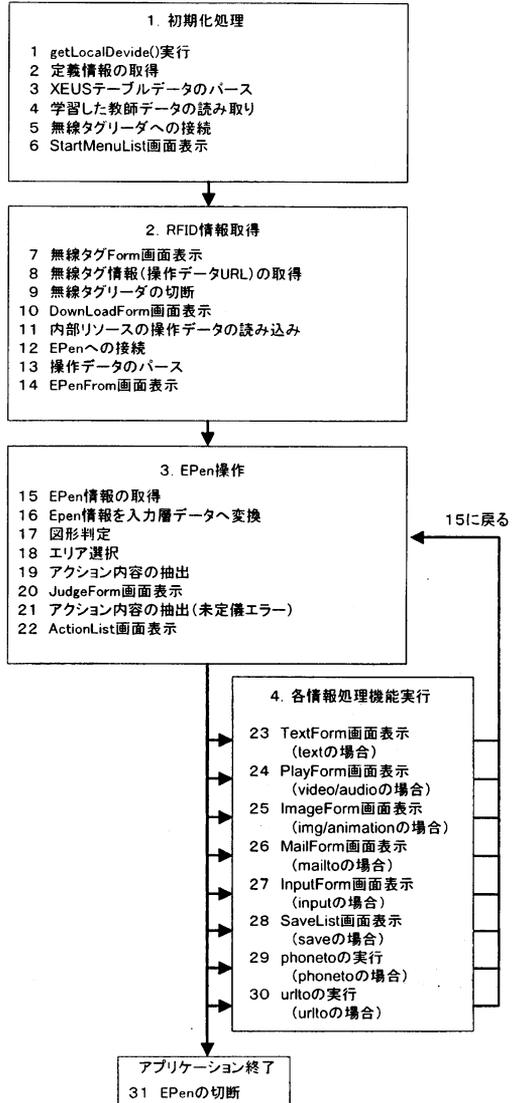


図 6：処理フロー

なお、4 ブロック目の phoneto に関しては、今回の試作のベースとなった携帯電話端末に搭載済みの機能を流用したものであり、今回の試作の範囲外の機能であったため、議論の対象とはしない。

5. 2, 5. 3 節では、これらの処理時間の原因と改善の可能性に関して、評価を行う。

表 1 : 処理時間

測定項目	時間(ms)	
	平均	割合
getLocalDevice()実行	7206.8	10.89%
定義情報の取得	337.4	0.51%
XEUSテーブルデータのバース	7648.2	11.56%
学習した教師データの読み込み	2449.8	3.70%
無線タグリーダへの接続	8991.0	13.59%
StartMenuList画面表示	7876.4	11.90%
無線タグForm画面表示	602.4	0.91%
無線タグ情報(操作データURL)の取得	2840.2	4.29%
無線タグリーダの切断	684.2	1.03%
DownloadForm画面表示	1031.4	1.56%
内部リソースの操作データの読み込み	1039.0	1.57%
E-Penへの接続	3406.6	5.15%
操作データのバース	2949.0	4.46%
E-PenForm画面表示	1783.2	2.69%
E-Pen情報の取得	2009.6	3.04%
E-Pen情報を入力層データへ変換	397.0	0.60%
図形判定	383.2	0.58%
エリア選択	84.8	0.13%
アクション内容の抽出	141.4	0.21%
JudgeForm画面表示	926.8	1.40%
アクション内容の抽出(未定義エラー)	121.0	0.18%
ActionList画面表示	981.4	1.48%
TextForm画面表示(textの場合)	280.6	0.42%
PlayForm画面表示(video/audioの場合)	1541.8	2.33%
ImageForm画面表示(img./animationの場合)	680.8	1.03%
MailForm画面表示(mailtoの場合)	628.4	0.95%
InputForm画面表示(formの場合)	1111.6	1.68%
SaveList画面表示(saveの場合)	614.0	0.93%
phonetoの実行	4287.4	6.48%
urltoの実行	2303.8	3.48%
E-Penの切断	839.8	1.27%
合計処理時間	66179.0	

5. 2 Bluetooth 関連処理の評価

ここでは、Bluetooth 関連処理の評価結果を示す。

初期化処理 (1 ブロック目) では、getLocalDevice()実行、無線タグリーダへの接続が合計 16 秒程度と大きな割合を占めている。また、2 ブロック目以降でも E-Pen への接続や無線タグ情報の取得、E-Pen 情報の取得で、それぞれ 2~3 秒の時間が費やされていることがわかる。

まず、携帯電話端末からの Bluetooth の接続時間について、携帯電話端末の Native 層上と Java 層上で測定を行った。2つの異なる Bluetooth 機器 A,B に対し、携帯電話端末から A→B→A→A→B の順に接続、切断するシーケンス¹を 4 回実施し平均を求めた。実験構成を図 7 に、A,B の機器の詳細を表 2 に示す。

¹同 - 機器への再接続と異なる機器への接続切り替え時には、初期化処理等が異なり得るのでこのような試験を行った。

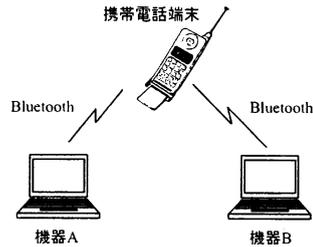


図 7 : 試験構成

表 2 : 試験機器 1

	A	B
Bluetoothデバイス	3comUSBアダプタ	ハギワラシスコムCFアダプタ
Bluetoothスタック	BlueZ	BlueZ
OS	PC-Linux	PC-Linux

試験結果を表 3 に示す。A,B 機器への 4 シーケンスの接続時間の平均が 1~5 である。最下段の平均は初回の接続を除く、2~5 への平均である。

表 3 : 接続時間 1

(4 シーケンスの試行平均値)

	Native	Java
1 A	1.580	6.281
2 B	2.211	2.981
3 A	1.979	1.952
4 A	1.451	2.583
5 B	1.647	1.924
2~5の平均	1.774	2.360

本結果から、Java の層が接続時に影響を与えていることが分かる。オーバーヘッドとしては 0.6 秒程度であることがわかった。その一方、Native 処理だけでも平均 1.7 秒の接続時間がかかっており、その高速化が望まれる。

次に試験機器 A,B の OS を Windows に変更し、同様の測定を行った。機器構成は表 4 の通りである。結果を表 5 に示す。

表 4 : 試験環境 2

	A	B
Bluetoothデバイス	3comUSBアダプタ	東芝BluetoothPCカード
Bluetoothスタック	3com製	東芝製
OS	Windows2000	Windows2000

表 5 : 接続時間 2

(4 シーケンスの試行平均値)

	Java
1 A	8.279
2 B	2.815
3 A	3.235
4 A	2.752
5 B	3.756
2~5の平均	3.140

表3の結果や表5の結果より、対向するBluetoothモジュールや機器の違いにより、通信の確立時の性能差が発生することが観測された。

また、これらの評価において、今回のJava試作携帯電話端末から、初めてBluetooth通信路を確立する際(e.g.,表3や表5の初回A)、JSR-82のBluetooth関連のクラスのロードにより、平均4秒程度のオーバーヘッドが追加されることを確認した。

以上の結果により、試作携帯電話端末のJavaプラットフォームには、NativeのC言語ベースのプラットフォームと比較して、(a)クラスロードに関するオーバーヘッドと、(b)通信確立時のオーバーヘッドがあることがわかった。また、対向するBluetooth機器によっては、確立処理時間が劣化する場合もあることが判明した。

これらの結果は、Bluetooth関連処理の向上に、携帯電話端末性能(e.g., CPUクロックの高速化、メモリ読み出し速度の高速化)やBluetoothモジュールあるいは対向機器の性能向上(e.g., 相互接続性能の向上等)が必要であることを示唆していると考えられる。

5. 3 XMLパーシング関連処理の評価

今回の実装では、ブロック1のXEUSテーブルデータのパーズ時間も大きな割合を占めている。また、ブロック2の操作データのパーズ処理に関しても3秒程の時間がかかっている。

XMLのパーズ処理に関しては、前回の試作時に使用したXMLパーサ(NanoXML[11])に比較した結果、5.5秒程度の処理時間の短縮を確認している。

また、XEUSをC言語ベースのプラットフォーム上(BREW[12])に移植し動作させた場合、同テーブルデータのパーズが20ms程度で完了することを確認している。従って、今回の試作における処理時間はJavaプラットフォーム固有のオーバーヘッドであると考えられる。

5. 4 認識率評価

前述の通り、携帯電話端末へ実装を行うため、ニューラルネットワークの構成を変更した。変更後の認識率を各パターン50回ずつE-Penから入力し、調査した。測定結果を表6に示す。

表6：認識率

ユーザパターン	被験者A	被験者B
○	98.0%	98.0%
/	84.0%	88.0%
—	100.0%	100.0%
?	72.0%	86.0%
平均	88.5%	93.0%

PC上での認識率として、82.86%という結果を既に報告[1]しているが、これと同等あるいは上回る結果となった。これは、今回処理を軽くするために行った認識パターン数の削減において、誤認識の起こりやすい、“□”、“△”、“～”のパターンを実装しなかったためと考えられる。

5. 5 モジュールサイズ

実装時のモジュールサイズを表7に示す。

表7：モジュールサイズ

モジュール区別	サイズ(byte)
AEE部	16,414
XEUS部	11,620
アプリ部	42,189
内部リソース部	21,252
合計	91,475

異種メディア連携システムのモジュールサイズは合計で約91KBとなった。昨今の携帯電話端末では、100KB~200KBのアプリケーションが動作可能であり、このモジュールサイズは携帯電話端末で動作可能なサイズである。また、異種メディア連携システムの共通プラットフォームとして利用できる、AEE(Application Execution Environment:図3参照)部分が27KB程度(AEE内部にXEUS部が11KB)、これらを共通化できるプラットフォーム上ではさらに小さなサイズでの実装も行うことが可能となる。

6. まとめ

異種メディア連携システムの携帯電話端末への実装に関して、その評価結果を報告した。評価結果から、認識率、モジュールサイズ、ならびに応答性に関しては、現状の携帯電話端末でも動作可能な所望の性能が得られたと考えられる。ただし、Bluetooth通信関連処理やXMLパーシング処理に関する処理時間の評価では、Javaプラットフォームに起因するオーバーヘッドが挙げられ、改善の余地があることがわかった。今後は、本システムをBREW環境へ実装し、ポトルネットワークの改善確認を行う予定である。

謝辞 日頃ご指導いただき、KDDI研究所 浅見所長、和田執行役員に深謝する。

参考文献

- [1]例:KDDI株式会社, "au携帯電話における2次元コード読み込みをEZアプリ(BREW(TM))で実現", http://www.kddi.com/corporate/news_release/2003/1215.idx.html
- [2]清水章央, 太田慎司, 山崎昇一, 大橋正良, "異種メディア連携システムの提案と評価", DICOMO2003, 阿寒, 2003年6月。
- [3]Arai, T., Aust, D., and Hudson, S. "PaperLink A

- Technique for Hyperlinking from Real Paper to Electronic Content", CHI'97 Proceedings, 1997.
- [4]Stifelman, L. J."Augmenting Real-World Objects: A Paper-Based Audio Notebook", CHI'96 Companion, pp 199-200,1996.
- [5]Anoto AB. "Anoto". <http://www.anoto.com>.
- [6]OMRON 社. "OMRON 社製 V-700 シリーズ".
<http://www.omron.co.jp/rfid/>
- [7]キャノンアイテック株式会社. "Bluetooth シリアルドングル".<http://www.citech.co.jp/products/ub-6010.html>
- [8]E-Pen Inc.. "E-Pen". <http://www.e-pen.com/>.
- [9]JavaCommunityProcess.. "JSR-82".
<http://java.sun.com/j2me/>.
- [10]小林亜令 松本一則 井ノ上直己. "汎用 XML 文書符号化方式「XEUS」の性能評価", FIT2003.
- [11] cyberelf. "NanoXML":
<http://web.wanadoo.be/cyberelf/nanoxml/>
- [12]Qualcomm Inc..
"BREW".<http://www.qualcomm.com/brew/>.