平成18年度

筑波大学第三学群情報学類

卒業研究論文

題目 モバイルデバイスを介して利用する大画面情報アプライアンスの提案と試作

主専攻　　知能情報メディア主専攻

著者　　金 衍廷

指導教員　田中 二郎，高橋 伸，三末 和男，志築 文太郎

要　　旨

ユビキタス時代を開くキーとなる技術には、ネットワーク、センシング、コンテキストアウェアネス、セキュリティーなどがある。以上に挙げた技術が網羅された情報アプライアンスが生活環境のどこにでも偏在するようになったとき、我々はユビキタス時代に生きていると言えるのである。

本研究では、ネットワーク上の個人情報管理エージェントであるリソースマネージャを導入し、モバイルデバイスを用いて情報アプライアンスを利用するインタフェースを提案する。そして、提案したインタフェースに基づいて試作した大画面情報アプライアンスを紹介する。

目 次

第 1 章　序論 **1**
1.1　はじめに 1
1.2　本研究の目的 1
1.3　本論文の構成 1

第 2 章　背景と関連研究 **2**
2.1　ユビキタスコンピューティング 2
2.1.1　情報アプライアンス 3
2.1.2　グリッドコンピューティング 3
2.1.3　柔らかいストレージ 3
2.2　大画面 3
2.2.1　大画面の有用性の検討 4
2.2.2　大画面とモバイルデバイスとの連携 4

第 3 章　モバイルデバイスを介した情報アプライアンスの提案 **5**
3.1　ユビキタス時代における情報アプライアンスのあるべき姿 5
3.1.1　情報アプライアンスの特徴 5
3.1.2　情報アプライアンスの拡張 6
3.1.3　情報アプライアンスに対するユーザの状態 6
3.2　情報アプライアンスを利用するインタフェースの提案 7
3.2.1　リソースマネージャの導入 7
3.2.2　モバイルデバイスを介するインタフェース 8
3.2.3　大画面情報アプライアンス 9

第 4 章　大画面情報アプライアンスの試作 **11**
4.1　システムの構成 11
4.1.1　Large Screen 12
4.1.1.1　personal workspace 13
4.1.1.2　public application 14
4.1.1.3　public workspace 14
4.1.2　Resource Manager 15
4.1.3　Mobile Device 17

4.2 システムの機能 19
4.2.1 電子データの閲覧 19
4.2.2 電子データの共有 20
4.2.3 大画面が提供するサービスの利用 21

第 5 章 まとめと今後の展望 22
5.1 まとめ 22
5.2 今後の展望 22
5.2.1 モバイルデバイスの改善 22
5.2.2 リソースマネージャの機能拡張 22

謝辞 23

参考文献 24

図目次

2.1 ユビキタスコンピューティング 2

3.1 情報アプライアンスに対するユーザの状態 6
3.2 情報アプライアンス、リソースマネージャ、モバイルデバイスによる構成 . . 8
3.3 大画面情報アプライアンスのイメージ図 9

4.1 システムの概念図 11
4.2 Large Screen のスクリーンショット 12
4.3 ユーザ認証前後の personal workspace 13
4.4 personal workspace の表示モード 14
4.5 public application 14
4.6 Large Screen と Resource Manager 間の通信におけるプロトコルの流れ 15
4.7 インタフェース ResourceManager 16
4.8 rm.conf の例 16
4.9 Make RM Configuration の実行例 17
4.10 Mobile Device 17
4.11 システム実行例 18
4.12 閲覧できる電子データの形式 19
4.13 電子データの共有 20
4.14 public application の例：電車の時刻表を表示するアプリケーション 21

第1章　序論

1.1　はじめに

近年、コンピュータは一人一台の時代を超え、一人数台の時代になった。PDA, Smartphone などのモバイルデバイスを含むと、人間はコンピュータに囲まれた環境に生活しているように見える。さらに、ほぼすべてのコンピュータおよびモバイルデバイスは高速ネットワークで結ばれるようになり、各種のウェブサービスが利用でき、モバイルデバイス同士に情報交換を行うこともできる。

一方で、プロジェクタやプラズマディスプレイを利用した大画面がオフィスや研究室など様々な場所に普及してきている。プレゼンテーションのように一方的な情報提示の目的ではなく、複数のユーザが能動的に利用したい要望がある。

1.2　本研究の目的

本研究では、ユビキタス時代において情報アプライアンスがどのようにしてユーザを支援するかについて議論し、モバイルデバイスを介して情報アプライアンスを利用するインタフェースを提案する。また、情報アプライアンスがユーザの情報にアクセスするために、ユーザの情報を管理するリソースマネージャを導入したインタフェースを提案する。

そして、提案したインタフェースの概念を取り入れた情報アプライアンスの例として大画面情報アプライアンス試作する。大画面アプライアンスは複数のユーザが同時に利用して、それぞれのネットワークストレージにある電子データを自然に閲覧でき、自由に共有できることを目的とする。

1.3　本論文の構成

本論文の構成は、序論である本章を含めて５章で構成される。２章では、本研究に関する概念や背景、関連研究を説明する。３章では、リソースマネージャを導入し、モバイルデバイスを介して情報アプライアンスを利用するインタフェースを提案する。４章では３章で提案したインタフェースで試作した大画面情報アプライアンスについて説明する。最後に、５章で本論文をまとめる。

第2章　背景と関連研究

本章では、本研究の背景となる概念と関連研究について説明する。

2.1　ユビキタスコンピューティング

ユビキタスコンピューティング [1] とは、M. Weiser によって 1991 年に提唱された概念である。それによると、ユビキタスコンピューティングのビジョンは、コンピュータが「環境にすっかり溶け込み消えてしまう」ことである。すなわち、ユビキタス環境では多数のコンピュータが環境の中に溶け込んでいて、ユーザが必要とする場合のみ機能を提供する。ユーザにコンピュータを使っているということを意識させないこと、つまり、コンピュータが環境になり生活の一部になるようにすることが目的である。

環境の一部になった多数のコンピュータが連合してユーザに適したサービスを提供することが望ましい。そのためには、各コンピュータがセンサを持ち、ネットワークでつながることで、そのユーザの特有の情報を入手する必要がある。例えば、ユーザにより、又は同ユーザでもそのコンテキストにより、異なるサービスを提供するかも知れないからである。

図 2.1: ユビキタスコンピューティング

2.1.1　情報アプライアンス

D. A. Norman によると [2]、現在のコンピュータが汎用的であることに対して、未来のコンピュータは一つのタスクの処理に特化された**情報アプライアンス**と呼ばれる道具になる。

アプライアンス（appliance）というのは一般に冷蔵庫や洗濯機などの“ 白もの家電 ”のことを言い、テレビやオーディオなどの“ 黒もの家電 ”は含めない。単機能に特化したアプライアンスは、センサとネットワークを有するコンピュータを内蔵することでより強力な機能を発揮できる。

2.1.2　グリッドコンピューティング

グリッドコンピューティングとは、ネットワークを介して複数のコンピュータを結ぶことで仮想的に高性能コンピュータを造るための方法論である。すなわち、分散されているコンピュータを繋げて各々のコンピュータの資源を共有することを目指している。

グリッドイントラストラクチャの中にモバイルデバイスを取り入れた GridLite[7] というフレームワークがある。資源に制約のあるモバイルデバイスにおいて、適時必要と判断されるデータを大量の電子データを有するサーバから自動的に転送することで、ストレージ容量が無限であるかのように見せつけるシステムである。モバイルデバイスをより強力にすることを目的としていて、モバイルデバイスを介して他の情報アプライアンスでストレージを利用することは想定していない。GridLite はユビキタス環境におけるネットワークストレージの利便性と、それを管理するマネージャエージェントの役割に関する重要な手がかりになる。

2.1.3　柔らかいストレージ

柔らかいストレージ [8] とは、“ 体 ”をネットワークストレージのキーとして利用し、さらにストレージにアクセスするためのインタフェースにも体を用いる手法である。この手法によってユーザは、自分の体内に自分専用のストレージがあり、データを自分の体に記録する錯覚を持つ。ユーザは NS ミラー [8] の前に立つだけで、鏡に見立てた画面に移った体から飛び出る写真データを閲覧できる。しかし、NS ミラーがユーザを識別するために、予めユーザ登録をして置くように想定しているので、どこでも利用できるといったネットワークストレージの長所を妨げるようになってしまった。

2.2　大画面

近年、プロジェクタやプラズマディスプレイなどによる大画面が、研究室や会社のミーティングルームなどで大画面が普及している。大画面とは物理的なサイズが大きいディスプレイを言い、より多くの情報をより大きく見せるのを目的である。最近では、ただ大きいだけではなくサイズに伴ってピクセル数も多い大画面が続々現れている。また、複数の画面を組み合わせることにより大画面を構成する場合も多々ある。

2.2.1　大画面の有用性の検討

現在、大画面が利用される場面を考えてみると、制限的な用途でしか使われてないことがわかる。例えば、ノートパソコンの画面をそのまま大きく表示させて多数の人に大きく見せるといったプレゼンテーションの目的で使われることが多い。また、街中や駅のホームなどの人が集まる場において、広告のために使われることもある。

いずれの場合も、大画面は単に複数の人に見せるためのものでしかない。すなわち、ある特定のユーザのみが操作を行い、複数の人は操作をする権限を持たずに一方的に見させられるだけである。

大画面を利用してユーザ同士に電子データを共有するためのシステムとして、Dynamo[9][10] というものがある。しかし、Dynamo を利用するためには、ユーザがノートパソコンか USB フラッシュメモリなどの物理ストレージを持参する必要がある。閲覧したい電子データを物理ストレージに持ってなければどうすることもできない。

2.2.2　大画面とモバイルデバイスとの連携

モバイルデバイスとは、容易に持ち運べるようにデザインされたハードウェアを呼ぶ。たとえば、携帯電話や PDA、タブレット PC、ノート PC などが含まれる。現在もほとんどの人が携帯電話をいつも持ち歩いているが、技術の進歩に伴い、携帯電話は Smartphone と呼ばれる PDA 並みの性能を持つように進化しつつある。携帯電話サイズでノート PC 並みの機能性を持つモバイルデバイスを、誰もが持ち歩くような近未来の一場面が簡単に想像できる。

既存の携帯端末の LCD を用いて画面操作できる C-Blink[11] という技術がある。ディスプレイの色点滅によりデータ信号を送信し、カメラ側おいて色相差から信号位置を検出しデータを受信する。つまり、ユーザの ID と位置の検出を同時に行うことで、複数のユーザが同時に画面を操作することも可能である。

カメラ付きの携帯電話を用いて大画面を操作する技術に Sweep[12] と呼ばれるものもある。カメラ画像でからオプティカルフローを検出し、携帯電話をオプティカルマウスのように使うことができる。

モバイルデバイスの中にある電子データを大画面で閲覧し、他ユーザのモバイルデバイスに送信するするシステムがある [13]。また、モバイルデバイスのタッチパネルで起こるマウスイベントをソケット通信で大画面側に送り、画面操作も行う。しかし、Dynamo と同様に電子データを物理ストレージで持参するようにして、ネットワークストレージに目を向けなかった。

第3章　モバイルデバイスを介した情報アプライアンスの提案

本章では、ユビキタス時代において情報アプライアンスがどのようにしてユーザを支援するかについて議論し、モバイルデバイスを介して情報アプライアンスを利用するインタフェースを提案する。

3.1　ユビキタス時代における情報アプライアンスのあるべき姿

3.1.1　情報アプライアンスの特徴

ユビキタス時代における情報アプライアンスの特徴として以下のようなものが言われている。

- 到来するユビキタス時代には、至るところに情報アプライアンスが存在する。しかも、その多くは人の目に見えないところに隠れていて、例えば、壁の中や床の下、椅子の中にあるかも知れない。とは言え、すべての情報アプライアンスが目に見えないとは限らない。実際、現在の情報アプライアンスと呼べるもののほとんどは目に見えるところにある場合が多い。より簡単にいうと、身の周りにあるすべてのものが情報アプライアンスとなり得る。

- 情報アプライアンスはユーザが必要とするときのみにその機能を提供する。逆に、必要とされないときには何にもしないことになる。ならば、情報アプライアンスはどのようにして起動し機能するのであろうか。一つは、情報アプライアンスが何らかのセンサを持ち、そのセンサからの情報から状況判断をして、適切な機能をすることである。このためには**コンテキストアウェアネス**技術が要求される。もう一つは、他の情報アプライアンスから知らされることである。このためには**ネットワーク**に繋がっている必要がある。実際、ユビキタスコンピューティングにおいて、ネットワークとコンテキストアウェアネス技術の進歩が重要視される。

- ユーザが積極的に情報アプライアンスを利用しようとしなくても、情報アプライアンスの判断で自動的にサービスを提供する。ところが、このような情報アプライアンスの自律性により、ユーザにとって都合の悪いことが起こる可能性がある。したがって、情報アプライアンスの判断よりユーザの命令を優先することで、ユーザに拒否権を与えることを前提にしなければならない。

- 同じ情報アプライアンスでも対象となるユーザによって提供すべきサービスは異なるかも知れない。すなわち、情報アプライアンスはユーザを識別しなければならない。さらに、ユーザに適したサービスを提供するために、識別したユーザの情報を取得する必要がある。

情報アプライアンスが以上に挙げた特徴をすべて持つ必要はない。言葉の通り特徴を挙げただけであって、情報アプライアンスとなるための条件などではないからである。しかし、以上の特徴を理解するのが情報アプライアンスを理解するのに役立つことには違いない。

3.1.2 情報アプライアンスの拡張

情報アプライアンスは単機能に特化したコンピュータのことを言うが、コンピュータとは言わない**モノ**（thing）にセンサとネットワークを有するコンピュータをもたらすことで、情報アプライアンスへ拡張することが考えられる。

例えば、自動ドアを情報アプライアンスへと拡張することができる。普段は閉まっている自動ドアの前に、中に入ろうとする人が現れると、その人の顔を認識する。そして、出入りが許可されているユーザの顔リストとパターンマッチングを行い、マッチされるならドアを開ける。これは RFID タグの入ったカードを用いる出入り管理システムとあまり変わらないように見えるが、自動ドアアプライアンスの方がより自然なインタフェースであり、ユーザにコンピュータを使っていることを意識させないといったインビジブルコンピューティングの理念に近いと言える。

3.1.3 情報アプライアンスに対するユーザの状態

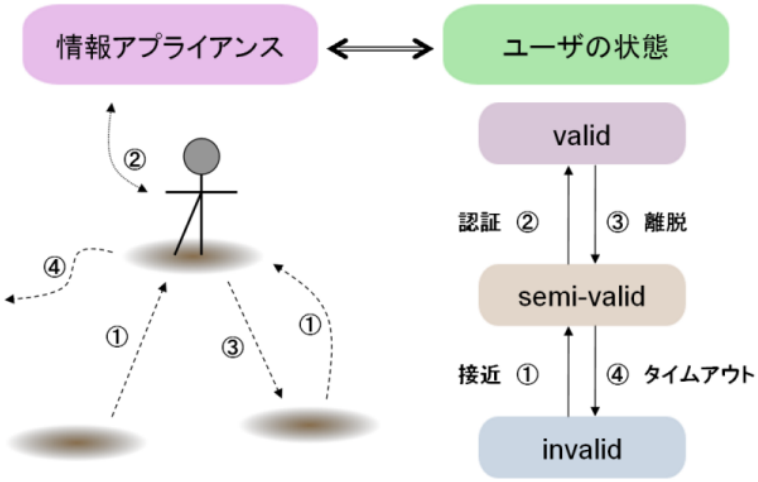


図 3.1: 情報アプライアンスに対するユーザの状態

情報アプライアンスに対するユーザの状態を **invalid**、**semi-valid**、**valid** の３つに分けることができる（図 3.1）。

まず、情報アプライアンスがユーザを認識してなく、当然にユーザは情報アプライアンスを利用することができない。この状態を invalid と呼ぶことにする。次に、ユーザが近づいて情報アプライアンスはユーザを認識すると、ユーザに対して基本的なサービスを提供することができ、 すべてのユーザは同じようなサービスを受ける。この状態では、この状態を semi-valid と呼ぶことにする。最後に、情報アプライアンスにネットワークを介してユーザの個人情報を取得できる許可を与えると、情報アプライアンスはそのユーザに適したサービスを提供することができる。この状態を valid と呼ぶことにする。

valid から semi-valid に・semi-valid から invalid に状態遷移が起こる場合としては、ユーザが意図的に情報アプライアンスの利用を止めた場合と、情報アプライアンスがユーザを認識できなくなった場合の２つが考えられる。

3.2　情報アプライアンスを利用するインタフェースの提案

情報アプライアンスの利用において、リソースマネージャ、モバイルデバイスを導入したインタフェースを提案する。

3.2.1　リソースマネージャの導入

情報アプライアンスがユーザごとに最適のサービスを提供するために、ユーザの個人情報にアクセスする必要がある。例えば、乗るだけで自動的に行き先の階まで運んでくれるエレベータ情報アプライアンスがあるとしたら、乗り込んだユーザのスケジュール情報にアクセスして、“ この時間にこの建物だと何階に行きたいんだな ”というふうに判断することが可能となる。

しかし、セキュリティーに注意が要されるので、どの情報アプライアンスにどのようなユーザの情報を提供していいかの判断をしなければならない。その判断を下すのがユーザであっては、むしろエレベータのボタンを押すほうが簡単である。したがって、ユーザに代わって判断を行うマネージャを導入する。そのようなマネージャを**リソースマネージャ**と呼ぶことにする。リソースマネージャはユーザの個人情報や電子データなどを管理していて、情報アプライアンスからの要請に応じて、ユーザの情報や電子データなどを返す役割を果たす。

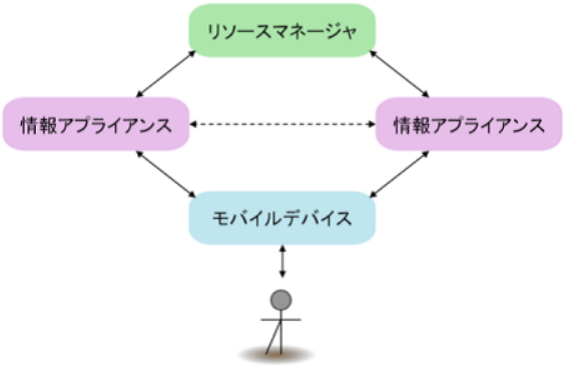
リソースマネージャはユーザに代わってユーザの様々な情報を情報アプライアンスに渡す。リソースマネージャが管理するユーザの情報には **unlocked information** と **locked information** ２種類ある。unlocked information は情報アプライアンスの要請があれば、情報アプライアンスに対して無条件に提供する。すなわち、semi-valid 状態の情報アプライアンスもアクセス可能な情報であるので、セキュリティーやプライバシーに触るような情報は該当しない。locked information は認証された情報アプライアンスに対してのみに提供する。すなわち、valid 状態の情報アプライアンスのみがアクセス可能な情報である。

ユビキタスコンピューティングにおけるユーザインタフェースの究極の目的は、ユーザにコンピュータやネットワークを使っているということを意識させないことである。したがって、ユーザはリソースマネージャを利用すると認識せず、頭脳の中から記憶を取り出すのと同じような、自然にストレージにアクセスすることが望ましい。

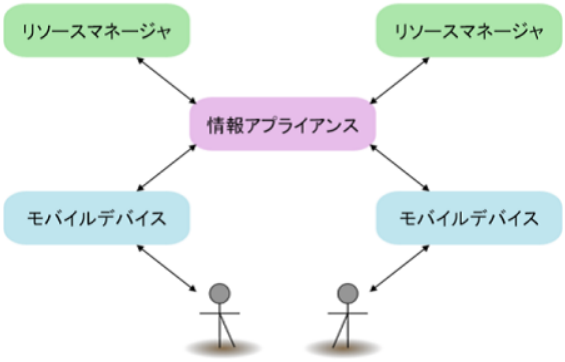
3.2.2　モバイルデバイスを介するインタフェース

ユビキタス時代を向かえ情報アプライアンスが生活周辺に溢れるようになっても、現在のパソコンやサーバ、そしてモバイルデバイスがなくなる。逆に、現在のコンピュータやモバイルデバイスが情報アプライアンスと連携して、我々の生活に役立てるようになると考えられる。

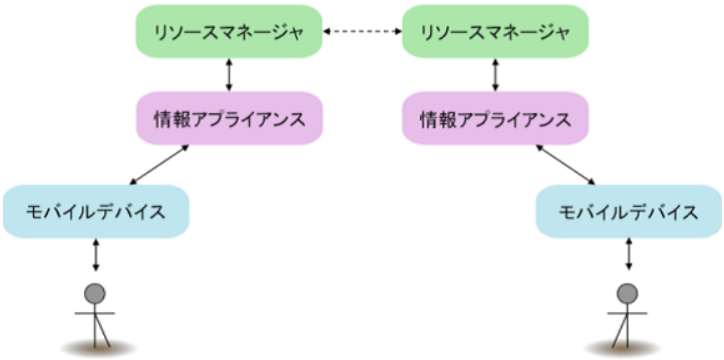
本研究では、モバイルデバイスを媒介とするインタフェースを提案する。



3.2(a)



3.2(b)



3.2(c)

図 3.2: 情報アプライアンス、リソースマネージャ、モバイルデバイスによる構成

まず、モバイルデバイスはユーザと情報アプライアンスの媒体として働く。情報アプライアンスはモバイルデバイスを通じてユーザを認識する。すなわち、モバイルデバイスを検出

したら、モバイルデバイスからユーザ名を受け取る。ユーザ名はユーザが複数いる場合、それぞれを識別するために必要である。これで、情報アプライアンスはユーザの存在を知ることになる。さらに、ユーザが情報アプライアンスを能動的に操作したい場合も、モバイルデバイスを用いて操作することができる。

次に、モバイルデバイスは情報アプライアンスとリソースマネージャの媒体として働く。情報アプライアンスはモバイルデバイスからリソースマネージャの IP アドレスを受け取り、リソースマネージャからユーザの情報を得ることもできる。

情報アプライアンスの利用において、リソースマネージャ、モバイルデバイスを導入したインタフェースを図 3.2 で示す。図 3.2(a) は 1 人のユーザが 2 つの情報アプライアンスを利用する場合の構成を表す。情報アプライアンス同士が互いにユーザの情報を共有できるようにすると、ユーザに適したサービスを効率よく提供することができる。図 3.2(b) は 2 人のユーザが同じ情報アプライアンスを利用する場合の構成を表す。同じ情報アプライアンスを同時に複数人が利用しても、それぞれの対して適切なサービスを提供しなければならない。ユーザを混同することがあってはならない。図 3.2(c) は 2 人のユーザがそれぞれ情報アプライアンスを利用する場合の構成を表す。リソースマネージャ同士が情報交換を行うことも考えられる。

3.2.3　大画面情報アプライアンス

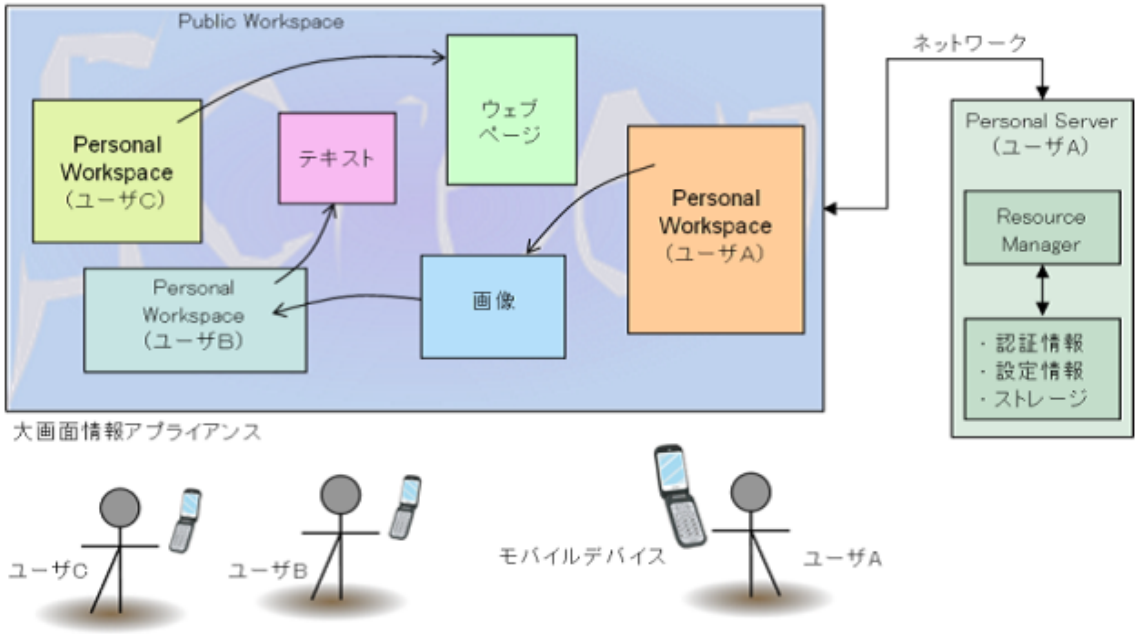


図 3.3: 大画面情報アプライアンスのイメージ図

本研究では、大画面を情報アプライアンスへ拡張する。大画面（large screen）はコンピュータからの外部出力により実現されるが、**画面（screen）**そのものはプロジェクタやプラズマディスプレイなどにより投影される**２次元平面のモノ**に過ぎない。このように考えた**モノ**を情報アプライアンスへと拡張したものを**大画面情報アプライアンス**と呼ぶことにする。大画面アプライアンスは複数ユーザが同時に利用して、電子データの閲覧・共有を行うことができる。

大画面情報アプライアンス情報アプライアンスの特徴とユーザの３状態を考慮し、リソースマネージャとモバイルデバイスを用いるインタフェースにする。すなわち、モバイルデバイスを用いて大画面情報アプライアンスを操作し、リソースマネージャの管理しているストレージを自然に利用できるようにする。

大画面情報アプライアンスの詳細については次の章で述べる。

第4章　大画面情報アプライアンスの試作

本章では、前章で述べたユビキタス環境における情報アプライアンスとして試作した大画面情報アプライアンスの実装について説明する。

4.1　システムの構成

大画面情報アプライアンスの実装は、大画面の本体である **Large Screen** 以外に、大画面が前章で述べたようなユビキタス環境における情報アプライアンスとして成り立つための **Resource Manager**, **Mobile Device** の３つのパートで構成される。

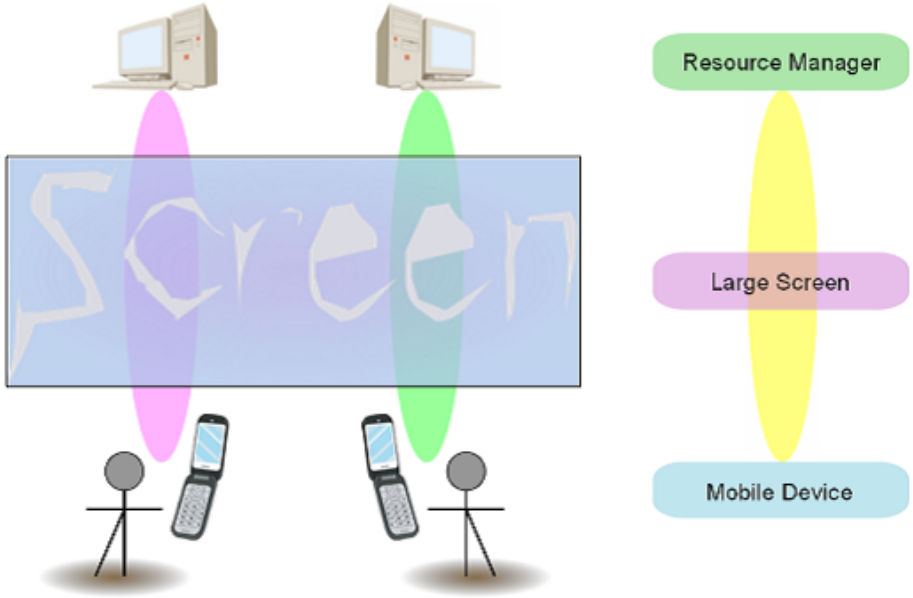


図 4.1: システムの概念図

図 4.1 で示すように、ユーザはモバイルデバイスを用いて大画面情報アプライアンスを操作して、Resource Manager から電子データを取り出し閲覧･共有ができる。Mobile Device を持つユーザが Large Screen に接近すると、制限的に Large Screen が利用できる（semi-valid）。Resource Manager に対してユーザ認証を行うと、Large Screen がフールに利用可能になり（valid）。

ユーザはモバイルデバイスを持っているが、大画面情報アプライアンスにて閲覧・共有する電子データは、モバイルデバイスや他の物理ストレージの中などに持っていない。ユーザは

いつどこにいても、Resource Manager を通してネットワークストレージの中に在る電子データにアクセスできる。

ソフトウェアのプログラミングには Java SE 1.6.0 を用いた。Large Screen と Resource Manager 間の通信におけるプロトコルは Java RMI（Remote Method Invocation）で実装した。Large Screen と Mobile Device の間の通信は QR コードを用いる。

4.1.1 Large Screen

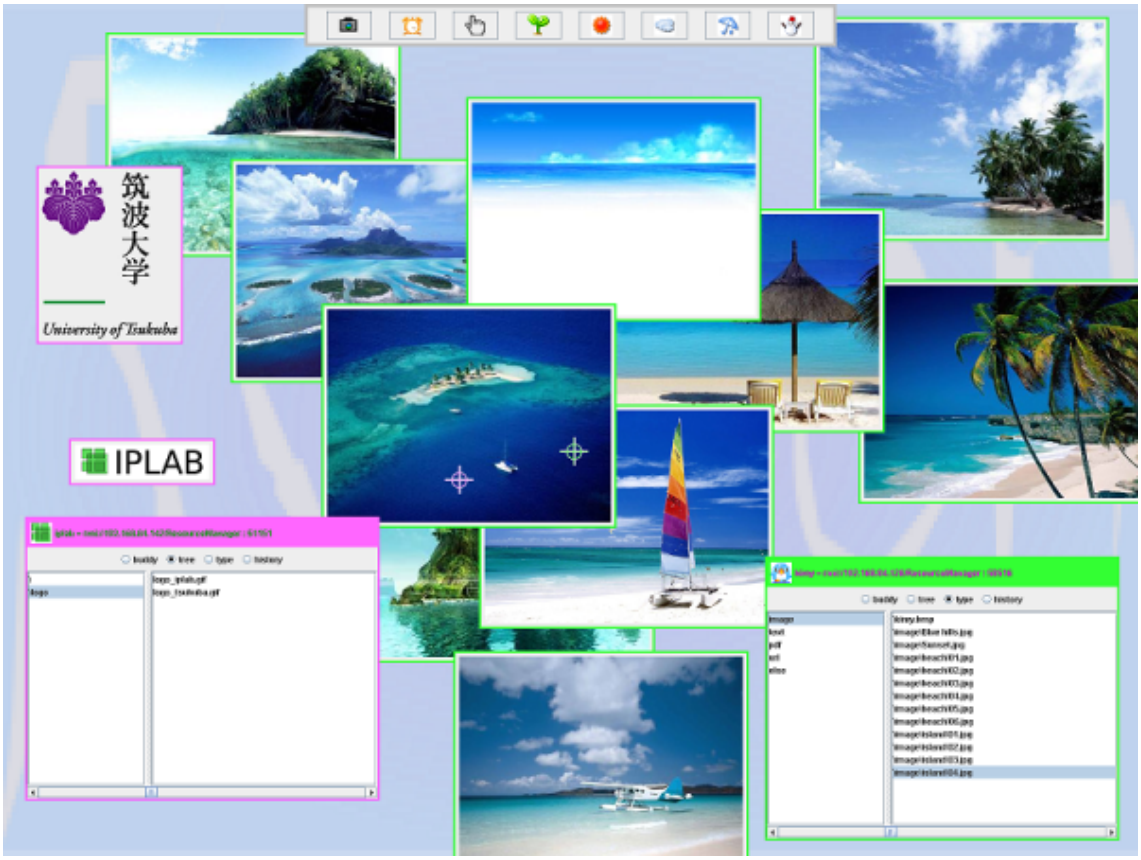


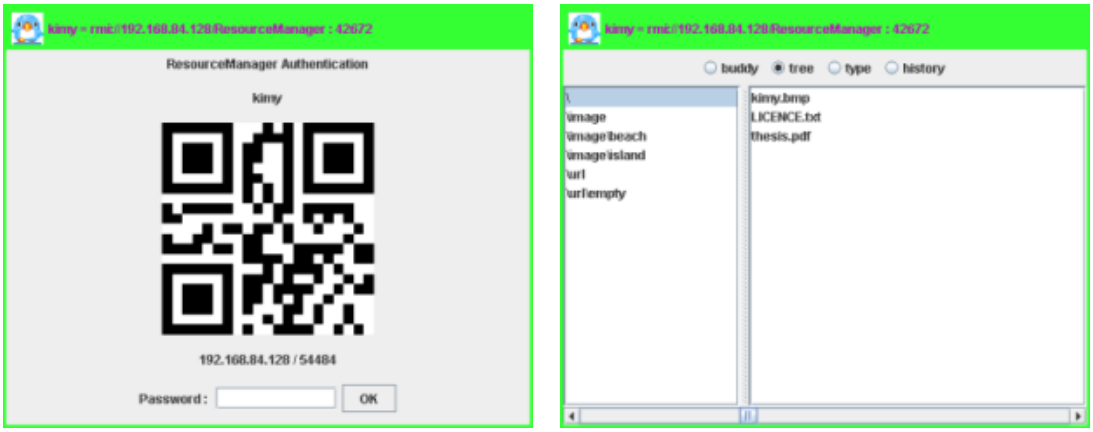
図 4.2: Large Screen のスクリーンショット

Large Screen は **personal workspace**、**public application**、**public workspace** の 3 つの領域で構成される。利用例のスクリーンショットを図 4.2 で示す。2 人のユーザがいて、それぞれ personal workspace とカーソルを持つ。personal workspace のボーダーラインとカーソルの色はユーザにより異なり、public workspace に置いてあるアイテム（表示された電子データを呼ぶ）のボーダーラインの色で生成したユーザがわかる。

4.1.1.1 personal workspace

Large Screen がユーザを識別すると、personal workspace が現れる。図 4.3 は personal workspace を示す。セキュリティー保障のため、他ユーザの personal workspace の中を操作することはできなくなっている。

ユーザが大画面情報アプライアンスに近づけば、図 4.3(a) のような personal workspace が現れる。これは Resource Manager からユーザの認証がされていない状態（semi-valid）である。semi-valid 状態になっただけで、カーソルが生成され public workspace の利用が可能になる。しかし、利用できる Resource Manager の機能は制限されて、認証なしで利用できる Resource Manager の機能は大画面情報アプライアンスが利用できるものに限り、実際にユーザが利用できるものは何もない。

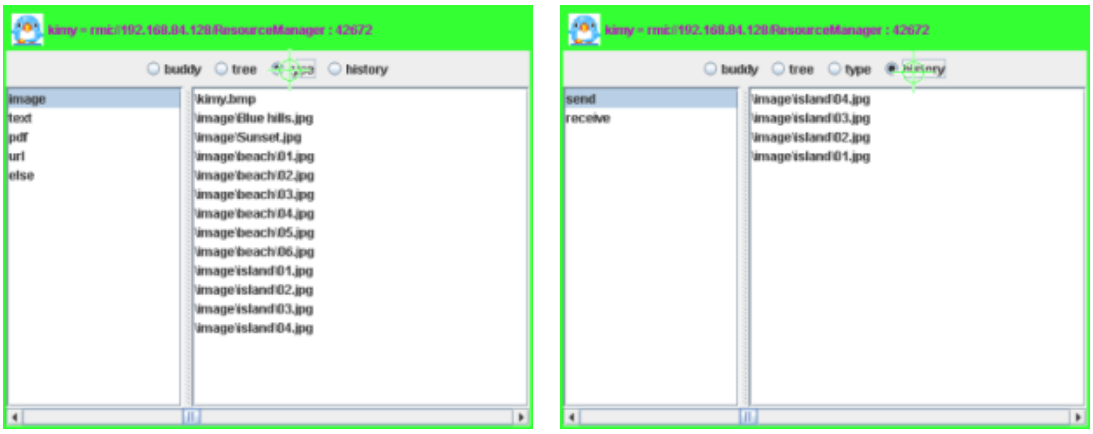


4.3(a) 4.3(b)

図 4.3: ユーザ認証前後の personal workspace

認証を行うと、図 4.3(b) のような personal workspace が現れる。これは Resource Manager からユーザの認証がされている状態（valid）である。personal workspace の中に表示される電子データは、実際には **Resource Manager** が管理しているサーバの中に存在する。valid 状態では personal workspace の中に表示される電子データを public workspace に表示することができ、他ユーザにコピーをあげることもできる。しかし、プライバシー保護のため、他ユーザの personal workspace を操作したり、他ユーザが public workspace に置いた電子データを勝手にコピーしたりすることは禁じられている。

public workspace の中の電子データの表示部が 2 つのウィンドウに分けられていることと、表示モードを **tree**, **type**, **history** の 3 つから切り替えることで、閲覧したい電子データを効率良く見つけることができる。tree モードではディレクトリ構造（図 4.3(b)）、type モードではデータの形式（図 4.4(a)）、history モードでは転送履歴（図 4.4(b)）で見ることができる。また、下部のスライドバーをスライドさせることで、左右 2 つのウィンドウを見やすいサイズに調整できる。



4.4(a) 4.4(b)

図 4.4: personal workspace の表示モード

4.1.1.2 public application

図 4.3 の上段に並んでいるいくつかのアイコン、又は、そのアイコンが並んでいるパネルを public application と呼ぶことにする（図 4.5）。また public application により生成されるアイテムをアプリケーションと呼ぶことにする。大画面情報アプライアンスにより提供したいサービスを予め用意したものであり、public application の種類は大画面情報アプライアンスごとに異なる。

図 4.5: public application

4.1.1.3 public workspace

public workspace は personal workspace, public application を除くすべての領域であり、personal workspace や public application から生成されるアイテムが位置する。

他ユーザの personal workspace は、操作は勿論、移動させることもできない。しかし、アイテムとアプリケーションは、アイテムの持ち主ユーザでなくてもドラッグ＆ドロップすることにより移動させることができる。

アイテムとアプリケーションは最後にタッチされたものほど手前に表示される。アイテムとアプリケーションは personal workspace より手前には表示されず、personal workspace は public application の手前には表示されない。

4.1.2 Resource Manager

Resource Manager は Large Screen とは別のコンピュータで起動していて、Large Screen から呼び出しがあるとそれに応答する。Large Screen によりユーザが認識されてから、ユーザが電子データを閲覧するまでの簡略なプロトコルを図 4.6 で示す。

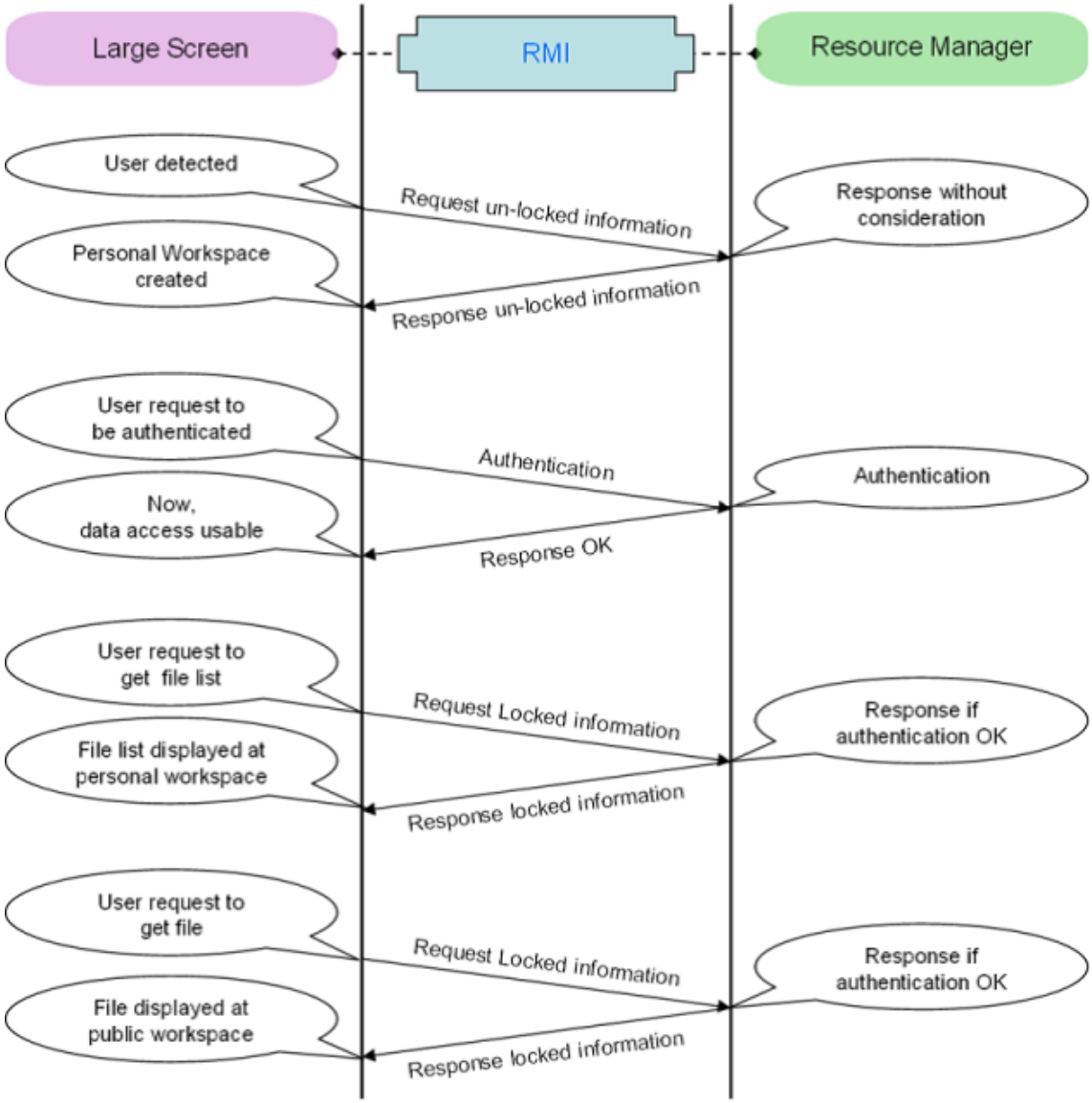


図 4.6: Large Screen と Resource Manager 間の通信におけるプロトコルの流れ

Large Screen が Resource Manager から得られる情報は、ユーザ認証の必要により 2 種類に分けられる。一つは、電子データやその一覧はユーザ認証が必要な **locked information** であり、valid 状態でなければアクセスできない。例えば、図 4.7 の青い文字で示されるようなメソッ

ドがそれに該当する。通常、valid 状態でアクセス可能な情報は、ユーザが能動的に Resource Manager から取得する場合が多い。もう一つは、ユーザの名前やアイコンなどはユーザ認証を必要としない **unlocked information** であり、semi-valid 状態でもアクセス可能である。例えば、図 4.7 の赤い文字で示されるようなメソッドがそれに該当する。通常、semi-valid 状態でアクセス可能な情報は、ユーザの意思ではなく情報アプライアンスの判断で自動的に Resource Manager から取得する場合が多い。

ResourceManager.java

```
1   public interface ResourceManager extends Remote {
2           public boolean signIn(String pass, int port) throws RemoteException;
3           public boolean signOut() throws RemoteException;
4           public String getName() throws RemoteException;
5           public Color getColor() throws RemoteException;
6           public ImageIcon getIcon() throws RemoteException;
7           public JList getFileList(String path) throws RemoteException;
8           public JList getDirectoryList(String path) throws RemoteException;
9           public byte[] getFile(String fileName) throws RemoteException;
10          public boolean setFile(String fileName, byte[] data) throws RemoteException;
11          ・・・
12  }
```

図 4.7: インタフェース ResourceManager

Resource Manager の実行の前に、ユーザの個人情報を環境変数のようなものを用意する必要がある。例えば、ユーザの色やアイコンなどの情報をまとめてファイル（図 4.8）に置いて、Resource Manager の実行時にコマンドライン引数でファイル名を渡すようにする。

rm.conf

```
1   NAME=kimy
2   HOME=C:¥Documents and Settings¥kimy¥デスクトップ¥LS_HOME
3   PASS=gogo1024
4   COLOR=-13369549
5   ICON=kimy.bmp
```

図 4.8: rm.conf の例

図 4.9 はテキストで編集しても構わないが、ユーザを手助けするツールとして Make RM Configuration（図 4.9）を作成した。特に、図 4.8 の４行目にある COLOR 項目は、ARGB の４バイトから成る integer 型を記入するようになっているので、ユーザが直接編集することは困難である（実際に、-13369549 は ARGB=(255, 51, 255, 51) を表す）。また、Make RM Configuration は図 4.8 の生成と同時に図 4.10(a～c) の画像も生成する。

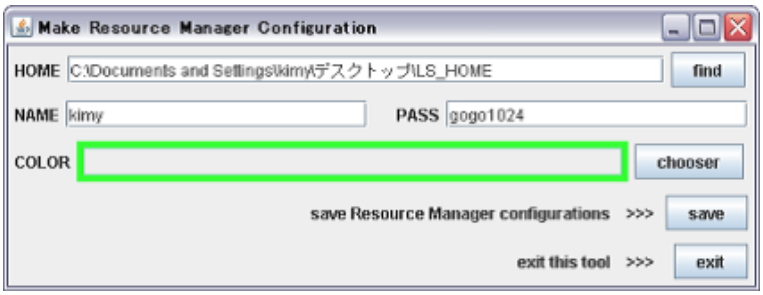


図 4.9: Make RM Configuration の実行例

4.1.3 Mobile Device

Mobile Device と Large Screen 間のインタラクション手法として次のようなものが検討された。

- wireless LAN-based（e.g. Bluetooth）
- IR-based（e.g. IrDA serial communication）
- image-based（e.g. visual marker）

今回は実装しやすい点から image-base の手法を用いた。つまり、Mobile Device と Large Screen 間の通信は QR コードと色の検出により行われる。そのためには大画面情報アプライアンスにウェブカメラを設置する必要がある。

4.10(a)

4.10(b)

4.10(c)

図 4.10: Mobile Device

図 4.10(a) を Large Screen のカメラに近づけると、Large Screen は QR コードをデコードして文字列「kimyMAKIFEIA」を得る。「kimy」がユーザ名、「MAKIFEIA」がユーザの Resource Manager の IP アドレスを表す。さらに、「MAKIFEIA」をデコードすると、「192.168.84.128」という IP アドレスが得られる。Large Screen は Resource Manager にアクセスして、図 4.3(a) のような personal workspace を生成する（semi-valid）。

その後、図 4.10(b) を Large Screen のカメラに近づけると、Large Screen は QR コードをデコードして文字列「kimygogo1024」を得る。「kimy」がユーザ名、「gogo1024」がパスワードを表す。Large Screen は Resource Manager にパスワードをもってユーザ認証を行い、認証されたら personal workspace は図 4.3(b) のように、電子データを利用できる状態（valid）になる。

図 4.10(c) をカメラの前で動かすと、マウスカーソルを操作ことができる。現在の実装ではユーザの色（「kimy」の場合、ARGB=(255, 51, 255, 51)）の検出を行い、検出された色分布の変化に合わせてカーソル移動を行う。また、同じ所に留まっていると、マウスボタンの押し・離しのイベントを発生させる。

本システムで IP アドレスを長さ 8 の文字列に変換するために、独自のエンコーダ・デコーダを作成した。これは QR コードでエンコードする文字列を短くし、QR コードの細かさを制限することで、モバイルデバイスの小さなディスプレイで移した QR コードを認識しやすくするためである。

図 4.11: システム実行例

図 4.11 で示す実行例は、モバイルデバイスで QR コードを表示して Resource Manager へ認証情報を送る場面である。背景が図 4.2 と異なるのは、背景をカメラ画像にする public application（図 4.5 の一番左側のアイコン）を実行しているからである。

ARGB=(255, 255, 102, 255) で象徴されるユーザ「iplab」（左上のところに personal workspace を持つ）が認証情報の入った QR コードを Large Screen のカメラに映している。Large Screen がその QR コードを認識すると、ユーザ「iplab」の Resource Manager に認証情報を送ってユーザ認証を行う。ユーザ認証に成功すると、ユーザ「iplab」の personal workspace は図 4.3(a) から図 4.3(b) のようになり、Resource Manager の管理しているストレージにアクセスできるようになる。すなわち、semi-valid 状態から valid 状態になる。

4.2　システムの機能

大画面情報アプライアンスは同時に複数のユーザが利用し、ネットワークを通じて Resource Manager が管理している電子データを閲覧・共有できる。また、大画面情報アプライアンスが予め用意したサービスも利用できる。

4.2.1　電子データの閲覧

personal workspace の中から閲覧したい電子データを public workspace にドラッグ＆ドロップすると、その電子データを表示するアイテムが public workspace に現れる。実際には、Large Screen が Resource Manager からその電子データを転送してもらうが、ユーザがそのような裏の処理を気にする必要はない。

アイテムのボーダーラインの色は personal workspace と同じになり、アイテムの持ち主ユーザが誰なのかが一目でわかる。もし、他ユーザのアイテムが邪魔になったら、ドラッグ＆ドロップで移動させることができる。personal workspace とは違い、持ち主ユーザでなくても移動は可能である。

4.12(a)

4.12(b)

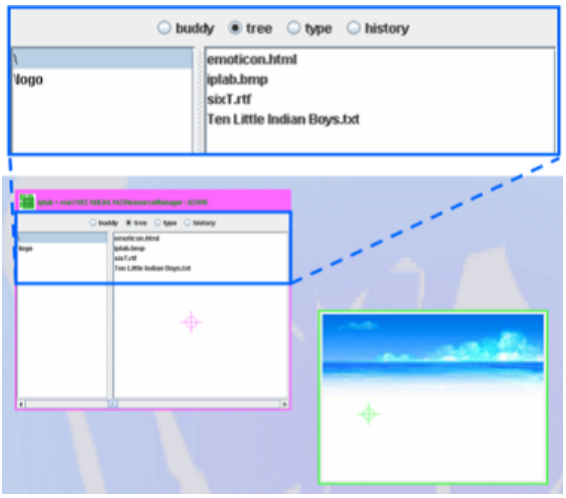
4.12(c)

4.12(d)

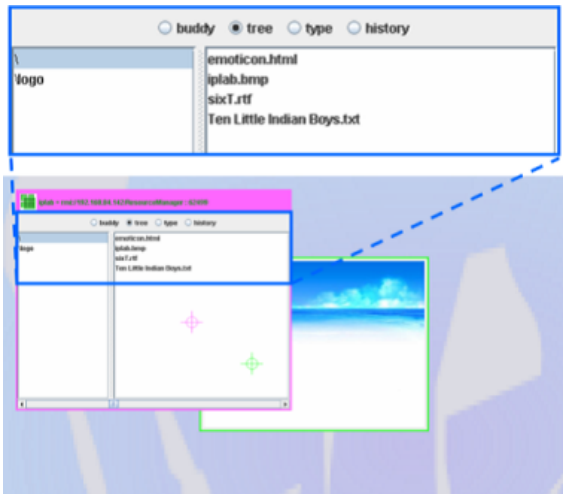
図 4.12: 閲覧できる電子データの形式

personal workspace から生成できるアイテムとして閲覧できる形式には、画像（jpg, gif, png, bmp）、テキスト（txt, rtf）、PDF 文章（pdf）、ウェブ（html, url）がある。それぞれ形式の電子データを表示するアイテムを図 4.12 の (a)〜(d) で示す。

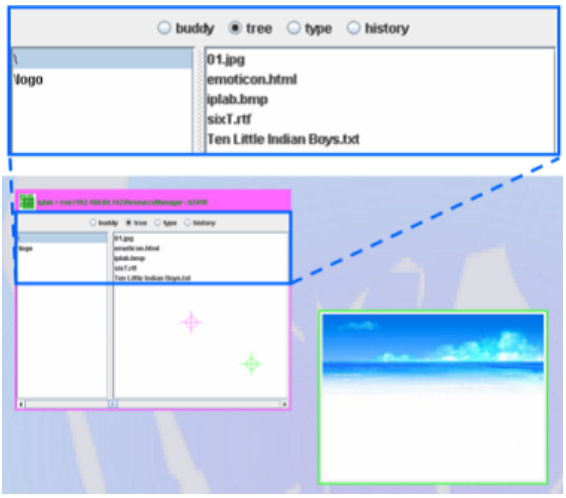
4.2.2　電子データの共有



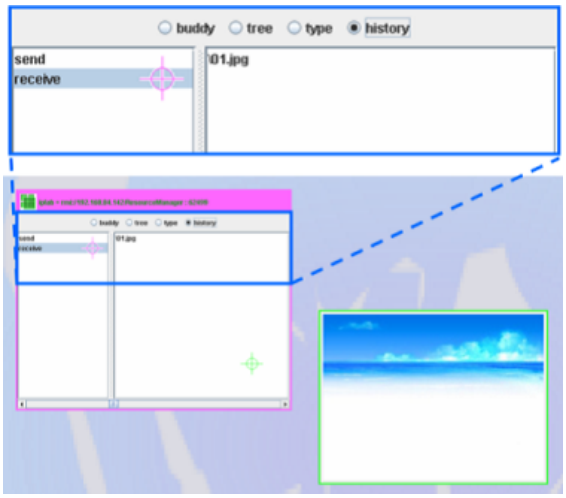
4.13(a)



4.13(b)



4.13(c)



4.13(d)

図 4.13: 電子データの共有

public workspace に表示してある電子データ（ファイル名 01.jpg）のアイテムを他ユーザの personal workspace にドラッグ＆ドロップすると（図 4.13(a)〜(b)）、電子データのコピーがそ

のユーザの Resource Manager に転送される。電子データを受け取った Resource Manager は、ホームディレクトリ（図 4.8 の 2 行目）直下にその電子データを置く。Large Screen は共有が行われるとすぐに、電子データを受け取ったユーザの personal workspace を更新する（図 4.13(c)）。なお、転送した・転送を受けた電子データの一覧は history モードで確認することも可能である（図 4.13(d)）。

しかし、共有の権限は持ち主ユーザにしか与えられない。すなわち、アイテムの持ち主ではないユーザが自分の personal workspace にドラッグ＆ドロップしても共有は行われない。図 4.13 でも、転送するアイテムのボーダーラインと転送を受ける personal workspace のボーダーラインが異なることから、それぞれの持ち主が異なることがわかる。したがって、電子データの共有ができるのである。

4.2.3　大画面が提供するサービスの利用

public application パネルのアイコンをクリックすると、それに相当するアプリケーションが public workspace の中で実行される。例えば、駅の構内に設置されている大画面情報アプライアンスなら、電車の時刻表や駅舎周辺の案内図が見れるアプリケーションなどが用意されているかも知れない。電車の時刻表を表示する public application の例を図 4.14 で示す。

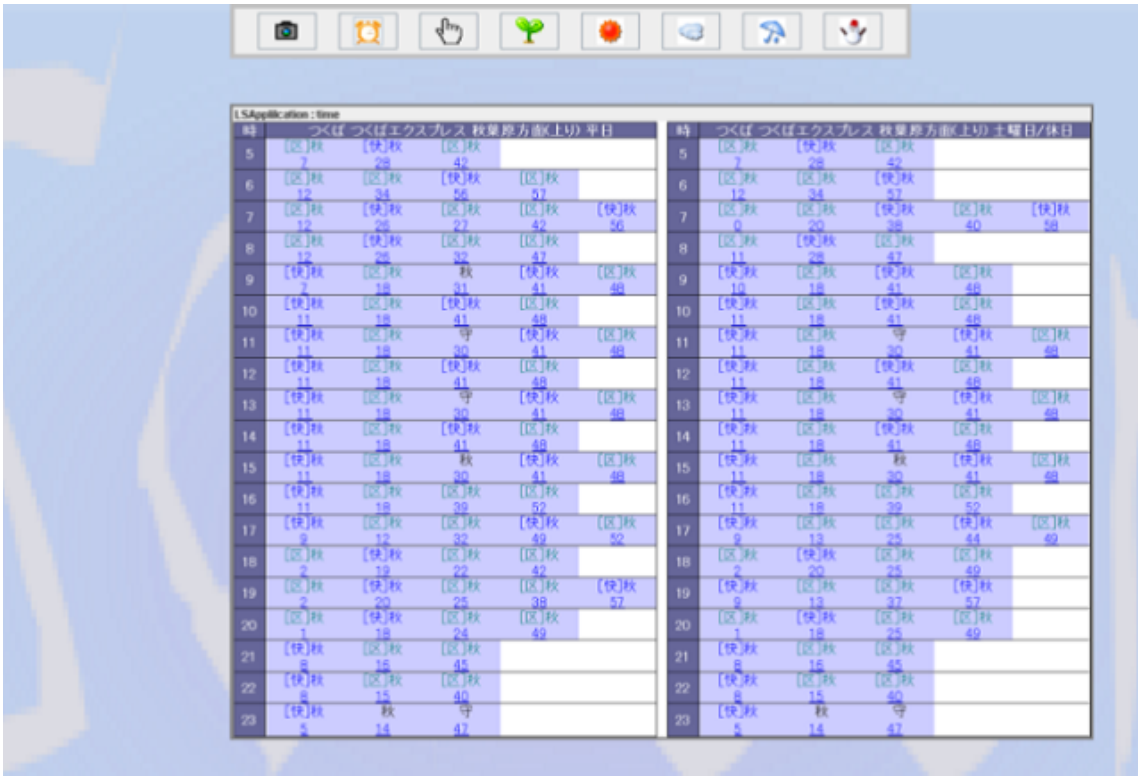


図 4.14: public application の例：電車の時刻表を表示するアプリケーション

第5章　まとめと今後の展望

5.1　まとめ

来るユビキタス時代に、至る所に情報アプライアンスが溢れるようになると考えられる。情報アプライアンスがユーザに適したサービスを受けるために、ユーザを識別しそのユーザのリソースマネージャにアクセスする仕組みを提案した。また、大画面を情報アプライアンスに拡張し、モバイルデバイスで操作してリソースマネージャの管理しているストレージを利用するシステムを試作した。

5.2　今後の展望

5.2.1　モバイルデバイスの改善

現在の実装では、モバイルデバイスで大画面情報アプライアンスのポインティング操作を行うために色の検出を用いた。しかし、ほとんどのモバイルデバイスのディスプレイ表面は光を反射しやすく、照明から悪影響を受ける場合が多い。まして、ユーザが操作に用いる色が大画面情報アプライアンスのカメラに映る背景に溢れているとしたら、ユーザの思うままに操作することが非常に困難である。この問題を解決するために、モバイルデバイスのディスプレイ表面に反射率を下げるシールを貼り付ける方法が考えられる。また、色の検出の代わりに、拡張現実感の実現を容易にするためのライブラリである ARToolKit[14] のマーカーを用いることで、ポインティングの精度を引き上げる方針を立てている。さらには、Bluetooth を用いることも考慮している。

5.2.2　リソースマネージャの機能拡張

リソースマネージャはいかなる情報アプライアンスからのアクセスにも対応しなければならない。例えば、ある情報アプライアンスはリソースマネージャからスケジュール情報を取得しようとするかも知れない。その場合、リソースマネージャからスケジュール情報が取得できる保障がないとならない。まず、リソースマネージャが提供できる情報かそうでないかだけでも、問い合わせできるメソッドが必要である。また、リソースマネージャが提供できる情報だとしたら、情報アプライアンスがスケジュール情報を理解できるように、形式情報を提供するメソッドも必要である。

謝辞

本論文を執筆するに当たり、指導教員である田中二郎先生、高橋伸先生をはじめ、三末和男先生、志築文太郎先生には、丁寧なご指導と適切な助言を多数いただきました。心より感謝申し上げます。

また、IPLAB(田中研究室) の皆様にもゼミなどを通じて大変貴重なご意見をいただきました。特にユビキタスグループの皆様にはグループでのミーティングだけでなく、日常的に多くのご意見やご指摘を頂きました。ありがとうございました。

最後に、私を支えてくれた家族、友人たちに、心より感謝いたします。

参考文献

[1] Mark Weiser. The computer for the 21st century. In *Scientific American*, pp. 94-104, Scientific American, 1991.

[2] Donald A. Norman. *The Invisible Computer*. MIT Press, 1998.

[3] Andrew M. Odlyzko. The visible problems of the invisible computer: A skeptical look at information appliances. In *First Monday, http://firstmonday.org/*, 1999.

[4] Guruduth Banavar, Abraham Bernstein. Software Infrastructure and Design Challenges for Ubiquitous Computing Applications In *Communication of the ACM*, pp. 92-96, 2002.

[5] Anind K. Dey. Understanding and Using Context. In *Personal and Ubiquitous Computing Journal*, pp. 4-7, Springer London, 2001.

[6] 石井裕. ユビキタスの混迷の未来. ヒューマンインタフェース学会誌, Vol.4, No.3, pp. 129-130, 2002.

[7] Raj Kumar, Xiang Song. GridLite: A Framework for Managing and Provisioning Services on Grid-Enabled Resource Limited Devices. In *HP Labs 2005 Technical Reports, http://www.hpl.hp.com/techreports/2005/HPL-2005-146.html*, 2005.

[8] 岩淵志学. ユビキタス環境における情報提示手法に関する研究. 筑波大学大学院博士課程システム情報工学研究科修士論文, 2006.

[9] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, Mia Underwood. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*, pp.159-168, ACM Press, 2003.

[10] Harry Brignull, Shahram Izadi, Geraldine Fitzpatrick, Yvonne Rogers, Tom Rodden. The Introduction of a Shared Interactive Surface into a Communal Space. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW '04)*, pp.49-58, ACM Press, 2004.

[11] Kento Miyaoku, Suguru Higashino, Yoshinobu Tonomura. C-Blink: A Hue-Hifference-Bbased Light Signal Marker for Large Screen Interaction via Any Mobile Terminal. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04)*, pp.147-156, ACM Press, 2004.

[12] Rafael Ballagas, Michael Rohs, Jennifer G.Sheridan. Sweep and point and shoot: phonecam-based interaction for large public displays. In *CHI Extended Abstracts*, pp.1200-1203, ACM Press, 2005.

[13] Chunming Jin, Shin Takahashi, and Jiro Tanaka. Interaction Between Small Size Device and Large Screen in Public Space. In *Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES2006)*, pp.197-204, IOS Press, 2006.

[14] ARToolKit Home Page. *http://www.hitl.washington.edu/artoolkit/*