

2007 年度 修士論文

アンビエント情報社会に向けた
Web ユーザインターフェースの開発

早稲田大学大学院理工学研究科
情報・ネットワーク専攻

横川 直人

学籍番号	3606U115-1
提出	2008 年 2 月 4 日
指導	成田誠之助 教授

概要

近年インターネットの利用者は年々増加し、2006 年現在でインターネット利用人口は 8,754 万人、人口普及率は 68.5%となった。ブロードバンドの普及率も高く 95%となっている。日本は世界有数のインターネット大国となった。インターネットの普及は情報通信社会における第 1 次革命といえ、現在は第 2 次革命であるユビキタス情報社会へと発展している。ユビキタス情報社会とは「いつでも」「どこでも」「誰とでも」というコンセプトを基にした情報通信社会の事を示す。日本では爆発的なインターネットの普及によってユビキタス情報社会の実現が近づいてきている。そのため 2000 年ごろからユビキタスに変わる次世代の情報通信社会の事を示す単語として「アンビエント」が用いられ始めた。アンビエント情報社会とは「今だから」「此处だから」「貴方だから」というコンセプトで IT がより自然に溶け込み、IT の方から人間にアクセスしてくる情報通信社会の事を示す。

アンビエント情報社会の実現のためにはハードウェアとソフトウェアの両方の面における様々な技術課題が存在する。ソフトウェアの面から考慮すると 2 つの技術課題が挙げられる。1 つ目はユーザに最適な情報を推定する技術を確立する事だ。アンビエント情報社会では常時ユーザに最適な情報を提供する事が求められる。2 つ目は情報をユーザに提示するユーザインターフェースを最適にする技術だ。アンビエント情報社会ではユーザが自分に最も適したユーザインターフェースで情報にアクセス出来る事が求められている。ユーザに適したとはユーザの状況に合わせて自分自身を最適化出来る事を示す。ユーザに最適な情報を推定する研究は盛んに進められており様々な推定技術が確立されている。それに伴い様々な情報システムが開発されている。しかし、ユーザインターフェースの問題にまで踏み込んだ研究はあまり進んでいない。また開発された情報システムはそれぞれが独自の方法で実装されており、連携して動くという事は容易ではない。アンビエント情報社会では様々な情報システムが連携して人間を支援する事が想定されており、この点についても考慮しなければならない。そこで、本研究ではアンビエント情報社会の実現に向けユーザの状況に合わせて最適化出来るユーザインターフェースの開発を目指す。また開発するユーザインターフェースには様々な情報システムを連携させる事の出来るプラットフォームとしての機能を持たせる事を試みる。

本研究では、WebOS を中心とした技術を用いてユーザインターフェースを開発する。WebOS とは、ブラウザ上で Ajax 等のリッチクライアント技術を用いて従来の OS の環境を再現する技術を示す。WebOS を中心とした技術を用いる事でユーザインターフェースを変形する事が可能になる。また開発したユーザインターフェースにはプラットフォームとしての機能を持たせ、プラットフォーム上で動作する Web アプリケーションをいくつか実装する。

本稿では、第 1 章でアンビエント情報社会と関連研究について述べる。第 2 章ではシステムの開発指針とシステムの概要について述べる。第 3 章では、システムの詳細について

述べる。第 4 章では開発者側から見た技術的側面について述べる。第 5 章では本研究のまとめと考察、今後の展望について述べる。

Abstract

Recently, the number of Internet users increases every year. The survey by Telecommunication white paper in 2006 revealed that the number of Internet users has exceeded 87 millions, and the Internet population diffusion rate has risen to 68.5 percent. 95 percent of the Internet users have broadband Internet connection, and Japan has become the IT superpower. The dissemination of the Internet is the first revolution in the telecommunication society, and currently, it is advancing to the second revolution, the Ubiquitous telecommunication society. The Ubiquitous, which recognized as “Always”, “Anywhere” and “Who” is the telecommunication society in the near future. With the explode in popularity of the Internet, people has started calling “Ambient” instead of “Ubiquitous”. In the Ambient telecommunication society, IT fits into the environment naturally, and it is time to speak to the human being from IT. The concept of the Ambient telecommunication society is developed for users’ accessing time, places, and personal information.

The two main technical difficulties to realize the Ambient telecommunication society are appeared; software and hardware problem. In this paper, the author solves software problems. Considering the software problem, the author finds two technical difficulties. The first is the technology that estimates best information for users. It is important in the Ambient telecommunication society to provide always best information for users. The second is the technology that develops interfaces which are suitable for users. It is necessary in the Ambient telecommunication society that people are able to access information with the most suitable interfaces for users. “Suitable for users” indicates that interfaces are optimized according to users’ situations. The research that finds the best information for users is actively advanced. With this advancement of the research, a variety of relating information systems is developed. The research, however, stepped into the problem of the user interface is progressed slowly. In addition, many developed information systems are mounted in the unique method so that cooperating the information systems is difficult. Cooperating various information systems and supporting human beings are assumed in the Ambient telecommunication society. This thesis aim at the development of the user interface which can be optimized in accordance with users’ conditions. The author implements functions to an interface so that various information systems can cooperate as a platform.

In this research, the author mainly used the WebOS technology to develop the user

interface. WebOS recreates the environment of the usual OS with rich client technology. A user can optimize the user interface by implementing the technology. In addition, diverse Web Applications are mounted on the system to confirm functions as the platform.

This article consists of six chapters. Chapter1 refers to the background about Ubiquitous telecommunication society and Ambient telecommunication society. Chapter2 presents the development guidelines of the system and the outline of the system. Chapter3 explains the details of the system. Chapter4 describes the technical side seen from the developer. Chapter5 discusses the summary of this research and the future directions.

目次

概要	1
Abstract	3
目次	5
1. 研究背景	7
1.1 ユビキタス情報社会の到来	7
1.2 ユビキタス情報社会の深化	9
1.3 ユビキタス情報社会からアンビエント情報社会へ	11
1.4 アンビエント情報社会に向けた研究機関の取組みと関連研究	12
1.4.1 アンビエント情報社会に向けた研究機関の取組み	13
1.4.2 アンビエント情報社会に向けた研究	13
1.5 研究の目的	15
2. システム概要	17
2.1 システムの開発指針	17
2.1.1 システムのプラットフォームとして WebOS を開発する	17
2.1.2 ネットワーク側からの情報配信を可能とする	19
2.1.3 ユーザに最適な情報を提案し行動支援出来る環境を構築する	20
2.2 システム概要	21
2.2.1 システム構成	21
2.2.2 WebOS(プラットフォーム部分)	22
2.2.3 Web アプリケーション	24
2.2.4 モバイル端末のプラットフォームと Web アプリケーション	24
2.2.5 本システムにおけるユーザインターフェースの変化	25
3. システム詳細	27
3.1 WebOS(プラットフォーム)詳細	27
3.1.1 ログイン処理	27
3.1.2 デスクトップアイコン関連の機能	29
3.1.3 スタートメニュー関連の機能	30
3.2 Web アプリケーション詳細	32
3.2.1 プレゼンテーションアプリ	32
3.2.2 ファイルサーバアプリ	34
3.2.3 notepad アプリ	35
3.2.4 スケジューラアプリ	36
3.2.5 周辺店舗探索マップアプリ	37
3.2.6 レコメンド RSS リーダーアプリ	39

3.2.7 テレビ番組表アプリ	41
3.2.7 街中案内システム	42
3.3 モバイル端末用プラットフォーム詳細	43
3.3.1 ログイン処理	43
3.3.2 トップ画面	44
3.4 モバイル端末用 Web アプリケーション詳細	45
3.4.1 プレゼンテーションアプリ	45
3.4.2 スケジューラアプリ	45
3.4.3 周辺店舗探索マップアプリ	47
3.5 本システムにおけるユーザインターフェースの変化詳細	47
3.5.1 モバイル端末用ユーザインターフェースの変化	48
3.5.2 WebOS のユーザインターフェースの変化	49
4.システムの開発	51
4.1 開発環境	51
4.2 本システム開発にあたっての技術的側面の詳細	51
4.2.1 Ajax について	51
4.2.2 WebOS について	53
4.2.3 Comet について	54
4.2.4 WebOS と Web アプリケーションの連携について	56
4.2.5 ベクトル空間法と TF・IDF 法について	57
4.2.6 Ajax ライブラリ「Bindows」について	59
4.2.7 Google Maps と Google Maps API について	59
4.2.8 PHP と PEAR について	60
4.2.9 Apache と MySQL について	60
5.システムの課題と今後の展望	62
5.1 まとめと考察	62
5.2 システムの課題	63
5.3 今後の展望	64
6.謝辞	66
7.参考文献	67

1. 研究背景

1.1 ユビキタス情報社会の到来

ユビキタスの語源はラテン語で「偏在するやどこにでもある」といった意味だ。ユビキタスという言葉自体は、ここ 10 年程度で一気に広まったのだが、情報化社会におけるユビキタスの概念が提唱され始めたのは 20 年程遡り、東京大学の坂村健氏が 1984 年から開始された TRON(The real-time operating system)の開発プロジェクトの中で「どこでもコンピュータ」という概念を提唱している。TRON は理想的なコンピュータアーキテクチャを目指した OS であり、現在もモバイル端末や家電製品に組み込まれている。この時の「どこでもコンピュータ」こそユビキタスという考え方を先取りしているものだった。アメリカでは 1988 年にゼロックス社パロアルト研究所のマーク・ワイザーがユビキタス・コンピューティングという言葉を用いて同様の概念を提唱している。

これらの概念を基礎に「ユビキタス」という言葉は近未来の情報化社会を指し示す代表的な言葉になったわけだが、その中核となる考え方は「いつでも、どこでも、誰とでも」情報をやり取りしコミュニケーションが取れるというものだ。

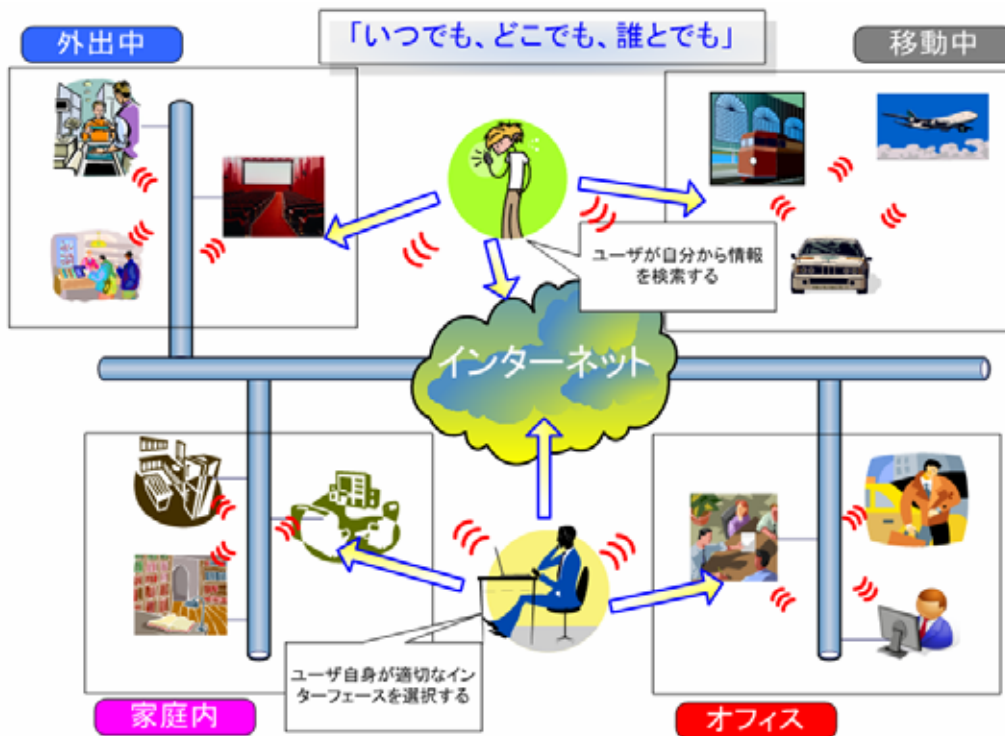


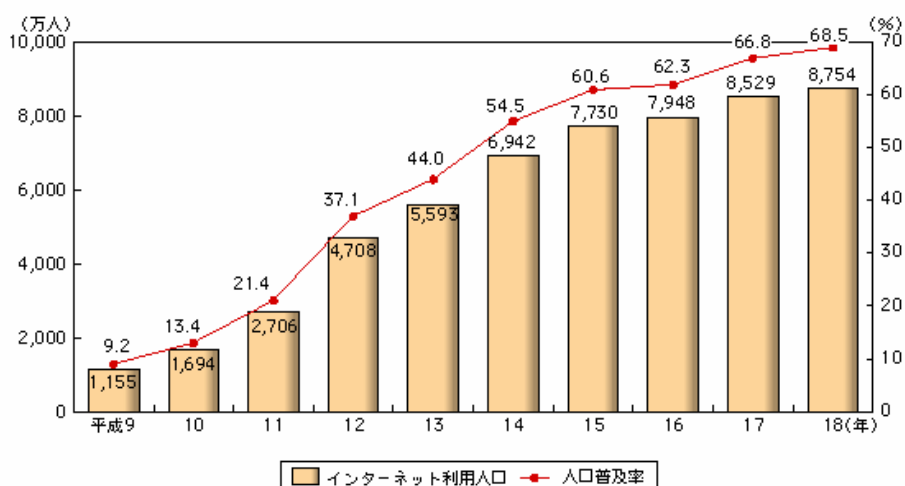
図 1.1 ユビキタス情報社会

本稿では「いつでも、どこでも、誰とでも」情報をやり取り出来る情報化社会のことをユビキタス情報社会と呼ぶ。

日本政府が最初に取り組んだのが e-Japan 政策だった。e-Japan 政策の重点政策分野は以下の通りで、

- ・超高速ネットワークインフラの整備および競争政策
- ・電子商取引の活性化
- ・電子政府の実現

まずはユビキタス情報社会に向けてネットワークインフラを重点的に整備する政策だった。その結果、インターネットの人口普及率は 68.5%、利用人口は 8,754 万人(平成 18 年現在)と世界でも有数のインターネット大国となった。またブロードバンドの普及率も高いいずれかのブロードバンドについては 95%、超高速ブロードバンドについては 84%と高水準になっている。



(出典)総務省「通信利用動向調査(世帯編)」

図 1.2 インターネット普及率(出展：平成 19 年版 情報通信白書)

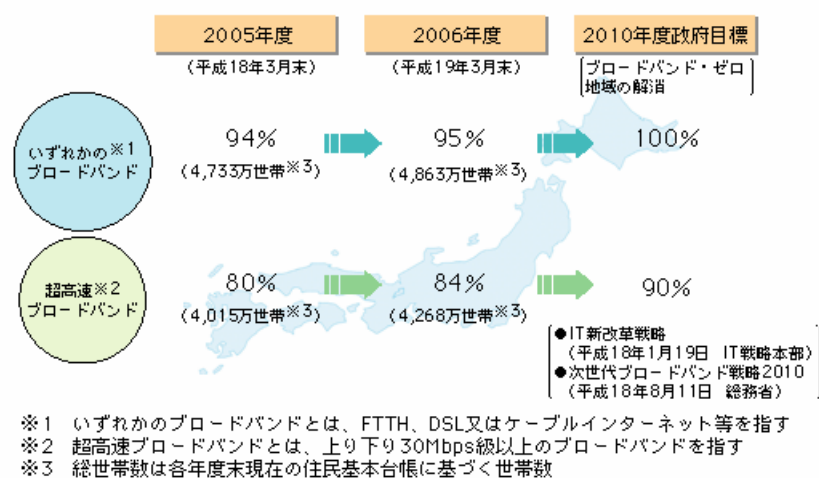


図 1.3 ブロードバンド世帯カバー率(出展：平成 19 年版 情報通信白書)

1.2 ユビキタス情報社会の深化

日本の情報通信環境は様々な技術変革を伴いながらより深化したユビキタス情報社会へと変わってきている。情報通信環境のモバイル化はユビキタス情報社会の深化をけん引する中心的な役割を担っている。携帯電話の急速な普及や、デジタル・オーディオプレーヤー、PDA 等の各種モバイル端末の高性能化やモバイルネットワーク環境の整備が大きな要因となっており、6 歳以上人口のうち携帯電話・PHS でインターネットを利用している人は平成 18 年末で 53.3%となっている。

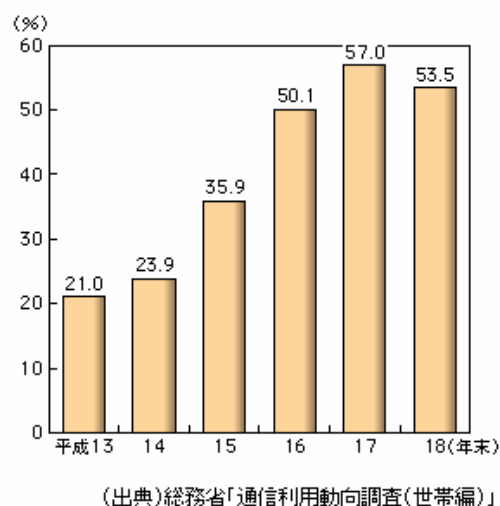
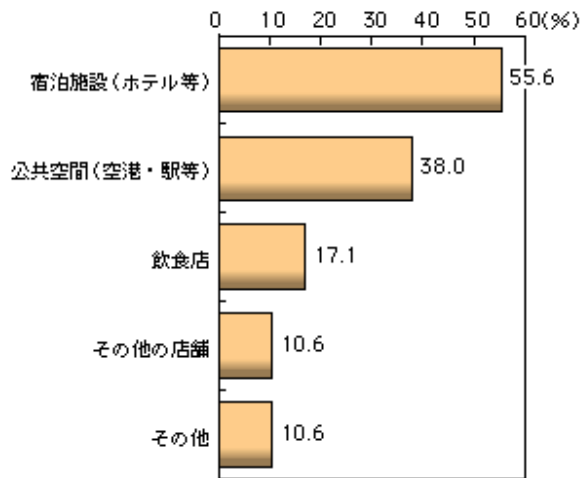


図 1.4 携帯インターネット利用率(出展：平成 19 年度版 情報通信白書)

また、無線 LAN スポットの展開も進んでおり、ホテル等の宿泊施設や空港等の公共の場所、飲食店から無線でインターネットに繋ぐ事が容易になっており、サービスエリアも拡大しているなどより利便性が高まっている。



(出典)総務省「平成18年通信利用動向調査(世帯編)」

図 1.5 公衆無線 LAN の利用場所(出展：平成 19 年度版 情報通信白書)

ここ数年の高性能モバイル端末の普及やブロードバンドの高い普及率と HTML の急激な変化による表現方法の広がりによって情報配信を行う個人のすそ野が急速に拡大している。その結果、ブログやソーシャルネットワークサービス(SNS)、口コミサイト等の消費者配信型メディア(CGM：Consumer Generated Media)が従来既存メディアだけが情報配信をしていた情報通信環境に大きな影響を与える様になった。消費者自身が積極的に情報を配信し「利用者参加」と「オープン指向」という特徴があるこれらの変化は一般的に「Web2.0」現象と呼ばれる。

「利用者参加」という観点から見れば、このような個人から配信される情報は多数の消費者によって情報が収集・増強され、ますます大きな社会影響力を持つようになる。多数の消費者が情報を配信する様になれば、既存メディア以上の影響力を持つようになる。価格.com 等の評価サイトではネット上に集まる消費者の声によって商品やサービスの価格、品質の妥当性が客観的に評価され第三者がそれを買うかどうか大きな影響を与えている。ウィキペディアに代表されるユーザ同士の協働作業による知の集合は当初の予想を遥かに超える規模になっている。

「オープン指向」という観点から見れば Amazon や Google に挙げられる企業は自らのデータを公開する事によって更に大きなメリットを得ている。自社に蓄積されているデータ及びデータへのアクセス(API)を公開する事によってより多くの消費者が自社サイトに集まる様になった。また多くの個人が様々なアイディアを持ち寄って新しいサービスやアプリケーションを開発して行くので消費者と企業のシナジーによってそれまでに無かった新しい産業が作られて行く事になった。

これらのトレンドによって配信される多様かつ豊富な情報がユビキタス情報社会を発展させる大きな源泉となっている。

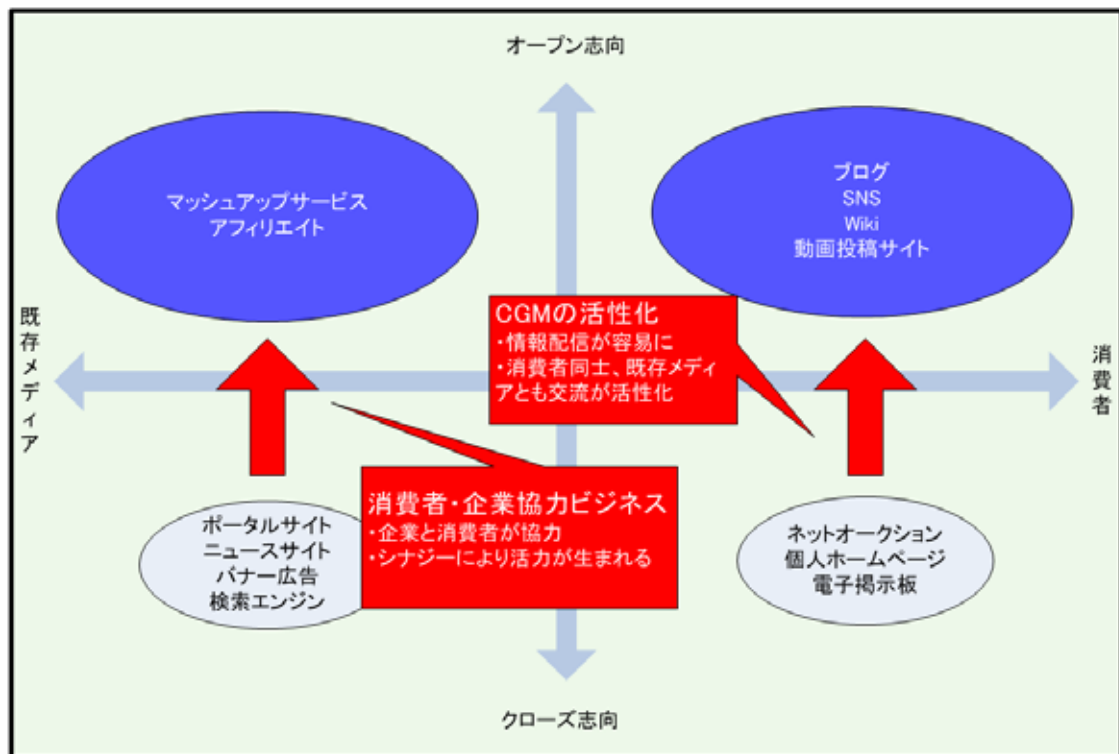


図 1.6 ユビキタス情報社会における「Web2.0」現象

1.3 ユビキタス情報社会からアンビエント情報社会へ

アンビエント(Ambient)とは英語で「周りの」とか「環境の」といった意味だ。日本でインターネットが爆発的に普及を始めてから約 15 年経ったが、その間に情報通信社会は大きく様変わりした。インターネットの普及は情報通信社会における第 1 次革命をいえ、現在は 1.2 で述べた第 2 次革命であるユビキタス情報社会へと深化している所だ。2000 年頃から次世代の情報通信社会を示す言葉として「アンビエント」という単語が使われ始めた。

「ユビキタス」の中核となる概念は「いつでも、どこでも、誰とでも」というものだった。「アンビエント」は「ユビキタス」に内包される概念と言えるが、情報通信社会において「アンビエント」とは「ユビキタス」の概念に大きく二つのコンセプトを追加して考えられている。

一つ目はより「自然に」というコンセプトだ。ユビキタス情報社会ではユーザが情報を必要とする時、適切なインターフェースを選択し情報を得る。例えばモバイル環境ではモバイル環境に適したインターフェース、携帯や PDA 等を使って情報を得る。それ以外では PC を使って情報を得る。この様にユーザが自身でインターフェースを選択する必要があった。「アンビエント」で追加された「自然に」というコンセプトではインターフェースとしてのデバイスが空間中(より自然に近い形で)に存在しユーザが選択するのではなく、自動的

に適切に選択される必要がある。これは主にセンサの技術革新を推し進めるためのコンセプトとなっている。

二つ目は「ネットワーク側から」というコンセプトだ。ユビキタス情報社会ではユーザつまり人間側から情報が必要な時に検索し情報を求める社会だった。「アンビエント」で追加されたコンセプトでは発想がユビキタス情報社会とは逆でネットワーク側から人間にアクセスし行動のアドバイスや情報の提案をしてくれる世界を目指すというものだ。これは主にソフトウェアやサービスの技術革新を推し進めるためのコンセプトとなっている。

以上の点をまとめると次の様になる。「アンビエント」という概念に基づいた情報通信社会は環境中のコンピュータから「今だから、此处だから、貴方だから」ユーザに最適な情報を提案したり、アドバイス出来るというものになる。本稿ではこの概念に基づいて実現される情報通信社会の事をアンビエント情報社会と呼ぶ。



図 1.7 アンビエント情報社会

1.4 アンビエント情報社会に向けた研究機関の取組みと関連研究

アンビエント情報社会を実現するために求められている技術課題には以下の様なものがある。

- ・ 情報環境と周辺環境を認識する技術
- ・ 自立的にネットワーク環境を構築する技術
- ・ 行動支援を可能とするコンテンツ作成技術
- ・ シームレスなネットワーク接続を可能とする技術
- ・ パーソナル情報をセンシングする技術
- ・ パーソナル情報を収集する技術 等

この節ではこれらの技術課題を解決するための取組みと関連研究について紹介する。

1.4.1 アンビエント情報社会に向けた研究機関の取組み

早稲田大学は、2007 年度から文部科学省が実施する「グローバル COE プログラム」において、情報・電気・電子分野のグローバル COE 拠点に選ばれた。このプログラムではナノテクノロジー(NT)を駆使した新材料と革新的デバイス、高度な情報サービスを創出する情報通信基盤ソフトウェア(IT)、および、誰でもが安全・快適に使い易い情報環境を提供するアンビエント技術(AT)を結集し、これらを統合化しチップ上に実現する SoC(Sensor, Software and Service on Chip)技術の国際拠点となる事を目指している。

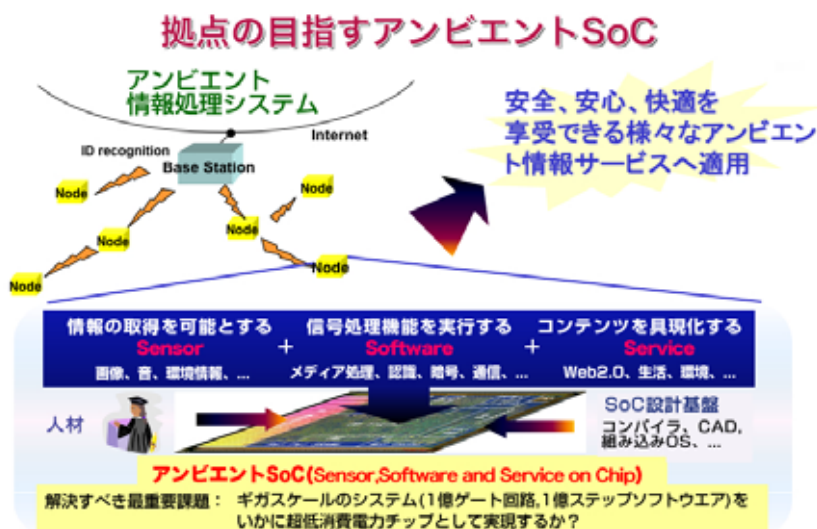


図 1.8 早稲田大学のアンビエント情報社会に向けた取組み
(出展：早稲田大学大学院理工学研究科 HP)

1.4.2 アンビエント情報社会に向けた研究

前節までで述べた通り、アンビエント情報社会とは「今だから、此处だから、貴方だから」といった IT がより自然に溶け込み、人間の方から情報を検索しに行くのではなく、ネットワーク側から情報がアクセスしてくる情報通信社会の事を示す。アンビエント情報社会を実現させるための研究には大きくわけて情報環境や周辺環境を認識させるための環境

カメラや環境センサといったハードウェアの研究と自立的ネットワークの構築やコミュニケーションを支援するサービスといったソフトウェア・サービスの研究がある。それぞれについていくつかの関連研究を紹介する。

環境センサを使った研究としては東京大学の森らによる「生活パターンを覚えて助ける知能住宅」[1]を紹介する。この研究では多数のセンサをネットワークで結合し、個人の「部屋」に着目し、その中で行われる生活パターンを分析し行動支援に生かしている。

ソフトウェア・サービスの研究では主に、周囲の環境からユーザに情報サービスをどのように推薦するかに重点を置いて研究が進められている。中央大学の宮崎ら[2]は各個人の自然な振る舞い(移動や動作)に着目し、実世界空間内に多数のモニタカメラなどのセンサを配置し、利用者に対して身体的・心理的負担をかけずにパーソナルデータを取得出来る環境を構築した。これらのパーソナルデータを解析する事で個人の嗜好や興味を推定し、個人ごとの行動履歴と地点の情報履歴、個人のパーソナルデータをもとに最適な商品を推薦するシステムを開発した。東京大学の金杉洋ら[3]はセキュリティの問題やユーザビリティの低さから誰の手元にも包括的な個人情報が存在せずより高度なコンテキストウェアサービスの実現に向けて障害になっている点に着目し、ユーザ自身が定常的・自発的にパーソナル情報を取得・蓄積・管理するためのプラットフォームを開発した。またこれらのパーソナルデータをもとにベイジアンネットワークを用いてユーザ行動のモデルを構築した。立命館大学の馬谷達也ら[4]は人間が何か興味を持つと通常とは違う動作をするという点に着目し、彼らが開発している街中の大型ディスプレイで個人に向けて情報を配信するシステム「Wonder Wall」の前で移動を停止した場合、ユーザが興味を持ったというパーソナル情報を蓄積するシステムを開発した。これまでに示したソフトウェア・サービスの研究例では主にユーザにどう最適な情報を推定するかという点に重点を置いて研究を行っている。周囲の環境からユーザに情報サービスを推薦するには、最適な情報を推定するだけでなく情報の表現方法である最適なインターフェースについても開発をしなければならない。

次にインターフェースに重点を置いている研究を紹介する。北海道大学の工藤峰一ら[5]は生活の場において自然で豊かな情報通信を行うには場所や環境に依らず、また、心理的・物理的に生活の障害にならないユーザ認証が必要であり利用者の好む装置や形式で通信機器を利用できる環境が重要であるとした。彼らの研究ではこれらの問題を解決するためパターン認識を導入し、電子メールに着目し初心者や高齢者でも簡単に無理なく扱え機能や入出力機能を利用者に応じて適切にパーソナライズすることでIT機器を使う場合の習熟困難性を解消した。北陸先端科学技術大学院大学の藤井崇史ら[6]はニュースサイト等で提供されているインターフェースをカスタマイズできるサービスに初期登録や自己入力の手間が問題である事を指摘し、ユーザの利用端末と閲覧履歴を用いたパーソナライズによる方法で表示部を動的に構築するシステムを開発した。

1.5 研究の目的

アンビエント情報社会の実現に向けては、様々な技術的課題を解決するため、ハードウェア・ソフトウェアの両方の観点から研究を進める必要性がある事は先に示した。本研究では特にソフトウェアの観点でアンビエント情報社会の実現に向けて研究を行った。

個人の持つ価値観はユビキタス情報社会の深化により急速に多様化してきている。これは1人1人が興味・関心を持つ対象や、嗜好等が異なるからである。更に必要とする情報が異なっていると同時に情報のやり取りに使う表現方法つまりインターフェースについても個人によって好みが異なる。アンビエント情報社会に向けた情報サービスでは、1人1人の価値観に基づく情報を推定し、最適なインターフェースで提供する事が最も求められている。先に示した様に様々な方法を用いて、ユーザに最適な情報を推定する手段と情報を提供するためのインターフェースについて研究が進んでいるがまだまだ問題も多い。筆者が特に注目している点が最適なインターフェースをどの様に選択し、提供するのかという点である。ここで考えなければならないのはユーザにとって最適なインターフェースというのは周囲の環境やユーザの状況によって最適化されなければならないという点だ。そもそもユーザにとってのインターフェース(以下ユーザインターフェース)とは開発者の立場からすると軽視されがちであるが、ユーザの立場からすれば情報システムそのものである。なぜなら、ユーザがある情報システムを利用する際に常に操作するのがユーザインターフェースだからだ。にもかかわらずユーザインターフェースが開発者から軽視されがちなのは、ユーザインターフェースの効率化によってユーザの利用効率の上昇よりも開発者がデータ処理の効率や手間かからない実装を優先してしまうぐらいユーザインターフェースの効率化というのが煩雑で手間がかかるからである。この様に煩雑なユーザインターフェースをアンビエント情報社会では周囲の環境やユーザの状況に応じて形態を変えて提供するので、アンビエント情報社会におけるユーザインターフェースの開発は大きな技術課題であると筆者は考えている。また先に紹介したソフトウェアの観点から見た研究ではそれぞれが独自のユーザインターフェースを実装している。もちろんそれぞれの情報システムに適したユーザインターフェースというのはあると思うがアンビエント情報社会においては出来る限り統一されたユーザインターフェースが提供されるべきだと筆者は考える。アンビエント情報社会においては「今だから、此处だから、貴方だから」というコンセプトで莫大な情報の中からユーザに最適な情報だけを提案するだけで無く、情報サービス自身が他の情報サービスと連携してユーザの行動を支援する様になる。よってアンビエント情報社会における情報システムのユーザインターフェースも統一された形態で提供されるべきだと筆者は考えている。

以上の事をまとめると、以下の通りになる。

ソフトウェアの面からアンビエント情報社会の情報システムに求められるものは次の二つである。

1. 「今だから」「此処だから」「貴方だから」ユーザに最適な情報が配信される情報システムの開発

2. 1 の情報システムが統一された形式で連携出来、ユーザに最適な形で情報を掲示できるユーザインターフェースの開発

1. に関しては先に述べた様に様々な研究が進んでいるが2. に関してはまだ研究が進んでいない。そこで本研究では2. のアンビエント情報社会に求められるユーザインターフェースの開発を目指す。

2.システム概要

2.1 システムの開発指針

本研究で開発した情報システムは、ユーザの状況に応じてユーザに最適な情報を最適なユーザインターフェースで提供する Web アプリケーションである。Web アプリケーションとして例を挙げれば google による検索システム等が想像されるが本システムでは技術的に Web アプリケーションに分類されるだけでコンセプト自体はまったく違うものになっている。従来の Web アプリケーションはユーザが情報を欲しい時に起動してインターネットから情報を引っ張ってくる事が主な機能となっている。本システムは発想が逆で極端に述べるとユーザに必要な情報はユーザが欲しいと思った時にはすでにインターネットから本システムは集まっており、ユーザが一切操作をしなくても必要な情報が必要な形式でユーザに配信されるという事が主な機能となっている。実生活で例を挙げればお昼ご飯を食べたいと頭の中に考えた時に、突然目の前にディスプレイが表れその日に食べたいお店の情報を最もわかりやすい形にして表現するといった具合だ。ただしアンビエント情報社会では情報伝達を視覚に頼るとは限らないのでディスプレイでは無く突然お店の料理の味が味覚を通じて伝えられるかもしれない。

上に述べた本システムのコンセプトはかなり極端な言い方をしているが、究極的にアンビエント情報社会に必要な情報システムというのは先に述べた機能が必ず必要となる。本研究では、この機能を目指すためにいくつかの開発指針を決め、それによってシステムを開発した。開発指針について次に説明する。

2.1.1 システムのプラットフォームとして WebOS を開発する

最初の開発指針はプラットフォームに何を使用するかという点について決めている。情報システムにおけるプラットフォームを決めるという事は、全ての動作についてその処理と処理に必要なデータをどこに置くかという基礎部分を決めるという事だ。現在、パソコンで本稿を閲覧しているのならば、閲覧しているオペレーティングシステム(以下 OS)がプラットフォームという事になる。論文のデータを OS が保存・管理しており、論文のデータを読む処理も OS に保存されている。現在は全ての情報システムのプラットフォームはこの OS という事になっている。WebOS とは、その名前の通り OS の機能をそのままインターネット上で動作する様に移植したものだ。従来の OS はパソコンが無ければ動作しないが、WebOS はインターネット上で動作するので Web ブラウザ上で動作する事になる。HTML を基本とした Web 技術の表現方法が技術革新した事により実現した技術であるが、WebOS の技術面に関する詳しい説明は 4.2.2 にて述べる。

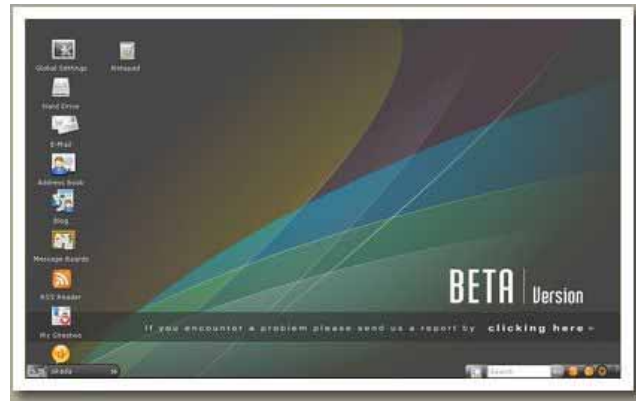


図 2.1 WebOS の一例(出展：Desktwo)

WebOS の特徴としては次の様な項目が挙げられる。

- ・ プログラム管理、ファイル管理、ウィンドウ管理等、従来の OS の機能が提供される
- ・ WebOS 自体はブラウザ上で動作するので従来の OS も必要となるが OS には事実上標準でブラウザが搭載されているので、WebOS を動作させるための条件は従来の OS を動作させるための条件と変わらない
- ・ 全ての処理の実行とデータの保存がインターネット側で行われる

ここで本システムに求められる機能について今一度検討する。1つ目はユーザに最適な情報を提供するという機能だ。この機能を実現するためにはユーザがどんな嗜好や興味を持っているかというパーソナルデータがまず必要になる。またパーソナルデータとマッチングするデータは当然インターネット上に存在する。この2つのデータについて検討してみると、従来のOSでは前者はローカルの端末上に後者はネットワーク側に存在する事になる。WebOSの場合前者も後者もネットワーク側に存在する事になる。ユーザに最適な情報を提供するにはパーソナルデータとインターネット上のデータをマッチングする必要があるので、この点においては従来のOSよりもWebOSの方が優れている事がわかる。

2つ目はユーザに最適なユーザインターフェースを提供する機能だ。問題をわかりやすくするため具体的に検討してみる。標準的なブラウザを持つ端末Aと端末Bがあるとする。更にユーザに最適なユーザインターフェースが端末A上でも端末B上でも同じであったとしよう。この時にそれぞれの端末にユーザインターフェースを実装する事を検討してみる。仮に端末Aと端末B上で違う種類のOSが動作していた場合、同じ動作をするユーザインターフェースを実装するとしても端末A上と端末BではプラットフォームであるOSの動作が違うため2つのユーザインターフェースのプログラムを実装しなければならない。WebOSの場合、標準的なブラウザがあればOSの種類が違ったとしてもプラットフォームであるWebOSが同じ動作をするので、端末Aと端末BのOSの種類が違ったとしてもユ

ーザインターフェースは 1 つのプログラムだけ実装すればよい。この点においても従来の OS よりも優れている事がわかる。

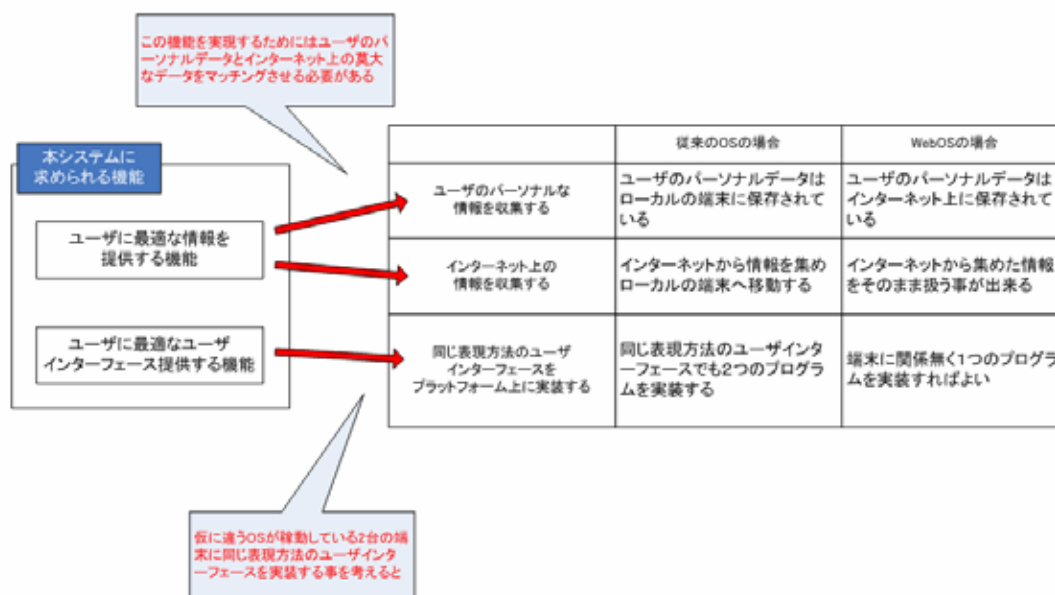


図 2.2 従来 OS と WebOS の比較

以上の事から本研究ではプラットフォームとして WebOS を開発する事とする。

2.1.2 ネットワーク側からの情報配信を可能とする

2 つ目の開発指針は情報配信の方向について方針を示している。先に述べた通り、アンピエント情報社会における情報システムはネットワーク側からも情報を提案出来るという事が必要条件になっている。本システムでも、ネットワーク側からの情報配信が出来る様に開発する。ただし 2.1.1 で述べた WebOS は通信プロトコルとして既存の HTTP を用いている。HTTP は基本的にユーザ(クライアント)側からネットワーク(サーバ)側にしかトリガを発信出来ない仕組みになっており、通常のサーバ・クライアント型 Web アプリケーションの実装方法ではネットワーク側からの情報配信は不可能になっている。そこで既存の HTTP を応用した Comet という技術を用いてネットワーク側からの情報配信を実現している。Comet の技術的側面に関する詳しい説明は 4.2.3 で述べる。

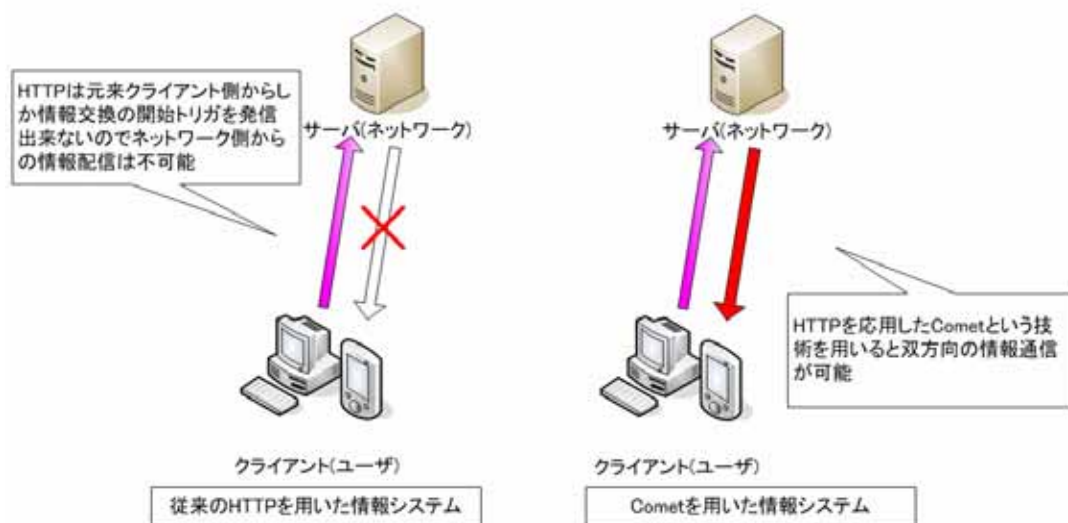


図 2.3 従来の情報システムと Comet を用いた情報システム

2.1.3 ユーザに最適な情報を提案し行動支援出来る環境を構築する

3 つ目の開発指針は本システムにおける最終的な目的を示している。本研究では先に述べた 2 つの開発指針に沿ってプラットフォームの機能を持ったユーザインターフェースを開発する。更にそのプラットフォーム上で動作するいくつかの Web アプリケーションを開発する。これらの Web アプリケーションと WebOS の関係が従来の OS とアプリケーションの関係になる。本研究では自動的に 4 つのパラメータ、時間情報・場所情報・端末情報・パーソナル情報をユーザに付加する。これらのパラメータによりユーザに最適な情報を最適なユーザインターフェースで配信し行動支援が出来る環境を構築するのが本研究の最終的な目的となる。

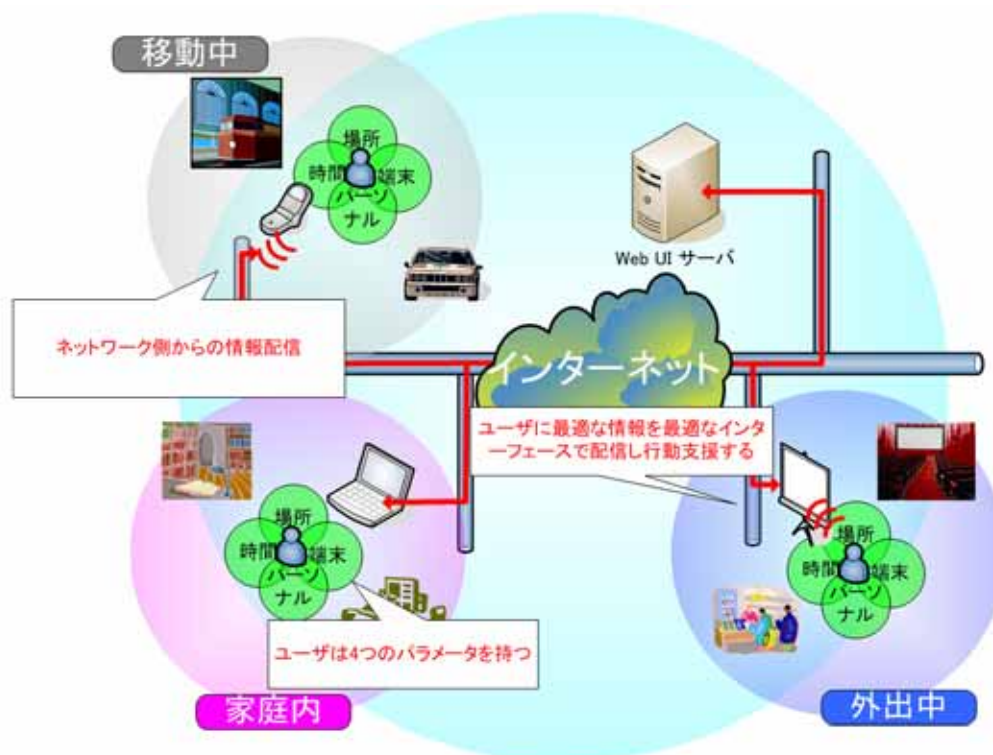


図 2.4 本研究で最終的に目指す情報システム

2.2 システム概要

2.2.1 システム構成

本システムは、インターネットを介したサーバ・クライアント型の一般的な Web アプリケーションの技術を用いて構築されている。ユーザは標準的なブラウザを使用すれば本システムの全てのサービスを受ける事が出来る。推奨ブラウザは Internet Explorer 6.0 SP2 と、主要キャリアのモバイル用ブラウザとなっている。ただし、通常の Web アプリケーションとはコンセプトが逆なので一般的な Web アプリケーションとはユーザインターフェースの様相がかなり違っている。ユーザに付加する 4 つのパラメータのうち端末情報に関してはモバイル端末・研究室の端末・家庭の端末・街中の端末という 4 つの値を取りうる設定にしている。街中の端末というのは東京電力[7]が渋谷区で提供しているタッチパネル方式の街中にある情報端末のようなものを示しており本稿では「街中端末」と呼ぶ。

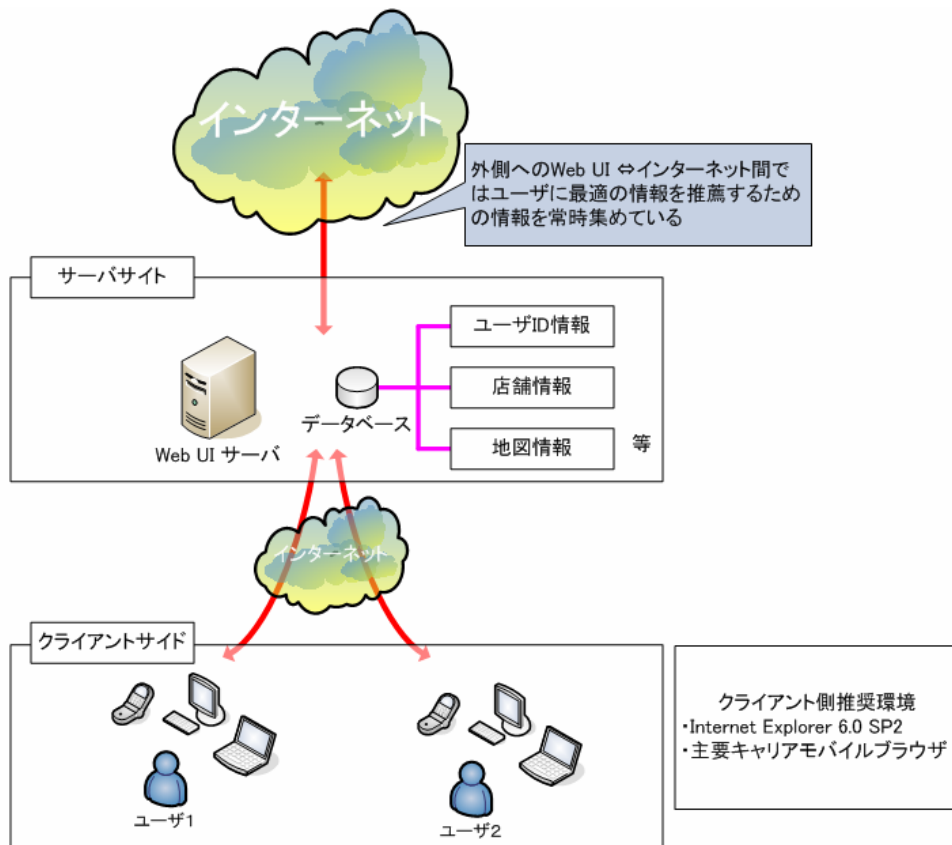


図 2.5 システム構成

2.2.2 WebOS(プラットフォーム部分)

この節では、プラットフォーム部分として作成した WebOS について簡単に説明する。本研究ではユーザインターフェースをユーザの適した形態に変化させ掲示するため、ユーザインターフェース自体が様々な形態をとる。またモバイル端末に関しては Web 技術を基盤とした表現能力がデスクトップパソコンに比べ貧弱なため WebOS の技術を使ってはいない。モバイル端末に関しては 2.2.4 にて説明する。また本研究ではユーザに 4 つのパラメータを自動で付加し、本システムを利用するとユーザに最適な情報を最適なユーザインターフェースで提案し行動支援が出来る環境を構築する事を目的としている。ユーザの状況によるユーザインターフェースの変化については 2.2.5 にて説明する。WebOS の更に詳細な機能については 3.1 にて説明する。WebOS の技術的な側面からの説明は 4.2.2 にて述べる。

最初にログイン画面から図 2.6 に示す。

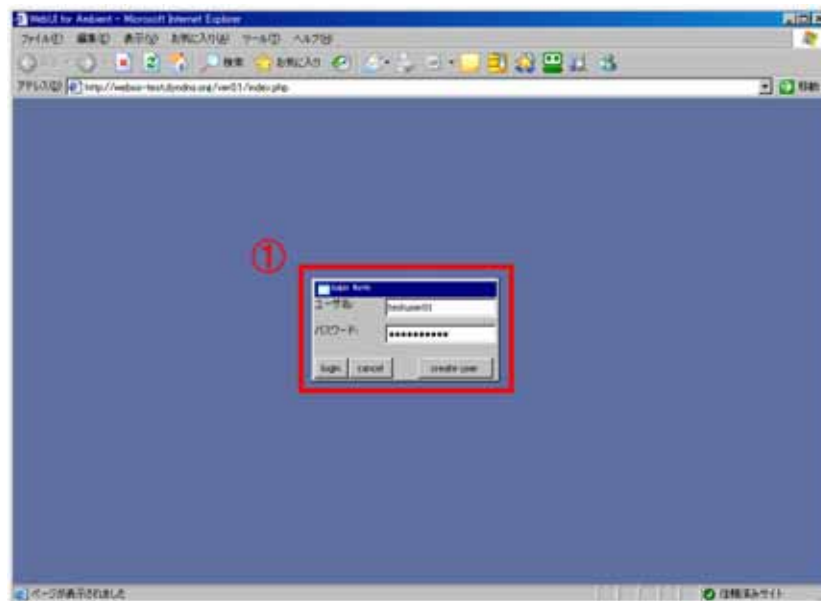


図 2.6 WebOS へのログイン画面

図 2.6 の がログインフォームになっている。ユーザ名、パスワードを入力して「login」を押すと WebOS のメイン画面に移動する事が出来る。次に図 2.7 に WebOS のメインとなる画面を示す。

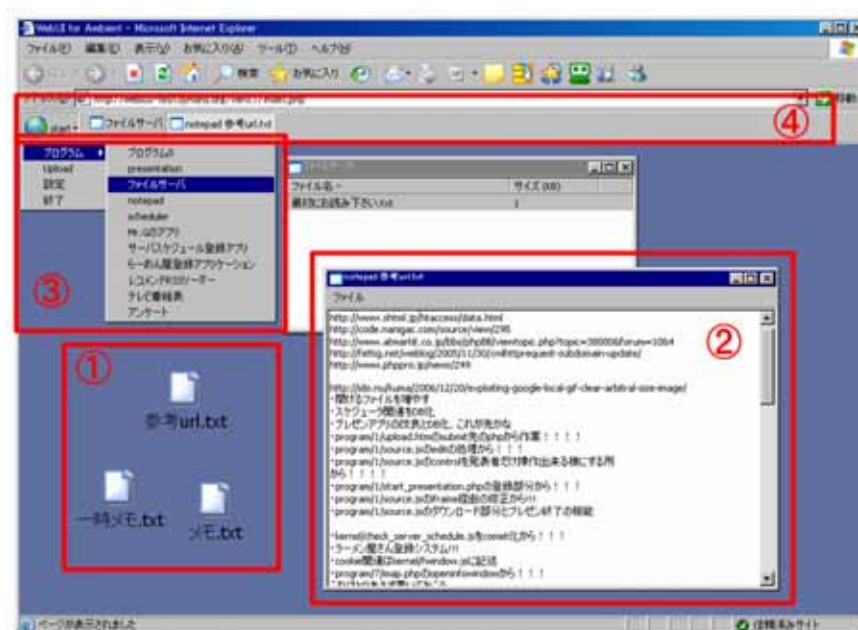


図 2.7 WebOS のメイン画面

見て頂ければわかると思うがローカルの GUI を用いた OS に非常に近いユーザインターフェースである。図中の はデスクトップ上のアイコンを示しておりこれらのアイコンを

使ってユーザは自身のリソースをインターネット上で管理・利用出来る。 はウィンドウである。これは Web アプリケーションの 1 つ「notepad」を実行しているが操作方法はローカルのそれとほとんど変わらない。 はメニューだ。ここから本システムのほぼ全ての機能を扱う事が出来る。 はタスクバーだ。起動している Web アプリケーションは全てここに表示されタスクバーから管理・利用する事が出来る。

2.2.3 Web アプリケーション

これらの Web アプリケーションが実際にユーザに最適な情報を掲示する機能を果たす。詳細な部分については 3.2 にて説明する。Web アプリケーションの技術的な側面からの説明は 4 にて述べる。表 2.1 に実装している全ての Web アプリケーションの表を示す。

Webアプリケーション名	機能
プレゼンテーション	オンライン上でプレゼンテーションを実行するWebアプリケーション
ファイルサーバ	オンライン上でファイル共有を行うWebアプリケーション
notepad	オンライン上でテキストファイルの編集を行うWebアプリケーション
スケジューラ	オンライン上でスケジュール管理を行うWebアプリケーション
周辺店舗探索マップ	オンライン上で様々な店舗情報を登録するWebアプリケーション
レコメンドRSSリーダー	Blogやニュースサイトの記事を読んだ時、ユーザが興味を持ったとしてユーザの嗜好データを抽出しユーザが興味を湧きそうな日記やニュースを推薦するレコメンドWebアプリケーション
テレビ番組表	上記レコメンドRSSリーダーから抽出したユーザのパーソナライズデータからユーザが興味を持ちそうなテレビ番組を推薦するレコメンドWebアプリケーション
街中案内システム	街中の情報端末を意識した、タッチパッド端末専用のWebアプリケーション

表 2.1 Web アプリケーション一覧

2.2.4 モバイル端末のプラットフォームと Web アプリケーション

モバイル産業は今や急速なスピードで進化しており、通話やメール、インターネットは当然の機能として、ワードやエクセル等のビジネスファイルを参照出来たりワンセグの導入によって複雑な処理も出来る様になってきた。

それらの流れを受けてインターネットに対するユーザインターフェースであるブラウザも進化を速め、現在はフルブラウザと呼ばれる HTML,CSS,ECMAScript に対応しプラグインの Adobe Flash Lite さえ使用可能なブラウザも搭載される様になってきた。

しかし、まだ通信速度の問題やコストの問題、メモリリソース等の問題から WebOS の様な既存の Web 技術を応用したよりインタラクティブなユーザインターフェースをモバイル端末に導入するのは時期早々と筆者は考え、本システムにおいてモバイル端末についてはデスクトップパソコンやノートパソコンといった端末とは違うプラットフォームを開発した。

本研究では全キャリア対応を考えユーザインターフェースの表現方法としては基礎的 HTML のみを用い小さな画面でも必要な情報を素早く提供出来る様にインターフェースを実装している。この節では簡単にモバイル端末に導入したプラットフォームと Web アプリケーションについて説明する。詳細については 3.3,3.4 で説明する。

図 2.8 にモバイル端末のプラットフォームと Web アプリケーションについての概略を示す。



図 2.8 モバイル端末のプラットフォームと Web アプリケーション

図 2.8 を見てもらえばわかるが一般的な携帯用の Web サイトの構成をしている。ログイン後のトップ画面では図中 にこのユーザ用にインストールされている Web アプリケーションの一覧が表示されている。図中 にはユーザの状況を判断して推薦される行動が表示される。 , どちらからでも Web アプリケーションが起動する様になっている。

次に表 2.2 に実装しているモバイル用の Web アプリケーションを示す。

Webアプリケーション名	機能
プレゼンテーション	WebOS版プレゼンテーションのモバイル版
スケジューラ	WebOS版スケジューラのモバイル版
周辺店舗探索マップ	WebOS版周辺店舗探索マップのモバイル版
レコメンドRSSリーダー	WebOS版レコメンドRSSリーダーのモバイル版
テレビ番組表	WebOS版テレビ番組表のモバイル版

表 2.2 モバイル端末の Web アプリケーション一覧

2.2.5 本システムにおけるユーザインターフェースの変化

この節では先に述べたユーザインターフェースが、ユーザの状況に応じてどの様に変化するのかについて簡単に説明する。詳細な説明は 3.5 で述べる。

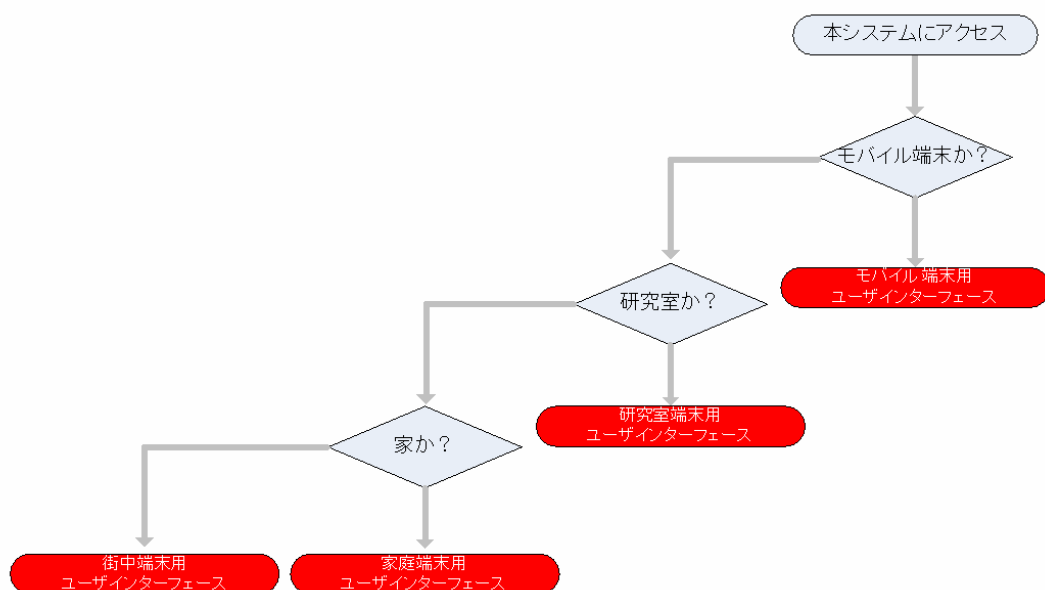


図 2.9 端末によるユーザインターフェース選択のフローチャート

図 2.9 で本システムに様々な端末でアクセスした際のデータ処理の流れを示した。

本システムはユーザの状況により図 2.9 の様な判断を行い、4 通りのユーザインターフェースの中からユーザに最適なものを選択し掲示する流れになっている。

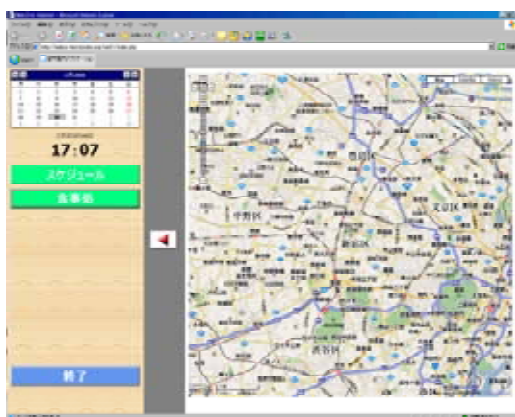


図 2.10 街中端末からアクセスした場合のユーザインターフェース

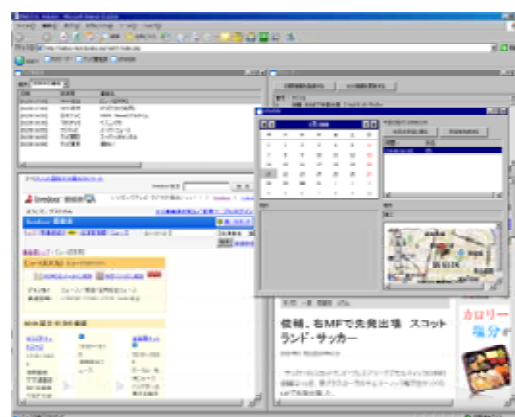


図 2.11 家庭用端末からアクセスした場合のユーザインターフェース

他にも「場所情報」「時間情報」「パーソナル情報」に基づいて最終的にユーザに最適な情報を最適なユーザインターフェースで掲示するのだが、詳細については 3.5 にて述べる。

3.システム詳細

この章では 2 で述べた、システムの詳細について述べる。最初にモバイル端末以外のプラットフォームとして開発した WebOS について実装している機能の詳細を説明する。次にその WebOS の上で動作する Web アプリケーションについて実装している機能の詳細を説明する。実際にユーザに情報を掲示するのはこれらの Web アプリケーションになるのでユーザがどのようなユーザインターフェースでどのような情報を得る事が出来るのかと言った事を中心に説明する。次に、モバイル端末用のプラットフォームと Web アプリケーションについて述べる。先にも述べたがモバイル端末用のプラットフォームは基本的な HTML で表現されているのでビジュアル的には WebOS に劣るが、モバイル端末では GPS を用いた「位置情報」を用いる事が出来るので「位置情報」に基づいた情報提供を中心に説明する。最後に本システムで開発した WebOS を中心としたプラットフォームとその上で動作する Web アプリケーションを連携させ、本システムを使用した場合のユーザインターフェースの形態の変化とユーザに提供する情報の変化について説明する。

3.1 WebOS(プラットフォーム)詳細

3.1.1 ログイン処理

モバイル端末以外で本システムのトップページ(webos.dyndns.org/ver0.1)にアクセスするとログイン画面に自動的に画面遷移する。家庭用端末と研究室端末に設定されている端末からアクセスした場合は図 3.1 の様にユーザ ID とパスワードを求められる。街中端末として設定されている端末からアクセスする場合、図 3.2 に所持している携帯電話で表示されている QR コードを読み取る画面に遷移する。



図 3.1 ID とパスワードによる認証



図 3.2 QR コードによる認証

図 3.1 において間違った ID とパスワード入力したり図 3.2 において過去に携帯で本システムにアクセスした事の無いユーザが QR コードを読み取ると図 3. の様にログインは失敗

する。

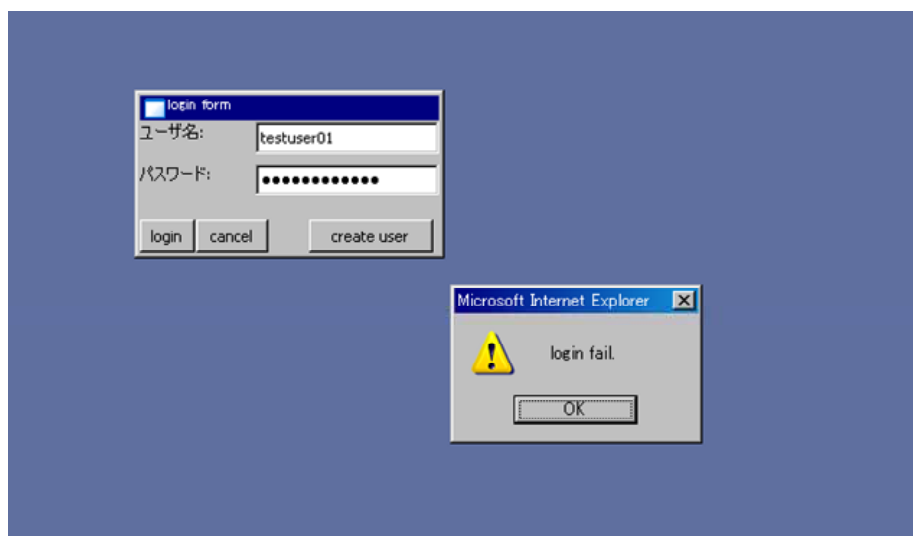


図 3.3 ログイン失敗

次に新規ユーザの作成について説明する。街中端末ではその場で新しいユーザを作るという事は想定していない。街中端末をそれだけ長時間占有する事は事実上難しい上、周囲に人が多いといった環境もセキュリティ上問題になると考えたからだ。図 3.4 の「create user」ボタンを押すとユーザ作成画面が立ち上がる。ユーザ名とパスワードを 2 度入力するだけでユーザは作られる様になっているが、当然ユーザ名が重複していたり、ユーザ名が空だとユーザの作成には失敗する。

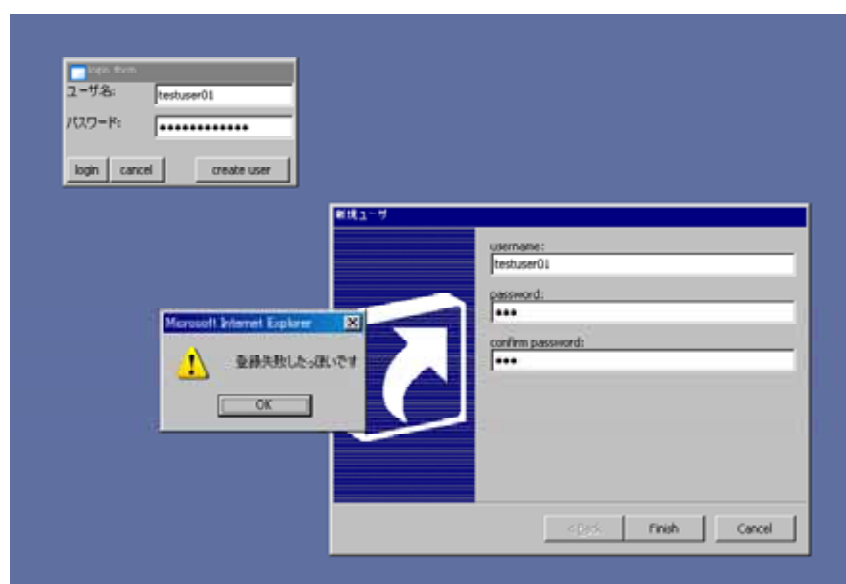


図 3.4 新規ユーザ作成画面と作成失敗

ユーザの作成やログインに成功すると図 2.7 の WebOS のメインユーザインターフェースの画面に遷移する。次からは WebOS の機能について説明をしていく。

3.1.2 デスクトップアイコン関連の機能

WebOS のメインのユーザインターフェースでは大きくわけて 2 つに分ける事が出来、上部のバーを「タスクバー」、下部の大部分を「デスクトップ」と筆者は呼んでいる。従来の OS におけるタスクバーとデスクトップとまったく同じ用途と考えて問題は無い。デスクトップ部分にはユーザが管理する事の出来るリソースとしてデスクトップアイコンがある。図 3.5 にデスクトップアイコンに対して起こせる操作を示す。

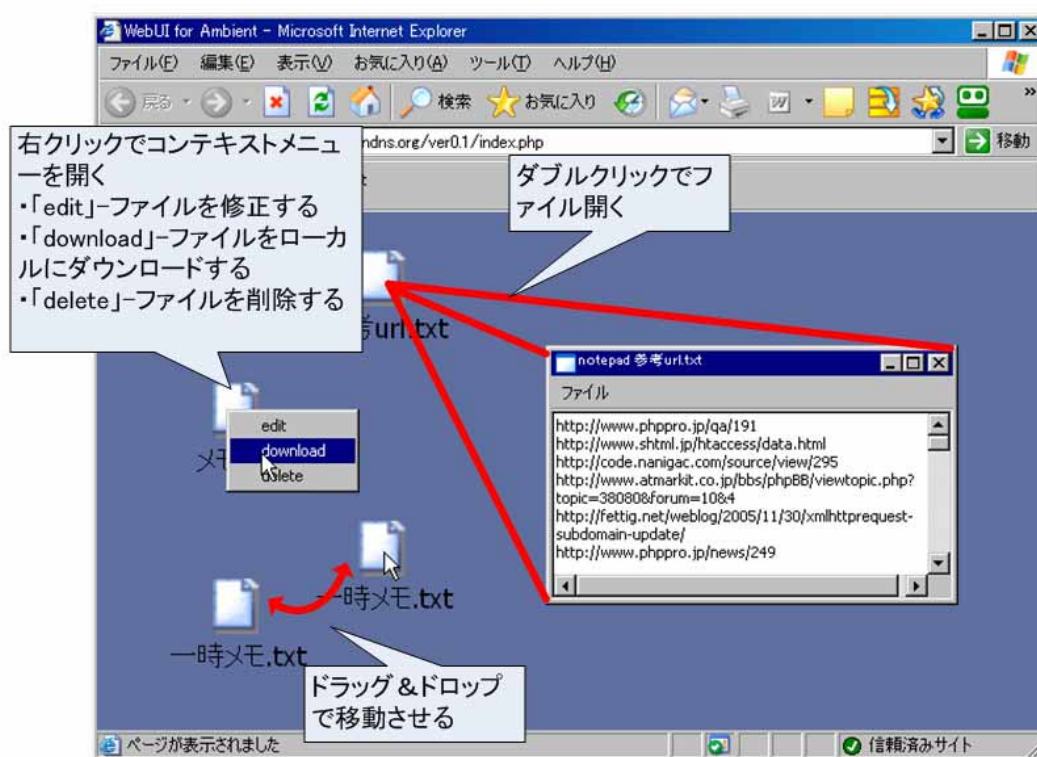


図 3.5 デスクトップアイコンに対して可能な操作

なお本システムで開く事の出来るファイルは以下の通りになっている。

- ・ xml ファイル
- ・ jpg ファイル
- ・ txt ファイル
- ・ ppt ファイル
- ・ doc ファイル
- ・ xls ファイル

基本的に従来の OS と同じ様な GUI を実装している。

3.1.3 スタートメニュー関連の機能

次にスタートメニューから操作する事の出来る機能を説明する。図 3.6 にスタートメニューから Web アプリケーションを起動する図を示す。

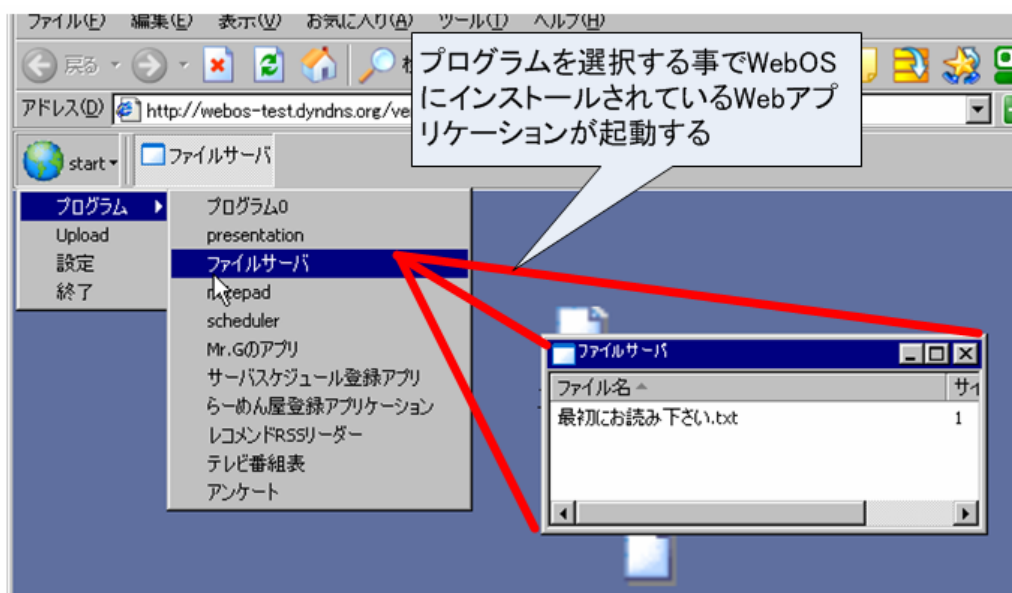


図 3.6 Web アプリケーションの起動方法

次に WebOS にファイルをアップロードする方法を図 3.7 に示す。

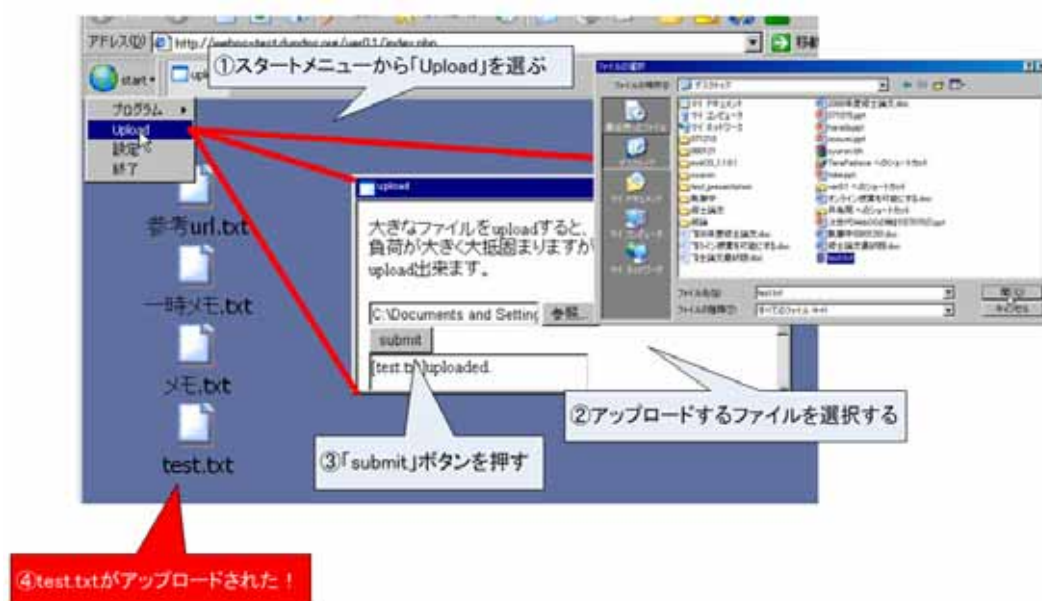


図 3.7 ファイルを WebOS にアップロードする方法

手順としては

スタートメニューから「Upload」を選ぶ

アップロード用の Web アプリケーションが起動するので「参照」ボタンを押しアップロードするファイルを選択する。

「submit」ボタンを押すと、アップロードが始まる。なおアップロード中でも他の作業は可能である。

アップロードが終わると WebOS のデスクトップにファイルが追加される。

この様に非常に簡単な操作でユーザはオンライン上でファイルの管理を行う事が出来る。

次に「設定」ボタンの動作について説明する。この「設定」ボタンは現在使っている端末がどこにあるという端末情報を付加するためのボタンである。その端末が街中端末であるかといった設定は非常に重要な作業なので通常のユーザは押す事が出来ない様になっている。押す事が出来るのはシステム開発者が管理者と認めたユーザだけであり、ユーザが管理者であるかどうかは「userdata.xml」ファイルに保存されている。

図 3.8 に「userdata.xml」ファイルと設定ボタンを押した時の動作を示す。

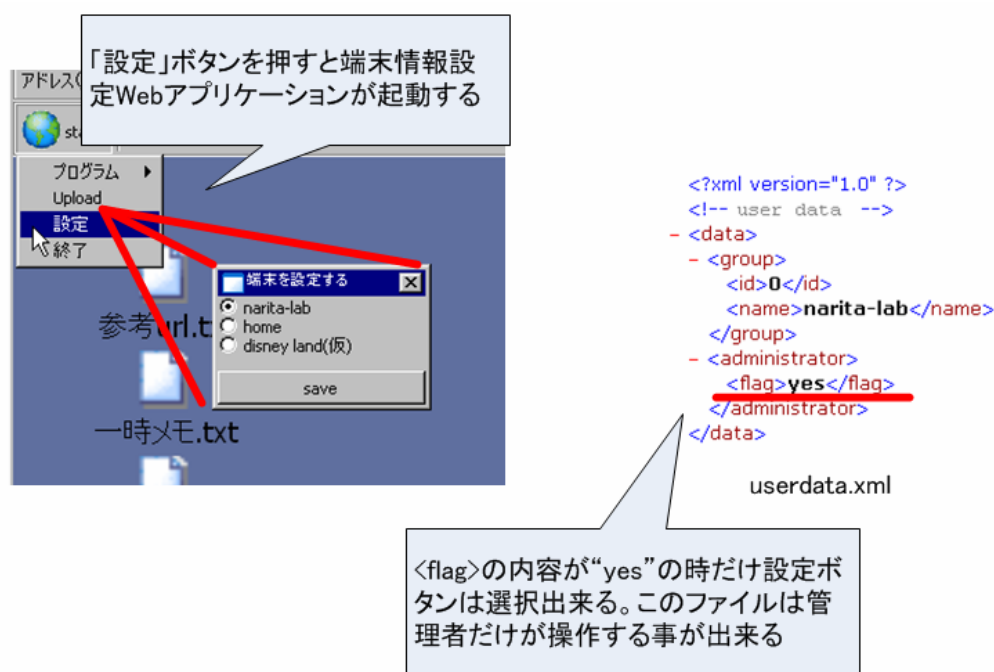


図 3.8 userdata.xml ファイルと設定ボタンの動作

ここで端末情報を設定する事により、今使っている端末の役割が家庭から研究室に移ったとしても対応する事が可能となっている。

3.1.4 タスクバー関連の機能

タスクバーに関する機能の説明をする。このタスクバーは複数のウィンドウを管理するために導入した。WebOS 上では Web アプリケーションが 1 つのウィンドウアプリケーションとして起動するので情報を効率良く管理しようとするとな必然的にタスクバーの存在が必要となった。図 3.9 にタスクバーを通して Web アプリケーションに対して行える操作を示す。

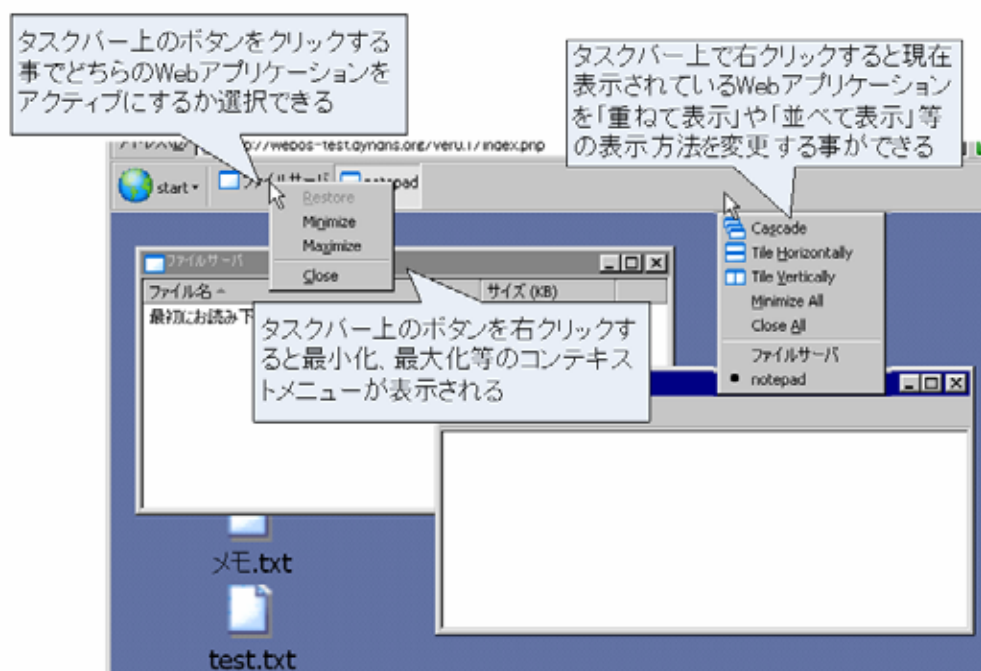


図 3.9 タスクバー上で Web アプリケーションに対して行える操作

タスクバー上での操作によって Web アプリケーションをより効率的に扱う事が出来るようになった。

3.2 Web アプリケーション詳細

図 2.1 において WebOS 上で実装している Web アプリケーションの説明を行ったがここではより詳しい詳細を説明していく。

3.2.1 プレゼンテーションアプリ

プレゼンテーションと言う名前の通り、この Web アプリケーションはオンライン上でプレゼンテーションを行うための機能を持っている。オンライン上でプレゼンテーションを行おうと考えた場合、専用のアプリケーションをインストールする必要があった。またネットワーク越しでの会議ではプレゼンテーションファイルのやり取りに大きな弊害が生じていたそれらの問題を解決するためにこの Web アプリケーションを WebOS に実装した。4.2.2 で後述するがこの Web アプリケーションでは Comet という新しい概念の技術を用い

ている。先に述べた通り HTTP という仕組みは基本的にネットワーク側から情報配信が来ない仕組みである。オンラインでプレゼンテーションを行おうと考えた場合クライアントから情報交換のトリガを発生させるわけでは無いので、従来の HTTP 通信では非常に実現が困難であった。それでも実装しようとした場合は Flash を導入するなど専用のプラグインが必要となる。ここでは詳しく述べないが Comet は従来の HTTP 通信を応用した考え方であり、特別なプラグインのインストール等は必要ない。図 3.10 にプレゼンテーションアプリを用いてプレゼンテーションを登録する方法を説明する。



図 3.10 プレゼンテーションを登録する方法

プレゼンテーションアプリは登録する機能に関しては図 3.10 の様に 2 つのウィンドウを用いて実現している。プレゼンテーションを登録する手順に沿って説明する。

プログラムよりプレゼンテーションアプリを起動する。アプリケーションが起動すると図中左側のメインインターフェースだけが立ち上がる。このメインインターフェースはプレゼンテーションの登録から実施まで全てを司っている。

次にプレゼンテーションを予定している日を選択し「予定を作成する」ボタンを押す。ボタンを押すと図中右側のプレゼンテーションファイルアップロードインターフェースが立ち上がる。これは名前の通り、プレゼンテーションファイルをアップロードするためのインターフェースだ。

ここでアップロードするプレゼンテーションファイルを指定するのだが、この Web アプリケーションでは PowerPoint ファイルを web ページとして保存した場合のみプレゼンテーション実行時に発表者とオーディエンスの間で同期が取れるという制限がある。プレゼンテーションファイルをそのままアップロードしても良いので発表者にどちらのアップロード方法を選ぶか選択させるようにしている。

最後に「submit」ボタンを押してプレゼンテーションファイルをアップロードしプレゼンテーションの予定を登録する。

次に登録されたプレゼンテーションの予定からプレゼンテーションを実施する機能について図 3.11 に示す。

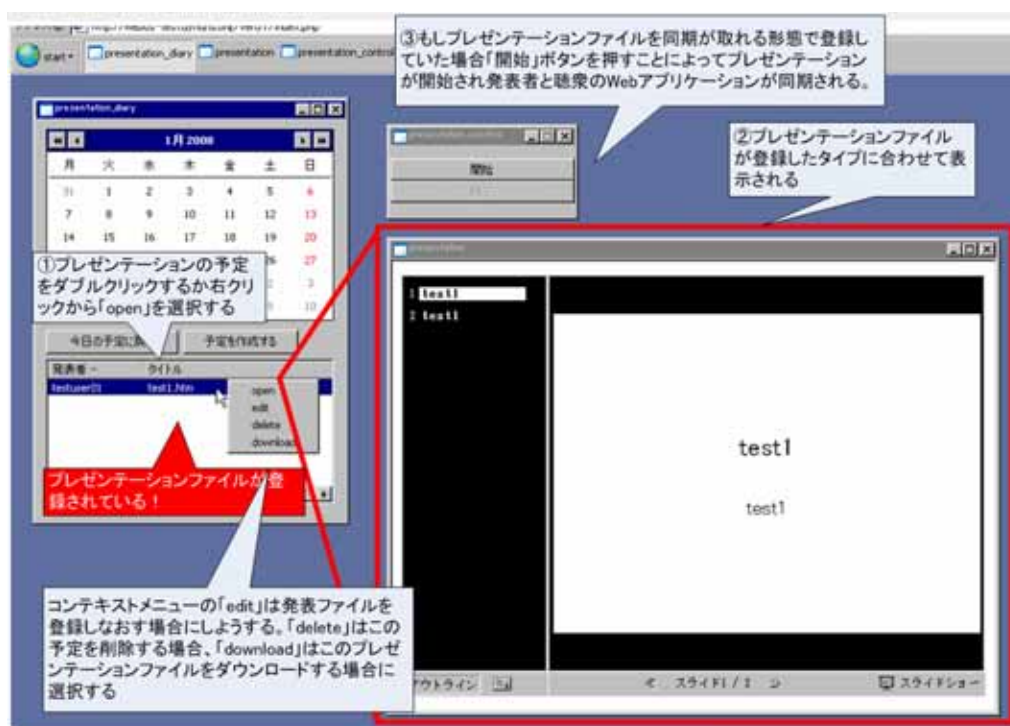


図 3.11 プレゼンテーションの実施方法

図 3.10 と比較するとメインインターフェースにプレゼンテーションが登録されているのがわかる。この予定の上でダブルクリックもしくはコンテキストメニューから「open」を選択するとプレゼンテーションファイルが右側に開かれる。この時 PowerPoint ファイルを web ページとして保存していた場合にはこの Web アプリケーションを介して発表者と聴衆の同期が取られるのでよりインタラクティブなプレゼンテーションが実施可能になる。

3.2.2 ファイルサーバアプリ

この Web アプリケーションはどちらかというと WebOS のプラットフォームとしての機能に近いかもしれない。名前の通り、WebOS 上にファイルサーバを建て、そこで本システムを使うユーザ同士でファイルの交換が出来る Web アプリケーションだ。ファイルサーバアプリの機能について図 3.12 に示す。

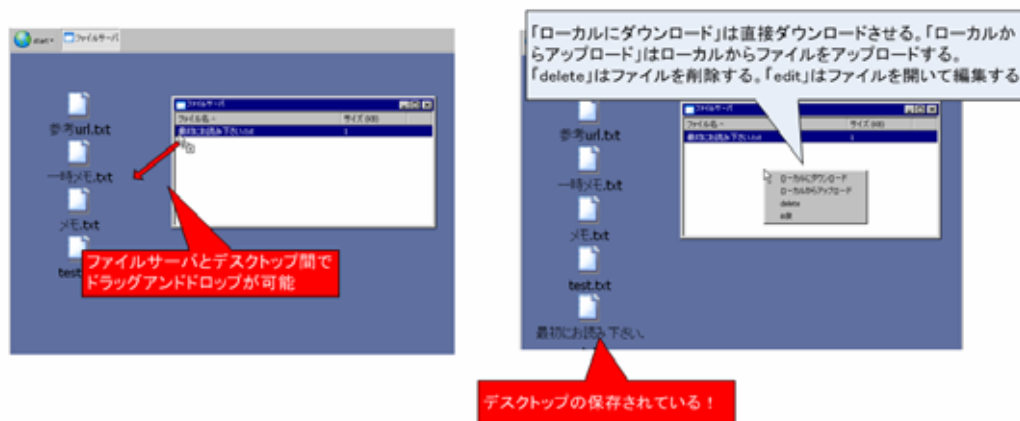


図 3.12 ファイルサーバアプリの機能

この Web アプリケーションは従来の一般的なファイルサーバと大きな違いは無い。デスクトップとファイルサーバ Web アプリケーション間ではドラッグ&ドロップによるデータの交換が可能となっている。

3.2.3 notepad アプリ

このアプリケーションをオンライン上でテキストファイルの編集を行うための Web アプリケーションだ。特別な特徴があるわけでは無いが、ドラッグ&ドロップによるデータの読み込みやコピー & ペースト等最低限のテキストエディタとしての機能を実装している。図 3.13 に notepad アプリの機能を示す。

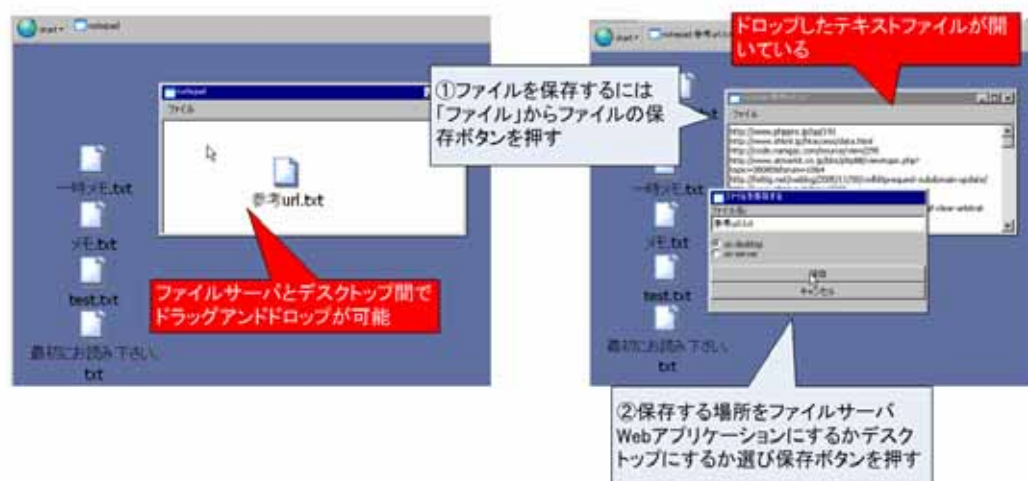


図 3.13 notepad アプリの機能

notepad アプリでテキストを保存するには、ファイルメニューから「ファイルを保存」ボタンを押し保存する場所を on desktop(WebOS のデスクトップ上)か on Server(ファイルサ

ーバ Web アプリケーション上)に保存するか選び保存ボタンを押せば保存される。

3.2.4 スケジューラアプリ

スケジューラアプリはオンライン上でユーザのスケジュールを管理するための Web アプリケーションである。ユーザのスケジュール情報からどのような情報をユーザに掲示すればいいのか推定出来る様になるため本システムにおいては非常に重要な Web アプリケーションとなっている。まずスケジューラアプリにおけるユーザインターフェースを図 3.14 に示す。

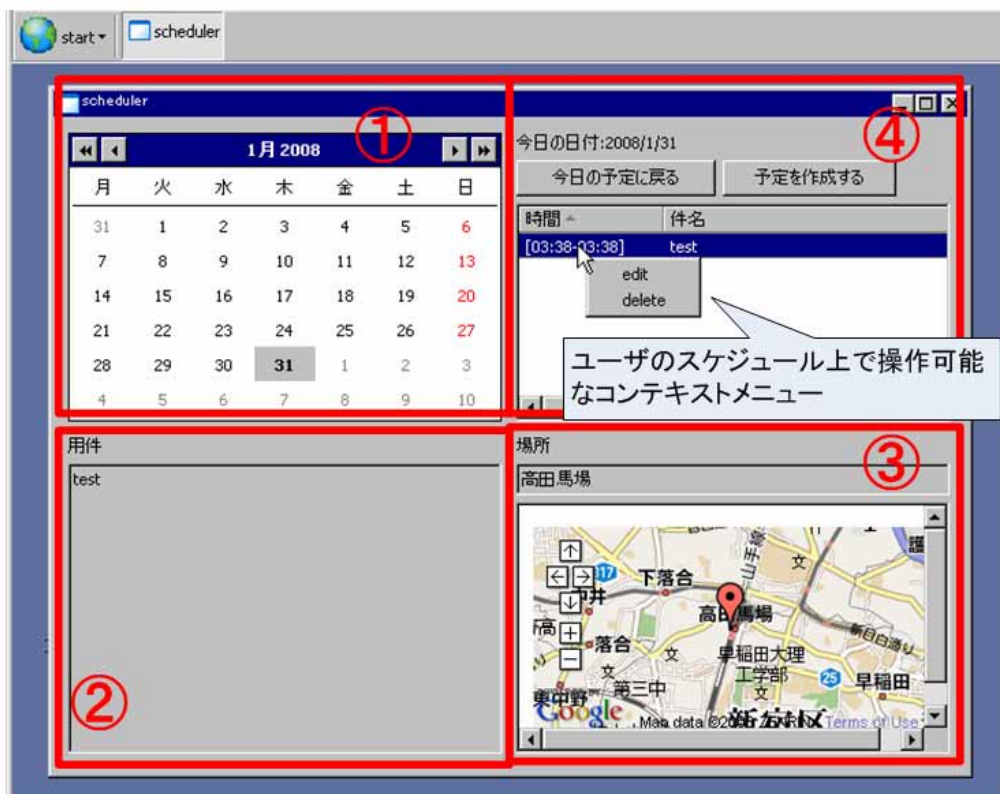


図 3.14 スケジューラアプリにおけるユーザインターフェース

スケジューラアプリは図 3.14 の様に 4 つのブロックから成り立っている。

は見てわかる様にカレンダー機能だ。スケジュールを管理するならば必須の機能だ。

と はそのスケジュールの用件と場所を掲示する部分になっている。なお地図情報に関しては後述する GoogleMapsAPI を使用してユーザに掲示している。 がスケジューラ Web アプリケーションの中核となる部分だ。基本的にスケジュールを登録したり、削除したりといったユーザが取りうる操作は全て の部分で行われる。

次にスケジューラアプリを用いて新しいスケジュールを登録する機能を図 3.15 に示す。



図 3.15 新しいスケジュールを登録する機能

スケジュールを作成する流れに沿って説明する。

まずカレンダーより予定を作成したい日を選択する。

次に「予定を作成する」ボタンを押す。

ここで予定作成用の新しいウィンドウが開くので、開始時刻・終了時刻・件名・用件・場所についてそれぞれ入力する。なお場所に関しては後述する GoogleMapsAPI を利用する事で住所からスケジュールに設定する地点の地図を作成する事が出来る様になっている。

最後に「予定を作成」ボタンを押すとスケジュールが本システムに登録される。

また、図 3.15 の様に登録されているスケジュール上で右クリックを行う事によりそのスケジュールを削除したり編集したりする事が可能になっている。

3.2.5 周辺店舗探索マップアプリ

周辺店舗探索マップは、本システムを使用するユーザが様々な店舗の情報を共有するために用いる Web アプリケーションだ。図 3.16 に周辺店舗探索マップアプリのユーザインターフェースを示す。



図 3.16 周辺探索マップアプリの
ユーザインターフェース



図 3.17 店舗名を入力して検索した場合

周辺店舗探索マップアプリは 3 つの部分から構成されている。地図部分と 店舗情報の表示部分、 地図情報の検索部分だ。地図については後述する Google Maps を利用しており、自由にスクロールや拡大、縮小が出来る。店舗情報には、「店名」「住所」「写真」「紹介文」という項目がある。地図情報の検索部分ではフォームに住所か店舗名を入力して検索ボタンを押すと図 3.17 の様に検索結果に基づいて地図が移動する。

次に店舗を地図に登録する流れについて図 3.18 に示す。



図 3.18 登録の流れ

図 3.19 登録結果

図 3.18 に沿って手順を説明する。

店舗情報を登録したいユーザは登録したい地点で右クリックをして「ここを指定」ボタンを押す。すると編集中心を示す白アイコンが地図上に表示される。

手順 2 で表示させた白アイコンを選択すると、店舗情報を入力するフォームが地図上に表示される。必要な情報「店舗名」「住所」「写真ファイル」「紹介文」を入力する。

最後に「submit」ボタンを押すと地図に店舗情報が登録される。

登録結果を図 3.19 に示す。

3.2.6 レコメンド RSS リーダーアプリ

RSS とはニュースサイトやブログサイトの最新情報を効率的に配信するための XML ベースの記述仕様だ。RSS リーダーとはこの RSS ファイルを読み込みユーザに掲示するアプリケーションの事だ。図 3.20 に一般的な RSS リーダーの概略を示す。



図 3.20 RSS リーダーの概略

本研究ではユーザがどのような事に興味を持っているのかというパーソナルデータを取得するために、この RSS リーダを使ってユーザにニュース等を掲示し、もし閲覧した場合はユーザがその記事の内容に興味を持っているとして利用している。レコメンドという名前が Web アプリケーションの前についているのは、ユーザから取得したパーソナルデータに基づいてユーザに最適な情報を表示する様に RSS リーダーが動作するからだ。どの様にマッチングしているかについては 4.2.5 にて詳細を述べる。図 3.21 にレコメンド RSS リーダーのインターフェースを示す。



図 3.21 レコメンド RSS リーダーアプリのインターフェース

レコメンド RSS リーダーアプリは 2 つの部分から成り立っている。図 3. の に示すタイトル表示部分と の本文表示部分だ。 に記事が表示される順番はユーザーに興味に近いものから表示される様になっている。Web アプリケーションの上部にある「初期情報を登録」というボタンでは静的にユーザーの興味を登録する事も出来る。

なお、この RSS 情報は以下のサイトより取得している。

- ・ "http://rss.asahi.com/f/asahi_newsheadlines" 朝日新聞ヘッドライン
- ・ "http://pheedo.nikkeibp.co.jp/f/nikkeibp_news_flash" Nikkei BPnet 最新ニュース
- ・ "http://a.feed.yahoo.co.jp/rss2/spnv/soccer.xml" スポーツナビ - サッカー
- ・ "http://a.feed.yahoo.co.jp/rss2/spnv/baseball.xml" スポーツナビ - 野球
- ・ "http://www.plus-blog.sportsnavi.com/feed/all/rss2_0.xml" スポーツナビ - ブログ
- ・ "http://headline.2ch.net/bbynews/news.rss" 2ちゃんねるヘッドライン

3.2.7 テレビ番組表アプリ

テレビ番組表アプリはその名前の通り、インターネット上が電子テレビ番組表を取得しユーザーに掲示する Web アプリケーションだ。図 3.22 にテレビ番組表アプリを掲示する。

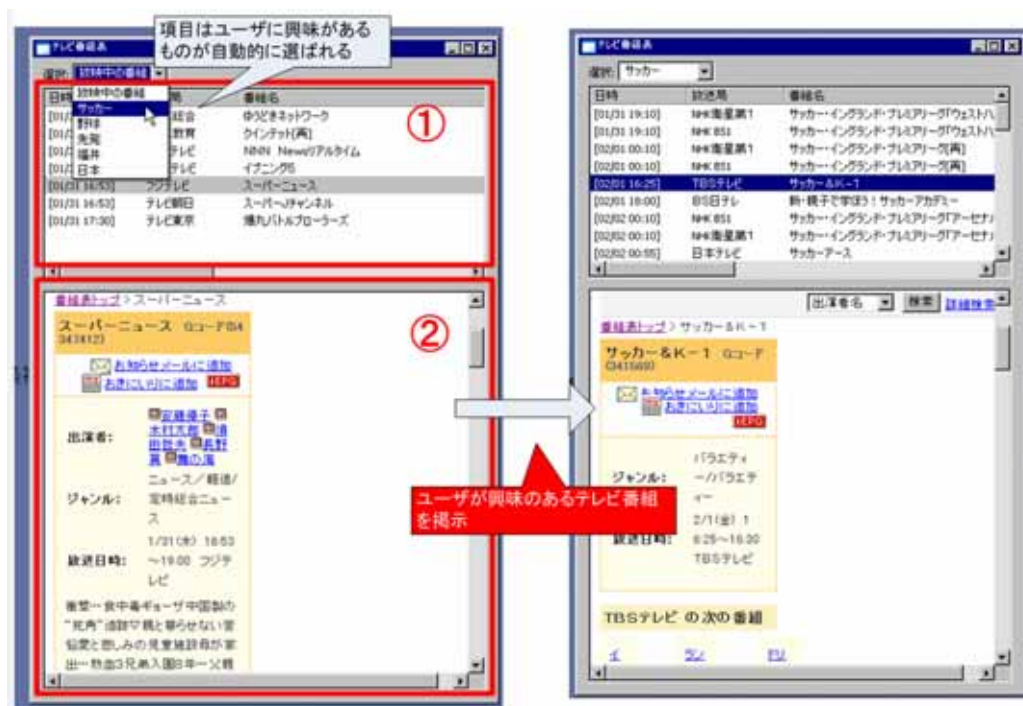


図 3.22 テレビ番組表アプリのインターフェース 図 3.23 ユーザーが選択する項目を変えた場合

テレビ番組表アプリは図 3.22 に示す様に タイトル表示部と 本文表示部の 2 つから成り立っている。Web アプリケーションの上部に「選択」というプルダウンメニューがありここで表示するジャンルを変更する事が出来る。ここに表示される項目は 3.2.6 で述べたレコメンド RSS リーダーアプリを使用した事により抽出されたユーザーのパーソナルデータに基づいて選択される。

なお、テレビ番組データは livedoor 社から配信されているものを利用している。

3.2.7 街中案内システム

街中案内システムは街中端末用 Web アプリケーションとして開発したものだ。その為街中端末と設定されている以外の端末からは立ち上げる事が出来ない。図 3.24 に街中案内システムのインターフェースについて示す。



図 3.24 街中案内システム

このWebアプリケーションは立ち上げると自動的に全画面に拡大図3.24の様な表示になる。本システムは開発中でまだ画面構成しか出来ていないのだが にログインしたユーザが認したい情報の入力欄が に で入力した情報の地図情報が表示される様な仕組みにすつもりである。

3.3 モバイル端末用プラットフォーム詳細

3.3.1 ログイン処理

モバイル端末で本システム(webos-test.dyndns.org/ver0.1)にアクセスすると、ID とパスワードを入力する画面に遷移する。ログインに成功すると位置情報取得画面に遷移し、ユーザの位置情報を取得する。次にトップ画面に遷移する。図 3.25 にログイン処理についての概要を示す。

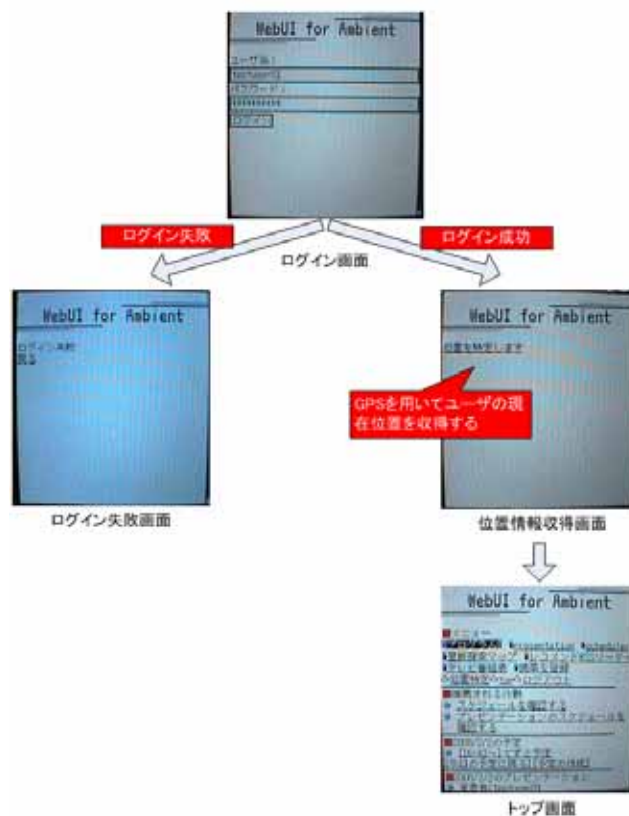


図 3.25 モバイル端末用プラットフォームのログイン処理概要

3.3.2 トップ画面

モバイル端末用プラットフォームのトップ画面からの操作を図 3.25 に示す。

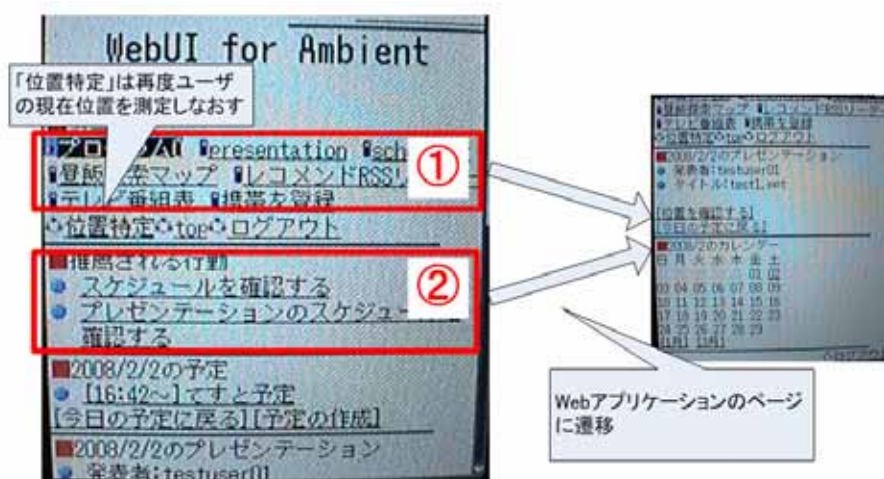


図 3.26 モバイル端末プラットフォームのトップ画面からの操作

図中 にメニューがあり、ユーザが起動出来る Web アプリケーションが並んでいる。起

動出来る Web アプリケーションは WebOS と連動しておりなるべく同じ情報にアクセス出来る様設計されている。メニュー下段の「位置特定」ではユーザの現在位置を再度測定する事が出来る。AU の端末に限れば、定期的にユーザの現在位置を測定出来るので今後自動化する方向に持っていく予定である。メニューの下部には図中の様にユーザの状況に合わせて推薦される行動が示される。メニューか推薦される行動、どちらからでも Web アプリケーションを起動する事が出来る。

3.4 モバイル端末用 Web アプリケーション詳細

先にも述べた様に WebOS とモバイル端末用プラットフォームでなるべく同じ情報にアクセス出来る様に設計している。モバイル端末用プラットフォームに実装した Web アプリケーションもほとんどが WebOS 用の Web アプリケーションを移植したものである。この節では WebOS 用 Web アプリケーションとは異なる部分について説明する。

3.4.1 プレゼンテーションアプリ

モバイル端末に実装したプレゼンテーションアプリは、予定されているプレゼンテーションの場所、タイトル、場所をユーザに掲示する Web アプリケーションである。図 3.27 にプレゼンテーションアプリの概要を示す。

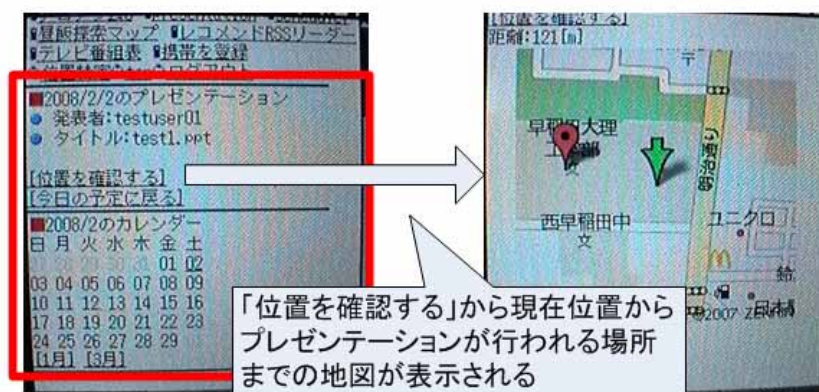


図 3.27 プレゼンテーションアプリの概要

プレゼンテーションアプリは図中左の様に「発表者」・「タイトル」の表示部とカレンダー表示部から成り立っておりカレンダー表示部から他の発表日のプレゼンテーションについても確認出来る。「位置を確認する」というリンクをクリックすると現在地からプレゼンテーションが行われる場所までの地図が表示される。

3.4.2 スケジューラアプリ

モバイル端末に実装したスケジューラアプリはほぼ WebOS 版と同じだが、地図が現在地

から登録されている場所までの地図になる点と、予定登録時に現在地で登録出来る点異なる。図 3.28 にスケジューラアプリで予定を確認する場合の概要を示す。

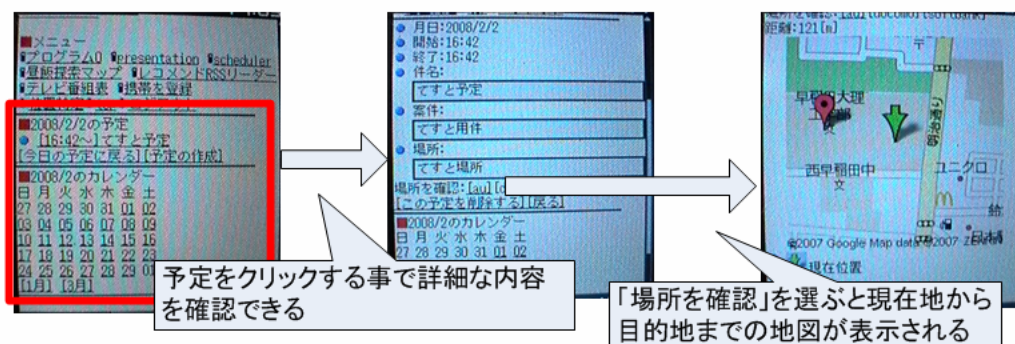


図 3.28 スケジューラアプリで予定を確認する場合

スケジューラアプリは図 3.28 左図の様に予定表示部とカレンダー部から構成されている。カレンダー部で月日を選択する事で指定日の予定が確認出来る。予定表示部に表示されている項目を選択するとより詳細な「開始・終了時刻」「場所」「内容」といった事が確認出来る。(図 3.28 真ん中)詳細な予定表示部の「場所を確認」を選択すると図 3.28 右図の様に現在地から目的地までの地図が表示される。

次に予定を作成する場合について説明する。スケジューラを用いて予定を作成する概要を図 3.29 に示す。

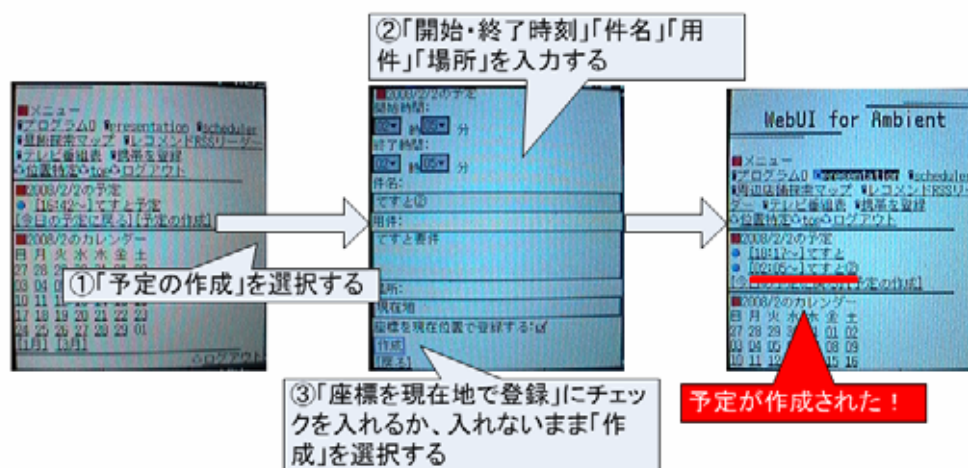


図 3.29 スケジューラアプリで予定を作成する場合

予定を作成する場合の流れについて図 3.29 に沿って説明する

最初にスケジューラアプリを起動した画面の「予定の作成」を選択する。

予定を詳細を入力する画面に遷移するので、作成する予定について「開始・終了時刻」「件名」「用件」「場所」を入力する。次に「座標を現在地で登録」にチェックを入れるか入れ

ないかを選択する。チェックを入れた場合、現在地の予定として登録される事になる。最後に「作成」を選択する。

3.4.3 周辺店舗探索マップアプリ

モバイル端末用プラットフォームではJavascript を使用していないので WebOS 版周辺店舗マップアプリ程、インタラクティブな操作を行う事は出来ない。そこで表示するマップの大きさに応じて現在地から近い順番に周辺店舗についての情報を掲示する様にした。図 3.30 に周辺店舗探索マップアプリの概要を示す。

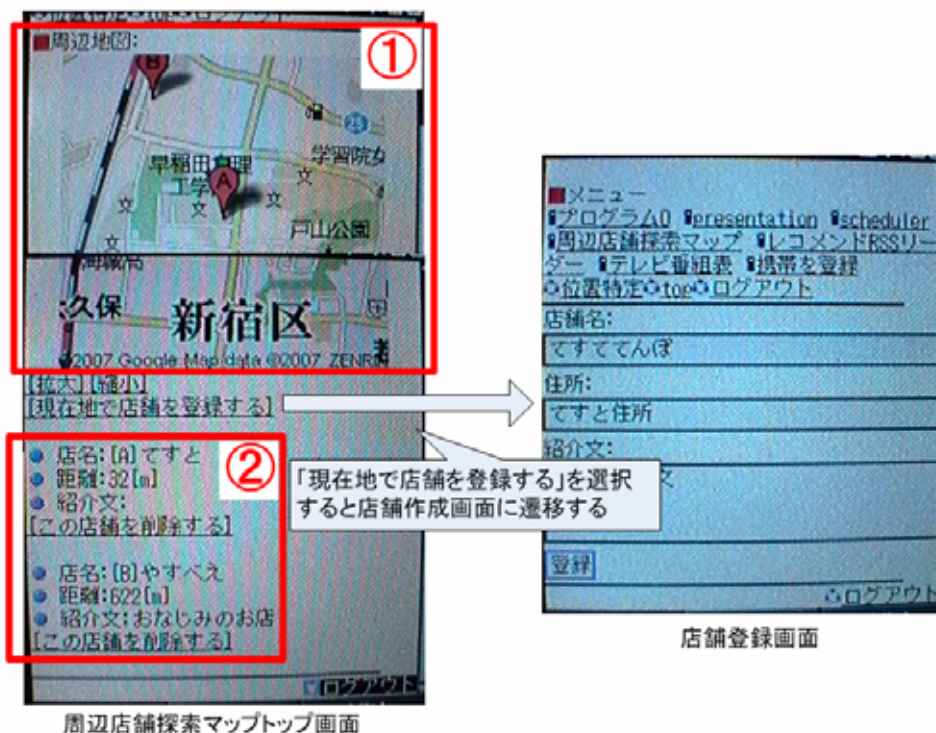


図 3.30 周辺店舗探索マップアプリの概要

携帯版周辺店舗探索マップアプリは図中左の様に周辺地図表示部 と周辺店舗表示部の 2 つから構成されている。現在地を中心として表示されている地図の範囲にある店舗を近い順にユーザに掲示している。新たに店舗を登録するには、画面の中段にある「現在地で店舗を登録する」を選択すると店舗登録画面に遷移する。

3.5 本システムにおけるユーザインターフェースの変化詳細

この節では、ユーザの状況によって本システムのユーザインターフェースがどの様に変

化するののかについて詳しく説明する。2.2.5 にて端末情報によりユーザインターフェースが「モバイル端末用ユーザインターフェース」「研究室端末用ユーザインターフェース」「家庭端末用ユーザインターフェース」「街中端末用ユーザインターフェース」に変化する事を述べた。それぞれの変化について詳しく説明する。本研究ではモバイル端末用ユーザインターフェース以外は最初に起動する Web アプリケーションによってユーザインターフェースの変化としているので、実際には起動する Web アプリケーションについての説明となる。

3.5.1 モバイル端末用ユーザインターフェースの変化

モバイル端末用ユーザインターフェースでは 3.3.2 で述べた様にトップ画面の「推薦される行動」によってユーザに最適な情報を掲示する設計になっている。図 3.31 にユーザの状況に合わせてユーザインターフェースが変化する概要について示す。

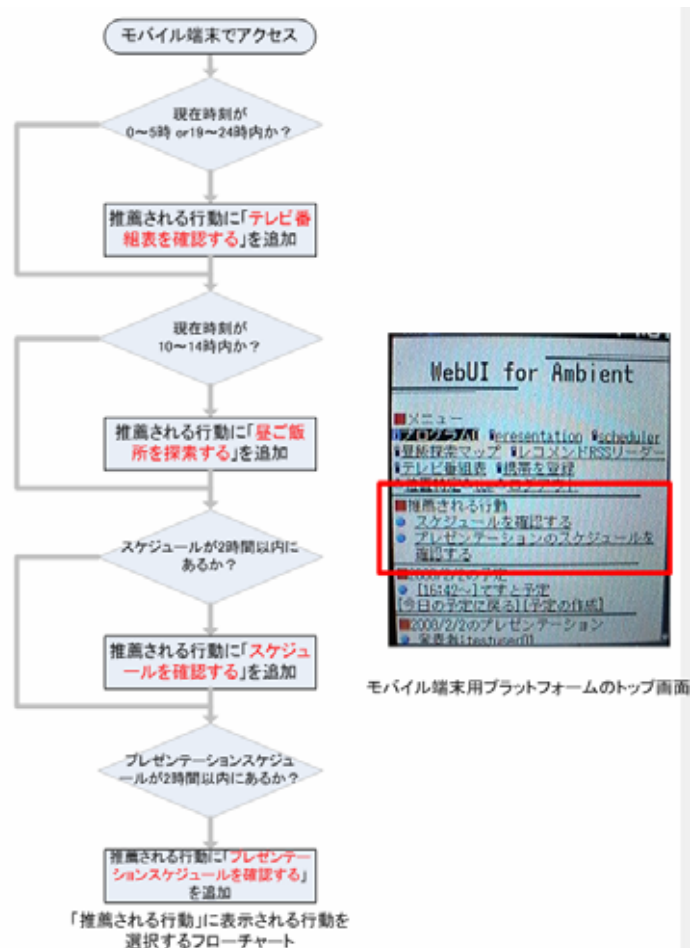


図 3.31 モバイル端末用ユーザインターフェースの変化の概要

モバイル端末用ユーザインターフェースの変化は図中左のルールに従って変化する。この事により例えばユーザがお昼休みの間にお昼ご飯をどこかで食べたいと考えている時に本システムにアクセスすると周辺店舗探索マップアプリの起動が推薦させるといった様な

使い方を想定している。

3.5.2 WebOS のユーザインターフェースの変化

本システムではモバイル端末用ユーザインターフェース以外は WebOS 上での変化となる。ここでは研究室端末用ユーザインターフェース」「家庭端末用ユーザインターフェース」「街中端末用ユーザインターフェース」について述べる。図 3.32 に WebOS 上でのユーザインターフェースの変化についてのフローチャートを示す。

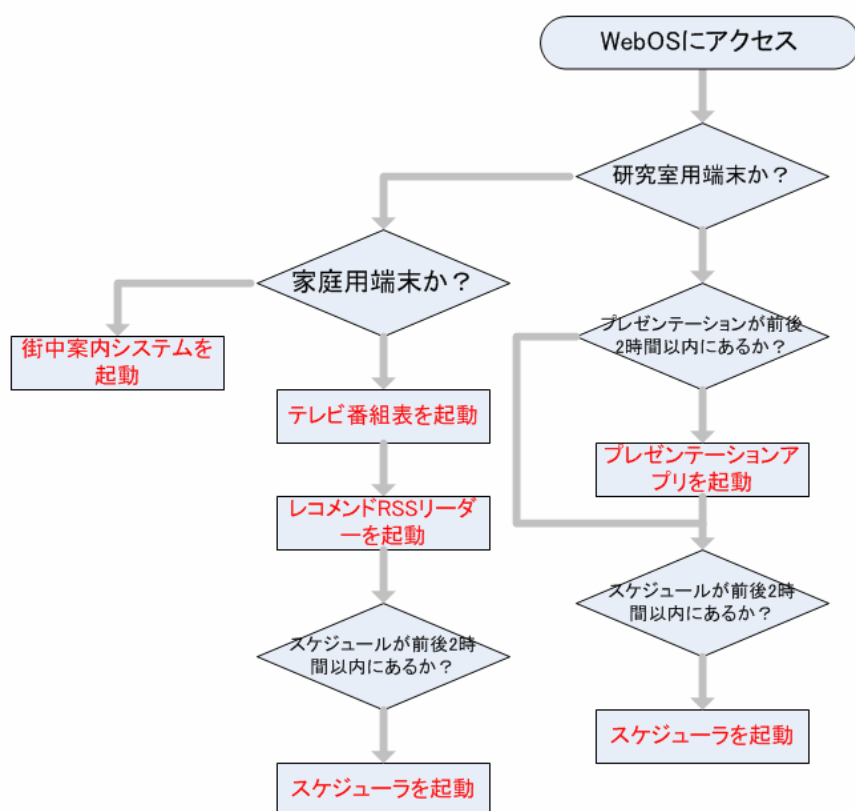


図 3.32 WebOS 上のユーザインターフェースの変化のフローチャート

図 3.33 にあるユーザがそれぞれの端末で WebOS にアクセスした場合の起動後のユーザインターフェースを示す。



図 3.33 WebOS に各端末でアクセスした場合のユーザインターフェース

図 3.32 を見てもらえれば図 3.31 で示したルールに従ってユーザインターフェースが変化している事がわかる。この事により例えば帰宅した後にテレビを見たいと考えているユーザが本システムにアクセスすると自動でテレビ番組表から情報を推薦出来るといった事が可能になっている。

4.システムの開発

2 から 3 にかけて、主にユーザの面から見た本システムの動作について述べた。この章では開発者の側から技術的な詳細に説明する。最初に開発環境について述べる。次に本システムを開発する上での技術的な側面について開発した WebOS や Web アプリケーションと関連付けをしながら述べる。

4.1 開発環境

開発環境について、表 4.1 に示す。

CPU	Intel Core2 6600 2.4GHz
メモリ	2GB
OS	Microsoft Windows XP Professional
サーバサイドスクリプト開発言語	PHP 5.2.4
Webサーバー	Apache 2.2.4
データベース	MySQL 5.0.45
クライアントサイド開発言語	Javascript,HTML

表 4.1 開発環境

一般的な Web アプリケーション開発環境を用いて開発を行った。

4.2 本システム開発にあたっての技術的側面の詳細

この節では、本システム開発にあたっての技術的側面について説明する。3 での説明順に沿って説明する。

4.2.1 Ajax について

Ajax とは Asynchronous JavaScript + XML を省略したものである。本システムでは主に WebOS 部分の開発に大きく関係しているものである。Ajax という概念が無い時代の Web アプリケーションユーザ側の操作が「開始 終了 開始 終了」と断続的になるのが当たり前だった。これは従来の Web アプリケーションのモデルが次の様なものだったからである。インターフェースを通じたユーザのアクションは最初にサーバへ HTTP リクエストを発生させる。Web サーバはデータの取得、数値計算、他データベースとのやり取り等の処理を行い、クライアントに対して HTML ファイルを返す。これはハイパーテキストを介する Web 本来の使い方だ。このモデルの問題はユーザが HTML ファイルが戻ってくるまでインターフェース側で一切の作業が出来ないのでユーザの操作が断続的に途切れてしまう

点だ。Ajax はクライアント・サーバ間の通信を Javascript によって作成される Ajax エンジンと呼ばれるものに仲介させる事によってユーザがサーバからデータが戻って来てなくてもユーザが操作を継続して行いう事を可能とした。図 4.2 に従来の Web アプリケーションと Ajax Web アプリケーションのモデルの違いを示す。

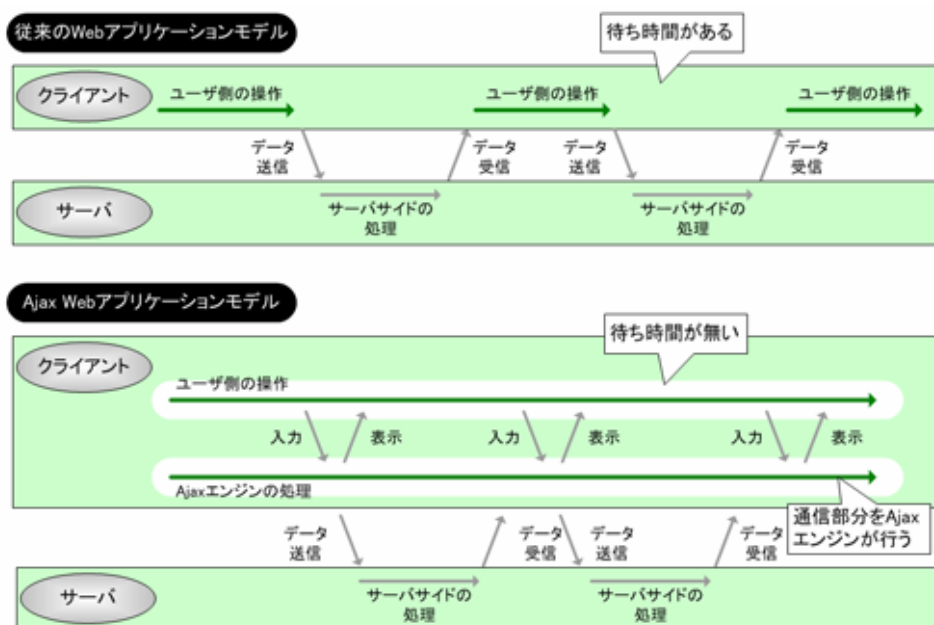


図 4.2 従来の Web アプリケーションと Ajax Web アプリケーションのモデル

Ajax を用いた Web アプリケーションの最大の利点はユーザの操作が途中で途切れ無いという所だが他にもいくつかの利点がある。1 つ目の利点として Ajax エンジンではページ全体ではなくユーザが必要なデータだけをサーバとやり取りするのでデータの通信量が減るという事が挙げられる。従来の Web アプリケーションではユーザが何かデータを要求するとページ全体を書き換えていたが、Ajax では DHTML という技術を使いページの一部分だけを書き換えるのでページ全体を書き換える必要が無くなった。2 つ目の利点は Flash 等とは違い専用のプラグインを必要としない事が挙げられる。Ajax は主にデータの通信と表示に Javascript と DHTML を使い、通信するデータ形式に XML を用いるのが従来の Web アプリケーションとの大きな違いだが、Javascript, DHTML, XML に関してもほぼ全てのブラウザとサーバが標準で扱う事が出来るため専用のプラグインを必要としない。ただし Ajax を用いた Web アプリケーションにもいくつかの問題点がある。1 つ目はクロスブラウザ問題と呼ばれている問題だ。Javascript は標準で使われているほぼ全てのブラウザで扱う事が出来るが現在の大部分のユーザが使う FireFox と IE の間で大きな言語使用の差が存在する。その為の 2 つのブラウザに実装する場合には最悪の場合、2 つのコードを記述する必要がある。この問題については様々な Ajax 用のライブラリが開発される様になっており後述する本システムが用いた bindows ライブラリもクロスブラウザ問題に対応している(全ての問

題についてではないが)。2 つ目はそもそも Javascript を扱えないブラウザの問題だ。先にほぼ全てのブラウザで Javascript を扱う事が可能と述べたがこれはデスクトップパソコンに限っての事である。モバイル端末にもフルブラウザ(パソコンと同じ様に Web サイトを閲覧出来るブラウザ)の実装等で Javascript が扱える端末が増えてきたがまだ多くのモバイル端末で Javascript を扱う事は現実的とはなっていない。本システムでもモバイル端末だけは Ajax を用いない形でインターフェースを実装している。3 つ目は現在の処理状況についてユーザが判断しにくいという問題だ。従来の Web アプリケーションでは操作のたびにページ全体が書き換わるので、良くも悪くもユーザが現在の状況について判断しにくいという問題は存在しなかった。しかし Ajax を用いた Web アプリケーションではユーザの操作が途切れるという事は無いので要求したデータはまたサーバから戻ってきていないが操作は可能という状況が考えられる。この場合ユーザは現在、データが戻ってきているのか戻ってきていないのかという判断が難しい。この問題を解決するには現在の処理状況を示す「now loading」の表示やプログレスバーの実装等が必要となる。

4.2.2 WebOS について

先にも簡単に WebOS については述べているが、詳細な技術的側面についてこの節では説明する。

Ajax は登場すると急速にインターネット上に広まっていった。Ajax が登場した当時、Web がデスクトップアプリケーションの様な豊かで応答性の良いインターフェースを提供出来ないという事が更なる発展の障害になっていたからである。Ajax の知名度に一気に広めたのは後述する Google Maps であるが、その後 Ajax を用いた Web アプリケーションの手法が研究される様になり操作性を究極的な所まで追求したのが WebOS という技術だ。

WebOS の基本的な概念は従来のローカルデスクトップ環境をそのままオンラインに移植させるというものだ。ローカルデスクトップ環境にはウィンドウ処理やアイコン処理等、Ajax が登場するまでは Flash 等を用いなければまったく実現する事が出来ない機能が多かった。

WebOS という概念が広まり始めたは 2005 年頃からだが、その後ローカルデスクトップ環境を移植するという試みやサービスは爆発的に広まって来ている。一般的に WebOS という概念で開発された Web アプリケーションには以下の特徴を持つ。

- ・ プログラムマネージャ・ファイルマネージャ・ウィンドウマネージャ機能を持つ
- ・ 端末にプラグインのインストールを必要としない
- ・ WebOS 上で他の Web アプリケーションとの連携が可能である
- ・ ユーザが各個人のデスクトップ環境を管理する事ができる
- ・ ユーザ同士でファイルの共有やチャット等の共同作業が可能である

本システムでモバイル端末以外のプラットフォームとして WebOS を選んだ理由はアンビエント情報社会と WebOS が非常に親和性が高いと考えたからだ。アンビエント情報社会は既存の Web 環境が更に発展した情報通信社会を示す。アンビエント情報社会では家電やセンサ等の現時点で将来確実にネットワークに繋がるようになるだろうと考えられている機器だけで無くセンサネットワークの発展によって時計や街灯等ありとあらゆるものがネットワークに繋がると考えられている。その際全ての情報を統一的に扱える技術が必ず必要になるが筆者はそれが現在の Web 技術が発展した技術だと考えている。HTTP,HTML,URI の 3 要素を基礎とした現在の Web 技術は問題がいくつかあるにせよ非常にうまく現在の Web 上に広がっている膨大な情報を扱える様にしている。今後も Web 技術を発展させながら情報通信社会を発展させていくという流れは変わっていかないと考えている。事実次世代の通信基盤となる NGN も現在の Web 技術を発展させた技術の上で動作する様に研究が進められている。そういった流れの中で WebOS は標準的な Web 技術のみを使い現在の OS の機能をそのままオンラインに移植しようとしている。ローカル OS は現在 Web だけでなくオフラインの作業も含め現在の情報通信社会における操作を行う事が出来る。オンラインで動作する情報システムとローカルで動作する情報システムの優劣はすでに述べたが、そのローカル OS をオンラインに移植しようという試みは情報通信社会の発展方向としては正しいレールの上を走っていると考えられる。

4.2.3 Comet について

先の述べた様に本システムの開発にあたっては、ネットワーク側からつまりサーバからの情報配信をする事がコンセプトの 1 つにあった。HTTP の仕様に従って従来の Web アプリケーションでこの機能を実装する事が非常に困難な事は先に述べたとおりである。ここではこの問題を解決するための概念である Comet について詳細に説明する。

Comet の基本となる考え方は一度クライアントからサーバに接続を要求すると、永遠に接続を維持し続けるというものだ。もし永遠に接続を維持し続ける事が出来れば、サーバ情報を配信したい時にその接続を用いてユーザに情報を配信すれば良いので、ネットワーク側からの情報配信が可能となる。例えばチャットアプリケーションを実装しようとした時、従来の Web アプリケーションの実装方法ではクライアント・サーバ間で実際のチャット上の発言があってもなくても、定期的に(リアルタイムでやろうとした場合非常に短い間隔で)通信を行う必要がある。一方 Comet は最初に 1 回だけ通信を行えば永遠に通信路を維持し続けるので、余計な通信がいらす通信量を大幅に減らす事が出来る。図 4.3 に通常の Web アプリケーションと Comet の概念を示す。

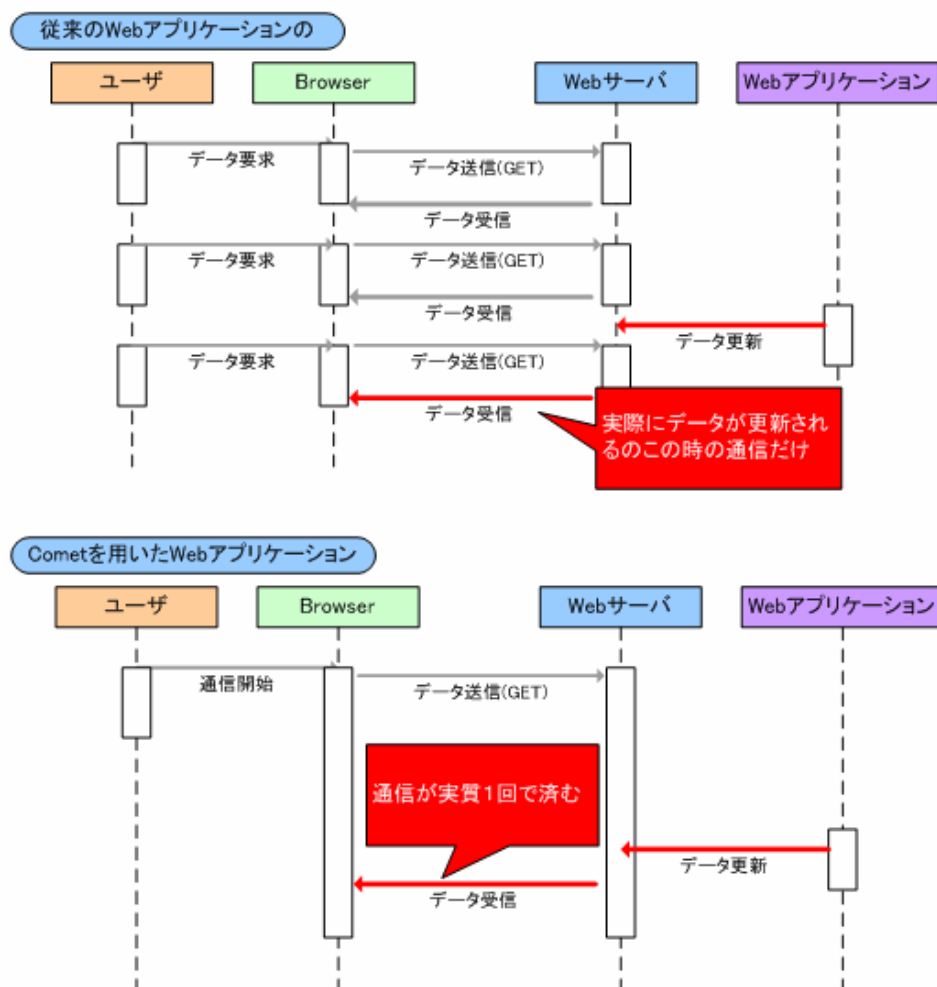


図 4.3 通常と Web アプリケーションと Comet を用いた Web アプリケーション

Comet の概念を実装する方法には 2 つあり 1 つは Web サーバを Comet 通信に対応する様に開発する方法だ。現在の Web サーバはこの通信方法に対応しておらず Web サーバを 1 から開発する必要がある。2 つ目は Ajax を用いる方法だ。Ajax はユーザとサーバ間の通信を代行するが、代行するだけでは図 4.3 上と同じ動作をするのでこの問題は解決出来ない。しかし、Ajax を用いて Web サーバと最初の通信を行った際 Web アプリケーション側で出力せず待機させておくと Ajax エンジンにデータが戻ってこないなのでこの通信は維持され続ける。Ajax を用いない Web アプリケーションでこの方法を用いるとサーバからデータが帰ってこないなのでその間ユーザがまったく操作出来なくなってしまうが、Ajax では Ajax エンジンにサーバからデータが戻って来なくてもユーザは操作可能なのでこの問題を解決出来る。通信を維持している間にサーバが情報を配信したい時にデータを Ajax エンジンに戻してやればユーザにも情報が配信されるのでネットワーク側からの情報配信が可能となる。

本研究では後者の Ajax を用いて Comet の概念を実装している。前者の Web サーバを開発した方がパフォーマンスは良いのだが、まだ開発手法が確立されておらず、時間的余裕

が無かったため本研究では取り入れていない。図 4.4 に Ajax を用いた場合の Comet の実装方法を示す。

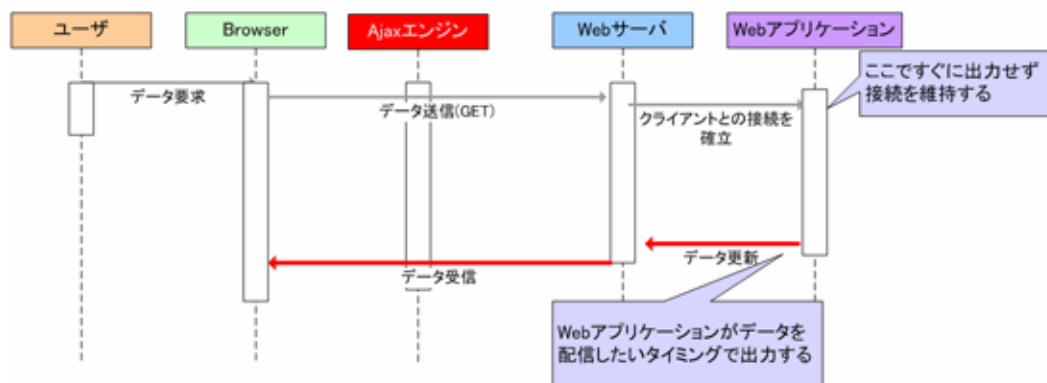


図 4.4 Ajax を用いて Comet を実装する方法

4.2.4 WebOS と Web アプリケーションの連携について

ここでは Web アプリケーションを WebOS 上でどのように動かしているかを開発者側の視点から説明する。図 4.5 にテスト Web アプリケーションとそのソースについて示す。

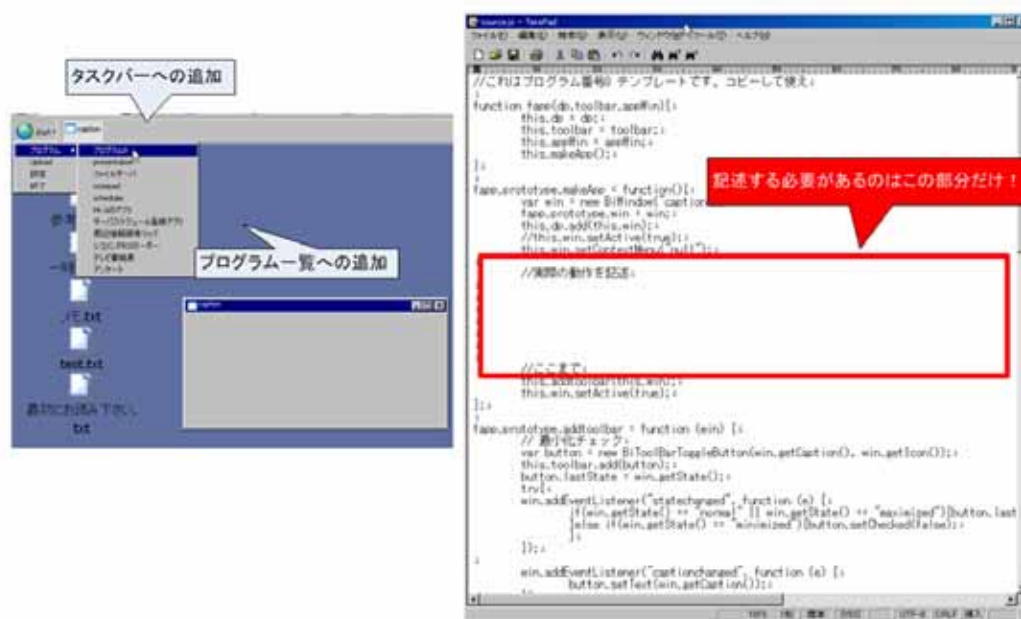


図 4.5 テスト用 Web アプリケーションとソースコード

WebOS 上で Web アプリケーションを動かそうとした場合、ローカル OS と同じ様にタスクバーへの追加やプログラム一覧の追加等の機能を持たせる必要がある。実際の Web アプリケーションのソースコードは図中右の様になっている。本システムでは最初から Web アプリケーション用のテンプレートが用意している。開発者はソースコード中段の「//

実際の動作を記述」というコメントから「//ここまで」という部分に実際の Web アプリケーションの動作を記述すればよい。

4.2.5 ベクトル空間法と TF・IDF 法について

ユーザに適した情報を推定する手法は、大きく分けると二つある。1 つはコンテンツに基づくフィルタリングと強調フィルタリングだ。コンテンツに基づくフィルタリングとは本研究で用いている様にユーザがニュースやブログの記事を閲覧した時の履歴等からユーザに最適な情報を推定する手法だ。強調フィルタリングではユーザと似た嗜好を持つグループを見つけ出し、そのグループに基づいてユーザに最適な情報を推定する。

ユーザに適した情報を推定する研究は先に述べた通り世界中の研究者が進めており様々な手法が存在する。本研究では筆者が「RSS 情報閲覧履歴を用いた TV 番組推薦システム」[8]にて有効性を確認しているベクトル空間法と TF・IDF 法を組み合わせた手法でユーザの情報を推定した。次にベクトル空間法と TF・IDF 法について説明する。

ベクトル空間法は元々情報検索において確立された代表的な手法で 1970 年代にサルタンによって採用された。ベクトル空間法は文章をいくつかのキーワードの「重み」によるベクトルで表現し、複数の文章のベクトルを比較する事でその文章の内容がどれだけ近いのかという事を判断する手法である。数式で表すと以下の様になる。ある文章群を D_1, D_2, \dots, D_n とする。この文章群全体で m 個のキーワード t_1, t_2, \dots, t_m を指定する。ここで文章 D_i におけるキーワードの重みを $w_{i1}, w_{i2}, \dots, w_{im}$ とすると文章 D_i は次のようなベクトル d_i で表される。

$$d_i = [w_{i1}, w_{i2}, \dots, w_{im}]$$

式 4.1 文章ベクトル d_i

比較する文章を Q とすれば同様に次のようなベクトル q で表される。

$$q = [w_{q1}, w_{q2}, \dots, w_{qm}]$$

式 4.2 文章ベクトル q

ベクトル空間法では文章ベクトル d_i と q をベクトル演算する事によって文章間の近さ、一般的には類似度と呼ばれる物を得る。類似度の定義は様々だが本研究では文章検索においてよく用いられるコサイン尺度を用いる。コサイン尺度は 2 つのベクトルのなす角度によって類似度を定義する。

$$\cos(d_i, q) = \frac{d_i \cdot q^T}{\|d_i\| \|q\|} = \frac{\sum_{j=1}^m w_{ij} w_{qj}}{\sqrt{\sum_{j=1}^m w_{ij}^2} \sqrt{\sum_{j=1}^m w_{qj}^2}}$$

式 4.3 コサイン尺度

図 4. にベクトル空間法によって類似度を求める概要を示す。

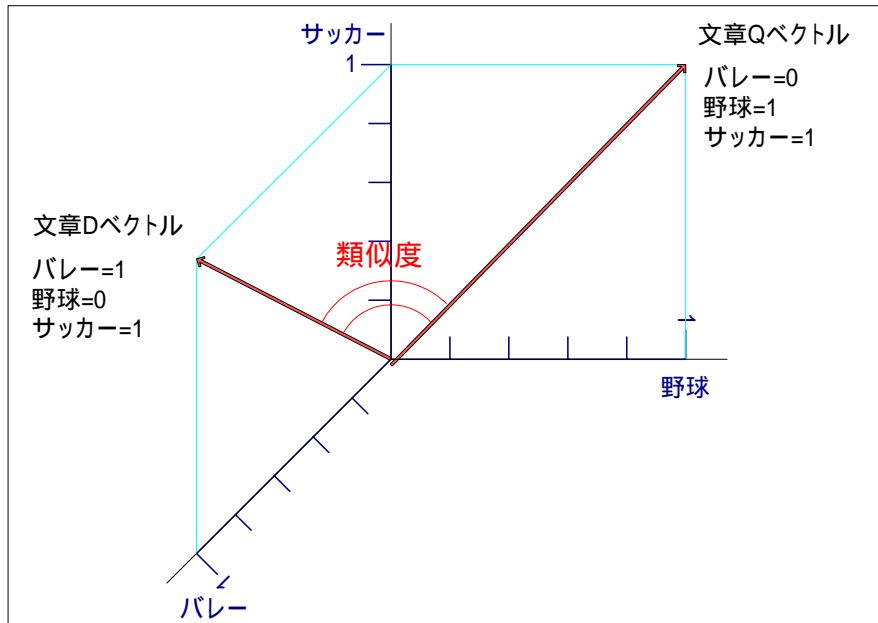


図 4.6 ベクトル空間法によって求められる文章間の近さ

次に TF・IDF 法について説明する。先に述べたベクトル空間法によってユーザに最適な情報を示すためには文章ベクトルを作成するためのユーザの嗜好を抽出する必要がある。本研究では TF・IDF 法を用いてレコメンド RSS リーダーアプリのユーザの閲覧履歴からユーザの志向を抽出する。

TF・IDF 法は TF と IDF という 2 つの指標を用いてキーワードの重みを評価する手法である。TF は term frequency の略称である単語がその文章中で何回出てきたのか出現回数がそのまま指標となる。IDF は inverse document frequency の略称で出現頻度の偏りを考慮して重みを決定するための指標である。多数の文章に共通して出現するキーワードの重みは低く評価し、特定の文章のみで出現するキーワードの重みは高く評価する。IDF は次に示す式で表される。

$$IDF_j = 1 + \log \left(\frac{N}{n_j} \right)$$

N : 文章の総数 n_j : キーワード t_j が出現する文章の数

式 4.4IDF

上で述べた TF と IDF をかける事で TF・IDF 法ではキーワードの重みとしている。

最後に Yahoo!JAPAN 社が提供している日本語形態素解析 Web サービスについて述べる。ニュースやブログの記事等の閲覧履歴からベクトル空間法と TF・IDF 法を用いてユーザに最適な情報を推定するには文章を品詞事に分解する必要がある。Yahoo! JAPAN の日本語形態素解析 Web サービスは API の後ろに解析オプションや文章をキーをして与える事で、日本語文を形態素に分割し、品詞・読みがな等の統計情報を XML データで戻してくる。本システムでも利用したので簡単に説明した。

4.2.6 Ajax ライブラリ「Bindows」について

Bindows は全てのブラウザに対応し、サーバサイドに様々なスクリプト言語を使用する事の出来る Ajax フレームワークライブラリだ。Ajax ライブラリとしては prototype.js が非常に有名だが、Bindows はライブラリというよりはフレームワークに近い。Bindows は以下の特徴を持っている。

- ・ ローカル GUI と同等の GUI コンポーネントが使用化
- ・ IE,Mozilla,FireFox,Netscape などブラウザ非依存で動作
- ・ 特別なプラグインのインストールは不要
- ・ Java,.NET,PHP などのあらゆるサーバに対応
- ・ Javascript と XML の知識以外はいらない

特に、1 つ目の GUI コンポーネントが豊富で Ajax を用いて WebOS の様なインターフェースに凝った Web アプリケーションを作るには最適にライブラリである。

4.2.7 Google Maps と Google Maps API について

Google Maps とは Google 社が提供している地図サービスである。Ajax を用いた Web アプリケーションの先駆けであり、そのインタラクティブな操作性は Ajax の知名度を一気に高めた。それまでは地図の方向を上下左右に移動したり、拡大縮小等を行おうとするとそのたびにページ全体を書き換えなければならず非常に使いづかった。一方 Google Maps は Ajax を用いて地図情報を常に送受信しながらユーザに掲示するので、ユーザは一々ページを更新する必要はなく、マウスでドラッグしたりスクロールしたりとほぼリアルタイムで地図情報を手に入れる事が出来る様になった。Google Maps はその操作性の高さから地図サービスにおいて圧倒的なシェアを誇っている。

Google Maps API は上記 Google Maps を他の個人や事業者が自由に使える様にと Google 社が公開した API である。Javascript と XML の知識さえあれば Web ページの中で

Javascript の地図オブジェクトを宣言すれば Google Maps を自分のサイトで公開する事が出来る。更に、縮尺表示・スクロールバーの表示等、地図をコントロールする機能のほぼ全てを追加する事が出来る。現在も Google Maps API は開発が進められており、アメリカのみに限るが開始地点と終了地点を設定する事でその間のルートを探索するといった機能を追加され始めている。Google Maps API の公開された結果、Google Maps を用いた新しい Web アプリケーションがユーザや企業によって次々と開発されている。

4.2.8 PHP と PEAR について

PHP とは Hypertext Preprocessor の略称で Apache や IIS などの主要 Web サーバ上で動作可能なサーバサイドスクリプトだ。関数主体のわかりやすい構文で急速にユーザ数が増加している。手軽なだけでなくオブジェクト指向言語であり PostgreSQL や MySQL, Oracle 等の主要データベースには全て対応している。また XML パーサ関数の豊富さも魅力であり Ajax を用いた Web アプリケーションの開発言語としては最適だと考え本システムの開発に採用した。PEAR ライブラリは PHP のライブラリで最も有名なスクリプトベースのオフィシャルライブラリだ。PEAR は他の C 言語のベースのライブラリとは異なり中身は純粋な PHP スクリプトである。自前で既存のライブラリを改変する事が中身を確認する事が容易な利点がある。更に専用のパッケージマネージャと呼ばれるインストーラも用いる事でインターネット上が必要なデータを自動で展開する事が出来るので非常に手軽である。PEAR の最大の魅力は豊富なライブラリの数だ。認証、データベース、暗号化、ファイルシステム、HTML、画像、メール、数値計算、XML と Web アプリケーションを開発する上でほぼ全ての分野のライブラリが用意されている。本システムで使用したライブラリの以下に挙げておく。

- ・データベースライブラリ MDB2
- ・HTTP ライブラリ HTTP_Download, HTTP_Request
- ・キャッシュライブラリ Cache_Lite
- ・ファイル形式ライブラリ File_Archive

4.2.9 Apache と MySQL について

本システムでは Web サーバとして Apache を用いている。Apache の利点を以下に挙げる。

- ・動作が非常に軽量である
- ・モジュールと呼ばれる部分を追加していくことで機能が豊富である
- ・Unix, Mac, Windows とプラットフォームを選ばない
- ・世界中で使われているので高い信頼性を誇っている

データベースには MySQL を用いている。MySQL の利点を以下に挙げる。

- ・ マルチユーザ・マルチスレッド SQL データベースである
- ・ 世界で最も普及しているオープンソースデータベースである
- ・ Linux,Mac,Windows とプラットフォームを選ばない
- ・ トランザクション機能をサポートしている

5.システムの課題と今後の展望

この章では、1~4 のまとめを最初に述べ、次にシステムの課題について述べる。最後に今後の展望について述べる。

5.1 まとめと考察

1 で本研究の目的はアンビエント情報社会におけるユーザインターフェースの開発に定めた。2~4 では最初にユーザインターフェースとして既存の Web 技術を発展させ WebOS を中心とした本システムのプラットフォームを開発した事を示した。次に開発したプラットフォーム上で動作する様々な Web アプリケーションを開発し図 2.4 に示す様な情報システムを構築した事を示した。

3.4 にて本システムのユーザインターフェースがユーザに対してどの様に变化して情報を掲示するかという事について説明した。実際に WebOS を中心とした本システムの動作を確認してみるとユーザに対して適切なユーザインターフェースで情報を推薦出来ている事がわかる。この結果はユーザ側から見た場合、WebOS を中心とした本システムがアンビエント情報社会におけるユーザインターフェースとして有効だという事を示唆している。次に開発者側から本システムを見てみる。1 にて現在「今だから」「此处だから」「貴方だから」というコンセプトに基づいたレコメンドサービスの開発研究が進んでいる事を述べた。これらの研究で開発された情報システムはそれぞれが独自の実装方法を取っている。これらの情報システムが統一されたプラットフォームの上で連携して動作する事の重要性は先に述べた通りである。本システムでは統一されたプラットフォームとして WebOS を中心とした情報システムを開発し、その上で実際に様々な Web アプリケーションが連携して動作している様子を 3.4 にて示している。この際 4.2.4 で述べた様に、Web アプリケーションの開発者は WebOS について特に意識する事なく開発を進める事が出来る。この事によりこれまで独自に開発を進めてきた各研究者の情報システムを容易に本システム上に移植する事が可能なため、本システムは開発者にとっても有益である事を示している。図 5.に本研究の成果の概略を示す。

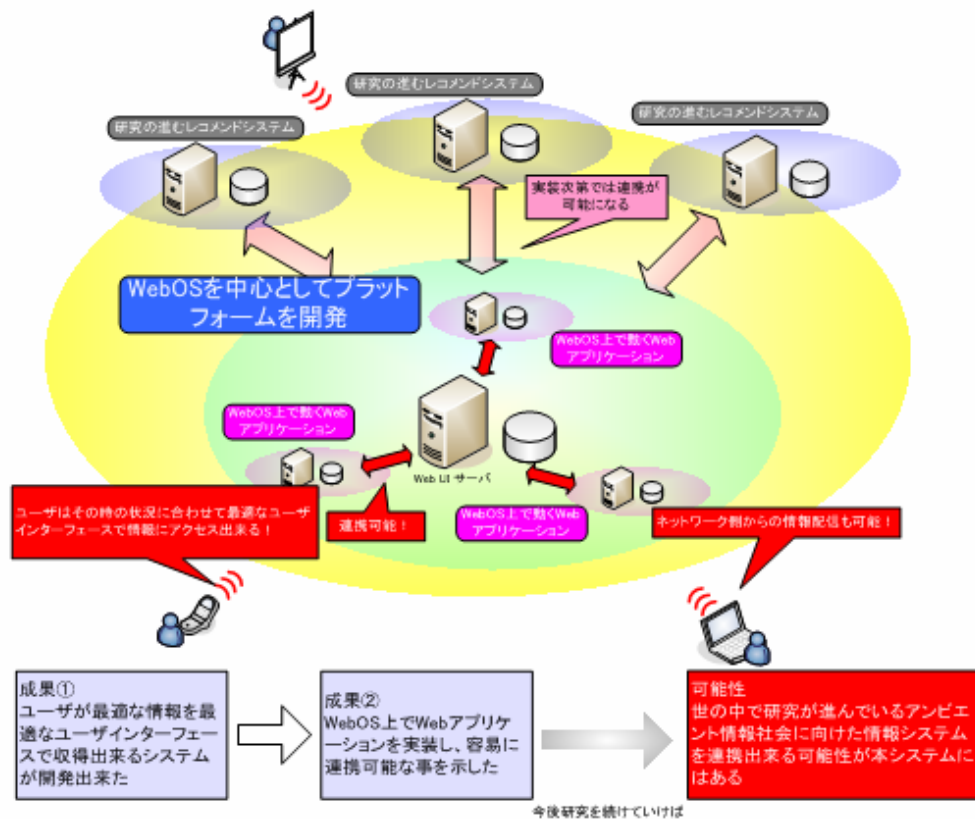


図 5 本研究の成果

5.2 システムの課題

本研究はアンビエント情報社会におけるユーザインターフェースを目指して開発を進めてきたが、実用的になるまでにはいくつかの課題を抱えている。それぞれについて問題と対策を述べる。

最初の課題は操作性の向上についてだ。Ajax を用いる事により、従来のローカル OS まったく同じ操作性を保持する事が最良だったのだが、主に Javascript の問題でまったく同等というわけにはいかなかった。特にファイルのアップロード関係の操作性については、問題が大きいと考えている。Javascript はセキュリティの問題上ローカルのリソースにアクセスする事が出来ない。そのため Ajax エンジンを使ってファイルを WebOS にアップロードする事が出来ず、従来のフォームを使ってファイルをアップロードしている。現在は iframe を使用する等してこの問題を解決しているがユーザーに対していつファイルのアップロードが終わるのか掲示出来ないため、プログレスバーの導入等を検討している。

次の問題はシステムの安定についてだ。先に述べた通り、ネットワーク側からの情報配信に Comet を用いているがこの部分が不安定となっている。Comet は先に説明した通り、サーバ・クライアント間で接続を維持し続けたまま、ネットワーク側からの情報配信を実

現する技術だ。しかしサーバ・クライアント間で接続が繋がっているうちに何らかの影響で接続が切れてしまうと Apache の httpd がゴーストとして残ってしまいサーバとの通信が出来なくなってしまう現象が起こっている。これは根本的には HTTP に関する問題でサーバ側でレスポンスを保留する部分に関しては HTTP 処理のカスタマイズにあたる。Jetty6 等ではこの問題も考慮に入れた設計も出来るようになっており、今後はそれらも検討していく。次はモバイル端末への対応についての問題だ。本システムでは先に述べた通りモバイル端末に関しては Javascript を用いず WebOS とは違うユーザインターフェースを実装した。ただし、今後はモバイル端末搭載のブラウザも高度な処理が出来る様に開発が進んでいく流れにあるので、モバイル端末にも WebOS と似たインターフェースを開発していく。最後に Web アプリケーションの拡充に関する問題だ。現在 8 つの Web アプリケーションを実装しているが、ローカル OS と同じ様に使えるためにはまだ実装しなければならない Web アプリケーションは多いと考えている。今後実装していこうと考えている Web アプリケーションは

- ・ Web メール機能
- ・ ファイルメッセンジャー機能
- ・ スクリプト言語開発機能

等である。

5.3 今後の展望

この節では、今後本システムをより発展させるための展望について説明する。最初に WebOS についての展望から述べる。WebOS についてはまずは WebOS と Web アプリケーションの連携部分をフレームワーク化する事がより本システムを発展させる事に繋がると考えている。4.2.4 で述べたプログラムインターフェースでは本システムを用いて Web アプリケーションを開発したい開発者はテンプレートファイルをコピーし、実行したいロジックを所定の場所に Javascript を用いて記述する形態を取っている。このプログラムインターフェースは非常にシンプルでわかりやすいが、ロジックを記述するには Bindows ライブラリと Javascript の高い知識が要求される。Javascript は非常にデバックがしにくい言語であるし、Bindows ライブラリについてもあまりサンプルが多くないので気軽に開発できる環境とは言いがたい。そこで XHTML 形式のタグ記述とシンプルな Javascript で開発が出来る様にプログラムインターフェースをフレームワーク化して行く事で、本システムで動作する Web アプリケーションが開発しやすくなると考えられる。更に 3.5 で述べたユーザに状況に合わせてユーザインターフェースと起動させる Web アプリケーションを変化させる部分のルールに関しては、現時点では筆者が静的に設定しているが、フレームワーク化するには Web アプリケーション側から設定する様にするべきだろう。またルール設定に関してはパターン認識を用いる等の動的な設定も検討して行く。

次に Web アプリケーションについての展望を述べる。Web アプリケーションに関しては、先に述べた開発予定の Web アプリケーションの他に強調フィルタリングを用いたレコメンド Web アプリケーションを開発するべきだと考えている。WebOS とローカル OS の大きな違いはオンラインで動作するかオフラインで動作するのだが、オンラインで動作することによる最大の利点はデータが全てサーバに保存されるので、ユーザ同士のマッチング非常に容易だという点だ。強調フィルタリングはユーザに似た指向を持つグループを探してそこからユーザに最適な情報を推定する手法であるから、WebOS との親和性はより高いと考えられる。強調フィルタリングを用い、精度の高い情報推薦がユーザに出来る様になれば本システムはより発展すると考えられる。

最後に、本研究では実際にユーザに使って貰い本当に WebOS を中心としたユーザインターフェースがアンビエント情報社会に有効なのかという検証を行う事が出来なかった。システムの課題が修正されしだい、本システムを公開しユーザに有効性を検証して貰い、更にユーザに適切な情報を配信出来るシステムに発展させていく予定である。

6.謝辞

本研究を進めるにあたり、沢山のご協力を頂き無事研究の成果を本稿にまとめる事が出来ました。

今年で成田誠之助教授が御退官という事もあり、全力で研究に取り組んで来たつもりです。3年間他研究室とは違い完全に自由なテーマで研究をさせて頂き、また私がテーマ選考に苦しんでいる時に貴重な助言をして頂いた成田誠之助教授に深く感謝いたします。

新しい発想で何かを研究するという事に不安を感じていた私が、まだまだ未熟ですが開発者としてここまで成長出来たのは成田誠之助教授のおかげだと思っております。

学部生として成田研究室に配属されてから、3年間お世話になった同輩の方にまずお礼を申し上げたいと思います。技術的に未熟な私にいつもの的確なアドバイスで開発を助けて下さった菅谷君本当にありがとうございました。最後の1年間は似た様なテーマという事で様々なアイデアを助言して下さいした後藤君本当にありがとうございました。的確な指摘をして下さった片野君本当にありがとうございました。

また、学部生の時代からプログラミングの基礎を叩き込んで下さった先輩方に厚く御礼申し上げます。

最後になりますが、3年間、成田研究室として様々な研究活動において助言やご協力を頂いた皆様に、この場を借りて心から御礼を申し上げます。

早稲田大学大学院 理工学研究科 情報ネットワーク専攻
成田研究室 修士2年
横川 直人

7.参考文献

- [1]森武俊,“生活パターンを覚えて助ける知能住宅-センシングルーム 2005-”,ネットワークロボット研究会 NR-TG-1-12, pp.20-24 (2005)
- [2]宮崎智大他,“個人に適合したユビキタス情報システムの設計と試作”,情報処理学会研究報告,2005-UBI-9,pp.13-16
- [3]金杉洋他,“自分情報に基づいたユーザ行動モデルの推定”,情報処理学会研究報告,2007-UBI-14,pp.23-28
- [4]馬谷達也他,“ユーザ状況に応じたプッシュ型 GUI の動的作成”,情報処理学会研究報告,2007-UBI-14,pp.9-16
- [5]工藤峰一他,“情報通信インターフェースとしてのユビキタスパターン認識”,第2回戦略的情報通信研究開発推進制度成果発表会予告集
- [6]藤井崇史他,“インターフェースをパーソナライズするニュース提供システムの提案”,情報処理学会研究報告,2006-GN-59,pp.13-18
- [7]東京電力,“「デジタル地図案内板による情報提供サービスの試験実施について～日本初のデジタル式街頭地図案内板～」”,プレスリリース,平成18年10月2日
- [8]横川直人,“RSS情報閲覧履歴を用いたTV番組推薦システムの提案”,成田研究室2005年度卒業論文,2006
- [9]総務省,“平成19年版 情報通信白書”,2007
- [10]山田祥寛,“独習 PHP”,株式会社翔泳社,2007.2.10
- [11]川崎克己,“ゼロから始める Ajax”,株式会社毎日コミュニケーションズ,2007.2.26