

パーソナルサイズのマルチエージェント統合環境の開発

Development of Personal-sized Multi-Agent Integrated Platform: airWeb

飯塚 豊¹⁾ 富樫 良成²⁾
Yutaka IIZUKA Yoshishige TOGASHI

- 1) 株式会社エアフロント (〒158-0096 東京都世田谷区玉川台 2 - 2 - 1 ヤマウラビル 2F
E-mail: iizuka@airfront.co.jp)
2) 株式会社エアフロント (〒158-0096 東京都世田谷区玉川台 2 - 2 - 1 ヤマウラビル 2F
E-mail: togashi@airfront.co.jp)

ABSTRACT. This describes the development of Personal-sized Multi-Agent Integrated Platform named airWeb. airWeb is a Web-based information aggregation software and has functionality to execute multi-agent. The platform also includes Agent Development Kit, Agent Distribution Server and practical agents.

1. 背景

Web が本格的に普及し始めてから 5 年以上経過し、インターネット上に蓄えられる情報は現在も加速度的に増えている。しかし、新しいサイトの発見や、決まったサイトの定点観測、コミュニティサイトへの参加、必要な情報の保存など、ほとんどのユーザは手作業で行っているのが実情である。一部自動処理 (ロボットや購読機能など) が行われているケースもあるが、たくわえる情報のフォーマットがまちまちであり、操作性も統一されていない。とくに個人の環境の場合、情報の保管場所が一元化されていないことによる情報の再利用が困難となっているケースも多い。

これは HTML が論理構造ではなく主としてデザインに重点を置いて記述するために使われていること、すなわち Web のデータが正規化されていない点に負う部分が多い。HTML は、その簡易な書き方とブラウザ上での表現力の高さから広く利用されてきたが、文書管理という本来の SGML の持っていた意味はほとんど失われてしまった。XML などのドキュメントの論理構造と表現を分離する手段や、WSDL に代表される Web サービスのような新しい形態の情報流通手段は現れてきているが、主流は依然として HTML と HTTP サーバにとどまっている。

このような状況を踏まえ、必要な情報を取得する手続きの複雑化、一貫性のない情報の氾濫を解決する手段のひとつとして、エージェント技術が注目されている。

エージェント技術とは、コンピュータ上に実現された人間の情報処理を代行する機構の総称である。エージェントは範囲が広い用語であるが、このプロジェクトでは、特にインターネットエージェント (インターネット上の情報検索・統合・配信の支援) を中心として取り扱う。

2. 目的

本プロジェクトは、Web サイトの違いを吸収し、必要な情報を自動で取得し、ローカルのデータベースエンジンで情報の正規化を行ない、情報の閲覧や配信、検索、あるいは情報の自動抽出ができる統合型環境 airWeb を実現するための研究・開発を行うことを目標としている。

同時に、airWeb 用エージェント開発環境、エージェント配布サーバ、および実用的なエージェントを開発する。

3. 開発概要

airWeb を特徴づける技術は、下記の 8 つある。

- 1) airWeb は仮想機械 (VM) を持ち、その上で複数のエージェントを動作させることができる
- 2) VM はエージェントが取得した情報を正規化されたデータベースで管理するレイアを持つ
- 3) VM は HTML/XML に対応した強力なページ解析機能を持つ
- 4) サイトのリビジョン管理機能を持つ (過去から現在にいたるまでサイトの差分を管理できる)
- 5) ユーザインタフェースはメーラー風で親しみやすい
- 6) エージェントをサーバより配信できる
- 7) エージェントを自動更新する仕組みを持つ
- 8) エージェントを開発する開発環境 (ASDK) が提供され、これだけで開発が可能である

airWeb は、エージェントを実行し情報のアグリゲーションを行うクライアントソフトウェアの名称であるが、エージェントを配布する ADS (Agent Distribution Server) やエージェント開発環境である ASDK (Air Software Development Kit) と合わせてひとつのプラットフォームを形成している。本プロジェクトでは airWeb 本体、ADS、ASDK、および実際に使用可能なエージェントをサンブ

ルとして30種類開発する。

(1) airWeb

airWeb 本体の構成は図1のようなレイヤ構造となっている。オペレーティングシステムを Air Kernel と呼ばれるアプリケーション部分が包み、その外側に Air VM (仮想機械) とマネージャが接している。各種エージェントや、エディタやブラウザ(これらも実はエージェントである)などのユーザの目に直接触れる部分は、この Air VM の外側に位置し、Air VM のサービスを利用して実行される。

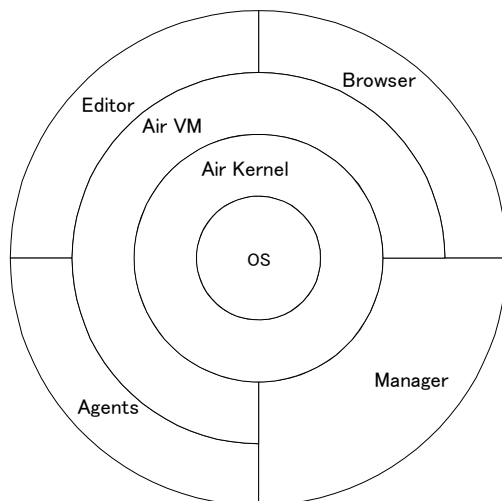


図1 . airWeb のレイヤ構造

Air Kernel の用意する機能は多彩である。これは、OS でサポートされていない高度な機能を集約したいわばライブラリ的なレイヤであり、以下のようなものが含まれる。

- TCP/IP 接続機能 (HTTP/FTP プロトコルを含む)
- ページ解析機能 (HTML/XML 解析)
- 検索機能
- データベース機能
- HTTP プロキシサーバ機能 (リビジョン管理機能)
- ブラウザ機能
- エディタ機能
- 各種コンポーネント

一方、Air VM は主として以下のような機能を持つ。

- Air VM Code 実行機能
- Object Pascal Script インタプリタ
- 外部スクリプトエンジン (VBScript や Ruby、Perl など) とのゲートウェイ
- エージェントのスケジューリング

(2) ASDK (Air Software Development Kit)

ASDK は、airWeb 用のエージェントを開発する環境で、以下のツールから構成されている。

- Air C Preprocessor (for Air C/C++ Compiler)
- Air C/C++ Compiler (for Air VM Code)
- Air Assembler (for Air VM Code)

- Air Link Editor (for Air VM Code)
- Air Designer (for Script Engine)

Air VM は Air VM Code とインタプリタスクリプトの両方をサポートしているため主として Air VM Code 生成系とインタプリタ系の2系統のツールを用意している。これは、できるだけ多くの開発者にエージェントを作ってもらうことを目的として、言語の壁をなくしたいとの考えから来ている。

Air C Preprocessor から Air Link Editor までは Air VM Code の生成に用いるコンパイラ系ツール群である。Air VM Code は実行速度が速いのがメリットである。現在は C/C++ しか用意していないが、Air VM Code は汎用であるので、他のコンパイラを用意することは可能である。

Air Designer の方は、インタプリタ系のスクリプトを生成するのに利用する。こちらはインタフェースビルダを持っているため、主としてユーザインタフェースの構築に威力を発揮する。

ただし、Air VM Code で利用できるサービスと、スクリプトで利用できるサービスに若干の違いがある。将来的にはこれらを統合して、Air C++ でもインタフェースを構築でき、スクリプトでも Air C/C++ の機能が使えようとする予定である。

(3) ADS (Air Distribution Server)

オンラインソフトウェアはもとより、パッケージソフトウェアであってもネットワークの普及により、ソフトウェアのオンラインアップデートはごく普通の事となった。

一方で、使用する数の多いユーザになると、毎日どれかがバージョンアップしているような状況で、ユーザは実際に全てをアップデートするどころかアップデート情報すら把握しきれないのが現実である。

airWeb のエージェントも同じ問題を抱えている。エージェントは情報ソースとなる Web ページが変更されただけで動作しなくなる可能性がある。エージェントは機能を維持するため、これに対応して修正を行わねばならない。

エージェントは頻繁にバージョンアップや仕様変更が行われる、ある意味特殊なソフトウェアである。幅広くエージェントを公開する開発者にとっては変更の都度それをユーザに何らかの方法で告知する必要があり、ユーザは自分の使用しているエージェントがバージョンアップされていないか常に気を配らねばならず、バージョンアップがなされれば新しいエージェントを入手してインストール作業を行わなければならない。

airWeb は一つのプログラムで対応しきれないものをそれぞれ専用の小型プログラムで対応しようというコンセプトであるため、ライトなユーザでも20~30種のエージェントを使用するものと予想される。

その数の多さそのものが障害となり、ユーザが自分の使用しているエージェントを全て最新の状態にしておくことは従来の手法では非常に困難なものになることは想像に難くない。

ADS (Agent Distribution Server) は airWeb のエージェントに関するこの問題を解決する事を目的として開発された。

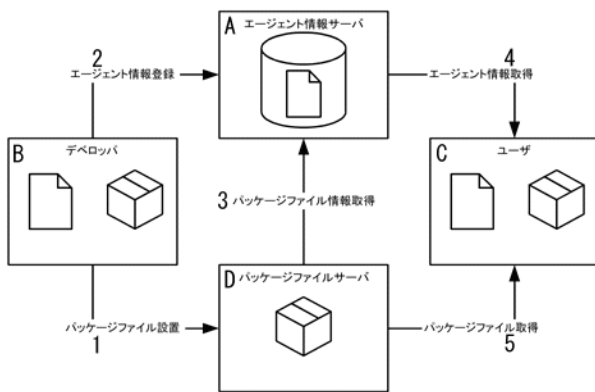


図 2. システム模式図

図 2 は基本的な ADS の仕組みを表したものである。A から D の各ブロックはシステムを構成する代表的な要素を表す。

- A. エージェント情報サーバ
デベロッパとユーザの橋渡しをするシステムの中核をなす要素である。
エージェントの情報を蓄え、ユーザの問い合わせに備える。
エージェント、デベロッパの情報を蓄積する独自のデータベースは XML を活用し、拡張性や可搬性に高い柔軟性を持つ。
デベロッパやクライアントとの情報通信にも XML を利用し、複雑で多様なエージェント、デベロッパ情報の交換のために役立てている。
エージェント情報サーバは専用のクライアントプログラムだけでなく WWW にもエージェント情報を提供する。
- B. デベロッパ
エージェントを登録し、配布する開発者。
デベロッパはパッケージファイルを公開し、エージェント情報サーバにエージェント情報を登録する。
この操作のためにデベロッパ向けのクライアントプログラム「エージェントパブリッシャ」があり、デベロッパ情報、エージェント情報の登録を可能にする。
- C. ユーザ
エージェントを入手し、利用するユーザ。
ユーザはエージェント情報サーバからエージェントの情報を取り出し、その情報に従ってパッケージファイルサーバからパッケージファイルをダウンロードしてインストールする。
この操作のためにユーザ向けのクライアントプログラム「パッケージマネージャ」と「オートアップデータ」があり、エージェント情報の検索、エージェントの導入をサポートする。
- D. パッケージファイルサーバ
パッケージファイルを集積し、ユーザに提供するサーバ。
HTTP や FTP 等の一般的なプロトコルを使用してパッケージファイルを取り出すことが可能なサーバであればパッケージファイルサーバ

として利用可能で、特別な機能は一切必要ない。

図中の 1 から 5 はデータやファイルの流れを表している。

1. パッケージファイル設置
デベロッパがパッケージファイルをパッケージファイルサーバに設置し、公開する。
2. エージェント情報登録
デベロッパはエージェント情報をエージェント情報サーバへ登録する。
3. パッケージファイル情報取得
エージェント情報サーバはパッケージファイルサーバをチェックしてパッケージファイルの存在を確認し、パッケージファイルのサイズを調べてエージェント情報に取り込む。
4. エージェント情報取得
エージェント情報サーバはユーザの問い合わせによってエージェントの情報を返す。
5. パッケージファイル取得
ユーザは得られたエージェント情報をもとにパッケージファイルサーバからパッケージファイルをダウンロードすることができる。

パッケージファイル設置はデベロッパがパッケージファイルを公開するために行う。これは従来のオンラインソフトウェアの公開と同じ方法であり、HTTP や FTP などの一般的なプロトコルでパッケージファイルがダウンロード可能な状態にする作業である。

エージェント情報登録はデベロッパがエージェント情報を公開し、パッケージファイルの入手法をユーザに告知する。

エージェント情報登録が行われると、エージェント情報サーバは新規登録と更新に分けて新着情報を作成し、ユーザに公開する。

エージェント情報登録は複雑なエージェント情報をエージェント情報サーバに転送するために XML を利用する。

エージェント情報取得はユーザがエージェントに関する情報を得るために広く使われる。ユーザが自分の必要なエージェントを探したり、興味のあるエージェントの機能を調べたり、既に導入しているエージェントの最新情報を検索する等利用頻度が高い。

パッケージファイル取得はエージェントのパッケージファイルを実際に入手する。パッケージファイルサーバから HTTP や FTP でパッケージファイルを転送する。パッケージファイルサーバの所在や転送に使用するプロトコル等のパッケージファイル入手法に関する情報はエージェント情報サーバから得ることができる。

(4) 実用エージェント群

エージェントプラットフォームの実用性を確認するために、30 種類以上のエージェントの開発を行った。今回開発したエージェントは、大まかに分けて以下のカテゴリに分類される。

- ニュースクリッピング
- 資料と情報源

- コンピュータ
- 読み物
- 掲示板
- ショッピング
- 巡回
- その他・ツール

実際にエージェントを動かすのと同時に、ADS を経由したエージェントの登録、配布、自動更新にも利用した。

4 . 開発内容

airWeb プラットフォームでの情報の流れは、図 3 のようになっている。これは、airWeb による処理の 1 例であり、Auction Server の位置には、ほかにもショッピングやマーケット情報、チケット予約など、同様に多様なサーバがはまり込むことになる。エージェントがこれらサーバとの橋渡しを行うため、本体の改造なしにエージェントの追加だけで対応ができる点もポイントとなっている。

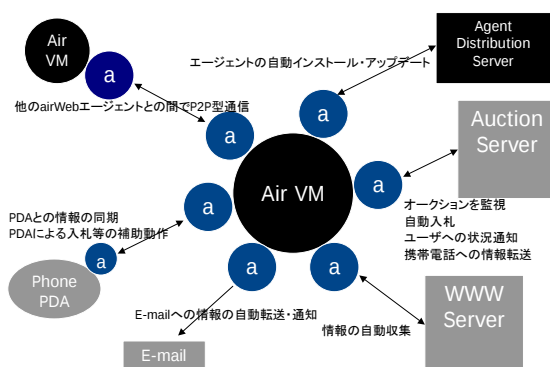


図 3 . airWeb の動作例の模式図

airWeb で実行されるエージェントには大まかに分けて 2 種類ある。

1 つは Air VM Code で構成された Air VM Agent。

もう 1 つは、Object Pascal Script や JavaScript など、スクリプトエンジンを利用した Script Agent である(図 4)。

Air VM Agent は Air VM 上で直接実行されるが、Script Agent は Air VM がインターフェイスとなって、インタープリタのスクリプトエンジンで実行される。スクリプトエンジンとして airWeb が標準で内蔵しているのは Object Pascal Script Engine のみである。VBScript Engine や JavaScript Engine などは OS が用意している機能であるため、Windows 以外のほかのプラットフォームへ移植する際には注意が必要である。移植性を考える場合は、Object Pascal Script Engine や Perl などを考慮した方がよい。

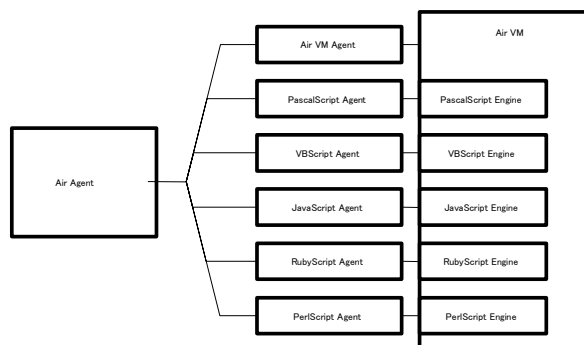


図 4 . Air Agent の種類

Air VM Agent は、Air C/C++ ツール群(Air Preprocessor、Air C/C++ Compiler、Air Assembler、Air Link Editor)を用いて開発する。

Script Agent は Air Designer を用いる。Air Designer は Windows 上の開発環境 Delphi とよく似ている。ただし、使用できる言語が Object Pascal Script や JavaScript、Perl などプラグインできる点が異なる。また、インタープリト機能は持つがコンパイル機能は持っていない。

(1) Air VM のサポートする関数群

Air VM は、大量の標準関数を持っているが、大別すると表 1 のようになる。

なお、将来的には、Air VM Code 系と Script Engine 系の関数を対称となるように用意する予定である。

表 1 . airWeb のサポートする関数群

Air VM Code 系	Script Engine 系
ANSI C 標準関数のサブセット	ユーザインタフェースコンポーネント
ページ解析関数群	データベース操作コンポーネント
TCP/IP ソケット関数群	TCP/IP ソケット操作コンポーネント
HTTP/FTP 関数群	HTTP/FTP 操作コンポーネント
データベース操作関数群	エージェント間通信コンポーネント
設定値データベース操作関数群	
エージェント間通信関数群	

(2) ページ解析関数

ページ解析関数は、HTML/XML ページの構文解析を行ない、エレメントの抽出を自動で行う。しかし、airWeb のページ解析関数は、一般に用いられるようなツリーウォークのような方法を取らず、集合論に基づいたエレメント演算を特徴としている。これは、ツリーウォークよりもより編集強度の高い(すなわち対象の HTML/XML 文書のエレメントの階層構造や順序が若干変わっても解析できる可能性の高い) 情報抽出手段だからである。一般のツリーウォークがファイルシステムとすれば、ページ解析関数は SQL データベースに相当すると考えられる。

抽出されたエレメントを集合で管理し、集合間の演算によって必要なエレメントを特定する。抽出されたエレ

メントからアトリビュートやテキストを取り出すこともできる。

例として、例 1 に <http://www.asahi.com/> のトップページから、トップニュースのみを抽出して表示するコード断片をあげる。

```
1 // ページを解析
2 page = page_create(buffer, len); // page オブジェクトの生成(bufferにはHTMLが入っている)
3 // “A” エlement集合を取得
4 el1 = page_find_elements(page, NULL, “A”);
5 // Top 5 コメントで挟まれた領域をElement集合として取得
6 el2 = page_find_text(page, “Start of Top 5.*End of Top 5”);
7 // Top 5 で挟まれた領域に含まれる “A” Elementを抽出
8 el3 = page_elements_inside(page, el1, el2);
9 buffer[0] = 0;
10 for(i = 0 ; i < el3->count ; ++i) { // el3 のElementを列挙
11 // Elementのテキストを結合
12 strcat(buffer,page_element_text(page, &el3->items[i]));
13 strcat(buffer, “%r%nr”);
14 }
15 puts(buffer); // ヘッダラインリストを表示
16 page_free(page); // page オブジェクトの解放
```

リスト 1 . <http://www.asahi.com/> のトップニュースを抽出する処理

リスト 1 のコードが対象とするページは、以下のような特徴部分を持っている。

```
<!-- Start of Top 5 -->
■ <a href="0803/news/national03034.html">
式根島で震度 5 クラスの地震相次ぐ</a><br>
← 中略 →
■ <a href="0803/news/sports03009.html">
```

リスト 2 . <http://www.asahi.com/> の HTML 断片

このページには、大量の A タグやコメントなどが含まれているが、<!-- Start of Top 5 --> と <!-- End of Top 5 --> というコメントに含まれる A タグは上記のものに限られる。

そこで、まずコードの 4 行目でページElementの全体から、A タグの構成するElementを抽出する。これが el1 というElement集合 (リスト) である。

次に 6 行目で、この着目するコメントに含まれた領域を正規表現によって抽出し、擬似Elementを生成する (擬似Elementは、抽出されたテキスト断片の前後が と で囲まれたElement相当として扱うことができる)。これは el2 というElementの集合となる。

8 行目では、el1 のうち el2 の各Elementの領域内

に含まれているElementのみを抽出する処理を行う (重なった領域)。すなわちこれがトップニュースを含んだ A Elementの集合 el3 ということになる。

9 行目以降は、el3 のElementに含まれるテキストを結合し、最後に puts 関数でトップニュースのテキストを表示している。16 行目でページオブジェクトを解放する。

この例は非常に単純であるが、ページ解析関数はElementの加減乗除算や包含関係・前後関係に関する演算、正規表現によるタグの検出など多彩な機能を持っている。もちろん、XML にも対応している。ページの特徴の抽出が簡単であるため、サイトに特化したエージェントを短いステップ数で簡単に記述できるのは airWeb の大きな特徴である。

(3) エージェント情報サーバ

ADS はデベロッパによって登録されたエージェント情報をユーザに提供することが最も重要な機能である。エージェント情報サーバはその機能を担うキーコンポーネントに位置付けられる。

デベロッパによって登録されたエージェント情報はカテゴリに分類し、ユーザはカテゴリに分けられたエージェント情報の一覧から必要なエージェントを容易に見つけることができる。

デベロッパから新しいエージェントの登録やバージョンアップの情報更新があれば、新着情報として提供することで、ユーザは集約されたエージェント情報をリアルタイムに得ることができる。

ユーザ側にはインストール済みのエージェントのバージョンが保存されており、エージェント情報に含まれる最新版のバージョン情報とインストール済みのエージェントのバージョンを比較することで、最新版の存在を検出し、更新が必要なエージェントを自動的に判別することが可能である。

登録されたエージェントをユーザがダウンロードすればダウンロード件数を計数し、デベロッパがダウンロードの傾向を分析する事を可能にするとともに、ランキングを集計して実績、定評のあるエージェントを判断するための材料をユーザに提供する。

エージェント情報にはデベロッパ情報が付加されており、デベロッパの連絡先やそのデベロッパの公開しているほかのエージェントを知ることが可能となっている。

これらの機能はクライアントプログラム、登録プログラムで利用可能なだけでなく、WWW にも情報を提供している。

a) エージェント情報サーバの内部構成

ADS 情報サーバプログラムの内部は大きく分けて、

1. インターフェイス
2. データベース
3. ドキュメントジェネレータ

の三つから構成されている。

インターフェイスはクライアントからのリクエストを受け取り、必要な処理を判別し、データベースやドキュメントジェネレータを操作する中心的役割を持っている。

データベースはエージェントやデベロッパのデータを集積している。

インターフェイスや、ドキュメントジェネレータから登録や問い合わせの操作が行われる。

ドキュメントジェネレータはクライアント要求する情報をデータベースから引き出し、クライアントの要求する形式に整形する役割を持つ。

b) インターフェイス

現在、ADS のインターフェイスは CGI (Common Gateway Interface) である。Web ブラウザによる情報提供と airWeb のエージェントとしてクライアントプログラムを開発するのに都合が良いため HTTP を用いて Web サーバ経由でサービスを実施している。

エージェント情報サーバのインターフェイス部は、

1. エージェント情報の登録
2. エージェント情報の更新
3. デベロッパ情報の登録
4. デベロッパ情報の更新
5. エージェント情報問い合わせ
6. デベロッパ情報問い合わせ

これらの要求に応える機能を持っている。

エージェント情報サーバのインターフェイス部は CGI パラメータ 'action' の指定によってさまざまな機能を提供する。

メソッドは GET/POST の双方に対応し、action に動作を指定し、追加パラメータを与えることで利用する。

表 2. ADS 機能パラメーター一覧

action	パラメータ	機能
display	view	view パラメータで指定されたビューに対応するテンプレートを展開した結果を得る。 view の他、任意のパラメータを付加してドキュメントジェネレータに渡すことができる。
query	category	category パラメータで指定されたカテゴリに登録されたエージェントの一覧を得る。
	categories	categories に値を指定する必要はない。 ADS のカテゴリ一覧を得る。
	agent(*)	agent パラメータで指定した ID のエージェントの詳細情報を得る。

	arev(*)	arev パラメータで指定した ID のエージェントのバージョン情報を得る。
	developer(*)	developer パラメータで指定した ID のデベロッパの情報を得る。
submit	session	session パラメータに XML で記述されたエージェントやデベロッパ情報の登録や更新を実行する。
download	item	item パラメータで指定した ID のエージェントのダウンロード方法を得る。

c) データベース

エージェント情報サーバはエージェントおよびデベロッパの情報を集積するために独自のデータベースを内蔵している。

データベースはレコードを XML で記述したものをファイルシステム上の 1 ファイルに格納し、特に多用される検索キーのインデックスを数種類用意している。

レコードが XML で記述されているため、フィールドの追加や削除、統合や分割が容易であり、レコードの構成変更の際にもインデックスの再構築を必要としない。

また、レコードがファイルシステム上の 1 ファイルに対応しているため、バックアップや部分的な修正が容易で、トラブルの際もレコードが全滅する危険を回避することが可能である。

d) ドキュメントジェネレータ

ドキュメントジェネレータはインターフェイス部が受け付けた Web ブラウザ、専用クライアントからのリクエストを元に出力するドキュメントを生成する。

ドキュメントの内容、書式、メディアタイプはテンプレートとして用意されており、リクエスト時に指定されたテンプレートが展開されて返される。

テンプレートによっては引数を受け付けるものもあり、テンプレートさえ用意すれば多彩なリクエストに応えることができる。

e) ドキュメントジェネレータのテンプレート

テンプレートはデータのアクセスとデータの加工、整形をまとめて書くことができる。

変数が使用でき、CGI のパラメータや、環境変数も取得できる。また、原始的な条件判断やリストに対する反復処理も可能である。

ADS のドキュメントジェネレータはプログラム言語処理系のプリプロセッサに似た性格を持つ。

HTML ドキュメントを生成する目的で設計されたため、HTML としてデザインされた雛型を加工してテンプ

レートを作成する利便性を重視し、記述法は可能な限り HTML の文法に合わせたいため、テンプレートは Web ブラウザで開いてもある程度デザインの確認が可能である。

Content-Type: text/html

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="/ads/templates/ads.css">
<title>
LIST
</title>
</head>
<body>
<ads:assign name="BASEURL"
value="http://&HTTP_HOST;&SCRIPT_NAME;"/>
<ads:assign>
NEWS=ニュース
INFO=資料と情報源 READING=読み物
COMPUTER=コンピュータ
BBS=掲示板 SHOPPING=ショッピング
FINANCE=金融情報 BUSINESS=業務用
GATHER=巡回 MISC=その他・ツール
cuALL=総数 cuNEWS=ニュース
cuINFO=資料と情報源 cuREADING=読み物
cuCOMPUTER=コンピュータ
cuBBS=掲示板 cuSHOPPING=ショッピング
cuFINANCE=金融情報 cuBUSINESS=業務用
cuGATHER=巡回 cuMISC=その他・ツール
aPIDX=21
</ads:assign>
<p>
<ads:List id="cats"
value="ALL,NEWS,INFO,READING,BBS,AUCTION,MAR
KET,BUSINESS,GATHER,MISC">
<ads:foreach list="cats">
<ads:RecordList db="agent" table="&CurrentItem;"
pagesize="max" pageindex="1">
<ads:value name="cu&CurrentItem;"/>
(&RecordCount;)
</ads:RecordList>
</ads:foreach>
</ads:List>
</p>
<ads:RecordList id="test" db="agent" table="ALL"
pagesize="50" pageindex="&PIDX;" orderby="&order;">
<ads:List id="test.Pages">
<ads:foreach list="test.Pages">
<ads:numcompare id="chk" a="&PIDX;"
b="&CurrentIndex;">
<ads:eq compare="chk">[&CurrentItem;]</ads:eq>
<ads:ne compare="chk"><a
href="&BASEURL;?action=display&view=TOP&PI
IDX=&CurrentItem;">&CurrentItem;</a></ads:ne>
</ads:numcompare>
</ads:foreach>
</ads:List><br>
<ads:assign name="prev" value="&PIDX;"/>
<ads:assign name="next" value="&PIDX;"/>
<ads:sub var="prev" value="1"/>
<ads:add var="next" value="1"/>
<strong>
<ads:numcompare id="chk" a="&prev;" b="1">
<ads:lt compare="chk">PREV</ads:lt>
<ads:egt compare="chk"><a
href="&BASEURL;?action=display&view=TOP&PI
IDX=&prev;">PREV</a></ads:egt>
</ads:numcompare>
[&PIDX; / &PageCount;]
```

```
<ads:numcompare id="chk" a="&next;"
b="&PageCount;">
<ads:gt compare="chk">NEXT</ads:gt>
<ads:elt compare="chk"><a
href="&BASEURL;?action=display&view=TOP&PI
DX=&next;">NEXT</a></ads:elt>
</ads:numcompare>
</strong><br>
<ads:foreach list="test"><ads:Record db="agent"
serialid="&CurrentItem;">
<p>
<table border="1" width="600" cellspacing="0"
cellpadding="2">
<tr><td width="70%">&Id;</td><td>カテゴリ:
<ads:value name="&Category;"/></td></tr>
<tr><td width="70%">エージェント:
&Name;</td><td>バージョン: &Version;</td></tr>
<tr><td colspan="2">説明: &Caption;</td></tr>
<tr><td colspan="2">開発者: <ads:Record
db="developer" serialid="&DeveloperId;"><a
href="&BASEURL;?action=display&view=DEV&PI
DX=&DeveloperId;">&Name;</a></ads:Record>
<tr><td align="right"><a
href="&BASEURL;?action=download&item=&Id;">ダ
ウンロード</a></td><td>ダウンロード数:
&TotalDownloadCount; (&DownloadCount;)</td></tr>
</table>
</p>
</ads:Record></ads:foreach>
</ads:RecordList>
</body>
</html>
```

リスト 3 . 登録されたエージェントリストを HTML 形式で出力するテンプレートの例

HTML を踏襲した構文の制約上、他のサーバサイドスクリプトに比較して関数がないなど機能や柔軟性において劣るが、シンプルな構造のため展開にかかる負担も少ない。

例として、ADS に登録されたエージェントの一覧を表示するテンプレートをリスト 3 に示す。

(4) エージェント

一般のユーザが手にとってすぐに試せるエージェントを 3 0 以上開発をした。これらを現状でジャンル分けすると、以下のようになる。

- ニュースクリッピング
ニュースやリリース情報など、速報性のある情報を収集するエージェント
- 資料と情報源
地図や書誌情報、会社情報、統計情報など比較的ロングタームに使用される情報を収集・検索・ナビゲートするエージェント
- コンピュータ
ニュースやソフトウェア情報などで特にコンピュータ関連に属する情報
- 読み物
エッセイや小説など、読み物の購読を支援するエージェント
- 掲示板
Web 掲示板を巡回し、操作性を向上させるエージェント
- ショッピング
ショッピングの補助を行うエージェント
- 巡回
サイトの更新チェックやミラーリングを行うエージェント

- その他・ツール

airWeb の操作性を補助するエージェントや ADS 関連のエージェントなど、ツールのエージェントと、その他の分類されないエージェント

実際にエージェントを動かすのと同時に、ADS を経由したエージェントの登録、配布、自動更新にも利用した。これらエージェントは<http://www.airclub.org/> にて内容を確認することができる。

5. 今後の展望

現在、airWeb は<http://www.airclub.org/> にてベータテストを行っている。

本プロジェクトにより、エージェントを開発、公開、配布、実行、更新する全体の仕組みが動作する環境を作ることができた。その上で、2001 年 10 月末現在で 70 種類以上のエージェントが稼働している。

実用的なエージェントが動作することを確かめられたため、今後はエージェントニーズの開拓を行うとともに、プラットフォームの強化を行いたいと考えている。これには、エージェント認証というセキュリティの仕組みの導入やナレッジマネジメント機能の追加、移植などが含まれる。

(1) ユーザインターフェイスの向上

プラットフォームの機能としてはすでに実用化段階にあるが、より幅広いユーザが利用できるようにユーザインターフェイスデザインの向上を行う。

これには、Web ブラウザとして使用した場合の、Web 操作履歴の自動管理機能や、既に実装されているマルチパネルブラウザのブラッシュアップ、ツールバーの整理なども含まれる。

(2) セキュリティ機能の向上

エージェントの普及とともに、セキュリティ機能のより一層の向上が求められる。本体のセキュリティ機能を向上させるほかに、エージェント用暗号 API の充実、エージェントコードの保護、エージェント配布パッケージの署名、ADS へのパッケージ認証機構の追加など、エージェントを安全に普及させていく仕組みを作る必要がある。

(3) Web サービスとの相互乗り入れ

W3C(World Wide Web Consortium)により策定中の WSDL(Web Services Description Language)と親和性を向上させるために内部 API を整備し、エージェント機能を強化して、Web サービスの活用を活性化させる。

(4) 情報の自動抽出機能と自然言語処理

airWeb には初期の版から情報を自動抽出してレポートにまとめるトピックスと呼ばれる機能がつく予定だが、最初の版ではあいまい検索や日付検索などの比較的単純な条件しか装備していない。これを、多変量分析などによる自然言語処理を取り入れ、よりユーザの意向に密着した情報のキャッチを行えるよう、精密化をはかっている。トピックス機能の向上は、すべてのエージェントの機能向上にもつながるため、絶え間ない改良を続けたい。また、同様の自然言語処理エンジンをエージェントに解放し、エージェントのユーザイ

ンタフェースをより人に親しみのあるものにしたい。

(5) 情報クリッピングテンプレート

ASDK を用いた本格的なエージェント開発を行わず、テンプレートの記述だけで Web サイトや Web サービスの情報を簡単に抽出できる機能を追加する予定である。

実際にはテンプレートを解釈できるエージェントという形で実装されるが、この機能により、プログラミングの経験のない者でも、比較的自由に情報のウォッチや自動収集を行う機能を追加できるようになる。また、ADS を通してこのようなテンプレート集もエージェントとともに配布できるようにする。

(6) Air VM の移植

airWeb 本体の Windows 依存部分を排除し (VM の API を整備し)、airWeb を Linux や MacOS、TRON、PalmOS など他の OS への移植を進める。つまり、Air VM の汎用化である。

airClub は、すべてのプラットフォームのユーザに共通に利用できるようにする。

エージェント開発言語としては Air C/C++ コンパイラ、Object Pascal Script、Ruby (フリーソフトウェア)、Perl (フリーソフトウェア) を推進するが、さらに新しい言語を開発することも検討している。Air VM は言語の設計と独立なので、時代とともに言語をバージョンアップさせていくことが可能である。

(7) 言語の充実

Air VM に対応した言語を増やす。Java や C# など、他社の考案した言語であっても評判のいいものや普及しているものであれば、積極的に実装を行ない、開発者に選択肢を与える。また、VM の基本スペックの公開を行ない、他の開発者による言語の開発を可能とする。

(8) 啓蒙活動

エージェント開発の促進のために、ウェブの構造やネットワーク上で提供されている情報の特性の分析などの初歩的な事項を含めて、インターネット上に教室を開いて、底辺の拡大をはかりたい。

実世界の仕事を取り込むためにあらゆるアプリケーションソフトウェアが現れたように、ネットワーク上に分散するリソースを取り込むために多種多様なエージェントを育てる必要がある。そのためには、製品内容をより魅力的にすると同時に、地道に啓蒙と教育によって開発者に働きかけていくしかない。

6. 参加企業および機関

飯塚豊
株式会社エアフロント

契約件名
「パーソナルサイズのマルチエージェント統合環境
AirWeb」