データベースシステム 116－30
（1998. 7. 9）

モーバイルオブジェクトシステム PLANET を用いた Web 検索ロボットのモーバイル化

染谷 祐一† 阿部 洋丈†† 松原 克弥‡
東村 邦彦† 加藤 和彦‡§

† 筑波大学大学院 博士課程 工学研究科
†† 筑波大学 第三学群 情報学類
‡ 筑波大学 電子・情報工学系
§ 科学技術振興事業団

モーバイルオブジェクトシステム PLANET を用いて，Web 環境におけるモーバイルソフトウェアロボットを開発するための方法について述べる．PLANET は言語独立な階層構造，モーバイルオブジェクトのネイティブコード実行，非同期オブジェクトパッシングという特徴を有している．本稿では，この特徴を利用した，モーバイル Web ロボットを実装するための方法を提案する．また，インターネット環境において行った実験に基づき，その効果を検証し議論を行う．

Implementation of Mobile Web Robots for PLANET mobile object system

Yuuichi Someya† Hirotake Abe†† Katsuya Matsubara‡
Kunihiko Toumura† Kazuhiko Kato‡§

† Doctoral Program in Engineering, University of Tsukuba
†† College of Information Science, University of Tsukuba
‡ Institute of Information Sciences and Electronics, University of Tsukuba
§ Japan Science and Technology Corporation

The paper describes a framework to develop mobile software robots in the Web environment by using the PLANET mobile object system we have developed. Among many proposals of recent mobile object systems, the system is characterized by language-neutral layered architecture, native code execution of mobile objects, and asynchronous object passing. Fully utilizing the characteristics, we propose an approach to implement mobile Web robots. We verify and discuss its effectiveness based on experiments performed in the Internet environment.

1 はじめに

WWW システムの急速な普及に伴い，大量の有用なデータが世界中のウェブサーバに格納されているが，ユーザがそれにアクセスするためには，必要とする情報の URL を指定する必要がある．目的とする情報の URL を得るために，インデックス機能が利用されている．現在では，自動的にインデックスを生成する方法として，ソフトウェアウェブロボット技術が用いられている．実際のシステムをあげると，World Wide Web Worm [7]，WebCrawler [9]，Lycos [6]，Harvest [1]，AltaVista [10]，WISE [12] 等がある．

ソフトウェアウェブロボットとは，ウェブのハイパーテキスト構造を走査し，参照されている全文書を再帰的に調査するコンピュータプログラムである．それらは，プログラムであるが故に，人間では難しいサーバに対する連続的なデータアクセス要求を行う事が可能である．それにより，ソフトウェアウェブロボットはインターネットのネットワークバンド幅やウェブサーバの計算能力を超えてしまう可能性がある．

本稿では，我々が開発を進めている PLANET モーバイルオブジェクトシステムを用いて，ウェブ環境においてソフトウェアロボットをモーバイル化するための方法について述べる．モーバイルソフトウェアロボットは，少なくとも次の点で従来のソフトウェアロボットに対して利点がある．第一に，いくつかの場合は，HTML ファイルの一部や何らかの処理結果を転送するだけで十分なので，ソース HTML ファイル全体をインターネットを介して転送する必要はない．第二に，ウェブサーバは訪問してきたモーバイルロボットの実行をスケジューリングする事で，ロボットの要求を処理する度合をスケジュールする事が可能である．本稿では，提案した方法に基づいてプロトタイプを実装し，従来のソフトウェアロ

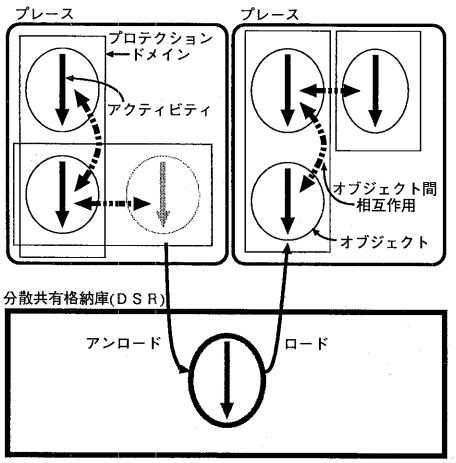

図 1: PLANET のシステムモデル

ボットとの比較を行う．

以下，第 2 章では，モーバイルオブジェクトシステム PLANET について簡単に述べる．第 3 章では，モーバイルソフトウェアロボットの実装に用いた言語 PLANET/C++ について簡単に述べる．第 4 章では，PLANET 上でどのようにモーバイルソフトウェアロボットを実現するかについて述べる．第 5 章では，実装したモーバイルソフトウェアロボットを用いて，現在の方式のロボットと比較実験を行う．最後に第 6 章で，まとめと今後の課題について述べる．

2 PLANET の基本概念

PLANET のシステムモデルは，オブジェクト，DSR (Distributed Shared Repository; 分散共有格納庫の略)，アクティビティ，プレース，そしてプロテクションドメインという 5 つの基本抽象概念を用いて説明される (図 1)．

- **オブジェクト** — 処理の対象であるデータと，そのデータに対する操作を一つにまとめたもの．
- **アクティビティ** — 計算状態を抽象化したもので，オブジェクト上を走ることで計算が進む．「スレッド」と似ているが，アクティビティはアドレス空間の壁を越えて動くことが可能である点や，永続性を付加できる点が異なる．
- **DSR** — オブジェクトを永続化しておく場所で，広域ネットワークと永続的記憶空間を統合・抽象化したもの．広域ネットワークにより接続され，磁気ディスク装置などの永続的記憶装置を持つサイト群によって実現されることを想定している．
- **プレース** — オブジェクトが処理を行う場所で，ローカルエリアネットワークと揮発的記憶空間を統合・抽象化したもの．ローカルエリアネットワークにより接続されたサイト群によって実現されることを想定している．
- **プロテクションドメイン** — プレース上のオブジェクト保護の単位．一つのプロテクションドメインは必ず一つのプレースに属し，同時に二つ以上のプレースに属することは出来ない．メモリ保護やアクセス制御の保護のポリシーなどはプロテクションドメイン単位で設定される．

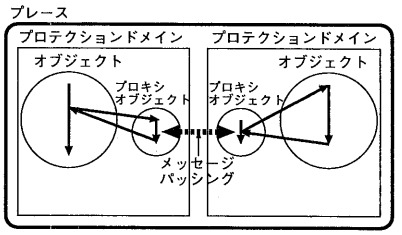

図 2: プロキシを用いた遠隔手続き呼び出し

PLANET のシステムモデルでは，オブジェクトは DSR またはプレースのどちらかに存在している．DSR 内にあるオブジェクトは，DSR 内で一意で位置独立な名前で識別される．

オブジェクトを DSR からプレースに移動する操作をロード操作，その逆にプレースから DSR へ移動する操作をアンロード操作と呼ぶ．ネットワーク全体の環境においては DSR 空間は一つしか存在しないが，プレースはいくつも存在する．オブジェクト間の相互作用はオブジェクトがプレースにロードされている時のみ可能である．

一般に，オブジェクトを常に同じアドレスに配置することはできない．そのため，PLANET ではオブジェクト内のアドレス情報を動的に再配置することにより，プロテクションドメイン内の任意の位置にオブジェクトを配置することを可能としている．

オブジェクトはプレース上にロードされる時に一つ以上のプロテクションドメインに関連付けられていなければならない．プレースにロードされて最初に関連付けられたプロテクションドメインを，そのオブジェクトのホームプロテクションドメインと呼ぶ．オブジェクトとホームプロテクションドメインの関係は，そのオブジェクトが DSR へアンロードされるか，そのオブジェクトが消滅するまで変わらない．

オブジェクトをプロテクションドメインに関連付ける操作をアタッチ操作と呼ぶ．オブジェクトを複数のプロテクションドメインにアタッチした場合，オブジェクトのコピーが複数のプロテクションドメインに同時に存在することになる．ユーザは，システムが提供する同期プリミティブを使い，オブジェクトの一貫性の管理を行うことになる．

オブジェクトは，アタッチされたプロテクションドメインが持つアクセス制御の下で実行される．各々のプロテクションドメインは独立した仮想記憶空間を持ち，アクセスを不正に行えないように厳密に保護されている．

同じプロテクションドメインに存在するオブジェクト同士の相互作用は，直接的かつ効率良く行うことができる．仮想記憶空間の切り替えやシステムコールなどを行う必要がないのがその理由である．

3 PLANET/C++

PLANET は，ネイティヴコードでモーバイルオブジェクトを実現している．そして，OS や言語に依存しないように設計されている．その特徴を生かす為，PLANET/C++ は PLANET 上での C++ による

モーバイルオブジェクトの記述を，できるだけ C++ 言語に変更や制限を加えずに行えるようにしたものである．

PLANET は medium-grain なオブジェクトを想定しているシステムなので，C++ のオブジェクトを全て PLANET のオブジェクトとすると，オブジェクトの粒度が小さすぎてしまう．そこで，PLANET のオブジェクトを生成するクラスと，C++ のオブジェクトを生成するクラスとに分け，C++ オブジェクトは PLANET オブジェクトの内部で生成と消滅が行われるようにしている．

3.1 オブジェクトのメソッド呼び出しと保護

PLANET では，実行時にプロテクションドメインによってオブジェクトの保護を行うため，メソッド呼び出しはオブジェクト同士が同じプロテクションドメインへアタッチされているか否かで 2 種類が存在する．ここで，もし同じプロテクションドメインにおけるメソッド呼び出しと異なるプロテクションドメインにおけるメソッド呼び出しを，別の枠組みで別の命令を用いて行うようにすると，相手のオブジェクトが同じプロテクションドメインに属しているかどうかでメソッド呼び出しの仕方をプログラマが別々に記述しなければならなくなり，プログラムの再利用性などを損なうことになってしまう．

また，C++ では，オブジェクトを定義する際にオブジェクト内の変数やメソッドをオブジェクト外部から参照可能にするか不可能にするかを記述できる．このオブジェクトの隠蔽概念によって，オブジェクト内部のアクセスされては困る変数へのアクセスをプログラミング時に禁止し，オブジェクトの保護を行っている．

PLANET では，オブジェクト指向プログラミングにおけるオブジェクトの隠蔽概念とは直交的にプロテクションドメインによる保護を行う．隠蔽概念と保護概念が直交的であるとは，オブジェクトの隠蔽概念とは独立に，オブジェクトを任意の保護のドメインに割り当て可能で，その割り当て状態に依存せずにオブジェクトのメソッド呼び出しが可能であることを言う．つまり，オブジェクトがプレースにロードされる時にどのプロテクションドメインにアタッチされようとも，オブジェクトのメソッド呼び出しはソースレベルでは同じ記述で行えることを意味する．これにより，PLANET/C++ プログラマはプロテクションドメインを意識せずにプログラムを記述出来る．

3.2 直交的プロテクションドメインの実現

異なるプロテクションドメインにアタッチされたオブジェクト同士は，全く別の仮想記憶空間に存在するので，直接メソッド呼び出しを行うことは出来ない．そのため，この場合には何らかの通信機構を用いて遠隔手続き呼び出しを行うことでメソッド呼び出しを実現する必要がある．

PLANET では，メソッド呼び出しの相手オブジェクトに対応するプロキシオブジェクトをシステムが各々のプロテクションドメインにアタッチすることにし，オブジェクトはそれに対してメソッド呼び出しを行うことで遠隔手続き呼び出しを行っている (図 2)．プロキシオブジェクトとは，メッセージパッシングのためのプリミティブ呼び出しを行い，遠隔手続き呼び出しを実行するオブジェクトである．プロキシオブジェクトのプログラムコードは，PLANET/C++ コンパイラが対応オブジェクトの記述を基にして自動生成する．

そのため，PLANET/C++ プログラマは，プロテクションドメインの存在を気にすることなくプログラムを書くことができ，かつ，作られたオブジェクトは変更すること無く，実行時に任意のプロテクションドメインにアタッチして実行することができる．

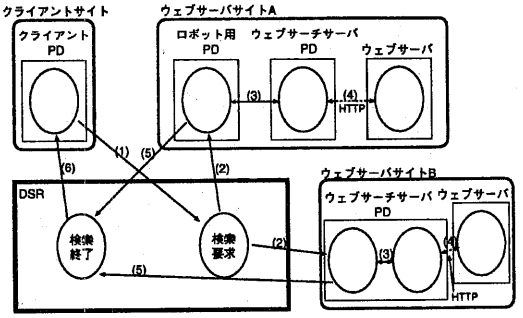

図 3: PLANET 上のモーバイル WWW ロボット (PD — プロテクションドメイン)

4 WWW ソフトウェアロボットのモーバイル化

一般的に，現在のウェブサーチロボットはロボットが動作しているホストから目的のウェブサーバが動作しているホストへ通信を行い，HTTP を用いて必要な HTML ファイルを取得し，それを解析するという動きを繰り返し行っている．

それに対し，PLANET 上のモーバイルサーチロボットは，図 3のように動作させることにする．ここで，ウェブサーチロボット，ウェブサーチサーバ，ウェブサーバの各用語を以下のように定義しておく．

- ウェブサーチロボット — ウェブサーチサーバと相互作用を行うことで目的の HTML ファイルを取得して処理を行うロボット．ロボット自身は HTTP を用いてのウェブサーバとの通信は行わない．
- ウェブサーチサーバ — ウェブサーチロボットの代わりに HTTP を用いてウェブサーバと通信し，その結果をウェブサーチロボットへ渡すサーバ．
- ウェブサーバ — 一般的な HTTP サーバのことを表す．

PLANET 上のロボットは以下のように動作する．図 3の括弧中の数字は，以下の処理順序に対応している．

1. ウェブサーチロボットを送り出すプログラムは，目的とするウェブサーチサーバが指定する名前でロボットを DSR へ置く．

2. ウェブサーチサーバは，ロボットが指定した名前で現れたらDSRから取り出して，処理を行わせるためにプロテクションドメインへ入れる．その際に，2通りの方法がある．一つは，そのロボットが信頼できる場合に行える方法だが，ロボットをウェブサーチサーバと同じプロテクションドメインへ入れる方法である．この場合はロボットとサーバが直接相互作用を行える為，オーバヘッドはかからない．しかし，ウェブサーチサーバはロボットから保護されない為，ロボットがサーバに対して手を出すことが可能になる．もう一つは，そのロボットが信頼できない場合の方法だが，ウェブロボットサーバとは異なるプロテクションドメインを生成し，そこでロボットを実行させる方法である．この場合は，ロボットとサーバ間で遠隔手続き呼び出しを用いて相互作用を行うのでオーバヘッドがかかる．しかし，ロボットによりサーバが書き換えられる等の影響は避けることができる．

3. ロボットは，サーチサーバに対して相互作用を行い，必要な情報(HTMLファイル等)を送るように要求を出す．

4. ウェブサーチサーバは，HTTPを用いてウェブサーバに要求を出し，ロボットから要求された情報をロボットに返す．

5. ウェブロボットは，そのサイトでの処理が済んだら，異なるサイトへ行く場合はそのサイトが指定する名前で，ロボットを送り出したサイトへ戻る場合には送り出したプログラムが指定した名前でロボット自身をDSRへ置く．

6. ロボットを送り出したプログラムは，ロボットに対して指定した名前でロボットが戻って来たら，DSRから取り出して結果を得る．

従来のロボットと比較して，この方法を用いた場合のロボットには以下のような利点がある．

- ネットワークの負荷 — ウェブロボットを送り出したサイトと，ウェブサーバが存在するサイト間では，ロボットの移動時にのみネットワークが使われる．従来のロボットでは，必要なHTMLファイル全てが両サイト間のネットワークを用いて転送されているので，その場合と比較すると両サイト間のネットワークの負荷を軽くしていると言える．

- 保護 — インターネットのような開かれたネットワークから来るロボットを実行するのは，かなり危険が伴う．しかし，プロテクションドメインで監視しながら実行を行うため，ロボットが問題のある行動をしようとした時にそれを止めることが可能である．

- 非同期性 — PLANETは，システムレベルで永続性を保証しているので，ウェブロボットを記述する際には相手のウェブサーバが停止している場合を考慮する必要はなく，ただ相手のサイトにロボットを送ればよい．それに対し，従来のロボットは，相手のウェブサーバが停止していた場合にはそのサーバに対する処理を中断し，後でサーバが再開したら処理を再開するように記述しなければならない．

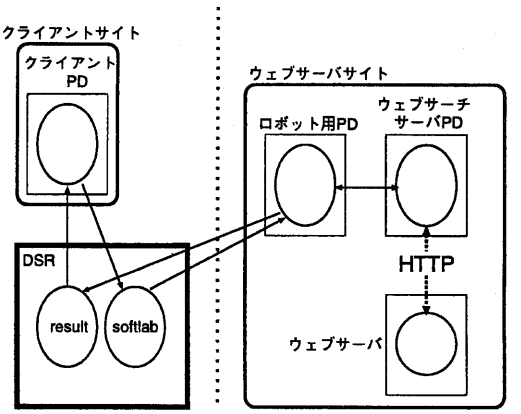

図 4: 実験時のオブジェクトの動き (PD — プロテクションドメイン)

	クライアントサイト	ウェブサーバサイト
CPU	UltraSPARC (167MHz)	UltraSPARC-II (296MHz)
メモリ	64MB	128MB
OS	Solaris 2.5.1	Solaris 2.5.1

表 1: 実験に使用した計算機の仕様

5 実験

前章で述べたロボットをPLANET/C++を用いて実装し，従来の方法におけるロボットとの実験結果の比較を行った．又，ネイティヴコード実行とバイトコードインタープリタ方式の違いを見るため，Javaを用いて実装されたロボットでも同様の実験を行い，PLANET上のロボットとの実験結果の比較も行った．

5.1 実験条件

以下の条件で実験を行った．

- ロボットは，HTMLファイルから張ってあるリンクを単に全て収集するものを実装した．

- サーチロボットを送り出すサイトを東京大学(東京都文京区)に，送り出されたロボットを受け取るサイトを筑波大学(茨城県つくば市)にし，筑波大学のウェブサーバ(http://www.softlab.is.tsukuba.ac.jp/)に対して探索を行わせた(図4)．各計算機の仕様は表1である．

- Javaでモーバイルウェブロボットを記述するために，Voyager [2, 8]を利用した．用いたJDKはJDK1.1.5である．

- 従来の方法におけるロボットでの実験は，東京大学から筑波大学のウェブサーバに対して行わせた．

試行	実験 1	実験 2	実験 3
1	1,117.83	705.12	643.55
2	1,141.59	851.26	783.66
3	1,132.67	865.74	819.42
4	1,135.64	868.50	816.09
5	1,139.88	866.45	818.17
6	1,132.17	866.65	814.04
7	1,139.15	892.86	834.52
8	1,188.18	875.25	843.35
9	1,136.92	886.03	820.03
10	1,157.45	878.32	827.53
平均	1,142.15	855.62	802.04

表 2: 実験結果 1:(単位は秒)

又，今回は筑波大学 1 か所のウェブサーバのデータを収集し終えたら，ロボットを送り出したサイトにすぐに戻るようにしてある．

5.2 従来方式のロボットとの実験結果の比較

実験結果を表 2に示す．又，表 2をグラフにしたのが図 5である．実験の時点では，ソフトウェア研究室のウェブサーバには約 4,500 個のファイルがあり，ロボットの走査対象になる HTML ファイルの分量は約 19Mbyte あった．それに対して，ロボットが走査後に持ち帰る URL データの量は約 320KByte であった．

表中の実験 1，2，3 は以下の条件で行った結果である．

- 実験 1 — 従来のウェブロボットの方式と同様に，東京大学でロボットを動かし，直接筑波大学のウェブサーバと HTTP を用いて通信を行った場合の実験
- 実験 2 — 前章で述べたモーバイルウェブロボットを用いた実験．東京大学から送り出されたロボットを，筑波大学側で受け取ってウェブサーチサーバと異なるプロテクションドメインに入れて実行した場合の実験
- 実験 3 — 前章で述べたモーバイルウェブロボットを用いた実験．実験 2 と異なり，筑波大学で受け取ったロボットを，ウェブサーチサーバと同じプロテクションドメインに入れて実行した場合の実験

結果を見ると，実験 1 と実験 2，3 との差が約 25% から 30% と開いていることが分かる．これは，ロボットを移動させることで，19Mbyte のデータを広域ネットワークを介して転送させることを避け，320Kbyte のデータ転送で済ませたことに対する差であると考えられ，モーバイル Web ロボットの優位性を示す結果と言える．

又，実験 2 と実験 3 の間にも約 5% 程差が存在する．これは，実験 3 が同じプロテクションドメイン内にロボットとサーチサーバが存在するので直接的に相互作用が可能であるのに対し，実験 2 は異なるプロテクションドメインに存在するので相互作用が遠隔手続き呼び出しで行われていることによるオーバヘッドであると考えられる．

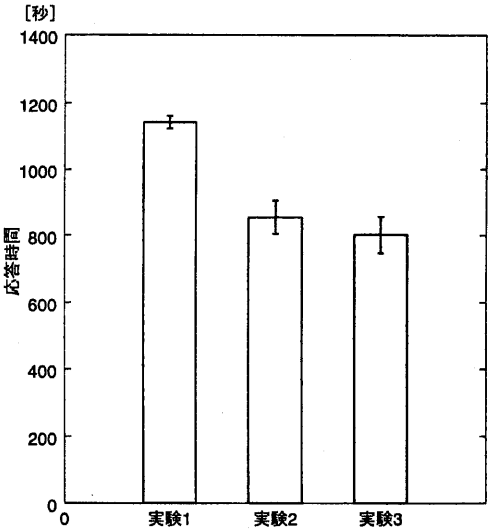

図 5: 実験結果 1

5.3 Java で実装されたロボットとの実験結果の比較

Java を用いて実装されたロボットの実験結果を表 3に示す．又，表 2と表 3をグラフにしたのが図 6である．

表中の実験 4，5，6 は以下の条件で行った結果である．

- 実験 4 — 従来のウェブロボットの方式と同様に，東京大学で Java で記述されたロボットを動かし，直接筑波大学のウェブサーバと HTTP を用いて通信を行った場合の実験．先の実験 1 と対になる実験である．
- 実験 5 — 前章で述べたモーバイルウェブロボットに対する比較のための実験．東京大学から送り出された Java で記述されたロボットを，筑波大学側で受け取ってセキュリティマネージャによるセキュリティチェックをかけた場合．先の実験 2 と対になる実験である．
- 実験 6 — 前章で述べたモーバイルウェブロボットに対する比較のための実験．実験 5 と異なり，筑波大学で受け取ったロボットに対してセキュリティチェックをかけずに実行した場合．先の実験 3 と対になる実験である．

結果を見ると，実験 1 と実験 4 の差ははっきりと出ている．つまり，現在のウェブロボットの領域における PLANET の優位性を示しているといえる．又，実験 3 と実験 6 との差は 12% となっている．この方式はセキュリティチェックを行っていない為，LAN 内での使用のみで保護を行う必要が無い場合である．最後に，セキュリティチェックを行っているため最も利用されると考えられる実験 2 と実験 5 の差は，約 23% であり，この差が無視可能ならば Java，PLANET どちらも選択可能である．Java には，現時点では PLANET にはない幅広いインター

試行	実験 4	実験 5	実験 6
1	1,539.08	1031.06	908.46
2	1,545.04	1038.95	884.85
3	1,556.13	1012.68	858.53
4	1,587.28	1031.07	881.38
5	1,664.99	1068.37	865.79
6	1,617.30	1110.64	886.47
7	1,635.15	1047.53	865.51
8	1,847.00	1046.91	872.50
9	1,689.34	1057.99	968.19
10	1,663.66	1047.56	966.51
平均	1,634.50	1049.28	895.82

表 3: 実験結果 2:(単位は秒)

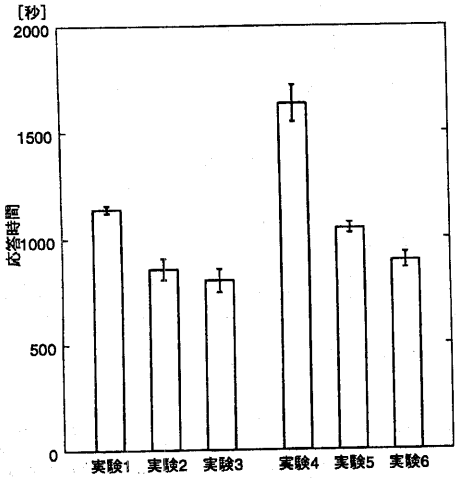

図 6: 実験結果 2

オペラビリティがある．しかし，PLANET にはネイティヴ実行や，現在は Java に無い非同期的なオブジェクトパッシングメカニズム等があり，異機種環境に対する対応も進めているので，十分選択肢となり得ると考える．

6 おわりに

本稿では，モーバイルオブジェクトシステム PLANET 上において，ウェブソフトウェアロボットをモーバイル化するための方法について提案した．また，モーバイルオブジェクト言語 PLANET/C++ を利用して WWW モーバイルソフトウェアロボットのプロトタイプを実装し，従来方式のロボットとの比較実験を行った．

今後の課題としては，第一に，今回の実験においてはロボットを一つだけ用いたが，複数個のロボットを並列動作させた場合の影響についての研究，第二に，移動してきたロボットに対して，計算機資源を必要以上に使い過ぎないように制限をかけられるようにする方策の研究が挙げられる．

参考文献

[1] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. The Harvest information discovery and access system. In *Proc. of the 2nd Int. World Wide Web Conf.*, pages 763–771, Oct. 1994.

[2] G. Glass. ObjectSpace Voyager: the agent ORB for Java. In *In Proc. of 2nd Int. Conf. on Worldwide Computing and Its Applications*, volume LNCS-. Springer-Verlag, 1998.

[3] K. Kato, K. Matsubara, K. Toumura, S. Aikawa, and Y. Someya. Object passing and interaction mechanism of the PLANET mobile object system. In *Proc. of France-Japan Workshop on Object-Based Parallel and Distributed Computation*, 1997.

[4] K. Kato, K. Toumura, K. Matsubara, S. Aikawa, J. Yoshida, K. Kono, K. Taura, and T. Sekiguchi. Protected and Secure Mobile Object Computing in PLANET. In *Special Issues in Object-Oriented Programming*, pages 319–326. Dpunkt-Verlag, 1997.

[5] K. Kato, Y. Someya, K. Matsubara, K. Toumura, and H. Abe. An Approach to Mobile Software Robots for WWW. Technical Report ISE-TR-98-154, Institute of Information Sciences and Electronics, University of Tsukuba, 1998. Submitted for publication.

[6] M. L. Mauldin. Measuring the Web with Lycos. In *Proc. of the 3rd Int. World Wide Web Conf.*, Apr. 1995.

[7] O. McBryan. GENVL and WWWW: Tools for taming the Web. In R. Cailliau, O. Nierstrasz, and M. Ruggier, editors, *Proc. of the 1st Int. World Wide Web Conference*, 1994.

[8] Inc. ObjectSpace. http://www.objectspace.com/.

[9] B. Pinkerton. Finding what people want: Experiences with the WebCrawler. In *Proc. of the 2nd Int. World Wide Web Conf.*, 1994. http://webcrawler.com/WebCrawler/WWW94.html.

[10] R. Seltzer, E. J. Ray, and D. S. Ray. *The AltaVista Search Revolution.* McGraw-Hill, 1997.

[11] 東村邦彦, 染谷祐一, 加藤和彦. モーバイルオブジェクトシステム PLANET におけるオブジェクトの隠蔽概念と保護の直交的な導入. 日本ソフトウェア科学会主催 第 13 回オブジェクト指向計算ワークショップ WOOC97 論文集, 1997.

[12] B. Yuwono and D. L. Lee. WISE: a world wide web resource database system. *IEEE Trans. Knowledge and Data Engineering*, 8(4), Aug. 1996.