平成20年度 卒業論文

論文題目

著作権管理のための
モバイルエージェントの実装と応用

神奈川大学 工学部 電気電子情報工学科

学籍番号　200402774

杉山　陽一

指導担当者　木下宏揚　教授

目 次

図 目 次

表 目 次

第1章

序論

インターネットの爆発的な普及に伴って、ネットワークコンピューティングのソフトウェアー技術の重要性が高まり、エージェントの技術が注目されてきている。[6] 近年ではインターネットを用いた最近の情報システムがますます複雑になってきているため、新しいソフトウェア技術として「エージェント」が用いられるようになった。[14][15] 実際にエージェントは様々なネットワークアプリケーションの本質的な部分になりつつある。エージェントを用いることで、ネットワークの条件に変化があった際にも柔軟に対応しなければならないアプリケーションを、効率よく構築することができるためエージェントはネットワークコンピューティングのソフトウェアーである。[10]　本稿では, モバイルエージェントによる著作権管理という観点から考察する。例として電子商取引システムを用いて秘密情報を管理する。過去の卒業論文及び修士論文においては、モバイルエージェントによるコンテンツ保護の研究がされているが、個人の秘密情報は保護できない。モバイルエージェントには不正のホストからの攻撃など様々な問題点があり、個人情報の漏えいなどが起こりうる。[1] 電子商取引において不正なホストからユーザーの秘密情報を保護するために、ユーザー側のホストと仮想店舗側のホストを巡回させるエージェントを導入し第三者の役割をもたして管理する。モバイルエージェントの問題点の対策をすることにより、利用者の利便性の向上を図ることができるため、それをふまえモバイルエージェントの実装から応用までに至りたい。[2]

第2章

個人情報の保護

2.1　個人情報

2.1.1　個人情報とは

現代社会ではコンピュータの利用が一般的になり、様々な業務でデータの集積が進んでいるが、こうした情報が無制限に利用できるとなると、個人のプライバシーに関わる内容が第三者に容易に把握されてしまう危惧が高まってきた（例えば、クレジットカードの利用状況、出身校、勤務先、家族構成、通院歴など各種のデータが結合されてしまうと、個人の私生活が露わになってしまうおそれがある）。そのため、個人情報の取扱いに関心が高まり、規制が必要とされ、法制度の整備が行われてきた。しかし、職業上公開せねばならない情報も多々あるため現状ではあいまいな点がのこっている。しつこく相手の名前や住所を聞くとストーカーまがいな行動と思われるケースもある。[1]

個人情報には次のような条件を満たすべきである.

- 個人の氏名、性別、生年月日
- 住所、住民票コード
- 携帯電話の番号
- 勤務場所、職業、年収

- 家族構成、写真
- 指紋などの生体情報

などが該当する。上記のいずれかに該当しても、個人を特定することができなければ、個人情報には該当しない。例えば、年収と職業の 2 情報から、個人を特定することはできない。なお、生体情報については、技術の高度化に伴ってその個人特定性が徐々に強まる傾向があり、個人情報該当性の判断が難しい場合が見られる。メールアドレスについては、氏名が含まれるなどの場合には、明白に個人情報であるが、含まれない場合には個人情報ではないとする考え方もある。超流通のポイントは, ディジタルコンテンツ本体はカプセル化 (暗号化) されていて, それを解くための電子鍵を入手しない限り, 見ることも印刷することもできないということである. また, 電子鍵は自分のパソコンに固有の番号 (MAC アドレス) などと合成されて機能を発揮するため, たとえ電子鍵を他人に譲渡しても, 機能を果たさない. そのため, カプセルは自由にコピー, 譲渡が可能であるが, それだけを入手しても閲覧, 印刷は不可能ということになる. 現在, 超流通を基盤にしたシステムは, PC や携帯電話の音楽・映像配信, 車のカーナビゲーションシステムなど，様々な分野において応用されている.

個人情報の保護に関する法律の定義では、生存する個人に関する情報であって、当該情報に含まれる氏名、生年月日その他の記述等により「特定の個人を識別することができるもの」（他の情報と容易に照合することができ、それにより特定の個人を識別することができることとなるもの=例えば学籍番号など=を含む）をいう。

これらの個人情報は、現在ではコンピュータデータベースの形で記録されていることが多く、データが CD や DVD、USB メモリやハードディスクドライブなどのメディアに容易にコピーできるため個人情報漏洩が起こりやすい。

2.1.2 個人情報の漏えい

個人情報の漏えい事件は日々増加傾向にある。Webのアプリケーションのセキュリティホールを突かれた不正アクセスや、内部者による情報の持ち出しなど方法はさまざま。Security & Trustフォーラムが独自に行ったWebアンケートでも、情報漏えい対策は、不正侵入対策やウイルス対策を抑えて最も興味のあるセキュリティ関連分野となった。

また、一部施行されている個人情報保護法も2005年4月から全面施行となった。2005年4月から施行される第4章では、個人情報取扱事業者の義務が定められており、企業や自治体における個人情報保護対策が急がれている。2004年6月15日には経済産業省が個人情報保護法の適用についてまとめたガイドラインを策定した。個人情報取扱事業者は具体的にどのような対応を行えばよいのだろうかと検討している。先ほど述べたとおり個人情報の漏えいは種類が様々であり、すべてに共通する対策はない。例えばある企業の内部者による個人情報の流出はともかく、まずは不正なホストによる個人のクレジットカード番号等の個人情報の読み取り等、第三者による著作権の侵害や個人情報の流出を防ぐべきである。

もちろん情報の漏えいが発生した場合、被害者は当然個人情報を漏えいされた人物となる。一方、加害者はといえば、これは法的には漏えいを起こした人物となる。被害者は加害者に対してプライバシー権侵害による損害賠償請求が行える。また加害者には、捜査機関から不正アクセス禁止法などによる刑事告訴が行われる可能性が高い。

わが国における個人情報保護の法的枠組みは、大きく2つに分けられる。1つは、判例法により確立されたプライバシーの権利で、漏えい元の企業や組織には罰則はないものの損害賠償責任が課せられる。もう1つは個人情報保護法で、主務大臣に対する報告を怠ったり、虚偽の報告を行ったりすると罰則が科せられる。ただし、個人情報保護法では顧客に対する民事責任は定められていない。個人情報保護法は、あくまでも個人情報の適切な取り扱いに関するルールを定めたものであり、権利利益に対する侵害発生を未然に防ぐことを目的としている。

2.2 著作権管理

2.2.1 著作権について

著作権は特許権,商標権などの産業財産権とともに「知的財産権」と呼ばれる権利の一つである.産業財産権が産業経済の発展を目的としている制度であるのに対し,「著作権」は文化の発展を目的とし,音楽,絵画,小説,映画,コンピュータ・プログラムなどの著作物を保護することを目的としている.[5] 著作権法によると,「著作物とは,思想又は感情を創作的に表現したものであつて,文芸,学術,美術又は音楽の範囲に属するものであり,著作者はそれを創作する者を指す」(第2条) と定義している.

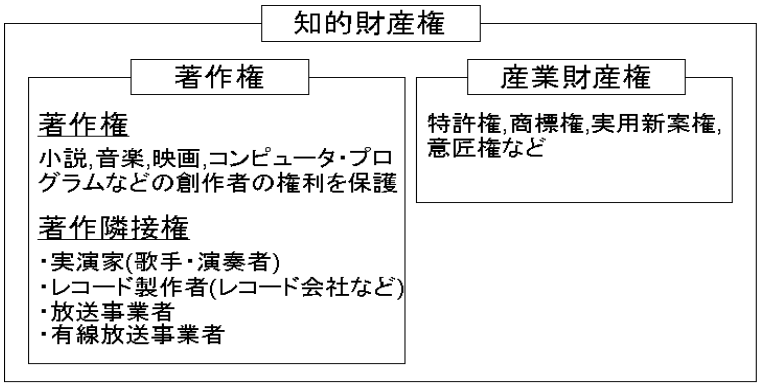

図 2.1 知的財産権

インターネットサイト、携帯電話向けサイト、データ放送等弊社が制作する一切のコンテンツは、弊社もしくは原権利者が著作権を保有しているため、これらのコンテンツを、許可なく複製する行為、またはインターネット上等で公衆が取得できる状態にする行為等私的利用の範囲を超えてのコンテンツの利用も著作権侵害にあたる。著作物の全部もしくは一部を権利者の許可なく改竄することも著作権法上禁止されているため。コンテンツの中で使用さ

れている映像、音声、写真、音楽等の著作物に関しても同様である。 こうした違法行為は刑事責任を問われたり、権利者から損害賠償を請求されたりする可能性があり、著作物の「引用」は、著作権法上許される行為にあたるが、「引用」と認められるには、「公正な慣行に合致」しており、同時に「報道、批評、研究その他の引用の目的上正当な範囲内で行われるもの」でなければならない。また主張したいことと引用される著作物との間に明確な主従関係の存在が必要であり、誰の何という作品かを表示する義務も課せられているため、安易な引用利用には十分注意する必要がある。[5]

2.3 プライバシー保護

プライバシーとは場所的・空間的領域概念であり、最も価値の高い部分である。プライバシー権とは、こうした空間に無断で介入することを拒否し、みずからの情報を提供することの可否を決定する権利（自己決定権）を包摂するものである。個人情報保護とは、管理されている情報の管理、利用、処分に関する基本的ルールであり、個人情報保護法とは、情報管理者規制・規律法であるため、法律的にはプライバシー保護法と個人情報保護法との二つが必要である。[13]

2.3.1 プライバシー保護への応用

プライバシー保護に必要な機能

- 流通している個人情報の利用条件の設定
- 個人情報の正当性の確認、修正、消去

著作権管理に必要な機能

- コンテンツの利用条件の設定
- コンテンツのバージョンアップ、有効期限切れによるコンテンツの消去

2.3.2 証券化商品への応用

複数の債権ん等を組み込んだ証券化商品に必要な機能

- 個別の証券の債権者を特定可能
- 個別の証券の評価額の自律的な評価

著作権の二次利用に必要な機能

- 二次著作物を構成する一次著作権物の権利者を特定可能
- 二次著作物の利用権の設定と調停が可能

2.4 モバイルエージェント

そもそもエージェントとはその語源そのものを辞書で調べると、「代理人」や「動作主体」といった意味がある。要は「ある動作の実際の行為者」ということを意味している。モバイルエージェントとは、ネットワーク上のコンピュータを移動しながらプログラム処理を進めることのできるプログラムである。モバイルエージェントが移動うする際にはそのプログラムの実行状態も移動されるという特徴がある。つまりは、移動先のコンピュータで移動前の実行状態から処理を継続可能なエージェントプログラムということである。[14]

2.4.1 エージェントの種類

エージェントには様々な種類があり、それぞれ用途によって変わってくる。エージェントの種類は主に以下の種類がある。

1. **自律エージェント**
   自己充足的であり、観測された環境に基づいて内部目標を達成するための行動を自己の判断で決定する

2. **知的エージェント**
   人工知能的機能を有するエージェントでユーザを補助し、繰り返し行うべきコンピューター関連のタスクをユーザーに代わって行う

3. **マルチエージェント**
   単体では目的を達成できず、複数のエージェントが相互作用をおよぼしながら動作する

4. **モバイルエージェント**
   ネットワークに接続されたコンピュータ間をプログラムが移動しながら処理を行う

またエージェントによって、様々な分野に分けられる。

- **人工知能分野**
  思考、意思決定、学習などの人間の知能を代行してくれるものはこの分野である。インテリジェントエージェントと呼ばれることが多い。

- **ヒューマンインターフェース分野」**
  ヒューマンインターフェースを高速化し、コンピュータを擬人化した高度なインターフェースや人工生物のような感情をもつコンピューターを目指した分野である。

- **分散処理システム分野**
  ネットワークでつながったプロセス同士が強調しあい、分散処理を実現する分野である。モバイルエージェントはこの分野に近い。

2.4.2 Plangent

モバイルエージェントのシステムとしてPlangent「プランジェント」というエージェントがある。Plangentは、モバイルエージェントと知的エージェントの特徴を併せ持つシステムで、ユーザーがネットワーク上で何をしたいのかを理解し、ユーザーの代理人となってタスクを遂行する。このときネットワーク

上での何処で何をすべきかといった行動計画はプランニングによってエージェント自身が決定するため、ユーザーは自分が必要とする情報やサービスが何処に存在し、どうアクセスすれば利用できるのかを知らなくてよい。Plangentでは、プランニングによって特徴となる知的動作を実現しており、その名前も「プランニングを行うエージェント」からきている。

2.4.3 一般的なモバイルエージェント

モバイルエージェントには大きく二つの種類に分けられる。

- **移動型エージェント**
  移動型のエージェントはエージェントそのものが移動するもので、利用者の代理として、ネットワーク上のホストを自律的に移動しながら特定のタスクを遂行するソフトウェアを示す。すなわち、そのソフトウェア自体が「動き回る代理人」を意味する。このエージェントは、ネットワーク上の複数のホストにまたがるタスクの遂行にあたって、利用者がそれぞれのホストに接続して作業していく代わりに代行してくれるものである。

- **利用者移動型エージェント**
  利用者移動型エージェントはモバイルコンピューティングの支援エージェントである。例えば携帯端末を持った利用者が外出先で移動を　繰り返しながら、行く先々で電話回線を使って遠隔地にあるサーバーに接続する状況で、利用者と接続先サーバーの間を仲介するものである。このタイプのエージェントは、その移動のために接続を切断している間、利用者のかわりに取り仕切ってくれる代理人を意味する。一般にモバイルコンピューティングにおいては、利用者とサーバーとの間の接続には、LANと比べて通信速度が遅くかつ不安定である。電話回線と、電話回線よりはまだましだが同様の傾向をもつ広域ネットワークが介在している。エージェントは電話回線を通してネットワークの内部のどこかのホストへとアップロード（派遣）される。このとき、エージェント内部に利用者からサーバーへのトランザクション郡を抱え込んで持ち運ぶ。利

用者は、エージェントをアップロードして起動してしまえばトランザクションの終了を待たずして、電話回線を切り、次の移動先へ移動することができる。このエージェントは、サーバーとの間に接続を確立してトランザクションを発行し、必要に応じて、利用者の代わりにサーバとの間でデータのやり取りをする。利用者は、次の移動先で電話回線からエージェントに接続する。利用者の移動中に、サーバでのサービスが終了していたら、エージェントが代わりに結果を受け取っておいてくれる。これによって、例えば利用者が飛行機に乗って移動している時間を有効に使うことができる。

以上の種類があり、一般的には利用者が移動せず、エージェントそのものが移動する移動型のエージェントをモバイルエージェントと呼ぶ。

2.4.4 なぜモバイルエージェントか

エージェント移動型エージェントは、ネットワーク上の複数のホストにまたがるタスクを遂行するのにあたって、利用者が接続して作業する代わりに代行してくれるものと考えられる。しかし、その目的のためだけなら、エージェントのコード自体がネットワーク中を移動する必要がなく、それぞれのホストと順次、遠隔通信機能を使って接続すればよい。接続先のホストに遠隔接続できるサービスが必要となるが、むしろその方が簡単である場合が多い。コード移送を行い、それぞれのホストでそのコードを実行するためには、セキュリティー保護、障害対策、管理の手間の増加といった問題が複雑化してしまう。つまりそうした問題を上回るメリットがなければならない。主にモバイルエージェントを用いる必要性が生じるケースは

- 1 不安定であったり、低速であったり費用の高い通信路が間に挟まる
- 2 遠隔地にある対象を長時間監視するための通信頻度が高い
- 3 負荷分散を行いたい

１では不安定であったり、低速もしくは費用の高い通信路の使用を極力避けることが目的である。そうした通信路を越えた相手側にモバイルエージェントを送り込むことで、その使用を減らすことができる。

２では通信路を越えた相手側にモバイルエージェントを送り込むことが行われる。一般に、監視を行う場合には、時間遠隔で検知を繰り返すポーリングループが行われる。しかしその時間間隔によっては通信の頻度が高くなりすぎてしまい、通信のトラックが増大してしまう、そこで、監視用のエージェントを監視対象と同じホストに派遣し同一ホスト内でポーリングループさせ、通知すべき状況になった時点で遠隔通信で通知すればよい。これによって遠隔通信を減らす事ができる。

３に関しては、対象となる問題に依存した制御が必要となる。モバイルエージェントを同時に複数ホストに派遣することで、一部のタスク遂行を並行化することができるため、負荷の分散ができているように見え、実際にその効果が得られる事が多い。特にたくさん分岐する経路に沿って探索するような場合には多重度が上がり、その効果も大きい。しかし大域的に負荷の検出とプランニングを行わなければ、逆にある特定のホストにモバイルエージェントが集中しすぎたり、エージェント間での通信や同期にコストが嵩むということもありえる。

2.5　モバイルエージェントの利点

モバイルエージェントの利点について、まずモバイルエージェントの特徴は次の通りである。

- 分散処理における通信遅延
- 通信回数の削減
- 負荷分散
- 耐故障性の向上

以上の特徴がある。また利点としては次のような例が挙げられる。

- **ローカルアクセス**
  情報量の多いデータベースから情報を検索する際、検索の結果全てをユーザーへ返すと回線に大きな負担がかかってしまうが、モバイルエージェントがデータベースサーバーから検索結果をフィルタリングして情報の選定を行い、ユーザーへ返すことで回線にかかる負担を軽減することが可能であり、情報量の多いマルチメディアの検索に有効である。

- **負荷分散**
  クライアントサーバーモデルが大規模な構成になった場合にはサーバーの分散化が必須であるが、特定のサーバーに負荷が集中したときに効率よくサーバー間あるいはクライアント間で分散させることが困難である。また機能がサーバーに集中しているため、サービスの追加変更、クライアントの追加削除システムを止めずに行う事が難しい。しかしモバイルエージェントを用いたシステムはクライアント、サーバーといった主従関係がなく、クライアントとサーバーの機能を自由に構成することができる。またモバイルエージェントはネットワークを移動するため、空いているリソースを探して処理を委託、サーバー機能を端末に一時的に委託したりといったことが可能になり、ネットワークのリソースを有効に利用することができる。インターネットの環境ではリソースが遊んでいる時間が多いにもかかわらず、サーバーのボトルネックでアクセス性が犠牲になっている場合が多い。ネットワーク上のリソース全体にわたって負荷分散が可能なモバイルエージェントシステムは重要な技術である。

- **故障性**
  モバイルエージェントは移動先のコンピューターでも処理を継続可能なため、所定時間後にシャットダウンするコンピュータ等において、耐故障性が向上する。

- **ネットワーク**例えば小型の携帯端末を使用する際には、不安定なネットワークであるため、通信先との通信回線削減が求められる。モバイル

エージェントの特性により常時接続の必要もなくなり、無駄な通信回線数を減らすことが可能である。利用者移動型タイプのモバイルエージェントにあたる。

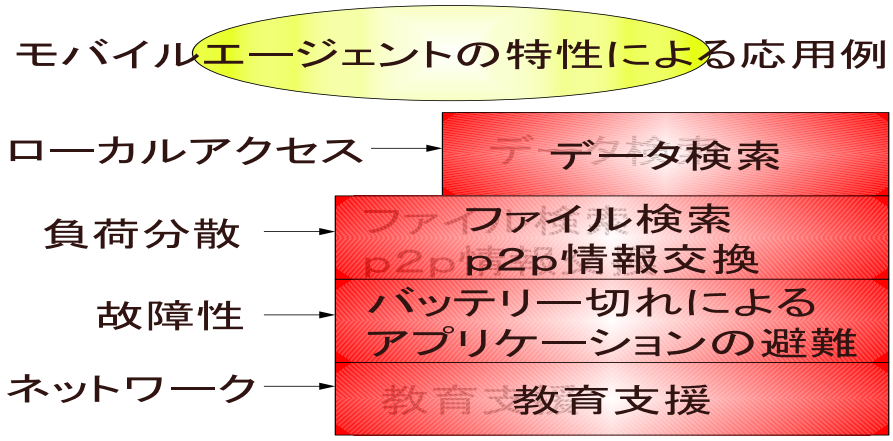

図 2.2 モバイルエージェントの特性による応用例

2.5.1 モバイルエージェントの問題点と対処法

モバイルエージェントは、計算機間を渡り歩くようなソフトウェアなので、管理上の問題を抱えている。大きく以下に三つを取り上げる。[6]

- **大きさ**

  個々のエージェントの肥大化によって引き起こされる。エージェントがネットワークを移動しながら情報収集活動を行う場合、収穫した情報が蓄積していくことで、一つのエージェントの内部のインスタンスデータが大きくなり、移送の負荷を大きくしてしまうことがあり得る。その対処法としては、そうしたデータを適宜、通信機能を使ってステーショナリーエージェントに送りつけることで、持ち歩くことを避ける事ができる。しかし、この方式を採用することは、モバイルエージェントを使う意義を薄めてしまう。

- **量**

  量の問題は、あるホストに対する攻撃として、大量にエージェントを送り付けて仕事ができないようにしてしまうというものが考えられる。こうした、エージェントの人口爆発は、悪意がなくても起こり得る。

- **セキュリティー**

  セキュリティーの問題は

  – エージェントとホスト間

  – エージェント間

  – ホスト間

  以上があげられる。ホスト間のセキュリティーは一般的な通信における問題に帰着でき、エージェント間のセキュリティーも通信の問題に帰着できる。そのためエージェントとホスト間のセキュリティーがモバイルエージェント特有の問題である。

  エージェントは他人の管理している計算機に移動してプログラムを実行する。そのため、その中にトロイの木馬のように、別の悪意のあるプログラムが仕込まれていた場合には、プラットフォームを破壊したり、リソースを不正取得することができてしまう。それを回避するために、移動してきたエージェントの実行権限を設定したりする。　また、プラットフォームから見てエージェントはデータにすぎない。つまりエージェント間通信も盗聴や改ざんの対象になりうる。セキュリティーについての問題の解決手段としては、暗号化が考えられる。

2.6　暗号化

暗号技術は、通信したいデータに対し、あるパラメータのついたプログラムで加工することにより、鍵なしではもとのデータを容易に得ることが不可能なデータを作成するものである。これらは秘密鍵と公開鍵に分けられる。

まず秘密鍵では図に示すように、発信者と受信者が同じ鍵を使って暗号化、復号化を行うことにより、秘密を保った通信を行う。その際、鍵は通信者同士以外に知られてはならないので、秘密鍵と呼ばれる。このため、この方式には、いかに秘密を保ったまま同一の鍵を共するか、という鍵配送の問題がある。

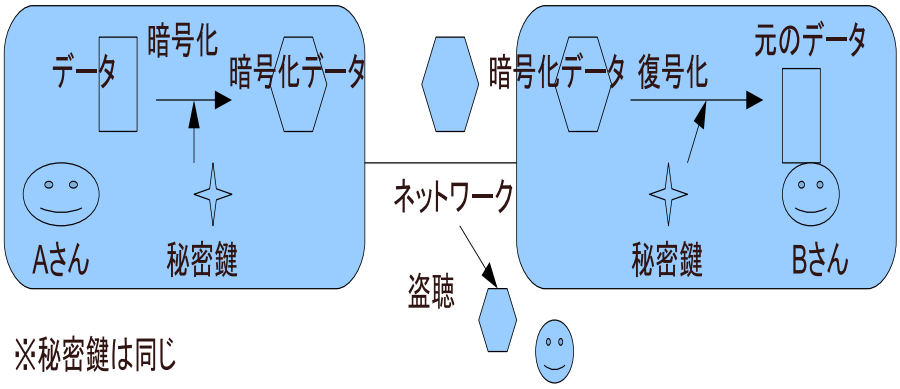

図 2.3　秘密鍵

公開鍵は図に示すように、発信者と受信者は異なる鍵を用いる。また、受信者の鍵（秘密鍵）から発信者の鍵（公開鍵）は容易に得られても、その逆は事実上不可能である。このために、たとえ公開鍵が広く知られても暗号文は解読されない、という特徴があり、前述の鍵配送の問題が解決できる。

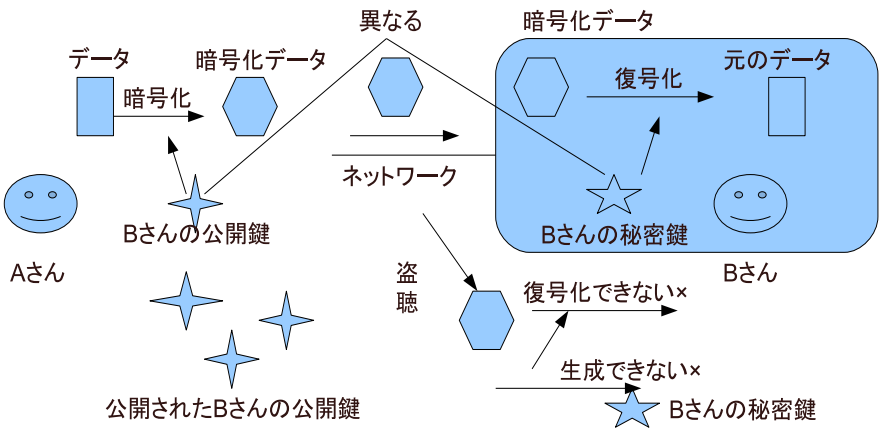

図 2.4 公開鍵

2.6.1 セッション鍵方式

セッション鍵方式 (ハイブリッド方式) は一般的な暗号方式で、共通鍵暗号方式と公開鍵暗号方式の長所をうまく組み合わせた暗号化方式である。セッション鍵方式では，メッセージの暗号化自体は共通鍵暗号方式で行うので高速処理が可能となる。また, 共通鍵を公開鍵暗号方式で暗号化して送信するので, 鍵の受け渡しが安全にできるなどのメリットがある.

＜セッション鍵方式の手順＞

1. 今回の通信だけに用いる使い捨ての共通鍵 (セッション鍵) を生成する
2. ”1”の共通鍵でメッセージを暗号化する
3. ”1”の共通鍵を受信側の公開鍵で暗号化する
4. ”2”と”3”をセットにして, 受信側へ送信する
5. 受信したデータから”3”の部分を取り出し, 自分の秘密鍵で復号することで共通鍵を得る
6. ”5”で得た共通鍵で”2”を復号する

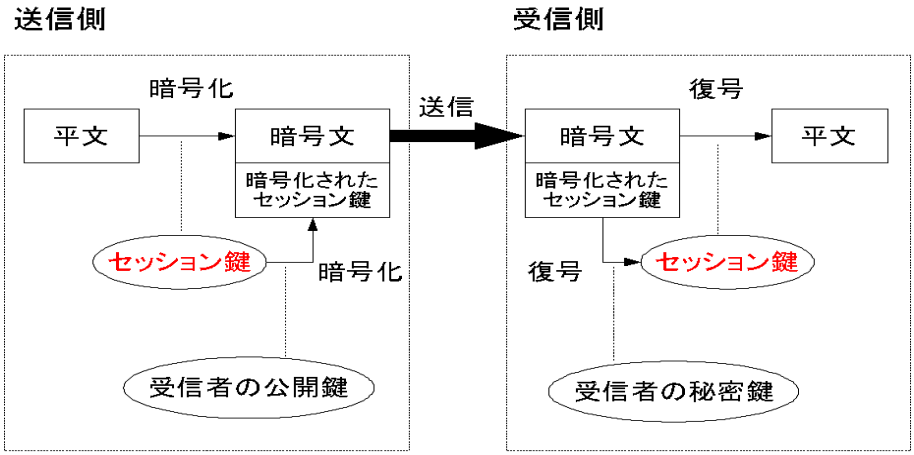

図 2.5 セッション鍵方式

2.6.2 SSL通信と電子証明書

SSL暗号通信とはSSLを利用することで、ネットワーク上で通信し合うクライアントPCとサーバの間で暗号化したデータをやり取りできるようになり、データの「盗聴」や「なりすまし」、「改ざん」、「否認」などさまざまなセキュリティ障害を防止できるようになるため、モバイルエージェントの通信経路をSSL通信化することでエージェントを保護することが可能である。
また電子証明書とはSSL暗号環境を構築する上で、必要不可欠であるサーバ証明書であり、電子証明書は誰でも作成可能だが、電子証明書の信頼性は認証局の信頼性に依存する。そのため本人確認が重要となる用途では、信頼のある認証局に電子証明書を発行してもらうことによって、データの出所を確実にすることが可能になる。

2.7 情報カプセル

コンテンツが不正複製されたり,暗号や耐タンパ性が破られたりすることでコンテンツが不正に流出した場合に,切り札となるのが電子透かし (digital watermarking) である. 電子透かしは,画像や動画,音声などのマルチメディアデータに,画質や音質にはほとんど影響を与えずに特定の情報を埋め込む技術であり,以下のような情報が埋め込まれる.[6]

1. **著作権情報**
   コンテンツの著作者名,使用条件などの著作権情報もしくはその管理 ID
2. **使用者情報**
   コンテンツの購入者やユーザごとに異なる,ユーザ ID などの情報
3. **コンテンツの識別情報**
   コンテンツを機械的に識別するための ID
4. **改ざん検出用情報**
   コンテンツが改ざんされているか否かを判断するための情報
5. **制御情報**
   コピーやその他の処理に関する制御情報

2.7.1 カプセル化

カプセル化 (encapsulation) とは,オブジェクト指向プログラミングが持つ特徴の一つであり,データとそれを操作する手続きを一体化して「オブジェクト」として定義し,オブジェクト内の細かい仕様や構造を外部から隠蔽することである. [6] 外部からは公開された手続きを利用することでしかデータを操作できないようにすることで,個々のオブジェクトの独立性が高まる. カプセル化の利点としては以下のようなものが挙げられる.[4]

- 不正な操作からの保護
- 複雑さの隠蔽
- 部品化/再利用性の向上
- 修正/変更に対する影響範囲の極小化
- バグの影響範囲の極小化

P2P ネットワークの普及などによるコンテンツの不正流通が社会問題となっている中, カプセル化はコンテンツを保護する手段として非常に有効である. カプセル化コンテンツ自体は超流通的に自由にコピー・転送され, 電子鍵と組み合わせることによって, コンテンツの閲覧・再生が可能となるという仕組みにすることで, 実質的にコンテンツの権利を保護することが可能となる.

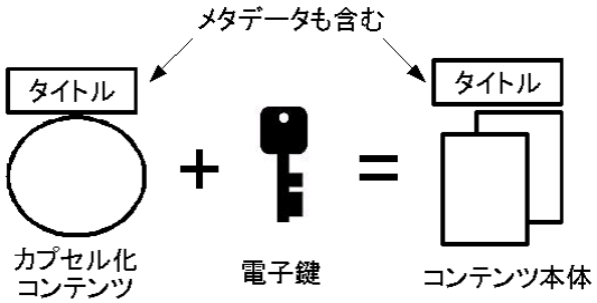

図 2.6 カプセル化コンテンツ

2.8 移動のメカニズム

モバイルエージェントは、ネットワーク上のホスト間を移動しながらタスクを遂行する。 通常、実行権限、セキュリティといった理由から、移動先のホストには、移動してきたモバイルエージェントを受け付けるプラットフォームプロセスがあらかじめ起動されてる事を前提とするものが多い。すなわち、エージェントホスト間を移動するというよりも、そのプラットフォーム間を移動すると言うほうが正確である。

＜モバイルエージェントの移動手順＞

1. 停止
2. 不活性化（データ化）
3. 移送
4. 再活性化（エージェント化）
5. 起動

となる。

階層アーキテクチャを考えると、例えば、データ化されたエージェントのデータ移送は、プラットフォーム層よりも下位層のトランスポートに任される。例えば、移送中のエージェントがもつデータが盗聴されないように、データ移送の前後に暗号化／複合化したりすることは、このトランスポート層もしくはその上のセキュリティー層で扱えばよい。セキュリティー層では、他に認証なども任せることができる。また複数のプラットフォーム間の相互運用を考えるのであれば、間に相互運用の層を設けるとよい。

2.8.1 継続実行

エージェントをデータ化するにあったっては、転送するためのデータとして、どこまでを対象とするかによって、何通りかに区別することができる。対

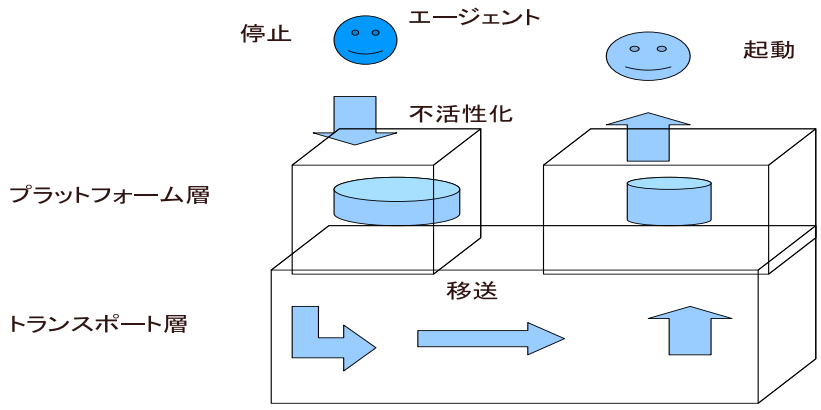

図 2.7 階層アーキテクチャ

象の候補には、エージェントプログラムのコード（メタ情報）、内部変数値（インスタンスデータ）、実行時内部データがある。コード（メタ情報）もさらに分けるなら、構造定義と振る舞い定義に分けられるが、オブジェクト指向をベースとするならば、この二つはクラス定義に集約される。実行時内部データは、例えばバーチャルマシンのもつプログラムカウンタや内部スタックなどに相当するものである。　これはエージェントの継続実行レベルに関係する。エージェントは移動前の状態を使って、移動後の処理を継続的に行うことができる。しかし、実際に実装されているモバイルエージェントのプログラミング言語を見ると、その継続のレベルによって二つに分類される。

1. **深い継続実行**

   深い継続実行は、エージェントの振る舞いを記述したエージェントのプログラムのコードと、インスタンスデータ、実行時情報のすべてをデータ化し移送する。これを強い移送といい、移送したデータからエージェントに再活性することで、移動前に実行していたプログラムの続きから、移動後に継続して実行できる。より具体的な、命令的なエージェント言語の場合には、移動前にｎ行まで実行していたとすると、プログラムカウンタの値ｎを移送しているので、移動後にｎ＋１行目から再開することができる。

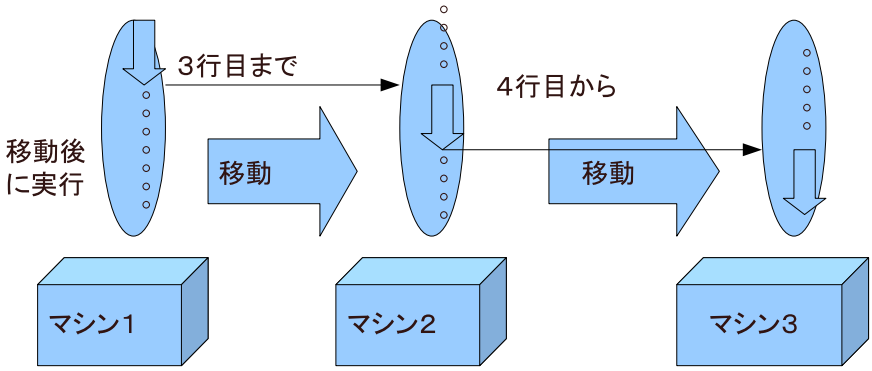

図 2.8 深い継続実行

2. **浅い継続実行**

浅い継続実行は、エージェントの振る舞いを記述したエージェントプログラムコードと、インスタンスデータはデータ化して移送するが、実行時情報は移送しない。これを弱い移送といい、プログラムカウンタを移送しないので、移動前に何行目まで実行していたかという情報は伝わらない。移動するたびに、移動前にどこまで実行していたかにかかわらず、ある特定のプログラム例を一行目から実行する。通常は、そうしたプログラムをとして、現在の所在地ごとに異なるコード断片を実行するような分岐文を与えるのが一般的である。プログラムカウンタを移送しないとはいえ、インスタンスデータは移送前の値を継続して使うことができるため、継続実行の一種として浅い継続実行と呼ぶ。

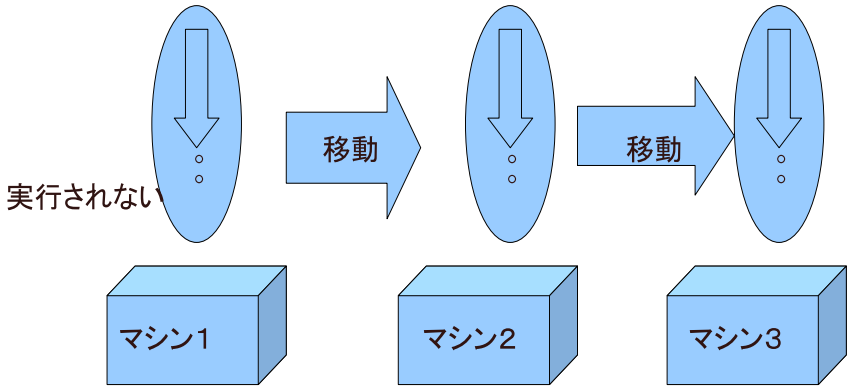

図 2.9 浅い継続実行

2.9 Agent Speace

モーバイルエージェントの研究開発用フレームワークを目的に開発されたシステムで、モーバイルエージェントシステムの動作原理の解析や、システム改造が容易で、他のモーバイルエージェントシステムと比較して、エージェント移動が高速であること、エージェントが自己完備化された計算実体であることなどの特徴がある。[15]

以下に例として、 AgentSpace を用いた移動型エディタープログラム「mail agent」の実行結果を示す。この例では, 同一コンピュータ上の異なるポート番号に対し, コードや状態が転送されている。

1. 通信ポート 5000 番において「mail.agent」を起動し、文字を入力し移動先のコンピュータの IP アドレスを指定して送る。

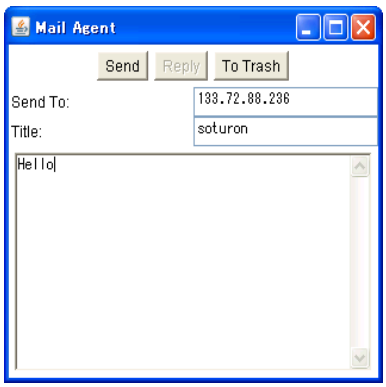

図 2.10 Mobile Agent Monitor (Port：5000)

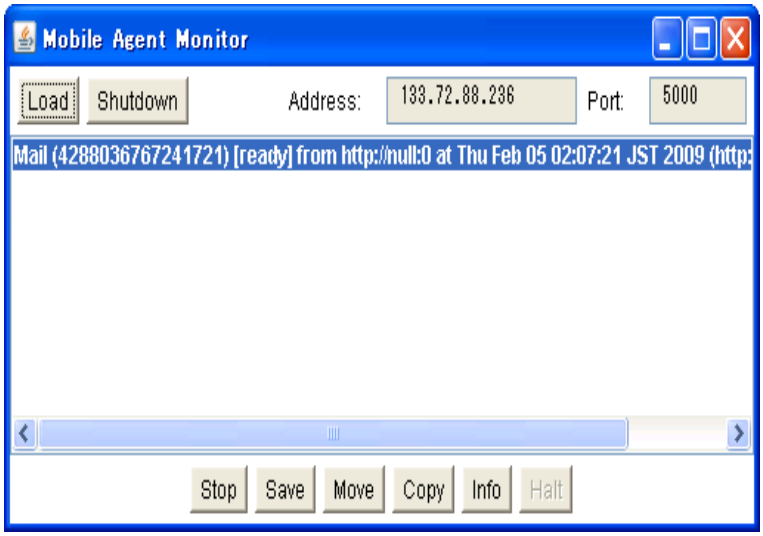

図 2.11 ”mail agent”の画面

2. 通信ポート 5001 番に「mail agent」の内部状態 (入力された文字, ウィンドウサイズ, 位置など) がコードと共に転送される.

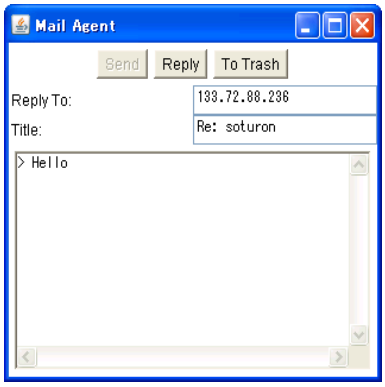

図 2.12 Mobile Agent Monitor (Port：5001)

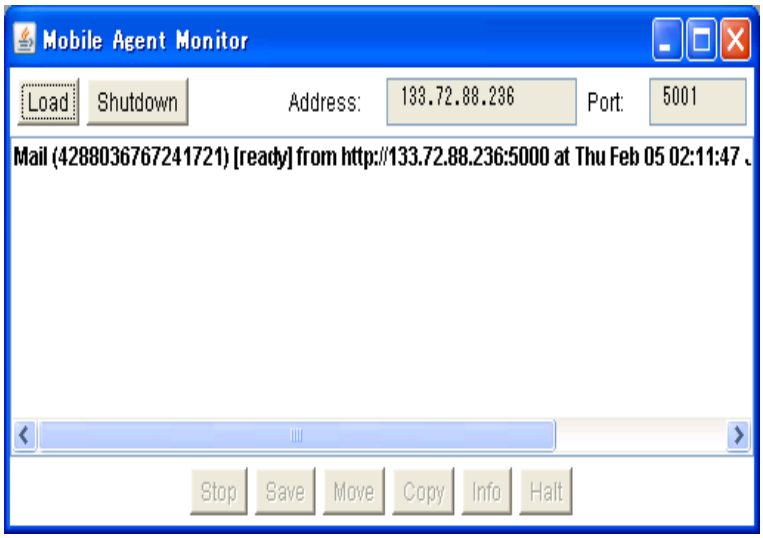

図 2.13 ”mail agent”の画面

結果から通信ポート５０００のコンピューター上から通信ポート５００１のコンピューターへの移動が確認できる。

2.9.1 モバイルエージェントの移動

エージェントのコンピューター間移動について、その過程を示す。またモバイルエージェントを実現するシステムは数多くあるが、エージェントのコンピュータ間移動は、遠隔手続き呼び出しの引数受け渡しと同様の方法で実現している。。

1. **移動対象となるエージェントの実行を一時中断する**

2. **エージェントの実行状態をデータ化する。データ化した実行状態とプログラムコードを転送可能なデータ形式に符号化する**

3. **データ通信プロトコルを用いて、符号化したデータを移動先のコンピューターに送信する**

4. **受信したデータを符号化し、エージェントに変換する**

5. **エージェントの実行を再開する**

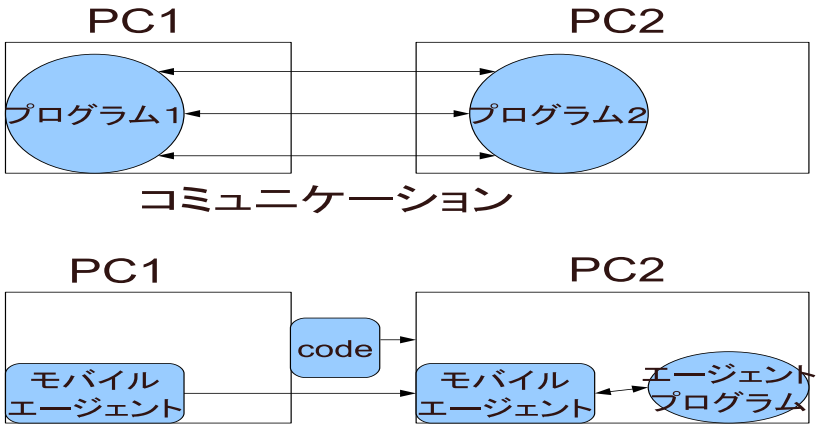

図 2.14 モバイルエージェントの移動

2.9.2 Freediaについて

Freediaとは、スマーティブプロジェクトにより開発されたエージェントフレームワークが「Freedia」である。.[12] エージェントによってコンテンツを管理するための研究が進められているプロジェクトが「Smartive Project」であり [10] Smartive Projectは、「自由でかつ安全なコンテンツの活用と流通の実現」を目指し、国立情報学研究所 アーキテクチャ科学研究系 本位田研究室が中心にパートナー企業・大学と推し進めているプロジェクトである。[11] スマーティブとは、コンテンツにかかわる人々の希望や要求をポリシーとして埋め込む技術であり、スマーティブによってコンテンツ流通の促進、コンテンツの表現力の向上、利便性の向上等の効果が期待できる。

Freediaには, エージェントを動作させる仕組み (プレース, エージェント間通信, 検索, ワークフローマネージャ, アスペクト記述等) やコンテンツを管理するためのライブラリやセキュリティ機能が組み込まれている. プレースは, XML言語で書かれたエージェント定義 (ワークフロー定義, ポリシー定義など) を解釈し, 実行することができる.XMLで定義されたエージェントは, Javaでつくられた機能を呼び出すことができる. そのため, Freedia基盤ソフトウエアもほとんどがJavaでつくられている.

2.9.3 Rubiret

エージェント実装システムとしてRubiretがある。
Rubiretは Rubyによって実装したモバイルエージェントシステムである。Rubiretでは、エージェントが動作するホスト上であらかじめエージェントのプラットフォームとなるランタイムシステムを動作させておく。エージェントはランタイムシステムを動作させておく。エージェントはランタイムシステムを利用して複数のホストを移動しながら与えられたタスクの実行を行う。Rubiretシステムはスクリプト言語を利用して、モバイルエージェントの動作を手軽に記述可能である。モバイルエージェントの動作ではネットワークの移動が含まれる。移動中は時間がかかるので、各ホストで大きな計算をしないでネットワークを頻繁に移動するようなモバイルエージェントの応用では、各ホストでの実行速度は大きな問題にはならないため実装には実用的である。またRubyでは組み込みクラスに高度な機能が提供されているので、そのような機能を用いるとモバイルエージェントの実現が容易である。[16][18]

2.9.4 要素

Rubiretにおいてエージェントの要素は

1. クラスコード
2. 実行状態
3. エージェントの管理情報

クラスコードとはエージェントの動作を記述したRubyのスクリプトプログラムである。ユーザー側はエージェントをRubiretクラスのサブクラスとして定義する。Rubiretクラスには

1. **initalize**
   エージェントの生成時

2. **run**
   エージェントの実行時

3. **bye**
   エージェントの実行終了時

でコールバックメソッドが定義されていて、ランタイムシステムはエージェント生成や終了などのタイミングでそれらのコールバックメソッドを呼び出してエージェントを動作させる。ユーザーはそれらのコールバックメソッドを再定義してエージェントのクラスコードを作成する。また、エージェントは移動する際にインスタンス変数を持ち運ぶ。エージェントの管理情報は

1. **name**
   エージェントの名前

2. **owner**
   エージェントの所有者

3. **birthplace**
   エージェントが生成されたホスト

であり、エージェントの名前やその所有情報などの情報は、エージェント生成時にランタイムシステムがエージェントに与える。

第3章

提案システム

本来、モバイルエージェントは暗号化しても、例えばユーザー側からあるホストに個人情報などを送信する際に、ホスト側では個人情報が埋め込まれているエージェントを実行するために復号化されてしまう、つまり電子取引等においては、ユーザー側のカード番号等の重要な情報を受け取り側のホストが不正なホストだった場合に秘密情報が漏れてしまう。つまりモバイルエージェントが持っている情報は、移動先のホストに対して秘密にできない。そこで本稿ではユーザーのホスト側で秘密情報を管理するエージェントと、電子取引やオークション等の決済の際に、仮想店舗のホスト側でクレジットカードのカード番号の個人情報等の秘密情報が必要となった時に、巡回しているエージェントによってカード会社がユーザの秘密情報管理エージェントにアクセスしての秘密情報を取り出し、巡回エージェントを介してホストに要求を送ることで、不正なホストに対する個人情報保護と著作権管理するシステムを提案する。

3.1　過去の提案システム

過去の研究で提案されたシステムはコンテンツ ID を埋め込むというもので、ディジタルコンテンツの流通モデルとして情報カプセルを導入し, エージェントがそれを管理するシステムが提案されている。しかしこの場合コンテンツは管理されても、個人情報の秘密情報の保護は管理されない。個人情報の秘密情報の保護のためには新たにシステムを考える必要がある。[3]

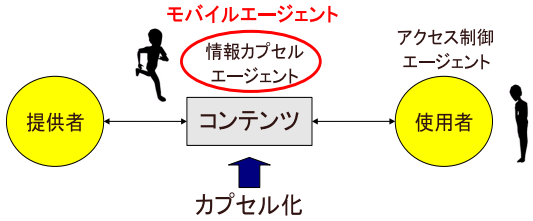

図 3.1　著作権保護のエージェントシステム

3.1.1　提案システムの利点と応用

巡回エージェントをユーザーとホスト間を巡回させ、ユーザーと仮想店舗等の取引の際、巡回エージェントにユーザーの秘密情報の仲介の役割を持たせることで、電子取引等での不正なホストに対するカード番号等の秘密情報の漏えい防止につながる。また巡回エージェントを用いることで、分散マーケットプレイスの実現が可能になる。分散マーケット上では、マーケットがネットワーク上の複数のサーバーに分散しており、ユーザーは巡回しているエージェントを用いてこれらのマーケット上で取引を行い、分散マーケットプレイスの形態を取ることが可能。例えば巡回してるエージェントユーザーの希望している商品を実際の複数のオークションサイトで入札を行うことでより安く、かつ確実に希望する金額の入札が成功するように機能する。

3.2 巡回エージェント

巡回エージェントはユーザー側が仮想店舗に秘密情報を送る際にあるカード会社のホスト、仮想店舗、ユーザー間を巡回しているため、巡回エージェントは情報の仲介者の役割を持つ。主な流れを以下に示す。

1. 仮想店舗側からユーザーの秘密情報を要求される。
2. ユーザは、秘密情報をアクセス制御リストとともに巡回エージェントに渡す。
3. 巡回エージェントは、アクセス制御リストに基づいて秘密情報のうち店舗に必要な情報のみを提供する。また、決済する金額を店舗から受け取る。
4. 巡回エージェントは、決済金額とアクセス制御リストに基づいた秘密情報をカード会社に提供する。
5. カード会社は、決済許可証を巡回エージェントに引き渡す。
6. 店舗は、巡回エージェントより決済許可証を受け取り検査する。

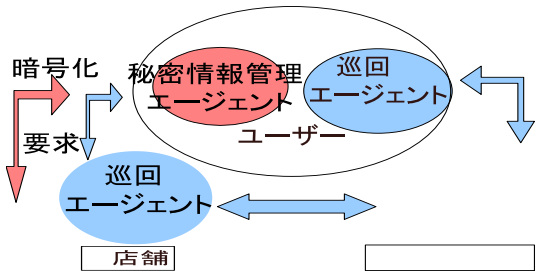

図 3.2 秘密情報管理エージェントと巡回エージェント

3.3 エージェントの動作

3.3.1 TAF

エージェントシステムの設計、実装、運用を支援するエージェントフレームワークとしてTAF[Training system for Agent Freamework] を利用する。
TAFで定められているエージェントのアーキテクチャに基づいて、エージェントプログラミング言語が提供され設計したエージェントの蓄積や管理、実行といった機能があり、これを用いてエージェントシステムの設計が可能になる。

3.3.2 提案システムの動作

前提としてユーザーと仮想店舗間を巡回するエージェントを動作させる。仮想店舗側からユーザーの要求があった場合にエージェントはアクセス制御リストからユーザーの秘密情報でない情報を取り出し、仮想店舗側の要求に適した情報を巡回エージェントに渡す。この場合はまず

1. **秘密情報管理エージェントから情報を取り出す**
   Object data = store.get(dataName);

2. **秘密情報管理エージェントにデータ情報として登録する**
   store.put(dataNsme,data,true)

3. **秘密情報管理エージェントのメソッドを呼び出す**
   CallMethod callInfo = newCallMethod(”methodName”,args); Object data = store.method(”ssl”,callInfo)

エージェントコード中の”store”はユーザーの秘密情報を巡回エージェントがアクセスする際にサポートするデータストアクラスのオブジェクトであり、データストアは巡回エージェントのデータベースとして機能する。つまりはユーザーのホストを出発後、セキュリティーポリシーにより秘密情報の暗号化と複合化をしユーザーの代わりに電子決済を行うことが可能である。
カード会社にアクセス制御に基づき、仮想店舗から受け取った決算金額と秘

密情報をカード会社に提供し、決済の許可証を巡回エージェントに渡し取引する。

ユーザーの電子決済情報を保護するセキュリティーポリシーの記述例は

＜ security ＞
＜ set ＜!–決済ー–＞
＜ payment-gateway ＞ CARD-NO ＜/payment-gateway ＞
＜ payment-gateway ＞ CARD-TERM ＜/payment-gateway ＞
＜/set ＞
＜ data ＞ ＜!. 個人データアクセスー–＞
＜ data-name ＞ CARD-NO ＜/data-name ＞
＜ data-name ＞ CARD-TERM ＜/data-name ＞
＜ data-name ＞ NAME ＜/data-name ＞
＜ data-name ＞ ADDRESS ＜/data-name ＞
•••••
＜ protect name=”set” /＞
＜ host name=”****”＞
＜ agent name=”CN=shop1, OU=yu-za, ••••• ”＞
＜ phase name=”****”＞
＜ access name=”****” /＞
＜/phase ＞
＜/agent ＞
＜/host ＞
＜/data ＞
＜/policy ＞
と記述可能である。

実行した結果を図として示す。

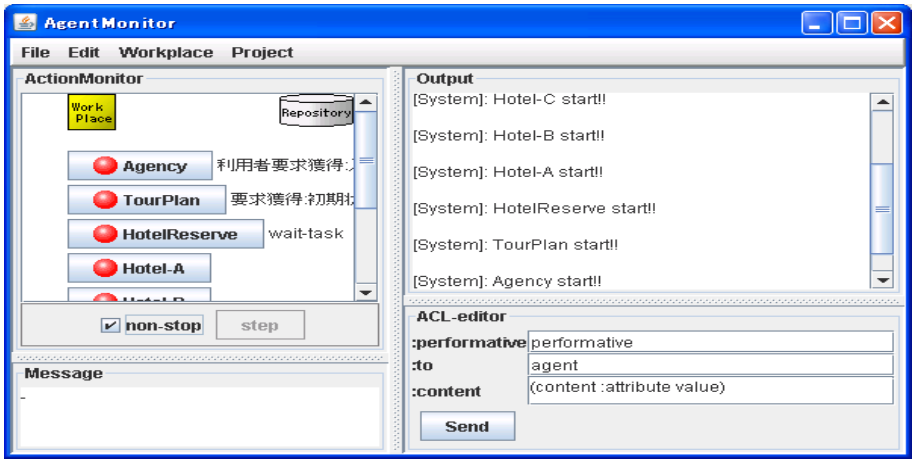

図 3.3　agent moniter

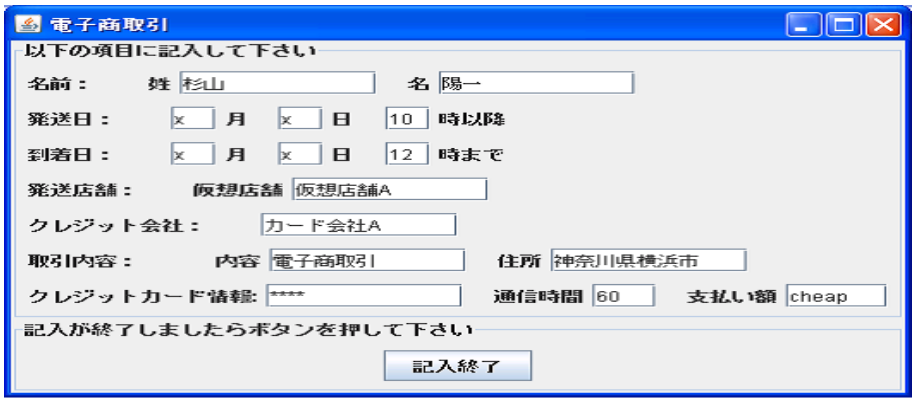

図 3.4　エージェント送信情報

3.3.3 エージェントの保護

モバイルエージェントシステムではプログラムとデータを移動させるため、エージェント自身の保護は必要である。保護の目的を以下に示す。[18]

- 通信路上のエージェントを盗聴者や改竄者から保護する
- 悪意のあるエージェントから他のエージェントを保護する
- 悪意のあるエージェントからホストを保護する

以上の要件を満たし、エージェントの保護をするため以下の機能を与える。

- エージェントの暗号化ではなく通信路の暗号化
- エージェントへの電子署名と承認
- エージェントのホストへのアクセス制限

通信路の暗号化にはSSLを用いる。また秘密情報に適用すべきセキュリティーポリシー、つまりは情報の公開先と情報の保護手法は例えば会員情報などは会員サービスを提供する店舗だけにSSLで通知し、クレジットカードの場合だとカードの認証機関だけに通知する。
[17]

3.3.4 評価

過去の研究では情報カプセルによってコンテンツの保護を提案していたが、コンテンツは保護できても個人情報や著作権等は保護しきれない。よって今回はユーザーと仮想店舗間に巡回する第三者のエージェントをおき、仲介させることにより個人秘密情報や著作権保護につながる。

3.4 他のシステムへの応用

エージェントに関しての他のシステムについて、今回のシステムとは別に他のシステムへの応用として巡回監視システムを提案する。

巡回監視システムは中央制御によるネットワークの監視は、ネットワークへの負荷が集中してしまう。モバイルエージェントの特性を用いることでネットワークへの負荷を減らし、機能の拡張を容易にすること可能である実際の応用例を挙げると、知的モバイルエージェント「Plangent」を用いることで制御機器上でユーザーが必要とする情報を抽出し、その情報のみを持ち帰ることで、ネットワークへの負荷を軽減している。また、機器固有の情報をエージェントの知識として保存することで、機器間の差異をエージェントが吸収することを可能としている。組込み機器上でソフトウェアを動作させる際には、厳しい資源の制限が問題になるため、動作に必要な最小限の要素のみを持って移動することで資源の少ないコンピューター上での動作を実現している。モバイルエージェントを用いた動的ルーティングシステムでは、モバイルエージェントを用いてルーティングテーブルを更新することで、制御通信量を削減し、エージェントを局所的に流したり部分的に種類を変える事で高速で柔軟なサービスが提供できる。[10]

第4章

結論

本研究では、本稿では、巡回エージェントを用いることで不正なホストからの秘密情報の管理によるシステムを提案した。本来の目的であったモバイルエージェントによる著作権管理、個人秘密情報の管理は電子商取引を例にして巡回エージェントの導入、SSL による通信路の暗号化、セキュリティーポリシーを記述し不正なホストから情報を保護することで達成した。
今後の課題として、今回は単純な記述形式のセキュリティーポリシーを用いたが、より柔軟なセキュリティー機構を実現するためには、セキュリティーポリシーの記述形式について考慮する必要がある。また他のアプリケーションに対する著作権管理や個人情報の保護の拡張も今後の課題となる。

謝辞

本研究を行なうにあたり、終始熱心に御指導していただいた木下宏揚教授ならびに鈴木一弘助手に心から感謝致します。また，公私にわたり良き研究生活を送らせていただいた木下研究室の方々に感謝致します。

2009年2月

杉山　陽一

参考文献

[1] ”ウィキペディア（Wikipedia)”
http://ja.wikipedia.org/wiki/

[2] 耐タンパ　モバイルエージェントを実現するセキュリティー機構の提案
www.jaist.ac.jp/library/thesis/is-master-2002/paper/makoto-h/paper.pdf

[3] 山田孔太, 木下宏揚, 森住哲也, 稲積康宏：
”エージェントベースの情報カプセルを用いたコンテンツ利用の
利便性の向上”, SCIS2007

[4] 山田孔太：2006
情報カプセルを用いた著作権管理, 2006

[5] ”JASRAC(社団法人日本音楽著作権協会)”
http://www.jasrac.or.jp/profile/copyright/index.html

[6] ”IT 用語辞典”
http://e-words.jp/

[7] 木下哲男:
”エージェントシステムの作り方”

[8] ”モバイルエージェント”
http://research.nii.ac.jp/ ichiro/download/satoh-nii010516.pdf

[9] ”AgentSpace”
http://research.nii.ac.jp/ ichiro/agent/agentspace.html

[10] 吉岡信和, 田原康之, 本位田真一：
”モバイルエージェントによる柔軟なコンテンツ流通を実現するアクティブコンテンツ” 情報処理学会論文誌 D Vol.44 No.18 pp.45-57 2003

[11] ”スマーティブプロジェクト”
http://smartive.jp/index.htm

[12] ”Freedia”
http://www.freedia.org/

[13] ”プライバシー保護と個人情報保護”
http://www.asahi-net.or.jp/ VR5J-MKN/point/privacy/

[14] ”エージェント技術”

[15] ”Mobile agent system”
http://research.nii.ac.jp/% 7Eichiro/agent/index.html

[16] 井上敦　竹内祐一　富永和人：
”オブジェクト指向スクリプト言語によるモバイルエージェントシステムの実装とその応用”　情報処理学会論文誌 D, Vol.102　No.602 pp.19-23 2003

[17] ”携帯 SSL について”
http://synth.jp/pg/2007/05/ssl.html

[18] ”Ruby によるモバイルエージェントシステムの試作と計算機管理への応用” www.teu.ac.jp/nsit/rubiret/paper/t-hide.pdf

質疑応答