

# Case Study: モバイルエージェントを用いた公開システム (Mobeet) の構築

† 吉岡信和      † 上野晃嗣      † 長野伸一      † 川村隆浩  
 † 田原康之      † 長谷川哲夫      † 大須賀昭彦      ‡ 片山卓也  
 † 株式会社 東芝 研究開発センター  
 ‡ 北陸先端科学技術大学院大学 情報科学研究科

## 1 はじめに

近年の急速なインターネットの発展とその利用形態の多様化にともない、WWW 上の情報を有効活用するための検索システムの需要が高まっている。そこで、われわれは得たい情報を効率よく探し出すためにマルチエージェントフレームワーク (Bee-gent[1]) を用いて Web 上から必要な情報を自動的に取得してくるエージェント (Mobeet[2]) を開発し一般に公開した。このようなインターネットの一般ユーザから広くエージェントを受け入れるサービスはこれまでになく、公開してサービスを維持するためにはどういった問題を解決しなければ良いか明らかになっていない。そこで、本稿では、Mobeet を構築、運営してことを通して得られた経験を Case Study として報告する。

## 2 Web 情報収集エージェント Mobeet

Mobeet は、知りたい情報をユーザの代わりに Web から集めてくれるエージェント (以下 Web 情報収集エージェント) である。ユーザが自然言語で知りたい情報を入力すると、エージェントはインターネット上のサーバに移動し、パソコンの電源を切っても、サーバーとの通信回線を切っても、ほしい情報を代わりに検索・収集し続けてくれる。ユーザは、エージェントを任意の端末から呼び戻して収集した情報を受け取り、閲覧することができる。

Mobeet は、Bee-gent を用いて構築されている。Bee-gent は、ネットワーク上のさまざまなアプリケーションを連携させて新しい分散システムを構築するためのフレームワークであり、システムをアプリケーション、ラッパーエージェント (以下ラッパー)、仲介エージェントの 3 種類で構築する。仲介エージェントはサービスを実現するためにアプリケーションを連携するモバイル

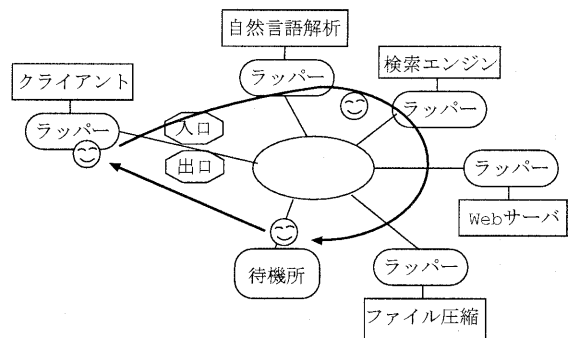


図 1: Mobeet の構成図

エージェントである。ラッパーは他のエージェントからの要求に応えるためにアプリケーションを立ち上げローカルなやり取りをエージェント通信言語に変換する。

Mobeet の場合、図 2 のようにアプリケーションとして自然言語解析アプリケーション、検索エンジン、Web サーバ、ファイル圧縮ソフト、クライアントで構成されており、それぞれについて、ラッパーが用意されている。Web 情報収集エージェントは、仲介エージェントに対応し、これらのラッパーを利用し情報を収集する。このエージェントはモバイルエージェントであるため、エージェント待機所を用意することで、情報を収集し終わったエージェントを他の端末からでも呼び戻せるようになる。

Web 情報収集エージェントは、まずユーザからの要求を受け取ると、自然言語ラッパーを利用して自然言語からキーワードを抽出し、そのキーワードを使って関連するページを検索エンジンラッパーを利用して調べる。次に関連のリンク先のページをキーワードが含んでいるかどうか調べながらまとめ取りする。検索が終わったらエージェントは待機所でユーザからのリクエストを待ち、要求があるとファイル圧縮ラッパーを利用しコンテンツをすべて圧縮した後、ユーザの端末に帰る。

本システムの様なインターネットの実際のサービスの連携し、一般ユーザから広くエージェントを受け入れるサービスはこれまでにないものである。そのため、安定動作させるにはどのような要因が重要になるかが明らかになっていなかった。そこで、われわれは、まずシステムのリスク分析を行い、安定動作させるためのさまざまな要因をリストアップし、その中で重要なものを

Case Study: Development of An Open Mobile Agent System, Mobeet

†Nobukazu YOSHIOKA, †Kouji UENO, †Shinichi Nagano, †Takahiro KAWAMURA, †Yasuyuki TAHARA, †Tetsuo HASEGAWA, †Akihiko OHSUGA, ‡Takuya KATAYAMA

†Corporate Research & Development Center, TOSHIBA

‡School of Information Science, Japan Advanced Institute of Science and Technology.

選択し、その項目について検討、対策を行った。

### 3 リスク分析

システムの要求として、サービスの安定動作、システムの一部が不都合があっても最小限の被害で復旧、不都合の早期発見、解決などがあり、そのための要因の列挙と重要度を分析するために、以下の手順でリスク分析を行った。

#### 1. サービスが不安定になる要因のリストアップ:

不都合による要因と効率・性能に関する要因についてフィッシュボーン図を用いてリストアップした。

2. サービスを大きく実行、運用、改良のフェーズに分け、そのフェーズ毎にプロセスマップを作成し、それぞれのプロセスについて不安定要素を列挙した。
3. 上記項目について、その原因、発生度、影響(エンド、ローカル)、影響度(性能、時間、コスト)、検出法、検出度について考え評価式によって不安定要因の大きいものを計算した。

この時、影響度については、「ユーザが気づかない」から「警告無しにユーザに悪影響を及ぼす」まで、発生度については、「発生の見込みはごくわずか」から「故障が確実」まで、検出度については、「常に検出」から「検出できない」までそれぞれ10段階で評価した。また、寄与度の評価は、発生度 × 影響度 × 検出度で計算を行った。

この結果として得られた不安定要因のうち、以下のものが重要項目として挙げられた。

- 処理が集中して不安定になりやすい
- システムの一部が不都合で止ってしまった時に、その復旧のために、システム全体が止まってしまう。
- 不都合が起きた場合の原因の追求が難しい

### 4 対策

リスク分析をふまえ、不安定要因の高いものの対策をとった。以下が、実際の対策の例である。

#### 4.1 処理の集中

処理の集中に対して、まずどこが実際に一番集中しやすいか実行プロセスから分析した。その結果、入口と出口、Webの検索部分であることと推測した。そこで、当初は1台のマシンでサーバーを開発していたが、実際の運営では、3台のマシンで負荷分散を行うことにした。3台のマシンの役割は、それぞれサービスの出入口、Web検索部分、その他の処理部分とした。Bee-gentのフレームワークを使っていたため、ラッパー、アプリケーションをそれぞれのマシンに再配置するだけで、容易に対応が可能であった。

#### 4.2 システムの一部復旧

システムの一部が不都合で止ってしまった時に、その影響が全体に伝播しないようにする必要がある。Bee-gentのシステムでは、連携部分の仲介エージェント、インターフェースとなるラッパー、アプリケーションの独

立性が非常に高く、ラッパーはアプリケーションの監視を行い、エージェントはラッパーが動いているかどうかチェックを行い、ラッパーが正常動作しているかどうかのチェック用プログラムを構築することで容易にシステムの監視が行えた。また、サービスの入口部分ではサービス内に存在するエージェントのリストを保有しているため、この部分が落ちるとシステム全体の整合性がとれなくなっていたが、動的にエージェントの存在を確認することで対応が可能であった。

Bee-gentでは、それぞれのラッパー、アプリケーションが別のプロセスとして立ち上がるため、復旧も落ちたプロセスを立ち上げ直すだけで対応可能であった。

### 4.3 不都合の原因追求

分散システムの場合、処理の継続を追うのが難しく、不都合の原因を突き止めるのが困難になる。しかしながら、Bee-gentのフレームワークでは、サービスの一連の処理の継続は仲介エージェントの動きに対応しているため、そのエージェントの動きを追うことで、不都合の原因追求が行える。そこで、サーバ内のラッパーのログから不都合がおきたエージェントの一連の処理のログを抜き出してくるツールを用意することで、原因の追究がしやすくなった。

### 5 システムの公開

前節のような対策を講じたのち、エージェントシステムを北陸先端科学技術大学院大学から公開を行った。公開から一ヶ月で2000件のエージェントを受け入れ、24時間体制でサーバーを動かすことができた。また、一日最大300エージェントの処理に耐えられるシステムが構築できた。

### 6 おわりに

モバイルエージェントのシステムはようやく実用段階に入ろうとしている。本稿では、このほど公開したモバイルエージェントサービス Mobeet を構築、運営してことを通して得られた経験を Case Study として報告した。このサービスでは、公開前にシステムの不安定要因を分析しその対策を行った。Bee-gent のフレームワークを最大限に利用することで、容易に対策を講じることができた。これにより、実用的なモバイルエージェントサービスが可能であることが示された。今後、サービスの内容、機能を充実していくにしたがって、モバイルエージェントのシステムとしてどのような問題が発生するかどうかを検討し、より広い分野で利用可能であることを示していきたい。

謝辞

この研究は東芝と北陸先端科学技術大学院大学の共同研究のもとで行われました。

### 参考文献

- [1] Bee-gent: <http://www2.toshiba.co.jp/beegent/>.
- [2] Mobeet: <http://beegent.jaist.ac.jp:8000/Mobeet/>.
- [3] 吉岡信和他: マルチエージェントフレームワーク Bee-gent による Web 検索システムの開発, 情報処理学会, 第60回全国大会論文集, 2000(3).