

移動エージェント相互運用の実現: ディレクトリサーバ

3S-4

別府祥之* 中島震* 糸野文洋** 長谷川哲夫*** 大須賀昭彦***
 NEC* 株式会社三菱総合研究所** 株式会社東芝***

1 はじめに

移動エージェント基盤間の相互運用を実現するためには、移動エージェントやその基盤に関する情報を統合管理するディレクトリサーバが必要である。実際、OMG MASIF では MAFFinder [1] を、FIPA では DF (Directory Facilitator) [2] を各々規定している。

本稿では、新たに開発した相互運用ディレクトリサーバについて報告する。特に、オープンなネットワーク環境下の異種プラットフォーム上でサービスの分散透明な利用環境を提供する ODP トレダ [3] [4] の考え方を採用したことを特徴とする。

以下、具体的な議論を行なうために既存標準案の問題点を指摘する。次に、相互運用ディレクトリサーバの構成、移動エージェントの位置情報を動的に取得する方法を説明し、最後に、提案サーバが MAFFinder と同等機能を実現できることを示す。

2 既存標準案の問題点

MASIF では、移動エージェントや移動エージェント基盤の位置情報を管理する MAFFinder について CORBA サーバの外部仕様を定義している。一方、FIPA では DF をイエローページサービスを提供すると規定している。以下、既存標準案の問題点を示す。

1. 分散する管理対象への対応不足

移動エージェント相互運用の対象システムは、複数の移動エージェント基盤を含む大規模なシステムである。管理対象である移動エージェント基盤はネットワークに分散して存在するので、単独のディレクトリによる集中管理を行なうことは難しく、複数ディレクトリで分割管理する。しかし、MAFFinder は複数サーバの連携方式を規定していない。また、DF では接続トポロジーに関する議論を行なっているが、DF 間の接続プロトコルは規定していない。

2. 管理対象の固定化・検索方法が貧弱

MAFFinder は、管理対象を AgentSystem、Place 及び Agent に限定しており、また or 条件を含む複雑な検索を

行なうことができない。しかし、相互運用の実現には、管理対象の拡張や、細かな検索条件指定を行なう必要がある。

3. 移動エージェント位置情報の取得方法が実現依存
 検索時点での移動エージェントの位置情報を取得する必要がある。しかし、既存標準案は、登録時以降に値変化する移動エージェント位置情報を追従する方式を規定していない。特定移動エージェント基盤に依存し、ディレクトリの可搬性が低下する。

3 相互運用ディレクトリサーバ

ODP トレダ [3] [4] は、多数の対象を複数サーバで分割管理しサーバ群を協調動作させること (フェデレーション)、管理対象を拡張する手段としてスキーマ定義を提供すること、及び or 条件を含む論理式による柔軟な検索を行なえることを特徴とする。ODP トレダの枠組を用いることで、第2節の第1及び、第2の問題点を解決できる。移動エージェント位置情報の取得方法は、動的に変化する値取得方法の特殊形として3.2節で説明する。スキーマ定義例と検索例を3.3節で示す。¹

3.1 システム構成

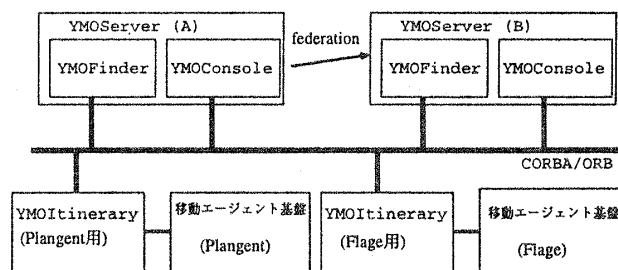


図1: システム構成例

相互運用ディレクトリサーバ (YMOServer) を含むシステムは、次の CORBA サーバから構成される (図1)。

YMOFinder 管理対象の登録、検索、削除を行なう機能を提供。

YMOConsole スキーマ管理、フェデレーション情報管理及び管理対象データの永続化機能を提供。

YMOItinerary 移動エージェントの位置情報を取得するためのインターフェースを規定。Plangent や

¹ フェデレーションの説明は省略する。

Interoperability Middleware for Mobile Agents: Directory Server

Yasuyuki Beppu, Shin Nakajima, Fumihiko Kumeno, Tetsuo Hasegawa, Akihiko Ohsuga

NEC Corporation, Mitsubishi Research Institute, Inc., Toshiba Corporation

Flage といった特定移動エージェント基盤ごとに自身が持つ移動エージェント位置管理方法を考慮して、本サーバを実装する。これにより移動エージェント位置情報の取得方法を統一でき、相互運用ディレクトリサーバの可搬性がよくなる。

3.2 エージェント位置情報の取得

YMOFinder は、管理対象の移動エージェント 1 つに対して、移動エージェント管理情報と移動エージェント位置情報を追随する YMOItinerary 名の組を保持する。以下、移動エージェント位置情報の検索要求時の動作を示す。

1. YMOFinder は、検索条件に適合する移動エージェント管理情報と YMOItinerary 名の組を集める。
2. 1. で集めた組を YMOItinerary 名ごとにグループ化する。
3. 各グループごとに移動エージェント管理情報の中から移動エージェントを一意に定める識別子を全て集めて、YMOItinerary に一括して渡す。
4. YMOItinerary は、3. で渡された複数の識別子に対応する移動エージェントの位置情報を一括して YMOFinder に返す。
5. YMOFinder は、移動エージェント位置情報を要求元に返す。

本方式では、検索時に各 YMOItinerary に対する問い合わせは高々 1 回である。通信回数を減らすことで、検索時間を短縮している。

3.3 MAFFinder 機能のエンコード

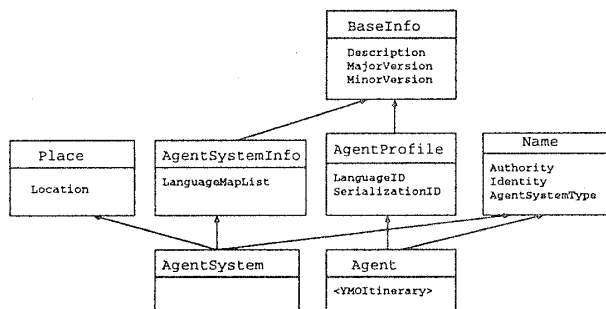


図 2: スキーマ定義例

本稿で提案する、スキーマ定義による方式の有効性を確認するために、MAFFinder が管理する対象を表現するスキーマ定義を与える (図 2)。スキーマ定義は、スキーマ名、継承する親スキーマ名、及び特徴を表す属性からなる。管理対象は、いずれかのスキーマに属する。移動エージェントや移動エージェント基盤を一意に定める Name は、Authority と Identity 及び AgentSystemType とい

う 3 つの属性を持つ。BaseInfo は、移動エージェントと移動エージェント基盤とに共通する特徴である。矢印はスキーマ間の継承関係を表す。移動エージェントに関する情報である Agent は AgentProfile と Name を継承し、さらに移動エージェント位置情報取得時に用いる YMOItinerary 名を持つ。

相互運用ディレクトリサーバでは、スキーマ名、検索条件等を指定して検索を行なう。

例 1 スキーマ名: AgentSystem

```

検索条件: Authority == "NEC"
           and Identity == "1111"
           and AgentSystemType == 1
  
```

例 2 スキーマ名: Agent

```

検索条件: MajorVersion == 2 or
           (MajorVersion == 1 and MinorVersion >= 5)
  
```

例 3 スキーマ名: Name

```

検索条件: Identity == "2000"
  
```

例 1 はスキーマ名と and 条件指定による検索であり、MAFFinder と同等である。例 2 は MAFFinder では表現できない or 条件を含む柔軟な検索例である。例 3 は、スキーマの継承関係を用いた例であり、Name 及びそのサブスキーマに属する管理対象で、Identity の値が "2000" であるものを検索する。すなわち、Agent と AgentSystem を同時に検索対象とすることができる。

4 おわりに

移動エージェントや移動エージェント基盤の情報を統合管理する相互運用ディレクトリサーバを提案した。本ディレクトリサーバは、適切なスキーマ定義を与えることで、MASIF が規定する MAFFinder 相当の機能を実現できる。

今後、相互運用ミドルウェアの他コンポーネントと統合してインカネーションを実現し、提案方式を評価していきたい。

本研究は、情報処理振興事業協会 (IPA) が実施している「次世代デジタル応用基盤技術開発事業」の一環として行なわれたものである。

参考文献

- [1] OMG: CORBA facilities: Mobile Agent Facility Submission, 1998.
- [2] Agent Management, FIPA98 Specification, <http://www.fipa.org/>, 1998.
- [3] ITU-T Rec. X.950-1: Information Technology - Open Distributed Processing - Trading Function -, Part 1: Specification, 1997.
- [4] Nakajima, S. and Futatsugi, K.: An Algebraic Approach to Specification and Analysis of the ODP Trader, 情報処理学会論文誌, vol.40 no.4, 1999.