

省メモリ・高速処理を指向した主記憶 MIB プラットフォーム

1Aa-1

登内 敏夫

中島 震

NEC C & C 研究所

1 はじめに

高度情報化社会の到来にともない、通信網の拡充が重要になっている。この際重要な役割を果たすのが、安価かつ高速なネットワーク装置 (NE) である。これらの装置は、通信網管理のために図 1 で示すようにマネージャとエージェントから構成する通信網管理システムを構成する。

我々は高速かつ安価な OSI 管理エージェントを開発中である。このようなエージェント開発の鍵は、MIB モジュールの高速化・低コスト化である。我々は、メインメモリをベースとする MIB [3] を採用することにより処理の高速化を試みる。さらに記憶領域の管理方法を工夫することにより必要記憶容量を減らし、コストの大半を占めるメモリの削減に取り組む。

本論文はエージェントプラットフォームのうち、MIB プラットフォームについて述べる。

2 アーキテクチャ

MIB プラットフォームは次の 2 つの特徴を有する。

1. データモデルに内在する従属性を排除することにより、記憶容量の削減を実現する。
2. 格納対象データの詳細を隠蔽することにより、データの性質に応じたデータ構造を実現することを可能とする。

前者は、関係データベースの正規化に関しても常々言われていることである。OSI モデリングでは、以下の従属性が存在する [5]。

- オブジェクトインスタンス (MOI) とオブジェクトクラス (MOC) の従属性 MOI と MOC は多対 1 の関係にあるという従属性がある。そのため、MOI 自身に MOC の情報を持たせることは冗長性を有することになる。このような冗長性を

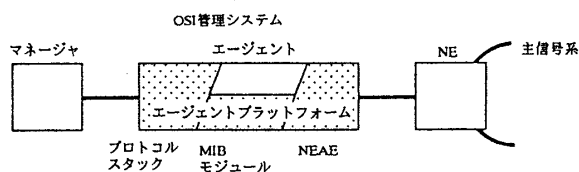


図 1: エージェントの全体アーキテクチャ

省くために、MOC に関する情報を 1 まとまりにして管理することにより、記憶容量の削減を目指す。

- 名前属性と包含木の従属性 OSI の包含木では、MOI の位置を示すのに名前属性 ID と名前属性値を指定する。名前属性を MOI と包含木で共有することにより (図 2)、記憶容量の削減を目指す。
- 包含木内の名前属性に関する従属性 包含木内の名前属性の種類は、NAME BINDING テンプレートによって規定されており、名前属性の種類による従属性が存在する。そこで、名前属性 ID と属性値を同時に管理するのではなく、別々に管理することにより記憶領域の削減を計る (図 3)。
- 属性を共有可能な MOI に関する従属性 OSI のモデリングでは、MOI 間で属性を共有するという概念はない。しかし、実質上、複数の MOI で同じ属性を共有していると考えて差し支えない場合が多く存在する。例えば、すべての MOI が有する objectClass 属性は、同じ MOC から生成した MOI はすべて同じ値を有する (図 4)。このような属性を共有することにより、記憶領域の削減を計る [2]。

次に後者の特徴について述べる。本特徴はオブジェクト指向フレームワークにより実現している。

次の 2 つの意味で、この特徴は重要である。

1. OSI モデリングでは、オブジェクトは静的なデータを有すると同時に動的な振舞いを持つ。振舞いはエージェントごとに異なる。そこで、プラット

"A Main-memory MIB Platform for Fast and Compact OSI Management Agents", Toshio TONOUCHI and Shin NAKAJIMA, C&C Res. Labs, NEC Corporation

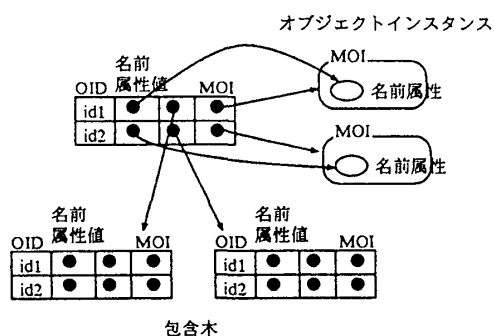


図 2: 包含木における名前属性の共有

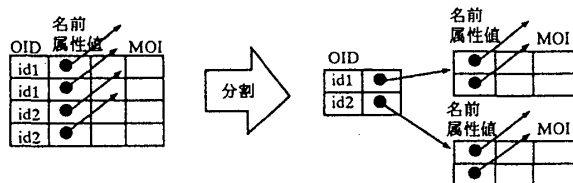


図 3: 包含木の節点の分割

フォームはこのような振舞いを実現することができる仕組みを用意する必要がある。

- データの性質がデータの種類によって大きく異なる。例えば、装置状態に応じて変更するデータや、マネージャがコンフィギュレーションのために設定するデータもある。このような多様多様なデータを一括して統一的方法で管理するのは、速度の低下や必要メモリの増大を招く。そこで、基本アーキテクチャを維持しつつ、かつ、多様な格納方式を実現する必要がある。

前者の例として、一定時間ごとに振舞いをおこなうオブジェクト (scanner) がある。MIB プラットフォームは、一定時間ごとにエージェントの virtual メソッドを起動するオブジェクト指向フレームワークを構成する。エージェントプログラマは派生クラスで virtual メソッドを定義することにより一定時間毎に任意の振舞いを起こさせることが可能である (図 5)。

後者の例として、logRecord オブジェクトがある。NE などからのイベントが起きる度に、logRecord オブジェクトは生成され、MIB 内に蓄積される。生成されたオブジェクトがある一定数以上溜ると、古いオブジェクトが削除される (ログフルアクション)。このようなオブジェクトを扱う場合、通常の OSI の包含木を実現するデータ構造を流用するより、リングバッファで管理した方が都合がよい。本 MIB プラットフォームは、通常の包含木とリングバッファの違いをオブジェクト

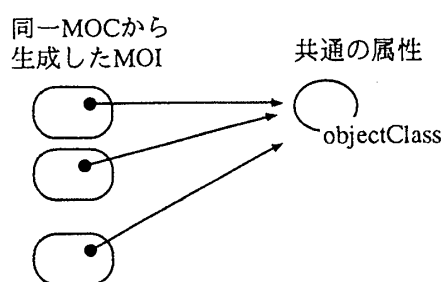


図 4: 属性の共有

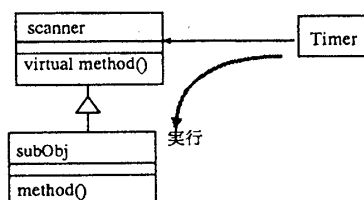


図 5: 定期的な振舞いの実現

により隠蔽し、両者を同様に扱うことが可能である。

3 関連研究

OSIMIS [1] はロンドン大学で開発された OSI 管理マネージャおよびエージェント開発プラットフォームである。OSIMIS は商用以外の使用が自由なので、様々なツール群が開発されている。

本エージェントと比較すると、速度的には 30 ~ 50 操作 / 秒と匹敵するが、記憶容量的には、1 インスタンス当たり 1kB 近く消費する。

4 まとめ

高速かつ低コストを目指して現在エージェントプラットフォームのテストベッドを開発中である。さらに目標性能の実証をおこなう予定である。

参考文献

- [1] Pavlow, G. et al. "The OSI Management Information Service User's Manual (ver. 1)", 1993
- [2] 佐々木 他, "リアルタイム OS を用いた TMN エージェントプラットフォーム", 信学技報 IN94-120, 1994
- [3] 清水 智明 他, "TNMS カーネルにおける MIB のオブジェクト指向データベースによる実装について", 信学技報 CS-93-115
- [4] 竹居 他, "MIB と管理対象装置の管理情報伝送", 信学会秋季大会, 1995
- [5] 登内 他, "OSI 管理用 MIB における記憶領域管理アーキテクチャ", 信学会秋季大会, 1995