オブジェクト指向によるネットワーク管理問題分析とOSI管理利用に関する考察

北橋 雅子* 野口 正一**

*情報処理振興事業協会 (IPA)　**東北大学応用情報学研究センター

まず、ネットワークの定義を明らかにする。それから、そのネットワーク管理に対する要求を示す。その上で、その要求を満たすために、オブジェクト指向問題分析を行う。さらに、その分析結果に基づいて、実際に管理システムを設計、実装する(プロトタイピング)。本論文は、オブジェクト指向問題分析、プロトタイピングを何回か繰り返した結果について述べるものである。ネットワーク管理要求は、最初は漠然としたものであるが、分析、プロトタイピングを繰り返すことによって、より具体的で詳細なものとなる。本論文で示す管理要求は、ある程度、具体化、詳細化が進んだものである。本論文では主に、分析結果について述べ、さらに、プロトタイピング時の留意点について言及する。

1. はじめに

何らかの通信手段でネットワークされた環境は、現在、あちらこちらに氾濫している。電話網、企業のコンピューターネットワーク、パソコンネット、等である。これらのネットワークは、利用者が増え、より高度なサービスを提供するようになってきている。そこで問題なのが、しだいに規模が大きくなり、複雑になってゆくネットワークを、どうやって管理すればよいのか、ということである。

非常に複雑な問題を正確に理解し、解決する方法として、オブジェクト指向問題分析が注目されている。そこで、これによって、マルチプロトコルネットワーク管理という問題を分析する。

その前に、マルチプロトコルネットワークの定義を明らかにする。それから、そのネットワーク管理に対する要求を列挙する。その上で、その要求を満たすために、オブジェクト指向問題分析を行う。さらに、その分析結果に基づいて、実際に管理システムを設計、実装する際の留意点について述べる。

2. マルチプロトコルネットワークの定義

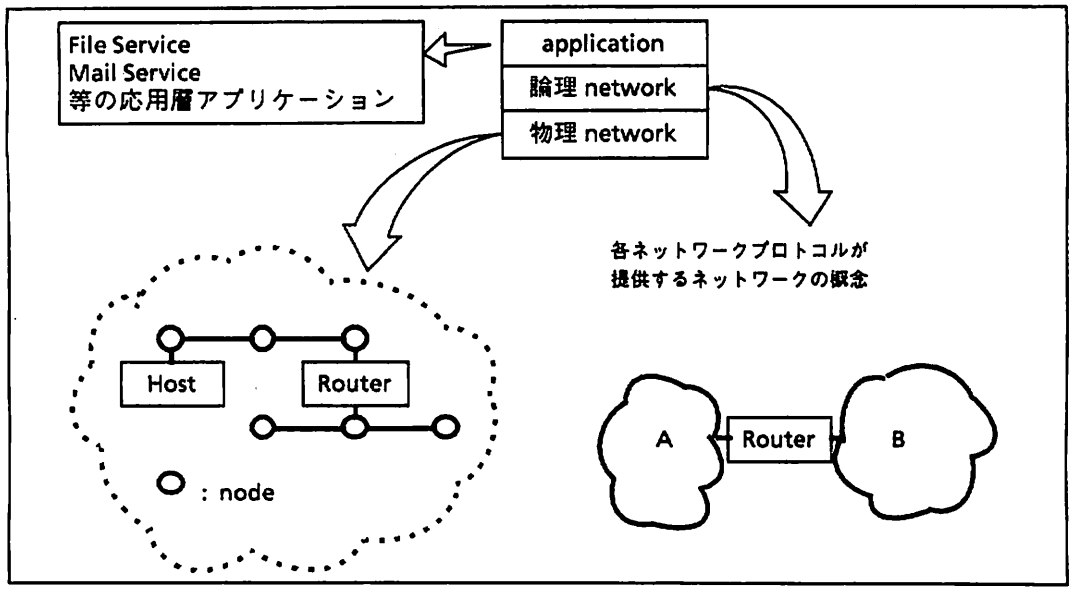
まず、ネットワークの定義を明らかにする。図1を参照。ネットワークとは、物理ネットワーク、論理ネットワーク、そして、ネットワークアプリケーションサービスの3層から成るものである。物理ネットワークは、ノードが、何らかのメディアによって物理的につながっている環境である。ノードとは、ワークステーション等のコンピューターホスト、ルーター、または、リピータ、ブリッジ等の機器である。メディアとは、イーサネットやFDDI等の線である。

物理ネットワークがあるだけでは何も起こらない。物理ネットワークを構成する機器が、メディアを介して通信を行うためには、各機器上に何らかの通信機能を載せなければならない。通信機能とは、TCP/IP[1]や、XNS[2]、または、OSI[3]等の各ネットワークプロトコルに従った機能である。通信機能を

Object oriented analysis for network management
and Consideration of using OSI Management
by Masako Kitahashi*, Shouichi Noguchi**
*Information - technology Promotion Agency, JAPAN　**Tohoku University

図1 ネットワークモデル

Fig. 1 network model



載せると、そのネットワークプロトコルが提供する論理ネットワークの概念が、物理ネットワークの上に重なって存在するようになる。論理ネットワークは、通常、ネットワーク番号というものによって認識される。図1では、ネットワークA、ネットワークBがあって、これらは、ルーターによってつながっている。このネットワークA、ネットワークBを総称して、ひとつの論理ネットワークと呼ぶ。

ネットワークアプリケーションサービスは、各ネットワークプロトコルが提供する、Mail Service、File Service等の応用層アプリケーションである。つまり、論理ネットワーク上に実現される何らかの通信サービスである。

ひとつの物理ネットワークに対して、複数の論理ネットワークがあるとき、それは、マルチプロトコルネットワークと呼ぶ。

3. 管理要求

より良いネットワーク管理を行うための具体的な管理要求を以下に示す。これは、現在ネットワーク管理を行っている人々にインタビューした結果をまとめたものであり、さらに、現在使われているネットワーク管理ツールが備えている機能を管理要求として考慮したものである。

図1のネットワークモデルの3つの層ごとに管理要求が存在する。

レベル1: 物理ネットワーク管理

① 物理ネットワークトポロジーの表示、又は変更。

レベル2: 論理ネットワーク管理

① 論理アドレスと、それに対応する物理アドレス、hostname等の表示。
② 各層ごとのEntityの状態表示、又は、変更。すなわち、層管理。
③ ネットワーク番号ごとの、ホスト、ルーター、メディアの表示。論理ネットワークを物理ネットワークに対応させて表示する。
④ 論理ネットワークトポロジーの表示。
⑤ エラーPacketの管理

⑥ 二重論理アドレスのチェック
⑦ ルーター、ブリッジ、リピータ等、メディアを中継している機器のテスト。テストとは、テストPacketを送信して、返事を受信することによって、正常に動作しているかどうか確認することである。
⑧ ネットワークの現状把握
高負荷talker, receiverの把握。
メディアの負荷状況の把握。
プロトコルごとのtrafficの推移、
broadcast packetの推移、等、統計情報による現状把握。

①②は、レベル1の物理ネットワークトポロジー上で表示したい、という要求である。④は、物理ネットワークトポロジーと関連付けて表示されなければならない。⑦のテストの結果、マシンがダウンしていることがわかった場合は、それは物理ネットワークトポロジーに反映されなければならない。

レベル3: ネットワークアプリケーションの管理

① 各サービスがどのホスト上にあるか表示。
② 各サービスが使用するネットワークプロトコルの表示。
③ ドメインごとのホスト、ユーザー、メールアカウント、サービスの表示。ドメインとは、Accountingにおける名前付けのメカニズムが提供するドメインの概念である。
④ ホスト、ユーザー、メールアカウント、サービスの追加、削除、変更。すなわち、Accounting。これは、物理ネットワークトポロジーに反映されなければならない。
⑤ ファイルサーバー等、重要なサービスを提供しているマシンのテスト。テストとは、テストPacketを送信して、返事を受信することによって、正常に動作しているかどうか確認することである。

①②③は、レベル1の物理ネットワークトポロジー上で表示したい、という要求である。

それぞれのネットワークアプリケーションは、論理ネットワーク上にどのようなサービスを提供するのか、というモデルを持っている。そのモデルに従って、アプリケーションごとに管理要求があるはずである。ここでは、その詳細について詳しく述べることは省略する。

4. マルチプロトコルネットワーク管理問題分析

オブジェクト指向問題分析手法である、コード・ヨードン法(4)、および、OMT(5)によって分析を行う。ここで言う分析とは、オブジェクトモデル化、動的モデル化、および、機能モデル化の3つを行うことである。オブジェクトモデルとは、問題を理解するために「オブジェクト」を用いて対象領域を抽象化したもので、問題領域の把握にとって本質的でない詳細を省略したものである。この抽象化は、ある目的に対して行われる。その目的にとって重要でないものは捨て去られる。ある目的に対して唯一の正しいモデルが存在するのではなく、いくつかの妥当なモデルが存在し得る。ただし、良いモデルは、問題の最重要の側面を捕らえて、その他は省略されたものであると言える。

ある程度、オブジェクトモデル化ができたら、動的モデル化、機能モデル化を行う。動的モデル化とは、ある目的を満たすための、オブジェクトからオブジェクトへの一連の操作の列を記述することである。動的モデルをシナリオと言うこともある。機能モデル化とは、オブジェクトの持つべき各機能について、そのデータフロー図等を記述することである。

最も重要なのは、オブジェクトモデル化である。「オブジェクト」とは、情報と機能をひとまとめにしてカプセル化したモジュールである。オブジェクトどうしがメッセージのやりとりをすることによってある目的が達成される。オブジェクトが他のオブジェクトにメッセージを送信する時、相手のオブジェクトのデータ構造や、手続きのアルゴリズム等に、全く関知する必要はない。これが、データ構造と手続きが隠蔽されているという

ことである。

先に示した管理要求を目的として、オブジェクトモデル化を行う。ところで、管理要求はより具体的に、より詳細に記述されなければならない。よりよいオブジェクトモデル化のために、具体的で詳細な目的が必要である。ところが、目的をより具体的に、より詳細に把握するためには、よりよいオブジェクトモデルが必要である。

まず、管理要求をできるだけ具体的に、詳細に記述する。そのためには、問題領域をできるだけ深く理解しなければならない。最初は、漠然とした部分があったり、正確でなかったりするが、とにかく、できるだけ努力して記述した管理要求を目的として、オブジェクトモデル化を行う。すると、その過程で、もっと詳細に認識すべき点や、誤って解釈していた点等がわかってくる。そこで、管理要求の記述を書き換えたり、付け加えたりする。つまり、オブジェクトモデル化と管理要求の具体化、詳細化は、ほとんど平行して進むわけである。

先に示した管理要求は、今までの分析作業によって、ある程度具体化、詳細化が進んだものである。では、これらの管理要求を満たすためのオブジェクトモデル化について述べる。まず、管理要求にプライオリティーをつける。プライオリティー1から5まで設定する。プライオリティー1が、最優先であるとする。

プライオリティー1. レベル1の①
プライオリティー2. レベル2の①②③④
レベル3の①②③
プライオリティー3. レベル3の④
プライオリティー4. レベル2の⑤⑥⑧
プライオリティー5. レベル2の⑦
レベル3の⑤

プライオリティー1から3までが、構成管理要求である。この中には、層管理要求であるところのレベル2の②も含まれている。プライオリティー4から5が、障害管理要求である。

まず、プライオリティー1の管理要求を満たすためにオブジェクトモデル化を行う。次に、矛盾が生じないように考慮しながら、プライオリティー2の管理要求を満たすためにオブジェクトモデルを拡張する。以下、プライオリティーに従って、順番に管理要求を考慮し、オブジェクトモデルを拡張する。

4.1オブジェクトモデル

図2.1から2.6に、先の管理要求を全て考慮したオブジェクトモデルを示す。図中において、オブジェクトはそれぞれひとつの箱によって示される。箱は3段に分かれており、1段目がオブジェクトクラス名、2段目が属性、3段目はそのオブジェクトの機能を示す。細い線の矢印は、部分全体関係を示す。矢印の指している方が全体、その反対側が部分、つまり、図2.1で言うと、オブジェクトEquipmentは、オブジェクトPhysicalNetworkの構成要素であることを意味している。太い線の矢印はスーパークラス・サブクラスの関係を示す。矢印の指している方がスーパークラスである。また、点線の矢印はその他の関係を示す。

4.1.1 PhysicalNetwork

まず、図2.1参照。プライオリティー1の管理要求は、物理ネットワークトポロジーの表示と変更である。物理ネットワークとは、ノードがメディアによってつながっている物理的な環境であった。従って、ノードとメディアをオブジェクトとして捕らえる。

メディアをPhysicalMediaというオブジェクトクラスで表し、ノードをEquipmentというオブジェクトクラスで表す。さらに、物理ネットワークをPhysicalNetworkというオブジェクトクラスで表し、PhysicalNetworkは、PhysicalMediaとEquipmentによって構成されているとする。

ところでPhysicalNetworkは、それを構成するメディアの種類によって、FDDINetや、EtherNet等に特殊化される。それぞれ、FDDINetを構成するメディアはFDDI、EtherNetを構成するメディアはEtherである。FDDI、Etherは、PhysicalMediaのサブクラスである。

PhysicalNetworkを構成要素とするMDomainというオブジェクトクラスを導入す

図 2.1 オブジェクトモデル PhysicalNetwork
Fig. 2.1 ObjectModel PhysicalNetwork

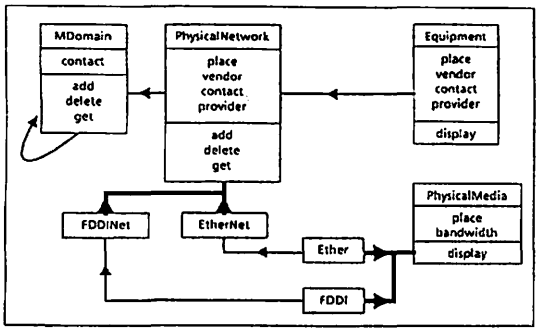


図 2.2 オブジェクトモデル Equipment
Fig. 3.2 ObjectModel Equipment

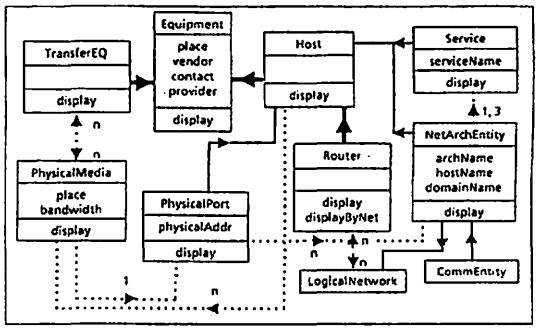


図 2.3 オブジェクトモデル CommEntity
Fig. 2.3 ObjectModel CommEntity

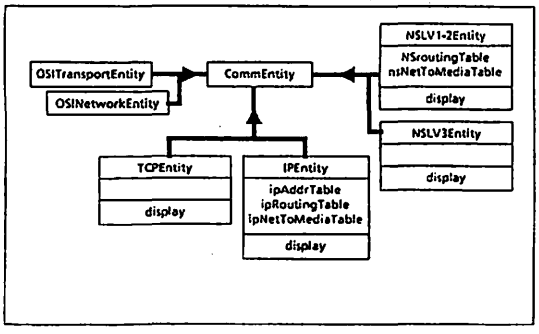


図 2.4 オブジェクトモデル Connection
Fig. 2.4 ObjectModel Connection

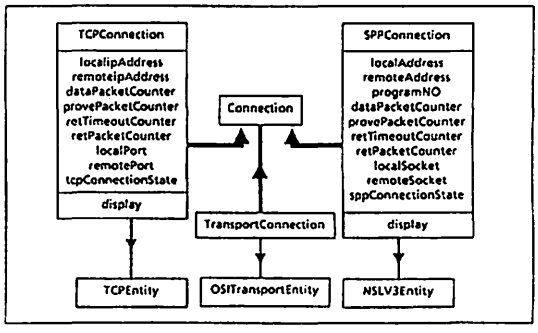


図 2.5 オブジェクトモデル Service
Fig. 2.5 ObjectModel Service

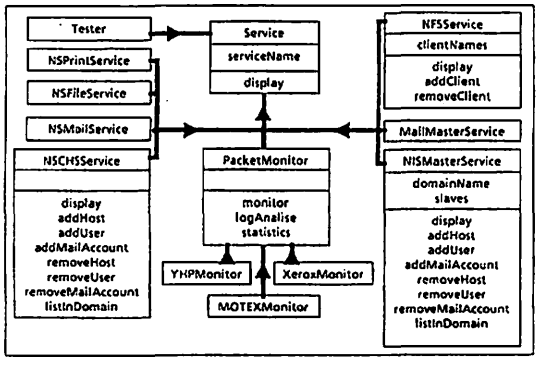


図 2.6 オブジェクトモデル LogicalNetwork
Fig. 2.6 ObjectModel LogicalNetwork

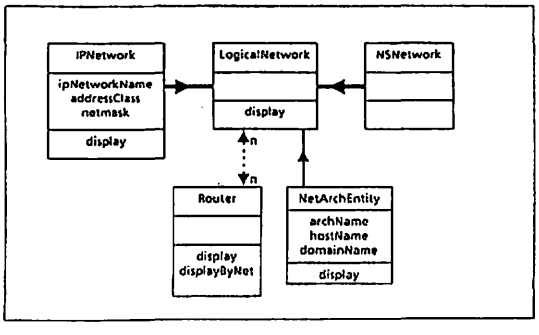
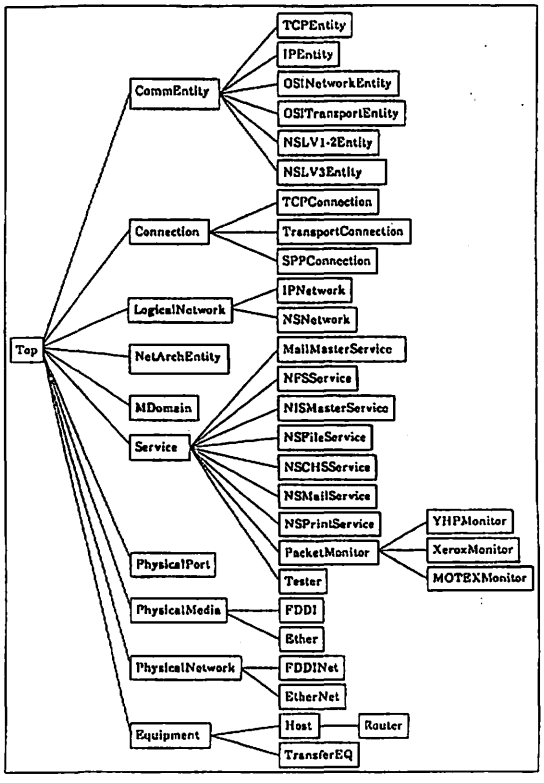


図 3 クラス継承木
Fig. 3 Class Inheritance Tree



る。MDomainは別のMDomainを構成要素とする場合もある。MDomainは、すなわち、物理的な管理領域を意味する。

管理者は、自分の管理すべき物理ネットワークを、MDomainにaddする。それぞれの物理ネットワークに機器やメディアをaddしたり、deleteしたりする。また、管理者は、必要に応じて、MDomainから、ある物理ネットワークをgetしたり、その物理ネットワークから、機器やメディアをgetして、それぞれの情報を得ることができる。

4.1.2 Equipment

次に、図2.2参照。Equipmentは、単なる中継機器であるところのTransferEQとcpuを登載したHostに特殊化される。プライオリティー1の管理要求が物理ネットワークの構成管理要求であったのに対して、プライオリティー2、3の管理要求は、論理ネットワークの構成管理要求である。論理ネットワークは、物理ネットワークを構成する機器上に、あるネットワークプロトコルに従った通信機能が実装されたときにはじめて実現する。論理ネットワークが物理ネットワーク上にどのように実現されているか、正確に反映するようにモデル化することが、管理要求を満たすことにつながる。TransferEQは単なる中継機器であるから、これに通信機能が実装されることはない。従って、Host上に通信機能が実装されることによって、論理ネットワークが実現される。では、Host上に実装された通信機能を、オブジェクトモデルとしてどのように表現するかを示す。

あるネットワークプロトコルに従った通信機能、あるいは、通信プログラムを、NetArchEntityというオブジェクトで表す。実際に通信するためには、HostとPhysicalMediaをつなぐためのポートが必要である。通信プログラムは、ポートを介してHostとPhysicalMediaがつながっていると認識している。従って、このポートを、PhysicalPortというオブジェクトで表す。

PhysicalPortとNetArchEntityはHostの構成要素である。NetArchEntityはあるネットワークプロトコルに従った通信プログラムであるから、あるHostが複数のプロトコルをサポートしている場合は、複数のNetArchEntityが、そのHostの構成要素として存在する。

NetArchEntityはn個のPhysicalPortを使用する。

PhysicalPortには、必ずただ一個のPhysicalMediaがつながっている。PhysicalMediaにはn個のHostがつながっている。TransferEQはn個のPhysicalMediaを中継しPhysicalMediaはn個のTransferEQによって中継される。

応用層サービスをServiceというオブジェクトクラスで表す。ServiceはHostの構成要素である。Serviceは1個以上、最大3個のNetArcEntityを使用する。この分析では、今のところ3種類のプロトコルしか考慮していないので、最大3個ということになる。

4.1.3 Router

図2.2では、Hostのサブクラスとして Routerが存在する。RouterはHostのサブクラスであるから、Hostの特性を全て継承する。それらに加えて、Routerとしての特殊な情報と機能を持っている。

ところで、ルーティング機能は、OSIでいうところの第3層の機能である。そこで、ルーティング機能をRoutingFunctionというオブジェクトクラスで表して、第3層に対応するエンティティが、これを構成要素として持つ、というふうに考えることもできる。そうすると、Routerという概念は存在しなくなる。

管理要求を満たすためにどちらの考え方が有効かというと、それは前者である。なぜならば、Routerとそれ以外のHostを区別して管理したいという強い要求があるからである。後者の考え方では、RouterとHostを区別しにくい。

Routerはn個のLogicalNetworkをルーティングし、LogicalNetworkは、n個のRouterによってルーティングされる。

レベル2の③「論理ネットワークごとに物理的構成を表示」という要求を満たすために、Routerには、displayByNetという機能

がある。

4.1.4 CommEntity

図2.3参照。プロトコルの各層ごとのエンティティをCommEntityというオブジェクトクラスで表す。CommEntityは、NetArcEntityの構成要素である。CommEntityは、プロトコルの各層ごとに対応したエンティティというサブクラスとしてそれぞれ具体化される。

各エンティティの間をデータが行き来することによって通信が行われるのであるが、各エンティティを関連付けるのは、このデータの行き来だけである。従って、各層ごとのエンティティは、基本的に、互いに独立である。上位層のエンティティが下位層のエンティティを包含したり、下位層のエンティティが上位層のエンティティから何かを継承したり、という関係は存在しない。

4.1.5 Connection

図2.4参照。プロトコルごとのConnectionの概念をConnection、および、そのサブクラスというオブジェクトクラスで表す。このConnectionは、Connectionをはる側がこれを生成するものとする。

TCPConnection,TransportConnection, SPPConnectionは、それぞれTCPEntity, OSITransportEntity, NSLV3Entityの構成要素である。

4.1.6 Service

図2.5参照。Serviceは、プロトコルごとのサービスとしてそれぞれ具体化される。たとえば、NSCHSService、NISMasterServiceは、Accountingに関する要求を満たすための機能を実現するサブクラスである。すなわち、Accountingを論理ネットワーク上のサービスとして捕らえているわけである。プライオリティー4、5の障害管理要求を満たすために、Tester、PacketMonitorというServiceのサブクラスがある。これらは、すなわち、論理ネットワーク上の障害管理サービスとして存在している。

4.1.7 LogicalNetwork

図2.6参照。プロトコルごとの論理ネットワークの概念を、LogicalNetworkというオブジェクトクラスで表す。

LogicalNetworkはNetArcEntityを構成要素とする。

図3に、以上の全てのオブジェクトクラスの、継承関係を示す。左側がスーパークラスである。

4.2動的モデル、機能モデル

全ての管理要求について、動的モデル(シナリオ)を作成しなければならない。動的モデルとは、時系列に従ったオブジェクトからオブジェクトへの操作の列、及び状態遷移図である。総てのシナリオ中一回も参照されない属性、利用されない機能があれば、それは必要ない、ということである。逆に、必要な属性、機能がない、ということがわかった場合は、どこか適切なオブジェクトにそれを付け加えるか、または、オブジェクトモデルの構成を考え直す必要があるということになる。

動的モデル(シナリオ)が完成したら、それぞれのオブジェクトごとに、その機能を、データフロー図等で詳細化しなければならない。すなわち、機能モデル化である。

5. プロトタイピング

プロトタイピングは、オブジェクト指向問題分析を行う上で欠かせない要素である。プロトタイピングとは、分析結果に基づいて、設計、プログラミングを試しに行ってみることである。その結果は、当然、分析に反映されなければならない。つまり、フィードバックがかかるわけである。最初の分析が完璧であることは、非常にまれである。

実装は、Smalltalk80TM(6)等のオブジェクト指向言語、または、オブジェクトデータベースを利用するものとして、その上で、前項で示した分析結果に基づいて設計を行う。

設計も当然、オブジェクト指向で行う。すなわち、実装言語や環境を意識して、オブジェクトモデル、動的モデル、機能モデルを詳細化する。

分析の結果、得られたオブジェクトのインスタンスを生成する時に、そのインスタンスが表す実際の対象、たとえば、実際のホス

ト、または、実際のルーター等と通信しなければならない。実際の対象が持っている管理情報ベースにアクセスする必要があるからである。このような通信のためのプロトコルとして、TCP/IP 上の SNMP(7) や、OSI の CMIP(8)等がある。

分析の段階では、このような管理のための通信は考慮する必要がなかった。しかし、設計の段階ではこれを考慮しなければならない。そこで、管理通信のためのオブジェクトクラスをオブジェクトモデルに追加する。

SNMPManager と OSIManager というオブジェクトクラスを定義し、それぞれ、SNMP agent、OSI agent と通信する機能を持たせる。Host、Router といったオブジェクトは、SNMPManager や OSIManager を使って通信する。

全ての通信手続きは、各オブジェクトの機能として隠蔽する。その手続きは、設計段階の機能モデルとして、各オブジェクトの機能ごとに記述される。

6. おわりに

オブジェクト指向によるネットワーク管理システムを設計、実装するにあたって、まず、何よりも重要なことは、オブジェクト指向問題分析を行うことである。その際、気をつけなければならないのは、分析の段階で、管理のための通信を必要以上に意識しないことである。管理情報を得るための通信手続きに惑わされることなく、問題分析を行うことが重要である。この基本姿勢を守れば、ネットワーク管理という主題から横道にそれることなく、分析、プロトタイピングを繰り返すことができる。

本文中にも述べた通り、ここに示した分析結果は完璧なものではない。しかしこれは、分析、プロトタイピングを何回か繰り返した結果であり、これからオブジェクト指向によるネットワーク管理システムを構築しようとする人々にとって、何らかの参考になれば幸いである。

本研究では、今後、分析、プロトタイピングを更に繰り返し、管理要求を更にブレイクダウンしていく。最終的には、プロトタイピングから一歩進んで、実用的なネットワーク管理システムの構築をめざす。

7. 謝辞

本研究を進めるに当たって、日本アイ・ビー・エム(株)小林 善和 氏、日本電気(株)勅使河原 可海 氏、富士通(株)高橋 修 氏、(株)SRA 佐原 伸 氏、富士ゼロックス情報システム(株)龍田 直紀 氏、羽生田 栄一 氏、また、INTAP NM WG の方々のご協力に感謝の意を表する。

8. 参考文献

(1) Douglas Comer : Internetworking with TCP/IP, P. 382, Prentice-Hall, Inc., 1988

(2) 上谷 晃弘 編著：ローカルエリアネットワーク -イーサネット概説-、P. 280、丸善(株)、1985

(3) ISO 7498: 1984 Information processing systems - Open Systems Interconnection - Basic Reference Model

(4) Peter Coad and Edward Yourdon : Object-Oriented Analysis (Second Edition), P. 233, Prentice-Hall, Inc., 1991

(5) J.ランボー/M.ブラハ/W.プレメラニ/F.エディ/W.ローレンセン著、羽生田 栄一 監訳：オブジェクト指向方法論 OMT モデル化と設計、P.544、Prentice-Hall, Inc. / (株)トッパン、1992

(6) adele Goldberg 著、相磯 秀夫 監訳：SMALLTALK-80 -対話形プログラミング環境-、P.480、オーム社、1986

(7) Marshall T. Rose : A Simple Network Management Protocol (SNMP), RFC1157

(8) ISO/IEC 7498-4, (X.700) Management Framework
ISO/IEC 9595 (X. 710) Common Management Information Services
ISO/IEC 9596 (X.711) Common Management Information Protocol