

## オブジェクト指向技術の伝送網管理への適用

榮 浩三      藤井伸朗  
NTT 伝送システム研究所

あらまし    オブジェクト指向の考え方が取り入れられた通信網管理のためのインタフェース設計が積極的に進められている。同時にオブジェクト指向技術をシステム記述に用いることも実現性を帯びてきた。

本論文は伝送網の構成管理をオブジェクト指向技術を用いて実現する場合の技術課題のうち、伝送網を構成するCircuitインスタンスへのメソッド配分の方法の比較検討を行い、Smalltalk-80の環境下においてインスタンス生成シナリオをクラスとインスタンスの双方に配置する方法がインスタンスの生成の効率を上げるために優れていることを示す。さらに処理言語の環境を拡大することによる効果も明らかにする。

## An Introduction of Object Oriented Technologies to a Transmission Network Configuratin Management System

Kouzou SAKAE      Nobuo FUJII  
NTT Transmission Systems Laboratories  
1-2356 Take Yokosuka-shi Kanagawa 238-03 Japan

**Abstract**    An object-oriented interface design has been discussed in an interoperable network management area, while various kinds of object-oriented technologies are available for uses in system designs.

This paper deals with the introduction of object-oriented technologies for a transmission network configuration management system. Among many technical issues, method allocations in a class and/or instances for CIRCUIT instance management are focused. Under the assumption that the Smalltalk-80 environment, method allocations both in a class and instances has an advantage in terms of the number of instances to be created in a period. In addition to the result, an effect caused by processing restriction improvement is addressed.

## 1. まえがき

コンピュータシステムのネットワーク化と平行して、コンピュータシステムによるネットワーク管理の研究がISOを中心積極的に進められている[1]、[2]。

一方、通信網の建設から運用保守までを含めた通信網管理のシステム化も通信網の運用コスト、サービス性を向上する上で不可欠なものとして幅広い検討が展開されている[3]、[4]。両者に共通することは管理を情報システムにおける情報処理としてとらえ、情報の表現とその操作の規定に統一的な手法を適用する点にある。システムの大規模化、情報の益々の複雑化が予想される中で情報のインテグリティを確保する上で優れているオブジェクト指向のアプローチがとられ、管理対象クラスの定義とインスタンス生成およびインスタンス操作により管理体制を構築しようとしている[5]。

オブジェクト指向の情報管理手法はERモデルを始めとしてデータベーススキーマ設計の領域で既に幅広く適用されている[6]。オブジェクト指向プログラミング技術は、主として情報処理のプロトタイプツールとして閉じた世界を提供するSmalltalk-80を出発点とし、AI技術としての適用、システム記述言語への適用など広範囲な応用に使用可能なものが普及しつつある[7]。さらにはデータベースの領域においてはオブジェクト指向データベース[8]、分散処理を考慮した平行オブジェクト指向言語[9]とははシステム設計の要求を満たすだけのオブジェクト指向パラダイムが形成されつつあるといえる。即ちインタフェース上の情報定義に限定することなく、システムの全般においてオブジェクト指向の処理系の利点を生かした設計が可能になってきたものといえる。

本論文は通信網において伝送システムと端局装置、クロスコネクト装置からなる階層構造を持つ伝送網[10]の管理システム構成にオブジェクト指向技術を適用する上での技術課題と適用例を示すものである。尚、適用性検討の第一段階として、適用するオブジェクト指向技術はオブジェクト指向のシステム記述ツールに限定し、ツール(言語)の仕様もSmalltalk-80を前提とした極めて入門的なものとした。まず、対象とする伝送網と管理システムの概要および目的を明らかにする。次にオブジェクト指向技術の適用を前提とした伝送網のモデル化、管理機能の記述およびシナリオを提案する。さらにはシナリオをオブジェクト指向技術で実現する上での技術課題を示し、課題の解決を含むオブジェクト指向処理系の設計例の提案と比較する。

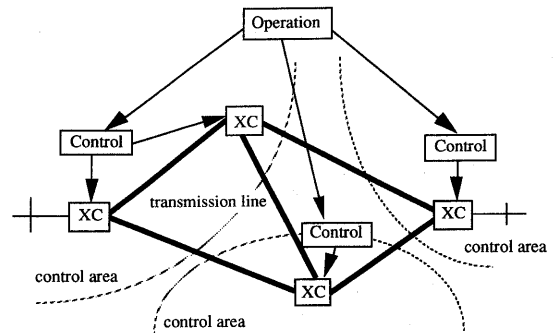


FIGURE 1  
TRANSMISSION NETWORK OPERATION

図1 伝送網管理システムの構成概要

## 2. 伝送網管理システム構築のねらい

### 2.1 伝送網管理システム概要

伝送網は伝送方式とクロスコネクト装置で構成されるネットワークで、ネットワーク上の任意の2点間に対して通信機能を提供する[10]。回線交換網とは異なり通信機能を計画に従い提供する。図1に伝送網管理システムの構成概要を示す。複数のクロスコネクト装置を地域的に監視制御する制御システムと、これらのシステムに対して制御条件を与え、オペレーション情報を収集し伝送網の運用状況を管理するオペレーションシステムにより構成する。オペレーションシステムは以下に示す機能を提供する。

- ・伝送網を構成する情報の生成、
- ・伝送網の構成するシナリオの実行、
- ・伝送網の個々の通信機能の性能の把握、
- ・伝送網の個々の構成要素の状態の把握、
- ・故障検出時の網的な処置。

### 2.2 OSIシステム管理の適用

OSI管理ではコンピュータネットワークの管理応用エリアとして、構成管理、障害管理、性能管理、会計管理、およびセキュリティ管理を挙げ、これらのアプリケーションが利用するサービスをOSIシステム管理の機能としている。これらのサービスの検討にはオブジェクト指向のアプローチが適用され、管理する情報を管理対象として定義し、共通情報管理サービスを用いてシステム管理サービスを個々に定義し、管理対象の操作により管理機能を実現する[1]。図2にOSI管理のモデルを示す。Managerの役割を果たすシステムとAgentの役割を果たすシステムが対向して管理情報サービスを用い、

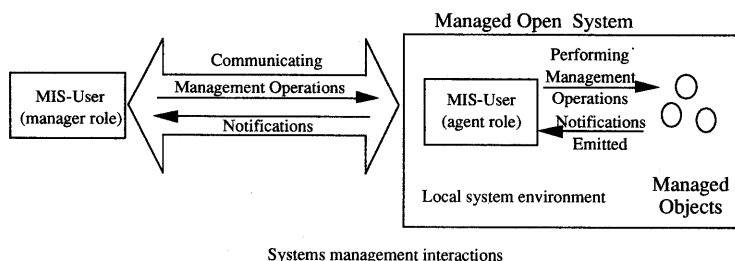


FIGURE 2  
OSI SYSTEM MANAGEMENT MODEL

図2 OSI管理モデル

Agentにより管理対象として具現化されたリソースを操作する。本モデルは伝送網管理システムの構成要素である、クロスコネクタ装置、制御システムおよびオペレーションシステムをオープンシステムとして捉えることにより、伝送網管理システムの構成要素間のインタフェース設計に直接適用することができる。CCITTにおいては既にOSI管理のインタフェースをテレコムネットワークを主な対象とした通信網管理システムへ適用しようしている[5]。

### 2.3 オブジェクト指向技術の適用性

伝送網管理システムのシステム構成要素間のインタフェースがオブジェクト指向の情報定義手法と操作手法を用いて規定される技術動向との親和性を考慮し、構成要素内部のシステム構成にもオブジェクト指向技術の導入を考える。オブジェクト指向技術ではオブジェクト指向プログラミング言語とオブジェクト指向データベースが技術の中心となる。管理システムをオブジェクト指向技術で構成する場合、ManagerシステムとAgentシステムでは設計条件が異なる。図3は管理システムを操作する操作者も含める処理のモデルである。操作者はオブジェクト指向のヒューマンマシンインタフェースに対して操作内容を指示する。ヒューマンマシンインタフェー

ス処理部は操作内容に従いシナリオを展開し、オブジェクト操作のシナリオを作成する。オブジェクト操作のシナリオはManagerのアプリケーションによりインタフェースのサービスを起動する。インタフェースのサービス起動によりAgentに伝達されたオブジェクト操作情報はAgentのアプリケーションによりオブジェクト操作に展開される。オブジェクト操作は管理情報ベースの操作が中心となる。管理情報ベースの情報の内容のうち、リソースと関係するものは他アプリケーションによりリソース状態データ/制御データに変換されリソースの状態を更新する。オブジェクト指向技術の提供領域は、

- ・ Manager部分のHMI機能の実現、
  - ・ Manager部分の個別管理シナリオ、
  - ・ インタフェース処理部分 (Manager/Agent)、
  - ・ Agent部分の個別管理シナリオ、
  - ・ Agentが管理するMIB構成、
  - ・ MIBとリソースのインタフェース部分、
- となる。

### 2.4 検討の前提条件

本検討は伝送網管理システムの構成にオブジェクト指向技術を適用し技術課題等を明らかにするものであるが、以下を検討の前提とした。

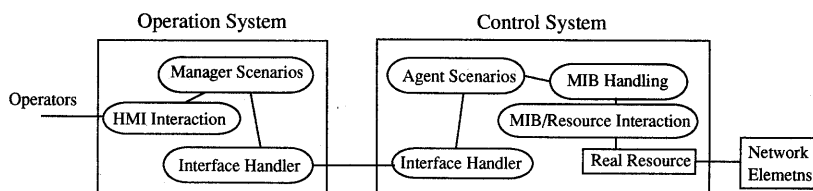


FIGURE 3  
AREA OF OBJECT ORIENTED TECHNOLOGY BEING APPLIED

図3 オブジェクト指向技術を適用した処理モデル

- ・ Agent部分の個別管理シナリオが対象、
- ・ オブジェクト指向プログラミング言語を適用、
- ・ オブジェクト指向プログラミング言語の条件はSmall-talk-80をベース。

尚、オブジェクト指向データベース、HMIなどは検討の範囲外とする。

### 3. オブジェクト指向伝送網管理シナリオ

#### 3.1 伝送網のモデル化

伝送網の構成要素は図4に示す管理対象クラスのインスタンスとして取り扱うことができる<sup>[11]</sup>。EQUIPMENT管理対象クラスは装置を代表するもので、クロスコネクト装置や装置内部の部分を表す。CIRCUIT管理対象クラスは2点間の通信機能を代表するもので、伝送容量の異なるパスを表す。CIRCUIT管理対象インスタンスは2つの端点をそれぞれaEndPointName, zEndPointNameと示し、構成要素をcomponentNamesとして示し、複数の任意の管理対象インスタンスと関係をもつことができる。EQUIPMENTおよびCIRCUITはインスタンスの部分を新たなインスタンスとする包含関係を定義することができる。特に包含されるCIRCUITのインスタンスをSubordinate Circuit、複数の要素を接続して構成したCIRCUITのインスタンスをComposite Circuitと呼ぶ。

各管理対象クラスの代表的属性は以下の通りである。

#### EQUIPMENT管理対象クラス：

equipmentType: 装置の種類、  
administrativeState: 外部から指示する管理状態、  
operationalState: 動作状態、

#### CIRCUIT管理対象クラス：

circuitType: circuitの種類、  
administrativeState: 外部から指示する管理状態、

operationalState: 動作状態、

bandwidth: 伝送容量、

aEndPointName: インスタンスの端点に位置する他のインスタンス名、

zEndPointName: 端点に位置する他のインスタンス名、

componentNames: 構成要素となる他のインスタンス名。

#### 3.2 管理機能一覧

伝送網の管理機能概要を以下に示す。

**構成管理：** XC装置、150M伝送路の設置とクロスコネクト装置制御による1.5Mパスの構成、

**故障管理：** 伝送路網構成要素が検出する警報の収集と分析、分析結果に基づく故障評定と復旧措置、

**性能管理：** 装置性能、伝送路の品質、パスの品質の各パラメタの収集と蓄積、性能評定と改善措置、

**会計管理：** 伝送網構成要素の使用状況の収集による伝送コスト分析、

**セキュリティ管理：** オペレーション機能、制御機能へのアクセスに対するスクリーニング処理、

上記管理機能は関連する管理対象インスタンスの生成、削除、属性値の変更、獲得による情報操作により実現する。尚、警報転送や性能情報の収集ではその機能をサポートする管理対象インスタンスを用いる<sup>[11]</sup>。代表的なサポート用のオブジェクトには、イベント報告用オブジェクト(EVENT REPORTING SIEVE)やイベントログ(EVENT LOG)、イベントレコード(EVENT RECORD)を挙げることができる<sup>[12]</sup>。

#### 3.3 代表的シナリオ

1.5Mパスの開通シナリオは次の通りである。

(前提条件)：クロスコネクト装置、150M伝送路が

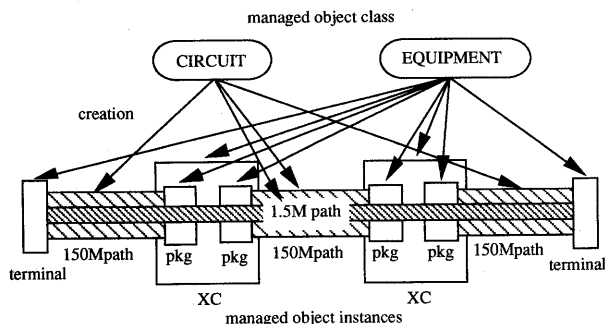


FIGURE 4  
A MODEL OF TRANSMISSION NETWORK

図4 伝送網モデル

既に接続され伝送網が形成されている。

(1 M) 1.5 Mパスの通過する経路を決定、

(2 A) 通過経路の150 Mパス内に1.5 Mの容量を持つインスタンスを生成、

[インスタンスの管理状態は使用不可状態]

(3 A) 生成したインスタンスを構成要素として1.5 Mパスのインスタンスを生成、

[構成要素の端点相互間を対応付けるXC制御をAgentが実行]

[インスタンスの管理状態は使用不可状態]

(4 A) 警報報告用、性能情報収集用のインスタンスを生成、

(5 M) 警報報告および性能情報を一定期間分析、

(6 A) 構成要素インスタンスおよび1.5 Mインスタンスの管理状態を使用可状態に。

以上のシナリオにおける(\*M)はManagerがローカルに行う処理で、(\*A)はManagerからの指示によりAgentが行う処理である。

#### 4. オブジェクト指向技術適用上の課題と対策

伝送網構成管理はSubordinate CircuitとComposite Circuitの生成により実現する。各インスタンスは共通のCIRCUITクラスから生成され、各々は包含関係を持つ。オブジェクト指向技術ではオブジェクトにインスタンス管理の機能を付与することにより、従来のデータと管理シナリオが分離された処理機能と比べて、情報のインテグリティが確保しやすい処理系を実現することができる。オブジェクトがインスタンス管理を行う場合の具体的技術課題と解決方法を考察する。

##### 4.1 オブジェクトクラスとインスタンス

オブジェクト指向プログラミング言語ではクラスとインスタンス各々に対してメソッドを定義することができる。オブジェクトによりインスタンス管理を実現するためにはこれらのメソッドのクラス、インスタンス間の役割分担を明らかにする。

包含関係を持つCircuitインスタンスの生成において、インスタンス生成のシナリオの記述位置による生成効率の違いにより、メソッドの役割分担を評価する。

##### 4.2 平行処理および処理機能の確保

伝送網管理を効率よく進めるためには、Managerは同時に複数のCircuitインスタンス生成メッセージをAgentに送り、連続したインスタンスの生成機構を実現する必要がある。オブジェクト指向技術の特徴を生かしAgentが介

在する処理を削減すると、Agentの役割はオブジェクトにManagerの要求を伝達するだけとなり、オブジェクトに対するメッセージが輻輳する。その結果、オブジェクトはその処理機構において以下の状況を解決しなければならない。

- ・オブジェクト上に滞る実行待ちのメッセージの処理、
- ・同時に届いたメッセージについての優先順位の決定、
- ・オブジェクトによる処理過程の管理。

状況を解決するためには、オブジェクト指向プログラミング言語の仕様の観点とオブジェクトの相互関係の観点がある。本検討ではオブジェクトの相互関係から平行処理により適したメソッド配分を評価し、続けて言語仕様の拡張による改善効果を明らかにする。

##### 4.3 Circuitインスタンスの管理機構

比較検討の対象とするCircuitインスタンスの管理機構を明らかにする。管理機構は

- ・Subordinate Circuitの生成機構、
- ・Composite Circuitの生成機構、

から成り立つ。Subordinate Circuitの生成機構は包含するCircuitインスタンスと包含されるCircuitインスタンスの関係を保つために生成のシナリオは全て包含するCircuitインスタンス上にだけ配備するものとしている。Composite Circuitの生成機構はクラスが中心となるシナリオとクラスとインスタンスが役割分担するシナリオの2つを挙げる。

各生成機構で用いるインスタンスおよびクラスの名前は以下の通りである。

CIRCUIT： オブジェクトクラス

CircuitA、CircuitB、CircuitC： クラスCIRCUITから生成されたインスタンス、

SubCircuit： インスタンスCircuitAの帯域を分割したものでCircuitAに包含され、クラスCIRCUITから生成されたインスタンス、

ComCircuit： インスタンスCircuitB、CircuitC、Equipmentを構成要素とする、クラスCIRCUITから生成されたインスタンス、

Equipment： クラスEQUIPMENTから生成されたインスタンス。

###### 4.3.1 Subordinate Circuitの生成機構

図5にAgentがCircuitAからSubCircuitを生成する過程を示す。各過程の処理内容は以下の通りである。

- (1) AgentはCircuitAに対し"create subordinate circuit <SubCircuit>"メッセージを送る。
- (2) CircuitAはクラスCIRCUITにインスタンス領域の生成

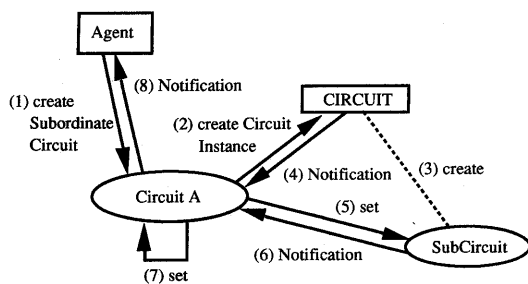


FIGURE 5  
SUBORDINATE CIRCUIT CREATION  
FLOW MECHANISM

図5 Subordinate Circuitの生成機構

メッセージ"create circuit <SubCircuit>"を送る。

- (3) CIRCUITはCircuitAのSubordinate CircuitとなるインスタンスSubCircuitの領域を生成する。
- (4) CIRCUITはインスタンスの領域を確保したことをCircuitAに通知する。
- (5) CircuitAはSubCircuitに属性値設定メッセージを送る。
- (6) SubCircuitはCircuitAへ設定完了を通知する。
- (7) CircuitAは自身の属性値の変更を行う。
- (8) CircuitAはAgentへSubCircuit生成完了を通知する。

#### 4.3.2 Composite Circuitの生成機構

図6にAgentがCircuitB、CircuitCおよびEquipmentを構成要素とするComCircuitを生成する2つの方法を示す。各方法の処理内容は以下の通りである。

[方法1]

- (1) Agentがメッセージ"create composite circuit <comCircuit>"をクラスCIRCUITに送る。
- (2) CIRCUITはComCircuitの領域を生成する。
- (3) CIRCUITはComCircuitと構成要素となる各インスタンスCircuitB、CircuitCに対して属性値の設定、書き換えを行う。
- (4) CIRCUITはAgentにComCircuitの生成完了を通知する。

[方法2]

- (1) 方法1の(1)に同じ。
- (2) CIRCUITはComCircuitの領域を生成し、属性値を設定する。
- (3) ComCircuit自身が構成要素となる各インスタンスCircuitB、CircuitCの属性値を書き換える。
- (4) ComCircuitはクラスCIRCUITに属性値書き換え完了を

通知する。

(5) CIRCUITはAgentにComCircuit生成完了を通知する。  
方法2では、クラスCIRCUITとインスタンスComCircuitがインスタンスの管理を分担する。

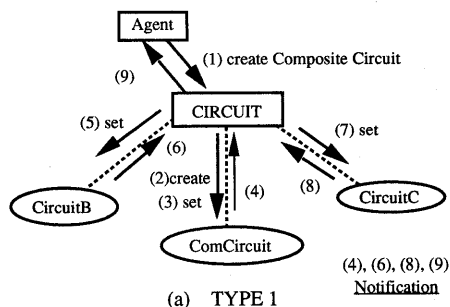
## 5 システム設計例

### 5.1 Smalltalk-80環境による処理機構の構成

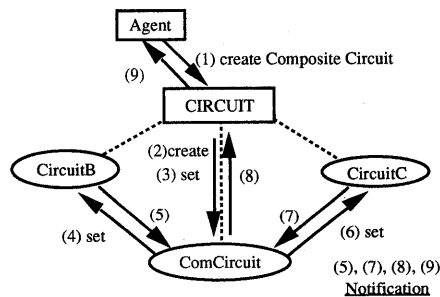
Smalltalk-80環境では、オブジェクトに次の制約が適用される。

- ・一つのメソッドが完了するまで他のメソッドを実行しない、
- ・メッセージを送ると送り先のオブジェクトからの応答待ちとなる。

以上の制約をSubordinate CircuitとComposite Circuitの生成機構に与えると、オブジェクト間でのメッセージの送受は図7-(a)のようになる。各メッセージ番号は4.3.1の処理内容に対応する。オブジェクトは処理の過程において、新たな処理が可能なunlock状態、新たな処理の受け付けができないlock状態の2つの状態を持つ。さらにlock状態は以下の2つの状態をとる。



(a) TYPE 1



(b) TYPE 2

FIGURE 6  
COMPOSITE CIRCUIT CREATION  
FLOW MECHANISM

図6 Composite Circuitの生成機構

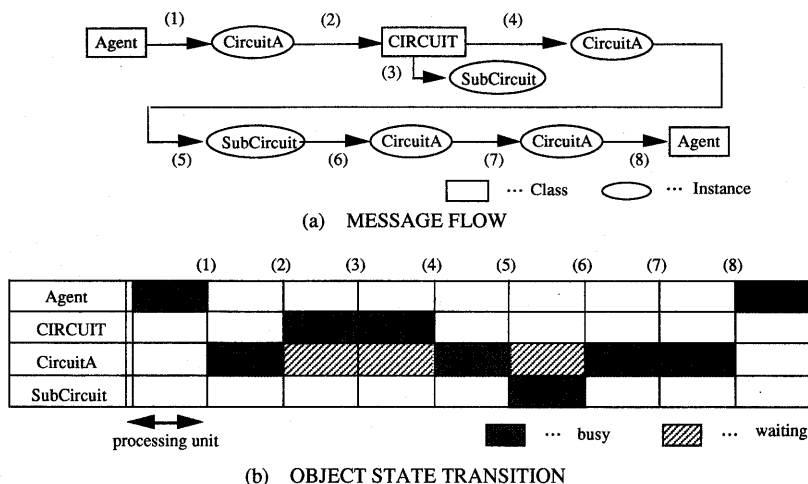


FIGURE7  
SUBORDINATE CIRCUIT CREATION FLOW  
図7 Subordinate Circuitの生成過程

busy：メソッド実行中、  
waiting：他オブジェクトからの応答待機中。

図7-(b)は1つのCircuitインスタンス生成におけるクラスとインスタンスの状態の変化を時間軸方向に表すものである。クラスまたはインスタンスに対するメッセージの入力、処理、同インスタンスまたは同クラスからのメッセージの出力を1つの処理単位（ユニット）として処理手順を評価する。CircuitAのlock状態は7ユニット続いており、このうちbusy状態が4ユニット、waiting状態が3ユニットとなっている。同様にSubordinate Circuit生成機構の方法1と方法2を図8、図9に示す。

## 5. 2 平行処理特性の比較

AgentがCircuitAとクラスCIRCUITにそれぞれSubordinate Circuit生成メッセージとCircuitB、CircuitC、Equipmentを構成要素とするComposite Circuitの生成メッセージを同時に送った場合を考える。図10にComposite Circuitの生成機構の方法1、方法2についてインスタンス生成完了までに必要なユニット数と各オブジェクトのlock状態を示す。両方法とも同一の16ユニットを必要としている。lock状態のうち、busy状態とwaiting状態にあるオブジェクトが処理可能なユニット数を比べると、クラスCIRCUITにおいて方法1ではそれぞれ7ユニットと3ユニット、方法2ではそれぞれ5ユニットと5ユニットとなっており、方法2は方法1に比べて応答待機に時間を費やしていることがわかる。尚、実際の処理系では各ユ

ニットの処理特性が異なることを考慮すると上記ユニット数比較は処理手順の定性的特徴を表しているものであるといえる。

## 5. 3 Smalltalk-80環境の拡張による処理特性の改善

Smalltalk-80環境下では、インスタンス生成時にインスタンス生成を管理するオブジェクトは必ずlock状態となる。Composite Circuit生成においてはその構成要素が増えるにしたがってlock状態となる機会が増え、多くのメッセージがオブジェクト上で受け付け待ちの状態となる。インスタンス生成に要する時間を短縮し、さらに多くのCircuitインスタンスを効率よく生成するためには、lock状態のうち"waiting"に費やしている時間を削減しなければならない。

メッセージの処理に関して以下の拡張を導入すると上記の時間削減が達成できる。

- ・メソッドの実行途中で他のオブジェクトにメッセージを送った時には、新しいメッセージを受け付けることができる。
- ・メッセージに優先度を付与できる。例えば、他のオブジェクトに送ったメッセージに対する応答と新しいメッセージとでは、応答が高い優先度を持つ。
- ・一つのオブジェクトは複数のオブジェクトに対して応答を待たずに連続してメッセージを送ることができる。

図11はメッセージ処理の拡張環境下でのSubordinate

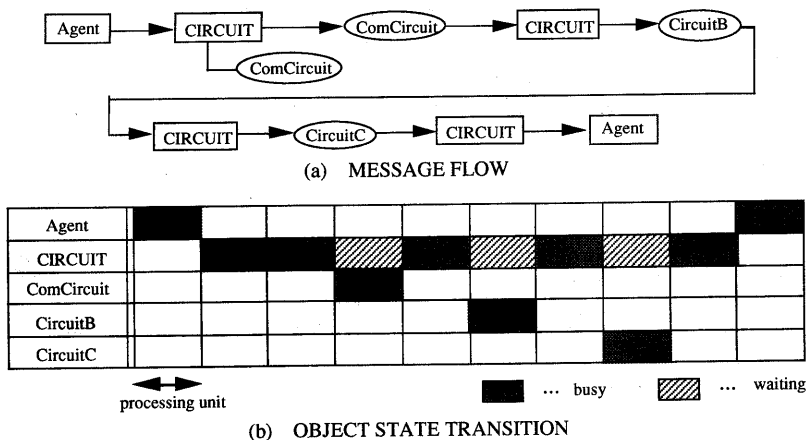


FIGURE 8  
COMPOSITE CIRCUIT CREATION FLOW (TYPE 1)  
図8 Composite Circuitの生成過程 (方法1)

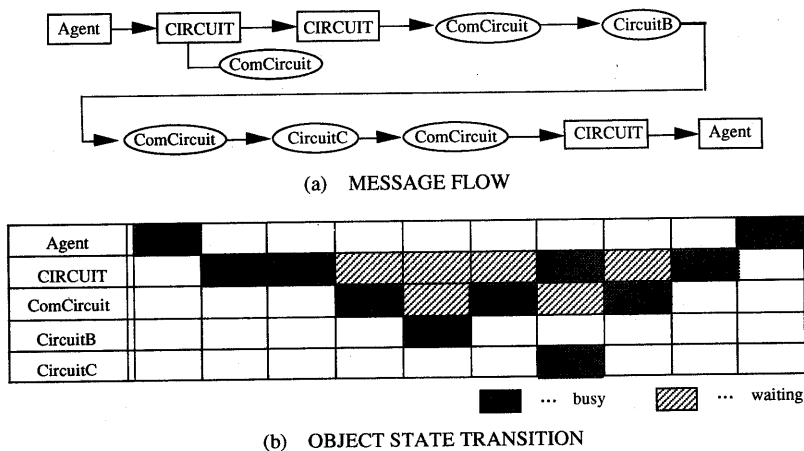


FIGURE 9  
COMPOSITE CIRCUIT CREATION FLOW (TYPE 2)  
図9 Composite Circuitの生成過程 (方法2)

CircuitとComposite Circuitの同時生成過程の各オブジェクトの状態の推移を示す。Circuitインスタンス生成完了に9ユニットを要している。表1は複数のインスタンス同時生成に必要なユニット数を、メッセージ処理機構の拡張前と拡張後について比較したものである。拡張後はユニット数が約半分となり、インスタンス生成が効率化できることがわかる。またComposite Circuitの生成機構についても、生成数を増加させると方法2では、クラスCIRCUIITにおけるunlock状態を他のインスタンス生成により多く利用できることから、方法1と比べて効率がよ

いことがわかる。

## 6 あとがき

伝送網管理システムに適用するオペレーションインタフェースのオブジェクト指向技術による設計に合わせ、システム内部の処理機構にオブジェクト指向技術を適用する上での技術課題を示し、伝送網構成管理において中心となるCircuitインスタンスの生成機構について処理効率を向上させる観点から比較評価を行った。

伝送網構成管理のCircuitインスタンス生成において平



- [9] 高田、米澤：“並列オブジェクト指向言語の分散環境における実現” コンピュータソフトウェア, Vol.6, No.1, pp.17-29, 1989.
- [10] 藤井星野、山口：“デジタルバスネットワークの制御” 昭和62年電子情報通信学会システム部門 全大288
- [11] OSI/NM Forum Object Library Issue 1.1 1990/6 OSI/NM Forum.
- [12] Application Service Document Issue 1 1990/6 OSI/NM Forum.

表1 インスタンス生成に必要なユニット数

TABLE 1  
THE SUM OF UNIT IN THE CASE OF MULTIPLE  
CIRCUIT INSTANCE CREATION

| CONDITIONS<br>EXAMPLES                      | NORMAL | EXPANDED |
|---------------------------------------------|--------|----------|
| TWO SUBORDINATE<br>CIRCUITS CREATION        | 1 6    | 1 1      |
| TWO COMPOSITE CIRCUITS<br>CREATION BY TYPE1 | 1 8    | 8        |
| TWO COMPOSITE CIRCUITS<br>CREATION BY TYPE2 | 1 8    | 1 0      |

unlock lock (busy) C...Composite Circuit S...Subordinate Circuit

|            |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |
|------------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|
| Agent      | C | S |   |   |   |   | C |   | S |  |  |  |  |  |  |  |  |  |  |
| CIRCUIT    |   |   | C | C | S | S | C |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitA   |   |   | S |   |   |   | S | S | S |  |  |  |  |  |  |  |  |  |  |
| SubCircuit |   |   |   |   |   |   |   | S |   |  |  |  |  |  |  |  |  |  |  |
| ComCircuit |   |   |   |   | C |   |   |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitB   |   |   |   |   | C |   |   |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitC   |   |   |   |   | C |   |   |   |   |  |  |  |  |  |  |  |  |  |  |

(a) TYPE1

|            |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |
|------------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|
| Agent      | C | S |   |   |   |   |   | C | S |  |  |  |  |  |  |  |  |  |  |
| CIRCUIT    |   |   | C | C | S | S | C |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitA   |   |   | S |   |   |   | S | S | S |  |  |  |  |  |  |  |  |  |  |
| SubCircuit |   |   |   |   |   |   |   | S |   |  |  |  |  |  |  |  |  |  |  |
| ComCircuit |   |   |   |   | C |   | C |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitB   |   |   |   |   | C |   |   |   |   |  |  |  |  |  |  |  |  |  |  |
| CircuitC   |   |   |   |   | C |   |   |   |   |  |  |  |  |  |  |  |  |  |  |

(b) TYPE2

FIGURE 11  
COMPARISON OF OBJECT STATE TRANSITION FOR  
MULTIPLE OBJECT INSTANCE CREATION

図11 Subordinate CircuitとComposite Circuitの同時生成時における  
Composite Circuit生成方法の違いによる各オブジェクトの状態遷移図