

階層化を考慮したオブジェクト通信図

植野直樹

横河・ヒューレット・パカード株式会社

階層化したオブジェクト通信図と、ユーザーがそれらを用いてシステムモデルを容易に理解するためのガイド機構について述べる。従来の通信図では、視点や階層構造という概念はほとんど考慮されていなかった。このため、大規模で入り組んだシステムでは、その表示が複雑となり、ユーザーは必要な情報を容易に得られなかった。そこで、彼らの視点から彼らの求める抽象度で情報が得られるように、オブジェクトの2種類のスコープ(直接メッセージ送信可能な世界)を通信図の表示単位とし、それらをインタラクティブに切り替えられるようにした。本稿では、そのスコープの定義および機能実現に関するガイド機構の構造・振る舞いについて述べる。

A Study on a Hierarchical Object Interaction Diagram

Naoki Ueno

Yokogawa-Hewlett-Packard, Ltd

1-3-2, Murotani, Nishi-ku, Kobe-Shi, Hyogo, 651-22 Japan

This paper describes hierarchical object interaction diagrams and the mechanism of the guide which helps users understand a system model easily by using them. In most of traditional diagrams, the concepts of viewpoint and hierarchical structure weren't taken much account of. So, in the case of an intricated system, the expression was too complicated for users to understand. Here, we define two kinds of object scopes (= the space where messages can be sent directly by the object) as unit of expression. Users can get needed information at the desired abstract level from the desired viewpoint by changing scopes interactively.

1. はじめに

最近、システム開発において、モデル化が自然に行え、そのモデルを分析から設計、実装、保守に至るまで一貫して使用できるという理由から、オブジェクト指向の開発方法論およびそのCASEツールが盛んに提案され、導入されるようになった。

一般にCASEツールは、システム設計者がシステムモデルを容易に構築するための支援ツールとしての側面と、システムの開発・運用に携わるすべての人がシステムに対して共通の理解を得るための支援ツールとしての側面の(少なくとも)2つの側面をもっている。ここでは後者の側面に着目する。

システムの開発・運用に関与する人は、クライアント(顧客)からプロジェクトリーダー、システム開発者、保守要員などさまざまである。彼らのシステムに対する見方、考え方は異なっている。たとえば、クライアントはシステムの細かな実装は知りたくないと思うだろうし、システムのある一部を開発・保守する人は、各自の担当分野以外には興味がないと思うかもしれない。そのようなさまざまな人々の要求を満たすために、彼らの視点でかつ彼らの求める抽象度で必要な情報が得られるように、彼らを適切にガイドしてくれる機構が望まれている。ここでは、上記ガイド機構のうち、機能実現に関するものに着目する。

われわれはこの機能実現の表記法として、オブジェクト間のメッセージ伝達(message passing)によって、それを表記したオブジェクト通信図(Object Interaction Diagram、方法論によってはそれをイベントトレース図⁴⁾ともシーケンス図²⁾ともよばれている。以後、それらを総称して通信図とよぶ)を採用している。ところが、従来の通信図では、視点という概念がなく、階層化に対する考慮もほとんどされていなかった。このため、大規模で幾重にも階層化されたシステムでは、ときにはその表示がたいへん複雑になり、ユーザーは必要な情報を容易に得ることができなかった。

そこで、われわれはある規則に則って通信図を階層化し、ユーザーを適切にガイドする機構を考

案した。その特徴は、2種類のオブジェクトのスコープを定義し、各オブジェクトのスコープを通信図の表示単位として、それをユーザーがインタラクティブに切り替えることにより、ユーザーは要求する情報を容易に得ることができる点にある。

以下では、まず現状の通信図における問題点を挙げ、次に通信図の表示単位となるオブジェクトの2種類のスコープを定義する。さいごに提案したガイド機構の構成やその特徴について例題を交えながら述べる。

2. 現状の通信図における問題点

現在、通信図はいろいろな方法論、たとえば、Objectory法³⁾やOMT法¹⁾などで使用されている。それらを一般化したものを図1に示す。それは、縦方向に時間軸を横方向にオブジェクト群を設定し、あるオブジェクトから発せられるメッセージが他のオブジェクトへどのように伝搬されていくかを示しており、以下のような事象を表現したものである。ある外部メッセージを受けて、オブジェクトAはオブジェクトBへメッセージを送り、それを受けてオブジェクトBはオブジェクトCへメッセージを送る。そしてそのメッセージを受けたオブジェクトCはある処理を行ってオブジェクトBへ結果のメッセージを返し、それを受けてオブジェクトBはオブジェクトAへ結果のメッセージを返す。最後にオブジェクトAは結果を外部へ返す。

一般に、メッセージとともにいくつかのオブジェクトも伝達され得るが、ここでは話を簡単にするために、そのようなオブジェクトを無視することにする。

現在のところ、このような通信図では以下のような問題点が存在する。

視点や階層化に対する考慮がされていないため、オブジェクト数や通信回数が増えたり、オブジェクトの構造が幾重にも階層化されている場合、その表記が繁雑になって、ユーザーが必要とする情報を迅速に見つけ出すことが困難になる場合がある。たとえば、図1において、ユーザーがオブジェクトAの振る舞いに着目する場合、その関心はオブジェクト

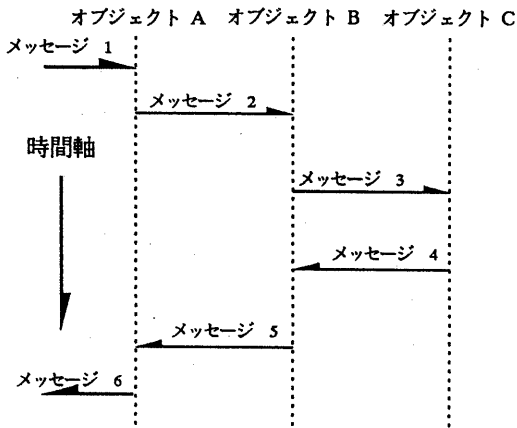


図1: 従来の発信図の一例

Bだけあって、オブジェクトCにはない。ところが、従来の発信図ではすべてのオブジェクトが見えてしまっている。これは現状の発信図に視点という概念がないために生じている。

われわれはこのような問題を解決するために、3章で述べる2種類のオブジェクトのスコープを定義し、それらを発信図に利用することにした。

3. オブジェクトのスコープ

あるオブジェクトから見える世界をスコープとよぶ。ここで見えるとは、メッセージを送ること(通信)ができることを意味する。図2はスコープの外側からメッセージが送られてきたときに、その要求を満たすために、外部もしくは内部の(内包する)オブジェクトへメッセージを送る様子を表わしたものである。各オブジェクトと各メッセージにはおのおの名前が表示されている。

この図では、上記の発信図とは異なり、表記平面から時間軸を取り去り、その平面すべてをオブジェクト空間に利用することにした。これにより、メッセージ伝達の並列性(同報通信など)を容易に表現することができる。いっぽう、時間軸のかわりに、各メッセージにシーケンシャルな番号が付加

されており、それにより、メッセージの発生順序を知ることができる。いくつかの方法論⁴⁾においても時間軸を取り去った発信図は使用されている。

図2からわかる通り、スコープには水平スコープと垂直スコープの2種類ある。以下では、メッセージと両スコープの定義を行う。

3.1 メッセージ

メッセージとは、他のオブジェクトへサービスを依頼するために送る要求のことである。

メッセージには、返信を必要とするもの(返信付メッセージとよぶ)と、しないもの(片道メッセージとよぶ)がある。さらに返信付メッセージには、返事の待ちに関して同期と非同期の2種類がある。ここでは、非同期の返信付メッセージについては考えないことにする。返信付メッセージの往信/返信部分をそれぞれ要求/結果メッセージとよぶことにする。

3.2 水平スコープ

水平スコープとは、その視点となるオブジェクトから外側に向いたときに見える世界のことをいう。そこでは、その視点となる核オブジェクト(太線で

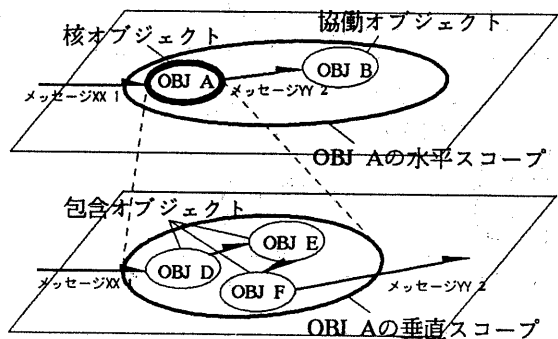


図2: オブジェクトのスコープ

表示)、それと直接通信関係のある外側の独立したオブジェクト(collaborators、以後それを協働オブジェクトとよぶ)と、それらの間のメッセージが表示される。一般に核オブジェクトと協働オブジェクトの関係はクライアント/サーバの関係(client/server relationship) とよばれている。

核オブジェクトに対する要求メッセージは常にスコープの外から入ってくる。それはまず核オブジェクトに伝達される。核オブジェクトは、独力でその要求を処理できない場合、協働オブジェクトにメッセージを送って処理を肩代りしてもらう。要求メッセージに対する結果メッセージはスコープの外へ放出される。

核オブジェクトに対する入力メッセージと出力メッセージの基本関係パターンを図3に示す。ここでいう外部/内部入力とは、メッセージがスコープの外部/内部から核オブジェクトへ入ることをいう。いっぽう外部/内部出力とは、メッセージが核オブジェクトからスコープの外部/内部へ出ることをいう。それらは6種類ある。水平スコープでは、これらのパターンを組み合わせたものが表示される。

これらの基本パターンを組み合わせる上でいく

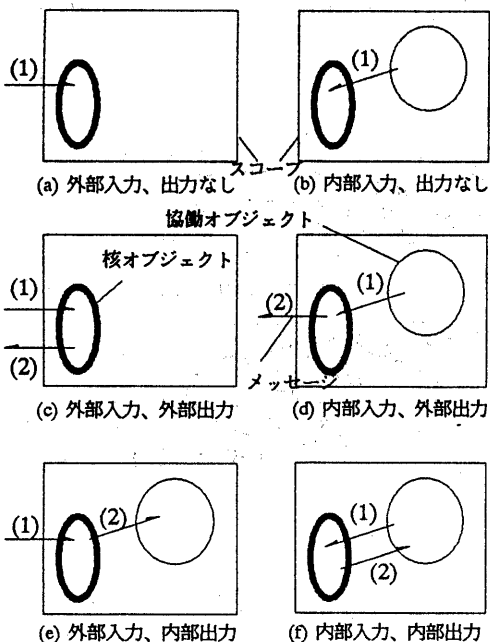


図 3: 水平スコープの基本パターン

つか制限があるが、ここでは触れない。

水平スコープで注意すべきことは2点ある。1つは、協働オブジェクト間でやりとりされるメッセージは水平スコープでは表示されない点である。あくまで、核オブジェクトと協働オブジェクト間でやりとりされるメッセージのみが表示される。もう1つは、核オブジェクトに対する要求メッセージの送信元も水平スコープでは表示されない点である。これは、メッセージに対する核オブジェクトのアクションは送信元のオブジェクトの種類に依存しないからである。

3.3 垂直スコープ

垂直スコープとは、その視点となるオブジェクトから内側に向いたときに見える世界のことをいう。そこでは、視点となるオブジェクトが内包するオブジェクト(contained objects、以後それを包含オブジェクトとよぶ)とそれらの間のメッセージが表示される。一般に視点となるオブジェクトと包含オブジェクトの関係は 集積関係(aggregation relationship) とよばれている。

視点となるオブジェクトに対する要求メッセー

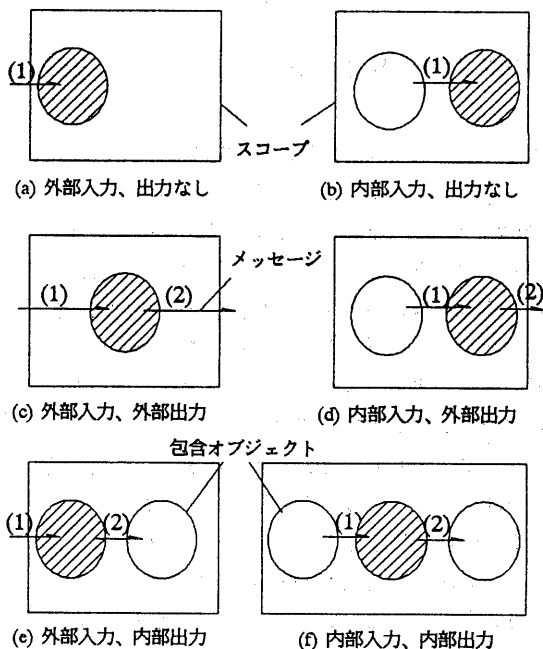


図 4: 垂直スコープの基本パターン

ジは常にスコープの外から入ってくる。それは1つもしくは複数の包含オブジェクトへ伝達される。

1つの包含オブジェクトに対する入力メッセージと出力メッセージの基本関係パターンを図4に示す。ここでいう外部/内部入力とは、メッセージがスコープの外部/内部から包含オブジェクトへ入ることをいう。いっぽう外部/内部出力とは、メッセージが包含オブジェクトからスコープの外部/内部へ出ることを行う。それらは6種類ある。垂直スコープでは、これらのパターンを組み合わせたものが表示される。水平スコープ同様、組み合わせる上でいくつか制限があるが、ここでは述べない。

4. 機能実現を理解するためのガイド機構

ここでは、3章で述べた2種類のスコープを交信図として用いて、ユーザーの必要とする機能的実現情報を容易に導き出すガイド機構について述べる。

このガイド機構は、システムに対してあるイベントが発生したとき、システムの内部でどのようなアクションが生じているかをユーザーに示すものである。

以下では、最初にユーザーがこのガイド機構に対して入力可能なコマンドの説明を行う。次にガイド機構の構造ならびに上記コマンドを与えたときの振る舞いについて述べる。

4.1 コマンド

ユーザーは以下の4つのコマンドを入力することができる。

4.1.1 メッセージ選択コマンド

メッセージ選択コマンドによって、ユーザーはトレースしたいメッセージを選択することができる。(ただし、一度に選択できるメッセージは1つに限定する)

4.1.2 視点変更コマンド

視点変更コマンドによって、ユーザーはあるオブジェクトから別のオブジェクトへ視点を移動したり、視点の方向を変えたりすることができる。このコマンドを与えたとき、選択したメッセージ

の送信先が核オブジェクトであれば、そのオブジェクトの垂直スコープが、そうでなければ水平スコープが表示される。

4.1.3 時間経過コマンド

時間経過コマンドによって、ユーザーは選択メッセージによって引き起こされる新しいメッセージやオブジェクトを見ることができる。そのコマンドは2種類ある。1つは現在表示しているスコープにおいて、変化が生じるまで時間を経過させる、ステップ時間経過コマンドと、もう1つは、一連の処理が完了するまで時間を経過させる、一括時間経過コマンドとがある。

4.1.4 履歴表示コマンド

履歴表示コマンドによって、ユーザーはメッセージやオブジェクトの履歴を残した状態で、メッセージ伝達のトレースを行うことができる。

4.2 構造・振る舞い

上記ガイド機構モデルの構造と振る舞いを図5に示す。図5からわかるとおり、ガイド機構は、入力部、オブジェクト計算部、メッセージ計算部、統括部および出力部から構成されている。以下で各部の説明を行う。

4.2.1 入力部

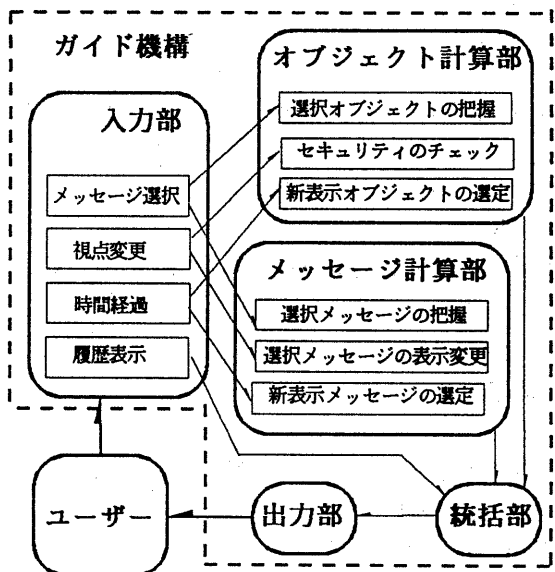


図5: ガイド機構の構造・振る舞い

入力部はユーザーから4種類のコマンド(メッセージ選択、視点変更、時間経過、履歴表示)を受け付け、所定の場所へメッセージを配送する。

4.2.2 オブジェクト計算部

オブジェクトに関する計算を行う。

視点変更コマンドを受けたとき、新しく視点に移るオブジェクトが原子オブジェクト(もうこれ以上分解できないオブジェクト)やセキュリティのかかったオブジェクトでないかをチェックする。もし、それが原子オブジェクトであったり、ユーザーに情報開示できないものであれば、警告表示を出す。

時間経過コマンドを受けたときは、新しくスコープ内に現われるオブジェクトの選択を行う。

4.2.3 メッセージ計算部

メッセージに関する計算を行う。

時間経過コマンドを受けたとき、その対象メッセージが受け付け可能な状態にあるかをチェックしたり、それが結果メッセージを返すかどうかをチェックしたりする。また、新しく発生するメッセージの選択などを行う。もし、結果メッセージを返さない場合は、警告表示を出す。

視点変更コマンドを受けたときは、その対象メッセージが新しいスコープ内でどのように表示されるかを計算する。

4.2.4 統括部

オブジェクト計算部で得られたオブジェクトやメッセージ計算部で得られたメッセージなどを総合して、ユーザーに表示するためのスコープ情報を生成する。履歴モードの管理はここで行う。

4.2.5 出力部

出力部は統括部で得られたスコープ情報をユーザーへ表示する。

5. 例題

具体的に例題(図1)をもとに提案したガイド装置の特徴を説明する。

システムは図1の3つのオブジェクトから構成されているものとし、メッセージ1をシステムに対する

操作(system operation、以後それをシステム操作とよぶ)とみなす。ここでは、ユーザーがこのガイド装置を使って、そのシステム操作に対するシステムの挙動を調べる場合について述べる。

初期時には、図6で示すようなシステムの水平ス

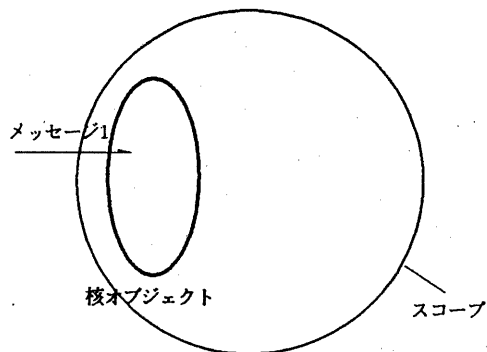


図6: 初期時のシステムの水平スコープ

コープが表示される。ここではシステム操作であるメッセージ1のみが現われている。(あらかじめメッセージ1は選択状態にあり、メッセージ通信の履歴が残るようにセットしているものとする。)

ユーザーがクライアントの場合のように、システムの詳細に興味がなく、システム操作に対する結果(たとえば結果メッセージに付随するオブジェクトの種類など)のみを知りたい場合、そのまま時間経過コマンドを与える。すると、結果メッセージ(メッセージ6)が現われる。(図7参照) これによりユーザーはシステムの行う複雑な処理過程を見なくて済む。

いっぽう、システム内の挙動に興味がある場合、

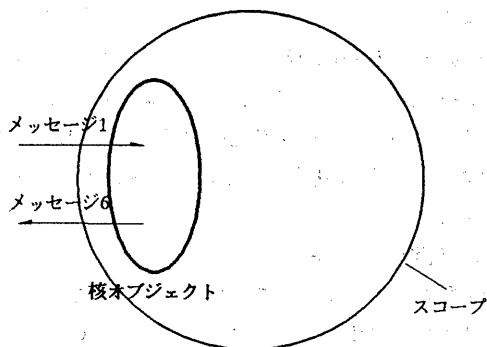


図7: 一括時間経過後のシステムの水平スコープ

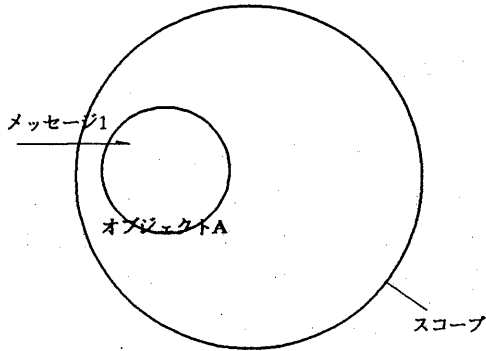


図8: 初期時のシステムの垂直スコープ

図6において視点変更コマンドを与える。この結果、図8で示すようなシステムの垂直スコープが現われる。そこではシステム操作のメッセージが直接送信されるオブジェクトAが現われる。

ユーザーがプロジェクトリーダーの場合のように、システム操作によって引き起こされるシステムの挙動を広く把握したい場合、図8において一括時間経過コマンドを与える。その結果を図9に示す。そこでは3つのオブジェクトならびにそれらの間でやりとりされるメッセージすべてが現われる。これによりユーザーは全体の動きを一目で知ることができる。

ユーザーがシステム開発者や保守要員の場合のように、システムのある一部分の挙動のみを把握したい場合、たとえば図8においてオブジェクトAの挙動に興味がある場合、視点変更コマンドを与えて

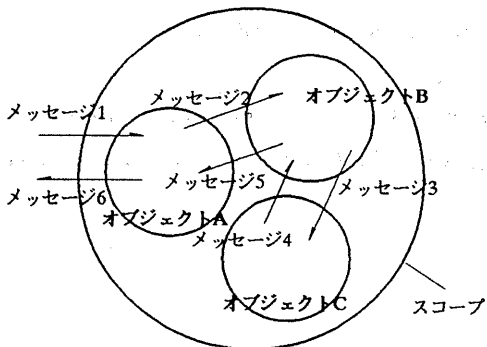


図9: 一括時間経過後のオブジェクトAの垂直スコープ

視点をオブジェクトAに移す。その結果、オブジェクトAの水平スコープが現われる。(図10参照)

図10において、メッセージ1に対するオブジェクトAの振る舞いを1ステップごと確認したい場合、ステップ時間経過コマンドを与える。いっぽうその振る舞いを一度に見たい場合、一括時間経過コマンドを与える。後者のコマンドを与えた結果を図11に示す。その図では直接関係のないオブジェクトCは現われない。これにより、ユーザーは不必要な情報

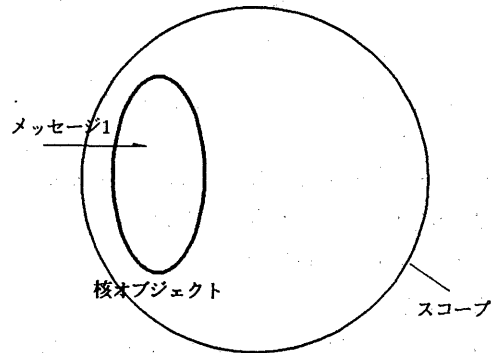


図10: 視点変更後のオブジェクトAの水平スコープ

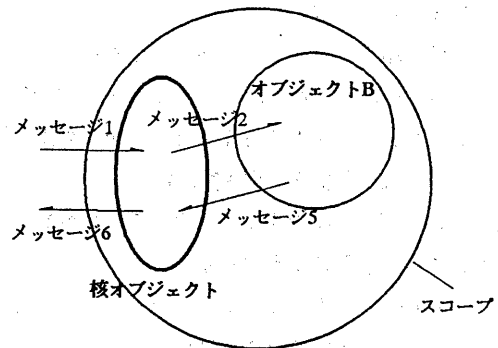


図11: 一括時間経過後のオブジェクトAの水平スコープ

を見なくて済み、本質的な情報のみを容易に把握することができる。

また図10において、メッセージ1を受けたときのオブジェクトAの内部の挙動について知りたい場合、視点変更コマンドを与える。その結果、オブジェクトAの内部を見ることができる。いまオブジェクトAがオブジェクトD、オブジェクトEとオブジェクトFから構成され、その

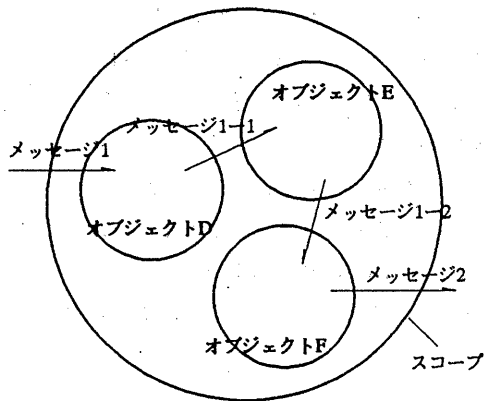


図12: 一括時間経過後のオブジェクトAの垂直スコープ

順でメッセージが流れるとして、図10で視点変更と一括時間経過コマンドを与えると図12のようになる。オブジェクトFから送信されるメッセージはメッセージ2となる。

もしオブジェクトAが原子オブジェクトであったり、セキュリティ上ユーザーに情報開示できないものであれば、警告表示が現われて、その視点位置は図10のまま変わらない。

図11では、結果メッセージであるメッセージ6はスコープの外へ放出される。そのメッセージ6を選択して視点変更コマンドを与えるとシステムの垂直スコープへ視点が戻る。(図9参照)

最後にこのガイド機構を用いたときの有効性を再度まとめると以下ようになる。

(1) 各種ユーザー(クライアント、プロジェクトリーダー、システム開発者や保守要員など)は、視点変更コマンドと時間経過コマンドを適当に与えることにより、各自の視点から望みの抽象度でシステムの挙動を見ることができ、必要な情報を容易に得ることができる。しかも、そのコマンド入力の方法は直観的でわかりやすい。

(2) オブジェクトのセキュリティチェックにより、限定されたユーザーのみにシステムの情報を開示することができ、情報の機密性を保持することができる。

6. おわりに

ここでは以下のことがらを行った。

(1) 交信図を階層化するために、その表示単位となるオブジェクトの2種類のスコープを定義した。

(2) それらを用いた機能実現を理解するためのガイド機構の構造および振る舞いについて述べた。

(3) 例題を用いながら、その表記およびガイド機構の有効性について述べた。

今後は、メッセージ送信時のオブジェクト移送問題などについて考えるとともに、CRCカード向などを用いてそのガイド機構の実装を行い、評価する予定である。

参考文献

- [1] J.Rumbaugh, M.Blaha, W.Premmerlani, F.Eddy and W.Lorensen: *Object-Oriented Modeling and Design*, Prentice-Hall, 1991
- [2] 野口健一郎: ソフトウェアの論理的設計法, 共立出版, 1990
- [3] I.Jacobson: *Object-Oriented Software Engineering*: Addison Wesley, Reading, MA(USA), 1992
- [4] D.Coleman, P.Arnold, C.Dollin, H.Gilchrist, F.Hayes, P.Jeremaes: *Object-Oriented Development: The Fusion Method*, Prentice Hall, 1993 (発刊予定)
- [5] R.Wirfs-Brock, B.Wilkerson and L.Wiener: *Designing Object-Oriented Software*, Prentice-Hall, 1990