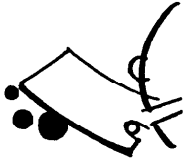


## 報告

## パネル討論会

## オブジェクト指向の展望

## 昭和63年度後期第37回全国大会†報告



## パネリスト

萩谷 昌己<sup>1)</sup>, 柴山 悦哉<sup>2)</sup>, 田中 克己<sup>3)</sup>  
 田中 譲<sup>4)</sup>, 安西祐一郎<sup>5)</sup>, 竹内 彰一<sup>6)</sup>  
 司会 所 真理雄<sup>7)</sup>

所(司会)「オブジェクト指向の展望」というのは、実はよく考えてみますと、いささかおかしな題目です。オブジェクト指向というのは形容詞ですが、

それを展望してしまうというわけです。このタイトルについて若干の説明を加えるならば、概念、パラダイム、理論、言語、オペレーティング・システム、データベース、ユーザ・インタフェース、人工知能、分散システム、グラフィックス、その他、ありとあらゆるアプリケーションがオブジェクト指向という枠組の中で考えられており、またこれからますます考えられるようになっていくのではないかと。そういう意味で、オブジェクト指向計算に関する展望というふうにご理解いただけたらよろしいかと思えます。

オブジェクト指向に関する定義というものはあるようでないわけでございますけれども、どなたもこれならば納得してくださるだろうというような定義としては、「オブジェクトとは自己完備で統一的な通信手段をもつ計算実体である」という程度かと思えます。これをベースにした計算がオブジェクト指向計算です。自己完備というのはオブジェクト内での計算を行うためのすべてが備わっているという意味です。それから統一的な通信手段というのは、どのようなオブジェクト間でも、通信の方法が一様で、だから、だれからだれへでも通信でき、送受信オブジェクト間で意味をもてばいい。こんな定義がなされています。

オブジェクト指向計算の特徴は「シェアリング・バイ・メッセージ」すなわちメッセージでもって情報を



シェアしながら計算を進めることにあります。情報システムというのは必ずデータをシェアしなければ話からはじまらないわけです。ひとり合点で自分だけのことをやっていたのではだめで、シェアするということの本質なんですが、メモリをアクセスするとかプロセスを実行するとかいうレベルではなくて、メッセージでもってシェアする。そのときお互いに名前を知っているだけでよいところが一番の基本ではないかと思っております。

僕の話はこれだけにしておきましょう。本日は輝々たるメンバーにパネリストとしてご登場いただきました。それぞれの方が、それぞれもうすでにオブジェクト指向をベースに実際のものを作っておられる方々です。6名の方々には、お二方ずつご発表いただいて、そこで他のパネリストの方々、もしくはフロアからの質問をお受けし、またその次の話題に移るといような形で話を進めていきたいと思います。

最初のセットはかなり全体的な概念に関する話で、京都大学の萩谷さんに、プログラミング・パラダイムの話、引き続きまして東京工業大学の柴山さんに、並行分散処理に関連したオブジェクト指向の話題でお話しいただきたいと思えます。

## オブジェクト指向の概念

萩谷 私は根が真面目なものですから、オブジェクト指向が広まるにつれて、オブジェクト指向の本質とは何かということを一生涯懸命考えていた記憶があ



るのですが、結局オブジェクト指向とは、「概念の闇鍋」ではないかという気がしているわけです。オブジェクト指向には抽象データ型とか、型の階層とか、並

†日時 昭和63年9月12日(月) 12:30~14:45

場所 立命館大学

1) 京大, 2) 東工大, 3) 神戸大, 4) 北大, 5) 慶大, 6) 三菱, 7) 慶大

列処理などのさまざまな概念が含まれています。そこで、これらを統一する概念がいったいあるのか、いろいろ考えたんですが、やはりないのではないかと、つまり、オブジェクト指向というのは、要するにこういういろんな概念が混じり合って、闇鍋のようになっているものではないかという気がしているわけです。

こう考えますと、他のプログラミング・パラダイムに比べて、オブジェクト指向は非常に一般的すぎるという気がします。そもそもオブジェクトとはモノということですから、これがモノだといっても全然情報がない、つまり、オブジェクト指向を利用したというだけでは全然情報量がないということです。だからオブジェクト指向という言葉は「アブストラクト・ナンセンス」の一つではないかと考えています。

最近パラダイムという言葉が叫ばれていますが、これは、コンピュータ・サイエンスの研究がまず、パラダイムが先にあって、次に、パラダイムをどのように応用するかを考える、という順序で行われているということです。私には、このような研究のし方が不健全に思えていて、本来はオペレーティング・システムやデータベースの研究が先にあって、各分野の中でどのようなパラダイムが有効なのかを考えるべきなのです。

私がやっていることを少し話しますと、GMW というウィンドウ・システムを作っているんですが、最近ではウィンドウ・システム上のアプリケーション構築をふくめた研究をしています。ウィンドウ・システム上のアプリケーションにはいろいろな特徴がありますが、重要なのは本体とユーザ・インタフェースを分離するという事です。これに並列オブジェクト指向の考えをあてはめると、本体とユーザ・インタフェースのそれぞれが並行オブジェクトというふうになります。そして、ウィンドウ・システムのイベントはオブジェクト間のメッセージというふうになります。並行オブジェクト指向では同期通信とか非同期通信、などいろいろな道具立てが用意されていますから、並列オブジェクト指向の枠組で、ウィンドウ・システムやその下のアプリケーションを組んでいくと非常にうまくいくわけです。

それで輝けるオブジェクト指向と書いてあるのです。しかし、ちょっと頭を冷やして考えてみると、結局みんなオブジェクトだというのは、要するにオブジェクトがごちゃごちゃとあって、それらがメッセージで結ばれていて、このへんにアプリケーションがある

ということです。しかし、これだけでは結局何も言っていないんじゃないかという気がします。

オブジェクト指向とは別に、昔からユーザ・インタフェースの分野では、ユーザ・インタフェースの機能を分析し、たとえば、ユーザ・インタフェースをプレゼンテーションとダイアログに分けるといったことが、いろいろと研究されています。結局、オブジェクト指向は第一ステップに過ぎず、これを使って旧来の概念をどう考え発展させるかということが大事だと最近では思っているわけです。

結論ですが、「思えば、二十代はパラダイムに踊らされていた。これからは本物をめざしたい」となります。つまり、ちゃんとアプリケーションを眺んでプログラミングのパラダイムとかというものを考えていくべきではないかというふうに考えています。

司会 どうもありがとうございました。では引き続きまして、柴山さんお願いいたします。

柴山 私に与えられた題目は並行分散、並列、とオブジェクト指向ということでございます。さて、並行オブジェクト指向あるいは分散オブジェクト指向とは何かと申しますと、闇鍋の中からの、自分に都合のいいものだけ取って食べちゃう話になるんです。その意味では、闇ではない鍋です。まず、カプセル化とかブラックボックス化という一般によく使う機能を、並列オブジェクト指向でもいただいてしまおう。それからポリモルフィズムは、性質のいい、シェアリングのメカニズムなので、これもいただいてしまおう。あとほかに並列問題を記述しようとする、どうしても同期のメカニズムがいりますから、これをよそから取ってきて入れてしまおう。さて、同期のメカニズムというのは、並列に計算してほしいけど、でたらめに計算してもらっては困る。適当な順番をつけて、並行計算の一部を逐次化するようなメカニズムの事です。

カプセル化・ブラックボックス化の機能は、違法なアクセスを排除し、安全性を保証するために使う。あとポリモルフィズムは、トランスペアレンシを保証するために使える。場合によってはインヘリタンスも、このために使える。

それで、私が思い描いているオブジェクトをかけと言われると以下ようになります。要するに二つのオブジェクトがあったとするならば、その間をメッセー



ジを使って交信する。しかも、メッセージしか使わずに交信する。オブジェクトには外界とのインタフェースの部分があって、ここが入出力を行うわけです。メッセージの入口になるインタフェース部は、メッセージを一般には実行時に自分で解釈しまして、ダイナミックに自分の中にもっている適当なプロシーダをコールする。そのプロシーダは内部のデータを書き換えたり、メッセージを送ったり、あるいはオブジェクトを作ったりする。インタフェース部にどんなものをつっ込むかによってオブジェクトの性質が全然違うものになる。データ抽象なら何もないに近いし、ここでメソッド・サーチをやってみるとか、あるいはケイパビリティの管理をやってみるとかすると、だんだんオブジェクトらしくなってくる。

さて、カプセル化の仕掛けというのは何がいいかというと、普通は内部のデータが内部のプロシーダからしかアクセスできないという保証がえられることです。私どもがやっております流儀では、このオブジェクトの内部は実質的に逐次的に処理が行われる。これによってオブジェクト内のデータが二つ以上のプロセスから同時にアクセスされることはない。こういう方向を考えておるわけです。

あまり時間がありませんので、いきなり並列並行分散のオブジェクト指向の展望の話をしませう。オブジェクト指向は閻魔というよりブラックホールになってしまて、まわりに使えそうな直交する概念があると、全部取り込んじゃう。逐次オブジェクト指向の場合ですと論理型のプログラミングであるとか、あるいはリスプであるとか、制約であるとかを全部取り込むことができる。並行オブジェクト指向が概念として発展するためには、この並行オブジェクト指向と直交するパラダイムが欲しいんです。ところが、この直交するパラダイムが逐次型オブジェクト指向の場合ほど簡単には見つからない。たとえば並列論理型を統合してうまくいくかということ、何か違うような気がしております。

オブジェクト指向は元来、いろんな分野でそれぞれかなり独立に、いろんな概念として出てきたものです。それが抽象的なレベルで考えてみるとみんな実は同じことだったのでオブジェクト指向と呼ばれています。それで、僕の将来展望として、今後はドメインデペンデントに分化してしまうのではなからうかと思っております。オブジェクト指向という概念は残るかもしれないけれども、そこから新しいパラダイムがどんどん分

化して進んでいくのではないかと。以上でございます。

司会 どうもありがとうございました。パネリストの方々からご意見がおありになりますでしょうか。

竹内 アブストラクト・センスという話はよく分かります。人に何の研究をしているのかと聞いたときに、オブジェクト指向の研究をしていますという答を聞いたりすると、やはり情報が乏しい。オブジェクト指向の研究をしているといわれると、何か方法論を述べているだけで、研究テーマを述べたことにならないという気がします。オブジェクト指向自身が流行みみたいになって割とそういう話を通るようになっていくということは疑問に思っています。



そこで一つ伺いたいのは、方法論より一歩踏み込んでオブジェクト指向じゃないとできない応用がないのだろうか、ということですが。基本的にはオブジェクト指向じゃないとできない応用なんていうのはないと思うんですけども、オブジェクト指向というのは、やっぱり一種のフィロソフィみたいなものがあるわけで、そこから出てくる新しい応用はないんでしょうか。

萩谷 非常に難しい質問なんですけど、僕が今まで勉強してきた感じだと、既存のシステムがあって、それをオブジェクト指向的に解釈すると、大変うまくいく。したがって、どうもオブジェクト指向はあとからきているという感じが強くあります。オブジェクト指向的にやってみようというのは、僕なんかよりもっと他の分野の方からいろいろ出てくるんじゃないかと思うんですけども、どうでしょう。

司会 ありがとうございます。

今お話しいただいたパネリストの方々、お二方とも、頭の中にあるのは並行オブジェクト指向にかなり近い概念をおもちだと思います。世の中でオブジェクト指向といっているオブジェクト指向人口を考えてみますと、ほとんどの方は逐次オブジェクト指向のことを考えておられるかもしれない。さて、逐次オブジェクト指向のほうはつきつめて考えてみるとアブストラクト・データ・タイプと何が違うのかよく分からん。ところが、並行オブジェクト指向になると、結構違いというものがあるに気が分る。もしくは実現上かなり大きく異なるなというのが分る。というのは私自身の経験から出ているところです。萩谷さん、柴山さんがおっしゃっておいでのことが逐次なのか並

行なのか、一言もいただければと思うのですが。

**萩谷** 私もやはり並行のことを念頭に置いていたんですが、ただ並行にしても従来からモニタとかそういう並行オブジェクト指向に近い概念があったと思うんです。そういうものと並行オブジェクト指向がどう違うのか、私はよく分からないんですけども、そのへんは所先生はどうお考えなんでしょうか。

**司会** その答はあとでしようと思いますが、柴山さんが答えてくれるかもしれません。

**柴山** そうですね、モニタはともかくとして、プロセスだと概念に結局近いものになってしまうと思うのですが。

それから、最初の質問は並行型について話していたか逐次型について話していたかということなんですが、大部分は並行分散オブジェクト指向について話しています。そういう題名をいただいていますので。

**司会** それでは萩谷さんの質問に、僕なりのお答えしようと思います。モニタというのは実は逐次型のプログラミング・パラダイムにおける排他の一つの方法だと思っております。それは何かというと、呼ぶ人の処理装置がモニタの実行をする。基本的な考えはそうだと思うんです。同様に、リモート・プロシージャ・コールも自分のほうで実行している処理装置が向こうへ行って何かの仕事をやる。たまたま計算機がネットワークでつながれているから、向こうへ行って仕事ができないから、向こうにある計算機にやらせるんだけど、やっぱりプロセスもしくは仮想的なプロセッサがあっちまで行って仕事して、また戻って来て自分の実行を続けるわけです。ですからリモート・プロシージャ・コールだとか、モニタを並行に走らせようとしたとたんに、フォークだとかパービギンだとか、そういう概念がもう1個絶対に必要になってくる。ところが並行オブジェクト指向というのは、一番の原理としてはメッセージをとばすわけだから、仕事を頼むよと言ったきり、本人はずっとここにいるわけで、そこらへんが一番大きな違いではないかと、僕は思っている。シングル・プロセッサ上を実現しているかぎり、逐次オブジェクト指向と並行オブジェクト指向とはそれほど違わないようにみえるんですが、分散環境で実行しようとしたとたんに、本質的に違ってくるんじゃないかなと、僕はそういうふうには思っています。

**田中(藏)** あとで OHP を用意してあるものですが、それでお話したいと思うんですが、どうもオブジェクト指向といったときに、オブジェクト指向パラ

ダイムというものとオブジェクト指向言語というものと、この二つのもの間に少し差があるのではないかなというふうに以前から考えています。

それで、今お二人の先生方の話を聞いたんですが、やはりそれはオブジェクト指向の機構面の話であってオブジェクト指向のフィロソフィの話ではないような気がするんですね。これについては、また後で申し上げます。

**司会** ありがとうございます。安西先生、何かございますか。

**安西** 私はいわゆるオブジェクト指向のユーザとしての立場でここに出ているんだと思います。ユーザとしての立場からみると、オブジェクト指向の言語、特に並行オブジェクト指向言語に関しましては、やはりきちっとした言語を作っていただきたいと思えます。また、システムと言語の関係がどうなっているかということ、最近のオブジェクト指向アーキテクチャなんかの話ともからめて、本当はもうちょっと聞ければよかったかなというふうに、ユーザの立場から思うわけです。

**司会** どうも大変ありがとうございました。

3 番目に、神戸大学の田中克己先生にデータベースに関連しまして、オブジェクト指向の展望をお願いしたいと思います。

### オブジェクト指向とデータベース

**田中(克)** 柴山さんのお話しの中で、オブジェクト指向という概念があって、そこから OS なら OS、言語なら言語、データベースならデータベースと、そのおのおののドメインに応じたオブジェクト指向というような形に分化していきだろうとおっしゃられたと思うんですが、私はむしろ逆の絵をこの OHP にかいております。ここでは、データベース、プログラミング言語、知識ベースの分野がオブジェクト指向の下に統合されるということを描いております。私が、まず言いたいことは、オブジェクト指向というのは、プログラミング言語、データベース、それから AI の研究者が並んで議論しなければいけない重要なトピック



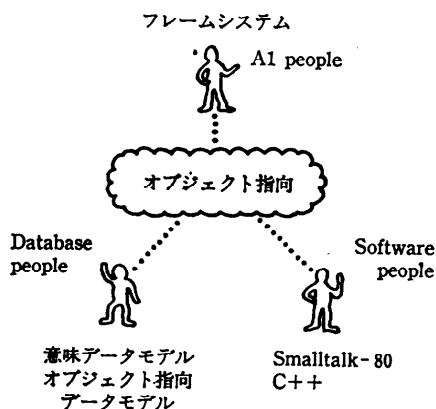


図-1 データベース・知識ベース・プログラミング言語の融合

だという認識をもっています。オブジェクト指向はたしかに闇鍋なんでしょうけれども、そういう意味では非常に魅力のある闇鍋ではないかというふうに考えます。

データベースの立場から言いますと、オブジェクト指向データベースというものが、ここ 2~3 年非常に活発に研究開発がされています。ここでは、データベースとプログラミング言語の統合を考え、次世代データベースを作る上で、ソフトウェア分野との融合が大変重要だと思っています。オブジェクト指向データベースが本当に次世代データベースかどうかは分かりませんが、そうだと思って現在研究活動をやっている人が国内外にたくさんいることは事実です。

それからデータベースの分野にはオブジェクト指向にかなり近いものが昔からあったということは、意外と知られておりません。データのモデリングという分野で、セマンティック・データモデルがあります。ここではほぼ 10 年くらい前からやられてきたことが、非常にオブジェクト指向に近い。

それから、これは後でも少しお話しますが、闇鍋の中には相当恐いものも入っておりまして、従来のデータベースの世界で金科玉条であったものに対してかなり挑戦的なことがらがあります。そういう意味でデータベースの分野自身の見直しも必要であろうと思います。私の究極の理想はオブジェクト指向を中心に、データベース、知識ベース、それからプログラミング言語、こういうものが歩み寄って行って、最終的には融合することです。

オブジェクト指向データベースが従来のデータベー

スのパラダイムと比較して相当異なったものがあるということをお話します。データベースの教科書を見ると必ず書いてあることですが、データベースは、データ独立を達成しようとするものであるというのが基本的な考え方です。データ独立とは、データの物理的な構造やデータベースの全体的な論理構造が変わっても、その変更がプログラムの書直しという形に波及しないようにしようという考え方なんです。それで従来のデータベースでは、むしろデータとプログラムを分離してしまおうという考え方だったわけです。これに対して、オブジェクト指向というのは、データとプログラムをまとめてカプセル化しようとしています。それではデータ独立という考え方はどうなるのかということが、これから非常に議論がいるのではないかと考えます。

それから、データの共有というのも、これはデータベースの中では金科玉条だと思うんですが、多数の人間がそのデータを共有するという環境を考えた場合に、いわゆるオブジェクト指向プログラミング言語だけではカバーできていなかったようないろいろな機能が必要になってきます。たとえば、マルチユーザのサポートですとかビューのサポートです。こういう機能をオブジェクト指向プログラミング言語をデータベース向きに拡張していくときに、やはり考えなくてはならないと思います。

データベース言語に関しても、従来はデータベース用の言語というのは何でもかんでもできる言語でなくてもよろしい、とくにデータを検索する、それからデータをアップデートする、そういう機能だけに限定して、言語を作りましょう、そういう考え方があったわけです。オブジェクト指向データベースでは、プログラミング言語と融合させようとする。従来のデータベース言語は、非手続的、それから集合論的な言語です。ところが Smalltalk などでは一度に一つのオブジェクトを処理するというふうな形態になっていて、あらかじめ定義された構造を巡航(ナビゲート)していく。従来のデータベースでは、非手続的で集合論的であったほうがいいと多くの方は、そう信じていたわけですが、これはそういうものとすんなり溶けこまないところがある。ですからそういう意味でも言語に関して相当議論が必要だろうと思います。

逆に、闇鍋の中にはデータベース屋さんにとっては非常に新鮮で面白いアイデアもあります。ソフトウェアの部品化とか差分プログラミングというアイディ

アです。これをデータベースのほうで言い換えてみますと、データベースそのものを部品化して流通させるという可能性が出てきているわけです。従来から実は DBMS そのものを部品化して、自分の好きなように組み立てられるようにしようという拡張可能 DBMS という発想はありますが、さらにデータベースシステムの構成要素であるデータ、データベーススキーマ、これらに対する基本的なオペレーションをクラスライブラリパッケージにして部品化できる可能性がある。これは従来にないようなデータベースソフトウェアの形態ではないかというふうに考えております。

それから最後にもう一つ、手続そのものをデータベースの中に放り込んでやろうという考え方です。これは従来にもなかった考え方ですので、これから非常に面白い分野になるだろうと言われております。ある人は、これをアクティブ・データベースというふうに呼んでいるようです。

**司会** どうもありがとうございました。それでは田中謙先生、引き続きましてご講演お願いいたします。

**田中(謙)** 私の話は「オブジェクト・オリエンテーションは夢の実現道具である」というタイトルで始めたいのですが、それで私自身はどういう夢を持っているかということ、また後でお話しさせていただきます。さきほどちょっと触れましたが、どうもオブジェクト指向パラダイムとオブジェクト指向言語というものが区別されて話されていないように思います。それがかなりいろんな誤解を生んでいるのではないかなと思うんです。オブジェクト指向パラダイムというのは思想であって、それを実現する機構として、オブジェクト指向言語というものがあつたわけです。現在存在しているオブジェクト指向言語というのはこういうパラダイムを育てる上で非常に使いやすい言語になっているわけです。

ただ言語というのは、言ってみれば機構ですから、言語を論ずる上では、たとえばメッセージ・パッシングであるとかクラスであるとかインスタンスであるとか、あるいはインヘリタンスであるとかいう機構がディスカッションの中心になってしまいます。

それに対して、そもそもオブジェクト指向パラダイムのモチベーションというのは、対象世界の事物と環境のモデルを事物を念頭に置いて計算機の中に作ってしまうということではなかったかと思うわけです。だとすると、非常に重要なことというのは事物のステートであるとか、事物のビヘービアであるとか、

事物の形状(シェープ)であると思います。ところが、これがあまり論じられていないように思うんですね。

私はデータベースのほうがそもそもの研究分野なものですから、どうしても、対象世界の事物のモデリングを計算機の中に作っていくというデータベースの考え方をしがちなわけです。ここでいう対象世界は、実世界であってもいいし、あるいは仮想世界であってもいいわけです。そういう意味でここに夢なんて書いていますけれども、まさに夢であってもかまわないわけです。

データの表現あるいはオブジェクトの表現の歴史をずっとみていくと、最初はステートだけを表現してデータタイプというのが出てくる。それに対して、ビヘービアを取り込んだアブストラクト・データ・タイプというのが出てくる。それに対して、さらにシェープまで取り込んだ形で MVC (モデル・ビュー・コントローラ) というのが出てくるというふうに思うんですね。

データベースの分野でもやはり同じように進んできたわけです。最初はステートに対応するものだけを格納していた。これは単なるデータベースですね、関係データベースというのも、このあたりです。それに対してビヘービアまで取り込んで、アブストラクト・データベースというものが出てきた。さらにはシェープまで取り込んだ形でオブジェクト・オリエンテッド・データベースが出てくるということになるのではないかなというふうに思います。

そういうものを考える上で重要なのは、マン・マシン・インタフェースということになります。いいマン・マシン・インタフェースというのはどういうものかということを考えてみると、どうやらそれは容易なプロトコルを提供するものであるというふうにいえると思います。容易なプロトコルというのは実は2種類あり得ると思うんですね。一つは単純なプロトコルの使用。単純だから容易であるといえます。他の一つは慣れ親しんだプロトコルを使うということだと思えます。単純なプロトコルを使うということは、できることが制限されてしまうことになって、これはあまりよくないというふうに思うんですね。それに対して慣れ親しんだプロトコルを使うとどうでしょう。さきほど特別講演のときに、佐伯先生がアフォーダンスということをおっしゃったと思うんですが、見ただけで何をやらなければならないかということが誘発されるようなプロ

トコルを使うことが重要ではないかと思ひます。オブジェクト指向パラダイムというの、これを可能にしてくれるような気がしているわけです。

じゃ、私の夢は何かというと、だんだんと年をとってきたので、研究能力がだんだん下がってくるものだから、研究支援システムを作りたいと思ひています。いってみれば思考支援システムです。ドキュメントとか道具、デスクワークに使う道具をすべて画面上で直接提示し、直接操作して、統合管理できるようなシステムを作りたいと思ひて、インテリジェント・パッドというシステムを作っております。

このシステムは三つのことをパラダイムに掲げています。第一に、すべてのものは紙であると考えております。ちょうどオブジェクト指向で、すべてのものがオブジェクトであるというのと同じで、すべてのものは紙である。それにはシェーブまで入れましょうということです。デスクワークで使うものだから、紙というシェーブを考えるわけです。第二に、すべての紙は固有の機能をもっている。おのおのの紙が機能をもっているの、インテリジェント・パッドと呼んでいるわけです。3番目が重要で、ある紙に別の紙を貼るということを許しています。貼ることによってレイアウトがまずできる、それと同時に二つの紙の機能の合

成ができるということを考えています。

これはインテリジェント・パッドの典型的な画面ですが、このシステムは富士ゼロックスの 1161 というマシン上で、Smalltalk-80 を用いて作ったわけです。この画面には、本のイメージをもったパッドが示されています。本の下部をマウスでクリックすると、本のページが変わります。この画面上には本のほかにもいろいろな部品パッドが作られています。本にはいろいろなパッドやパッドのマイコンをのせておくことができます。本にのっているマイコン自身も本になっていて構わないんです。

この画面には、デジタル・ディスプレイもあります。それから電卓用のボタンのようなものがありますね。レタリング用の紙もあって、文字や数字を一枚ずつはがして別の場所に貼ることができます。

こういうものを使うと、電卓を作るなんてのは簡単になってきてしまいます。電卓は倒と PC 板とディスプレイとボタンで構成されていますね。そこで、計算をするという機能をもっている紙を台紙として用意します。その上に、デジタル表示出力をする汎用ディスプレイを貼ってみます。それからパワー・スイッチを台紙に貼ります。次に、ボタンを台紙に貼っていきます。さらにそのボタンの上に文字を書き込むために、

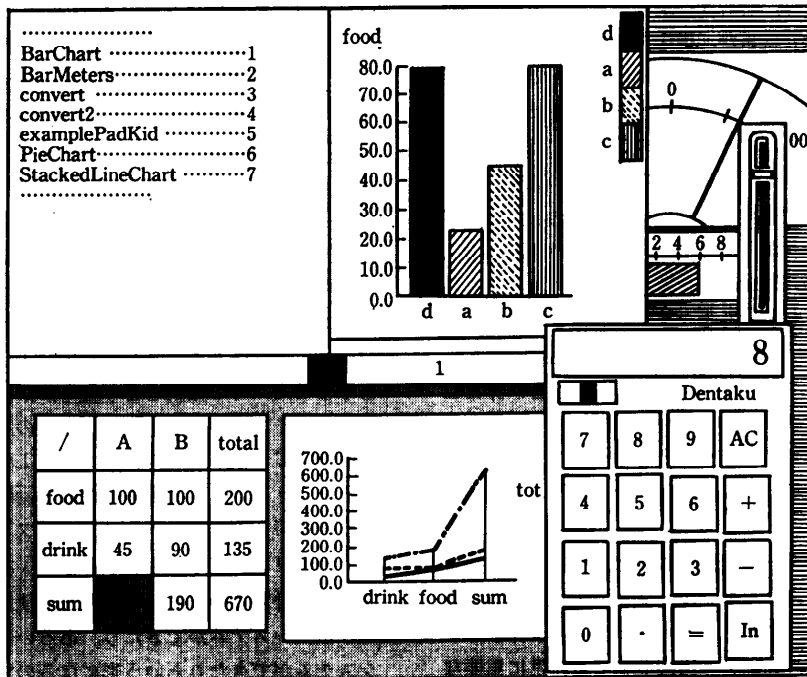


図-2 種々のパッドの画面コピー (左上方がパッドのカタログで、目次と第1頁を示す)

レタリング用の文字を貼り込んでいくわけです。この文字自体もパッドです。そうしてやると、パワー・スイッチをクリックすることで電卓の電源が入り、ボタンを押して計算していくと、 $0.7 \times 0.7$  が  $0.49$  なんてのが得られるわけです。パッドを貼り合わせただけでも動く電卓になっています。この電卓自身をまたコピーすることも可能です。

要するに、すべてを現実のものに即して考えていくんですね。紙に紙を貼っていくというだけでいるんなものが作れると。たとえばデジタル・ディスプレイの代わりに、これを引っぱがして、アナログ・ディスプレイを貼ってしまっても構わないわけです。

時間ですので、私の結論を言いますと、オブジェクト・オリエンテーションというのは、より多様な仕事ができる、より優しい計算機（ここはわざとこの字を使っています）を発想するに適している。計算機と仲良くなって、文化を育てることが肝要であると、オブジェクト指向というのは、こういう文化を育てるのに非常に適しているというふうに思います。以上で終わらせていただきます。

司会 どうもありがとうございます。パネリストの方々、ご質疑ありがとうございます。

安西 非常に面白いお話しでした。二つおききたいんですが一つは、いわゆるオブジェクト指向システムとか言語の研究結果が、田中先生の今のお仕事にどういうふうに反映されているかということです。

それからもう一つ、オブジェクト指向でよく言われることとしまして、オープンシステムということとの関連性について、一言お願いしたいと思います。

田中(誠) まず最初の話なんですけれども、ユーザ・インタフェースとかモデリングとかという面から、Smalltalk-80 と Flavors を考えた場合、私のところでは両方使ってます。あるシステムを作る場合に、最初 Smalltalk-80 で作って、それから Flavors にもっていったことがあります。そうしますと、非常に Smalltalk-80 的なシステムが作れました。それをもし最初から、Flavors で作ったとしたら、そういうような形にはならなかったんじゃないかというふうに思います。

そういうことを考えて感じるのは、私にとって一番面白いと思うのは、モデル・ビュー・コントローラ(MVC)の考え方です。モデル・ビュー・コントローラというのはオブジェクト指向の中でも非常に重要な概念ではないかと思います。決してクラス階層とかメ

ッセージパッシングだけがオブジェクト指向ではない。もちろんそういうものがあると MVC を非常に作りやすいわけですね。それからもう一つの質問はオープンシステムに関してでしたね。インテリジェント・パッドについて言いますと、どんどんどんプリミティブを作りこんでいって貯めていけば、それらはそれまでに作ったパッドと自在に貼り合わせが可能です。

たとえば、今まである種の数値表示メータがなかったとします。そこで、そういうメータを作ってやるとします。そうすると、それまで数値出力をするようなパッドを貼っていたところには、このメータを貼ることがいつでも可能になるわけですね。いろんなパッドをどんどん作っていけば、それだけいろんな部品が使えるようになるということになっていますから、オープン性というのは、非常に重要な概念として実現されているように思います。

司会 どうもありがとうございます。そのほかにございましょうか。

田中(克) 田中譲先生のお話しは、私も興味があります。MVC というオブジェクトそのものを表示させる機構は元々 Smalltalk という枠組の中で出てきた考え方だと思うんですが、それがたとえばオブジェクト指向データベースの分野にびったりと当てはまったようなメカニズムになっているのかということに関して、私は若干疑問をもっています。だから、これもさきほどの柴山さんのお話しではないんですけれども、オブジェクトを表示するというメカニズムというの、たとえば知識ベース、それからデータベース、それからそのプログラミング、おのおのにおいてやはりそれなりに考えていく必要があるという気もしているんです。そこらあたりはいかがでしょう。

田中(誠) MVC というのは、やはりメッセージパッシングであるとか、あるいはクラス階層とか、プロパティ・インヘリタンスだとか同じように機構でしかありません。その機構をどう使うかが問題です。たとえば今のインテリジェント・パッドの場合には、パッドを MVC を使ってインプリメントしています。パッドというメタファを実現するために、MVC を使っているわけですね。パッドというメタファを用いるというふうなワン・クッションが必要で、そこをどういうふうに設計するかもあって、非常に使いやすいデータベース・システムができたり、あるいは知識ベース・システムができたりという形になるのではないかというふうに思います。



質問 言葉の意味について、ちょっとご質問したいんですが、本当に Smalltalk の MVC のことをおっしゃっているのでしょうか。それとももうちょっと一般的なことをおっしゃっているのでしょうか。

田中(謙) MVC 的な考えで何かを作る場合は、Smalltalk-80 で作るのが一番作りやすいわけですが、思想そのものは他の言語でインプリメントすることもある程度可能だと思います。

司会 ありがとうございます。萩谷さんはウィンドウ廻りをオブジェクト指向でいろいろお作りになってご経験も豊富だと思えますけれども、今のことにに関して何かコメントがございませうか。

萩谷 大変興味深いお話しだったんですが、最近一つ問題に思っているのは、MとVの関係です。一つのMに対して複数のVを作ることがもちろんできますから、MとVは別なものなわけですね。その場合MとVの通信というのをメッセージでやろうと思うと非常に複雑になってしまいます。それで、たとえばアクティブ・バリューとかいろいろ研究されているんですけども、そのへんが一つの今後の課題ではないかと考えています。だから、そこは単なるメッセージのやりとりでなくて、もうちょっとそれを超える機構が必要ではないかと考えているんですけれども。

田中(謙) インテリジェント・パッドのレベルでは、それはほとんど問題ありません。MとVとの関係なんですけど、要するに二つのリンクが必要なんです。一つは、モデルのステートがチェンジしたときに、それをビューに反映させないといけないうから、アップデートを伝播するリンクが必要です。これは要するに changed update の二つのメッセージでやるやり方ですね。もう一つは入力しようという場合、たとえばビューを通してそのモデルに入力しようという場合には、そのビューでモデルをリファレンスすることが必要ですので、そのリファレンスのポインタが必要なんです。それで結論をいうと、その二種類のリンクがあれば、インテリジェント・パッドのようなものは実現できるというふうに思っています。

司会 どうもありがとうございます。それでは一つ前に戻りまして、田中克己先生のデータベースに関連して、ご質疑をお願いしたいと思います。

じゃ、ちょっと僕が、口火を切らせていただきますと、僕が田中先生にお願いしたときに、応用の分野で、というお話しをしたかもしれませんが、まあオブジェクト指向の究極はもうデータベースだと、僕はそ

ういうふうに思っています。オブジェクト指向では、対象にしているのは決してデータという1や0の列ではなくて、少なくとも情報なわけですね。場合によっては知識かもしれないし、場合によっては行動かもしれないです。オブジェクト指向はどうやったらモジュール化されたデータのおのおのに意味をもたせられるかということが重要だと私自身は思っております。そういった意味では、オブジェクト指向イコール、新しい形のデータベースなんじゃないか。そこにたとえ並行的なアクセスをどうしようとか、分散データベースはどうしようとか、ほとんどすべての問題は出てきちゃっている。その上に田中謙先生のように何かを組み立てるといふところまでくれば、もうそれが、実際にいうオブジェクト指向のパラダイムの実現じゃないかなと思っているんですが、いかがでしょうか。

田中(克) 半分賛成、半分反対です。昔のデータベースはデータとプログラムを分離するからオブジェクト指向と相容れないというようなご指摘がありました。しかし、データ独立という考え方がそもそもどうして出てきたかといいますと、これはデータを処理するアプリケーションの寿命をいかにして延ばそうかという知恵の一つとして出てきたわけです。もう20年ほど前ですね。ですから、私はその考え方自身は依然大事だと思うんですね。

それから、データベースというのはビッグ・アプリケーションの分野ですね。実現には非常に大きなアプリケーションの分野であり、新しい技術に対しても保守的です。それで、まだまだオブジェクト・オリエンテッド・データベースの技術が即実用化されるとか、そういう段階ではありません。このことも考慮すべき一つの重要な要素です。いずれにせよオブジェクト指向という観点からデータベースのテキストに載っていたようないろいろな事柄を見直す時期にきていると考えています。

司会 どうもありがとうございます。田中謙先生もご意見ございませうか。

田中(謙) 今、田中克己先生が大体おっしゃったんですが、データ独立性というのは、フィロソフィであって、それを実現する機構として、たとえばデータとプログラムは分けましょうという話になったんだと思うんですね。オブジェクト指向は、そういう機構とは相容れないかもしれませんが、そのフィロソフィとは非常に親和性がいいように思います。つまり、データ独立性というよりも、むしろ、オブジェクト独

立性という言葉でも作ればいいのかもありませんが、そういった考えに基づいてやっていけば非常に親和性が高いのではないかなと思うんですね。

司会 その他ご意見ございますか。

質問 最初の萩谷さんと柴山さんに代表されるようなプログラミング言語をやっておられる方はオブジェクト指向というのは計算のメカニズム、あるいは計算のモデルというふうにとらえているというふうに見受けました。一方、両田中先生に代表されるデータベースの方の場合は、データベースですから、多分田中譲先生がおっしゃったように、事物のモデル化というのが最初にあるというような考え方で、割と構造的な面からオブジェクト指向というものの特徴なんかとらえていらっしゃるような気がしました。オブジェクト指向でいろんな分野の方が共通して研究なさるのが非常に意味があるわけですが、お互いに他の分野とはこういうことが違うとか、あるいは違いというものがある程度鮮明にして研究されると、お互いにいいんじゃないかなと思います。お互いにどのようにとらえられているか、簡単にコメントいただければと思います。

司会 どうもありがとうございます。この点は実は今日のフロアの方及びパネリストの方の全員の興味のあるところで、これについてはやっぱりはっきりしてから帰ってもらいたいと、僕も思っておりますが、今の時点でよりは、もうあとお二方のパネリストの方にご登場いただいで、全員6人雁首を並べたところで、その議論をしていただきたいと思います。多分竹内彰一さんのお話でも、多少そんな話が聞けるのではないかなと思います。ちょっと、この質問はベンディングさせていただきまして、時間もちょうどいいころなので、さきに進ませさせていただきたいと思います。

最後のセットは、人工知能応用、特に分散人工知能応用に関しまして、慶応大学の安西先生、そのあとオブジェクト指向応用に関する展望として竹内彰一さんをお願いします。では安西先生お願いします。

### 人工知能とオブジェクト指向

安西 分散 AI とオブジェクト指向ということに関して、とにかく自分の考えているところをお話しさせていただきたいと思います。特に今質問が出たのでちょうどいいんですけれども、結局、オブジェクト指向というものを人工知能研究者としてどういうふうに見えるかということ、基本的なところだけお話しさせていただきたいと思います。

まず分散 AI というときに、本当の意味での分散 AI、つまり本当に局所的な計算モジュールたちが集まって、上からのモニタリングがほとんどなく、一緒に働いているだけで何か仕事をしてくれる、そういう AI システムですか、あるいは全体として一つの目的をもっているわけでもなくて、もっとマルチ・パースの AI システムであるとか、そういうものが考えられます。

それからもう一つ、協調的な問題解決システムというか、上にははっきりした目的があって、その下のモジュールたちが協力して、全体として一つの仕事をするような、そういう階層的な問題解決システムが考えられているかというふうに思います。

それで、分散 AI というときには、このような二つのシステムを含めて、なんとなくそういうふうになっているかと思いますが、オブジェクト指向という考え方が、AI の中でどの程度使われているのかということ、一步退いて考えてみますと、どちらのシステムでも、本当に分散 AI の応用として、これだというものはないのではないかな、少なくとも実用化という面では、AI の技術でオブジェクト指向が正面切って使われたということは、まだないのではないかなという気がいたします。

また、こうして考えると、むしろ分散型 AI といってもそれほど安定した概念ではないということでもあります。

では、AI の本当の基本のところは何であろうかということ、一言で、誤解を覚悟で言ってしまうと、これはダイナミック・リプレゼンテーションの研究であります。結局外界をどういうふうに表示するのか、しかも表現の中にダイナミックな計算という考え方を含める。そういう計算という考え方をもとにして、外界あるいは認知的な対象というものを計算機上にどういうふうに表示するかということが AI の根本問題であるかと思います。その点から考えると、あるデータとそれを使用するための基本的な操作手続をひとまとめにして、その合併体を一つのモジュールとみなしたものを、オブジェクトと言い、そして、構成しようとするシステムをオブジェクトの集まりとして表現しようとする考え方を、オブジェクト指向だ、というふうに定義できる。これは非常に広い定義で、さっきの萩谷さんに言わせると、アブストラクト・ナンセンスだとラベルを貼られかねないかなと思いますけれども、AI の観点からみると、やっぱりこの「表現」という

考え方が非常に重要になっているかと思えます。

それで、対象を計算機の上はどうやって表現するかということ考えたときに、オブジェクト指向に関して多分一番重要だと思う考え方は、半順序関係とそのダイナミックな変換だというふうに思えます。

半順序というのは、結局あるエレメントたちの集合があって、その中から二つのものを取り出したときに、その間に大小関係がつくかもしれないし、つかないかもしれない、そういう関係なんで、抽象的でどうでもいいような関係にみえるかもしれませんが、結局世の中の対象とそれに対する処理を AI システムとして実現したいと考えたときに、世の中の対象をどういうデータ構造で、あるいは表現構造でもって計算機上に表すことができるかということ、その点からみるとオブジェクト指向の考え方は、このぐらい抽象的なレベルに落ち着くかと思えます。

たとえば階層的な構造、これはネットワークでもいいし、あるいはもっと特殊な、たとえばツリーでもいいわけですが、そういうような表現形態というものがある。そのノード間のコミュニケーションというものをさっき言ったダイナミック・リプレゼンテーションで考えてみると、いわゆるマルチプル・インヘリタンスだとか普通のインヘリタンスというものに関連づけることができます。

また、たとえば、もっと手続的な階層を考えれば、リファレンシャル・コヒーレンスみたいな、そういう考え方もありますよ。さらに、多重アロケーションに関しては、Flavors とか CLOS みたいなものがあります。最近、Flavors とか CLOS みたいなものはオブジェクト指向と呼ぶとか呼ばないとか、そういう議論があるみたいですが、ここでは一つの例としてあげておきます。

それからインタフェース・マッチングについてですけれども、これは階層的なノードあるいはオブジェクトみたいなものをもうちょっとわれわれの認識的な目で、空間的な接続関係とか隣接関係とか、そういう関係でもってみるという考え方があるかと思えます。

たとえば、一つの例は有名な EMERALD のタイプ・コンフォーミティという考え方がそうだと思いますが、EMERALD にはインヘリタンスのメカニズムは基本的には入っていないと思います。タイプ・コンフォーミティというのは隣りあったオブジェクト同士の間パラメータだけのやりとりがあって、オペレーション自体はオブジェクトの中に隠されているよう

な、そういう考え方です。結局階層的な考え方をしないで、フラットな構造をとっているものだと思います。

このように、階層的な順序を入れる、半順序を入れる、あるいはインタフェース・マッチングという意味で半順序を入れる、そういう面でいろんなオブジェクト指向のシステムあるいはメカニズムをみていくことができる。しかも、そのメカニズムの表現を作っていくことができるんじゃないかと思うわけです。

もう一つ、重要な問題としてコンストレイントという考え方があります。有名な例として、アラン・ポニングのコンストレイントハイアラキーがあります。これは、コンストレイントアルゴリズムを使ったシステムをさらに拡張して、コンストレイントに階層を導入したシステムであります。これが 1987 年に発表されています。こういう制約という考え方も半順序とその動的変換という構造の上で考えられた表現の一つであるわけです。

とにかく対象をどういう構造としてみるかということを考えるのが、非常に重要です。つまり対象がどの程度の半順序をもっているかということをしきりと追っていくないと、どういうシステムになるかということをしきりと把握することは難しい。経験的にアドホックに作っていくことはできるかもしれないけれども、ある程度こういうレベル分けの考え方が必要ではないかと思っているわけです。

たとえば、順序がかなりついている例として、私どもでもやっておりました力学の問題解決システムというのがあります。力学の知識としては、物理的対象の概念があり、その中には動けるものがあり、また個々の対象がインスタンスとしてある、というように、階層的な概念の構造があります。

それからもう一つ別にルールの階層構造があります。この手続の階層と、それから概念の階層が一つのメカニズムで働くようなオブジェクトたちの中に書けるようになっていて、こういうかなりの順序をもった半順序構造として問題を表現できます。

一方では最近ちょっと流行りになっているニューラル・コンピュータという分野があります。ここの分野では、いわば半順序がほとんどついていないかのように見えるシステムを使っている。たとえば、画像認識を考えますと、画像の隣り合ったピクセルだけが関係あるというふうなみて、その隣り合ったピクセルの局所的な関係の計算だけを積み上げていって、

それで画像認識をやる、あるいは画像の修復をやるという、そういう考え方があります。

こういう順序があまりなくて小さいグレインのシステムと、力学の問題解決システムのような、大きなグレインでかなり順序がついているシステム、そういうのがいろいろあって、オブジェクト指向による AI と一般的に言っても、その中のどこどこをやっているかということ、はっきりさせなければいけない。

それで、最後に、そういうふうと考えていくと、はじめに申し上げましたけれども、AI のシステムを作るにあたって、基本的なところでオブジェクト指向ということが、あまり出てきてないように思う。それには二つの問題があるかと思っています。

まず、メッセージ・パッシングのオーバヘッドと、それからグラニュラリティの問題ということがあります。これは、いわばトレードオフであって、大きなグラニュラリティのモジュールを作れば、コミュニケーションのオーバヘッドは小さくなるし、小さなグラニュラリティのシステムだと、さっきのニューラル・ネットワークをオブジェクト指向でやるなんていうのは、まったくその典型的な例になるわけですけれども、コミュニケーションのオーバヘッドはものすごく大きくなります。こういう問題は、アーキテクチャのレベル、あるいは言語レベル、システムのレベルとかなり意思疎通をよくして、それで考えていかなければいけない問題だと思っています。

もう一つの問題は、大域的計算ということです。一番最初に分散 AI と申しましたけれども、いろいろなモジュールが一斉にワーと計算をすれば、それで仕事ができるかということ、なかなかそうはいかない。やっぱりモジュールたち全体を見渡すような何かの場といますか、そういう大域的計算機構がどうしても必要になってくる。これは、従来の AI ではブラックボード・アーキテクチャとか、プロダクション・システムでいえばワーキングメモリとか、そういうところで実現されているわけです。これがオブジェクト指向の概念と相容れるものであるかどうか、明確ではありません。

その典型例がニューラル・コンピューティングでしょう。こういう大域的な計算法と、それからオブジェクト指向のきわめてモジュール化された計算法というものが、互いに独立にやっていけるものなのかどうかということ、AI の基本的な問題を考えようと思っているものにとっては、非常に面白い問題であると思

います。一応これで終わります。

司会 どうもありがとうございました。それでは引き続きまして、竹内彰一さんにお話しをお願いします。

### オブジェクト指向とメディア

竹内 ちょっと大きな題目ですが、オブジェクト指向の展開について考えてみたいと思います。

オブジェクト指向というのはどちらかという手法として定着しつつある。しかし、私としてはここでオブジェクト指向自身は元々応用をめざして出発したものだということ、これを改めて申し上げたいと思います。

オブジェクト指向の起源というと Smalltalk とか、Actor とか Simula というのが必ず出てくるんですが、そこで出てきたいろんなオブジェクト指向のメディアが今やいろんな分野に波及しています。

そもそも、こういうふうなオブジェクト指向がいろんな分野に浸透している理由というのを考えてみると、大抵論文などに書かれているのは、オブジェクト指向でやるとモデリングが容易でしたとか、あるいは設計が容易だ、あるいはモジュラリティが高くてメンテナンスが楽とか、情報隠蔽がよいとか、階層化できたのでプログラム・モジュールの再利用ができて非常によかったとか、そういう話になるんですが、何度も言うようですが、オブジェクト指向というのはそういうものをめざしていたわけではない。オブジェクト指向のルーツ、Smalltalk、Actor、Simula、どれを取っていてもソフトウェアとはおよそ縁のない人たちが (Simula の場合は少しあったかと思うんですけど) が始めています。やっぱり別なことを考えていたわけです。たとえば Actor の場合は、今は並列計算モデルと考えるのが一番妥当なんでしょうけれども、Hewitt が IJCAI の 73 年に発表した論文を見ると基本的には知識の手続的表現のフォーマリズムとして提案しています。

Simula は名前のとおり、シミュレーションをめざしていたわけです。Smalltalk は、みなさんご存知だと思います。一番有名な話は、Alan Kay が考えていた Dynabook です。これは一言でいうと創造的活動を支援するものです。

どれ一つ取ってみても、ソフトウェア工学的なものはないわけです。別にモジュラリティが高くなるようにやろうとして、オブジェクト指向を考えついたわけではないわけですね。それが第一義的目的ではなかつ

たわけです。

オブジェクト指向は、どのようにして生まれたかということをも改めて考えようとして、この三つをまとめても簡単には答は見つからないのですが、あえて言えば、オブジェクトという概念を発見したことが三つに共通な誕生の契機ではないでしょうか。オブジェクトという概念の実体を計算機中に表現して、それを操作したい、そういう点でこの三つは共通のモチベーションをもっていただいたのではないかと思います。

三つを同時に議論していくのは難しいので、一つに絞らせていただきます。三つの中で、応用ということを考えてときに、一番大きなインパクトがあると思われる Smalltalk、そして Smalltalk の誕生の母体であった Dynabook について話を続けたいと思います。

オブジェクト指向には、何か応用のイメージがあってそれにドライブされていたと思うわけです。今後のオブジェクト指向を占うときに、ここではパーソナル・ダイナミック・メディアに象徴されるようなメディアの話を通じて考えてみたいと思います。Alan Kay と Adele Goldberg による "Personal Dynamic Media" という論文があるんですけども、そのイントロがメディアというものについて非常に深い洞察をしていて、印象に残ります。メディアとは情報や知識を貯え、リトリートし、マニピュレートしてコミュニケーションする道具です。また思考を、マテリアライズする道具でもあります。

それから、貯えられているものをレビューするか、レビューして、また自分の考えにフィードバックすることも可能です。メディアというものは基本的に思考のツールです。昔からメディアは当然あったわけですが、本のような昔からあるメディアは、基本的に非会話的であり受動的なものであります。つまり、本の場合ページを自分でめくって、そこにあるものを読み取り、理解するということができない。人間にとってみれば、一方向的なものであり、受動的なものです。

計算機が出てきて、メディアは変わりつつあります。今、技術予測などでマルチ・メディアとか言われていますが、そういう話とは違って、Alan Kay が言っていたのはもう少し哲学的な話なんです。計算機はまったく新しいメディアになり得るということです。従来のメディアと違って、計算機メディアには二つの特徴があります。一つはメタメディアということです。計算機というのは基本的に何でもシミュレートで

きます。たとえばアニメーションをディスプレイに出すこともできる。そういう映像メディアや文書メディアを計算機の中で真似することができる。そういう意味でどういうメディアでもあり得るということで、メタメディアといっています。それからもう一つは能動的メディアということです。つまり、本や映画のように一方的に流し続けるだけでなく、人間が質問をすれば、それに応じた反応をするという能動的なメディアです。要するに計算機で、思考のツールとしてのメディアを実現すると、こういったまったく今までなかった特徴をもつメディアを作ることができる。こういう哲学的な発想にドライブされて、Dynabook というものが出てきたわけです。

Dynabook は文章、詩、手紙、メモ、楽譜、音楽などいろいろなものを全部入れようとしています。それを入れるだけじゃだめなんで、ビットマップとかマウスを使い、リトリートしたり、マニピュレートしたり、問い合わせを出して答を貰ったり実験したり、人とコミュニケーションしたり、シミュレーションしたり、とにかくいろんなことをやろうとしています。こういうものをめざしている研究の中で Smalltalk が出てきたわけです。

こういうふうに歴史を辿って、あれはあであったこうだったと言ってもしょうがないんですが、私が思うにはオブジェクト指向の応用を考えるときには、やはりオブジェクト指向がもたらした新しい応用というのを考えてみると良いのではないかと、するとそれは能動的メディアであろうと思います。能動的メディアというのは、基本的に情報や知識を表現したり蓄積したり操作したり、レビューしたりということが基本になるわけですが、実際オブジェクト指向は、これを実現する枠組をもっています。そうするとオブジェクト指向言語も持っているクラスとかオブジェクト間のメッセージ通信とかブラウザとかいったものが、抽象的な言語機構としてではなくて、能動的メディアという応用に対して、すっぱりと位置が決まって、役割が明確になるんじゃないかと思います。

まず第一に「表現」という点ではクラスでやる。「蓄積」という点ではオブジェクト管理機構というものがある。それから「操作」ということではメッセージで操作する。あるいはマウスで操作する。レビューという点では、ビットマップというものがついていて、ブラウザのようなものがある。そういうことを考えていますと、やっぱりオブジェクト指向とい

うのは、単にソフトウェア工学的なアドバンテージを浸透普及させるだけではなくて、やっぱりそれ自身何かの応用を促進するのではないかと思えてきます。そしてその応用の中心は能動的メディアと知識情報処理を結合した知識メディアになるのではないかと考えています。

知識メディアというのは Mark Stefik が言い出した言葉でまだ耳慣れない言葉ですが、その役割は知識の伝播を早めることです。メディアというのは基本的には媒体ですから、知識メディアは知識を流す、そのスピードをはやめるものである。それから単に流すだけじゃなくて、知識の相互作用を高め新しい知識の出現を促進するものです。非常に社会学的なんですから、そして、こういったものは、今第5世代なんかでやっている知識情報処理技術を補完するものではないかというわけです。

AI の目標というのは大方の同意するところでは、知的で自律的な思考マシンを構築することです。これよりもより大きな応用の機会があって、かつこういう研究を補完する目標というのは、インタラクティブな知識メディアがどういうものか考え、それを開発することではないかというふうに、Stefik は彼の論文の中で書いています。

知識メディアに関して、現状はどうなっているかという、まだまだ寂しい状態です。知識を扱う能動的メディアといえるようなものは、いろいろ考えてもエキスパート・システム以外にありません。ただそれは知識を扱うという点でのみ、これしかないというのであって、エキスパート・システムだって能動的メディアかという、まだ足りないものがたくさんあるような気がします。

それから問題点としては、エキスパート・システムは互いに孤立化しているということです。互いに知識を伝播するためには、なんらかの形でつながっていないといけないんですが、まったく孤立している。したがって、コミュニケーションできない。それゆえ、知識間の相互干渉というものも生じさせていません。Stefik は今後知識のマーケットみたいなものができるであろうとか、そういうマーケットの中に、知識をサプライする人が出てくるであろうとか言っています。たとえば、CYC という人工知能で百科辞典を作ろうというプロジェクトがありますが、これは常識知識ベースのサプライ側になるわけですね。そういうものを、たとえば、みんなもらってくるなり買ってくるなりして、

何か知識処理をしたり、あるいは自分のものを追加して、他と比べて、もっと良いものにするかということが起きると、そのとき、問題になるのは知識表現の標準化でしょう。それについては、ここでは深く立ち入らないことにします。

今のエキスパート・システムは、さきほども言いましたように、能動的メディアというにはちょっと寂しいです。推論が中心で、知識を蓄積したり、伝達するメディアを実現するという発想があまりないように見受けられます。知識表現一つをとってもルール一本やりではなくて Dynabook が実現したのと同じような豊富な表現を許すことが必要ではないでしょうか。それを使った推論は二の次でも良い気がします。AI に対して能動的メディアはこれからいろんな影響を与えるんじゃないかと思います。

結論ですが、第一にオブジェクト指向を手法として考えると、基本的にオブジェクト指向本来の応用などというものは考えられなくなります。第二に、私としては、オブジェクト指向はそういうパラダイムとか手法以上のものではないかと思っております。第三に、しかし、オブジェクト指向の手法としての有利な点は、今後もどんどん認められて、いろんな方面に拡散していくのではないかと思います。第四に、ただ、その応用という面で見ると、一番重要でインパクトがあると思われるのは、能動的メディア及び知識メディアのようなものをどんどん進めることじゃないかと思えます。以上です。

司会 どうもありがとうございました。

田中(贖) モデリングという言葉を考えて場合に哲学的には、昔から二つの立場があるわけですね。一つは実体に着目して、それを記述していく、という立場。他方はまず関係があるんだとして、関係をまず記述して、その結果として関係を満たすような事物というものがあつたか考える立場。私自身は、オブジェクト指向という言葉自身は、きっと哲学という実体、事物というものを念頭にしてモデリングするという哲学的な立場を計算機屋さんが呼んだんじゃないかなというふうに理解しているんです。そういうふうに理解すると、それをうまく実現するために、オブジェクト指向言語として、いろんなものができたと考えられます。事物をうまくモデリングするのに必要な機構、たとえばクラスという考えとか、あるいはインスタンスという考えとか、クラスハイアラキをもってプロパティを継承するとか、そういう機構をたくさんもち込んで

だのではないかなというふうに理解したいと思っているんですが。

司会 安西先生いかがでしょうか。

安西 さきほど竹内さんがおっしゃっていました Stefik のナレッジ・メディアということについて、ちょっと申し上げたいと思います。私がさきほど申しましたのは、表現の問題であって、AI の基礎的な一つの問題であります。その面からみても、さっき竹内さんが言われていたような能動的メディア、特に知識に関するメディアというのは非常に重要な問題である。多分一番重要な問題の一つになるのではないかなというふうに思っております。ただ、そのときにオブジェクト指向というのは相性がいい「だろう」というのは、よく分かるんですけども、本当に相性がいいかどうかということとは、やってみないとちょっと分からない面もある。

たとえば、知識の統合みたいなことですね、そういうことは多分やらなくちゃなくなるだろうという気がします。オブジェクト指向という概念が、そういう操作に関して都合良くできているものかどうかということがもう少し分かりたいと思います。

#### おわりに

司会 そろそろ時間もきてしまったんで、そろそろまとめなければならぬので、収束させる方向に多少話を進めましょう。オブジェクト指向の概念がいろいろところで出てきているが、それはよく考えてみると同じじゃないか、そういう意味では、統一的な概念として考えることができるのではないか、ということが分かってきたのは 80 年代の後半の時期、ことに 85 年前後からではないかと思えます。今日のお話を聞くと、それはそうとして、やっぱりプログラミング言語だとかメソドロジーというのは、アプリケーションを書くため、そうなってみると、統一的な概念と

いうところにこだわりすぎるよりも、ドメインデペンデントなディフィニションをしてみるとか、ドメインデペンデントなアプリケーションを書いてみるとかというようなことがもっともっと必要ではないかなというように議論があったように思います。

それから、もう一つ、オブジェクト指向をいろんなところのドメインにもって行って料理して、うちのオブジェクト指向はこうなんだ、うちのオブジェクト指向はこうなんだというのもいいかもしれないけれども、やっぱりわれわれがほしいのは、これを踏み越える新しいパラダイムでしょう。たとえばニューラル・ネットワークのモデルとしてオブジェクト指向は、場合によたら良くなって、何か新しい拡散計算モデルみたいなものが新しいパラダイムになるかもしれない。MVC みたいなものをもっと発展させ、それがオブジェクト指向と似ているかもしれないけれども、やはり基本的な思想はここにあるんだよと主張できるような新しい考えが、どんどん出てきたらよいのではないかと思います。

一番最後に、自然界というのは、元々分散して並列して存在しているもので、計算システム自体が最終的には自然の世界、もしくは何かある原理に基づいて矛盾なく動いているある系というものをシミュレーションしている、すなわち、計算機というのは、広い意味で言うシミュレータということになります。そういう意味で 80 年代の一つのプログラミング・パラダイムとしてやはりオブジェクト指向の占める位置は重要だと思えます。また、これだけいろいろの分野の方々に、今日お集まりいただいて、オブジェクト指向という考えをベースにして、いろいろなディスカッションができました。今後のおの新しい応用分野に即した新しいパラダイムを、またお考えいただいて、そしてまた何年後かに、このようなディスカッションができるとよいと思います。