

TM-0727

オブジェクト指向言語の
レイアウト問題解決への応用と
CADデータベースの考察

渡辺俊典, 近藤秀文 (日立)

May, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

オブジェクト指向言語のレイアウト問題解決 への応用とC A Dデータベースの考察

わた なべ とし のり * こん どう ひで しみ *

渡 辺 俊 典 近 藤 秀 文

* 日 立 製 作 所 シ ス テ ム 開 発 研 究 所
川 崎 市 麻 生 区 王 禅 寺 1 0 9 9

キーワード： C A D (Computer Aided Design), レイアウト (Layout)
オブジェクト指向 (Object Oriented Paradigm),
データベース (Data Base)

1. はじめに

レイアウト問題は設計自動化等の分野で長い間研究されてきた問題のひとつであり、例えば、プラントにおける設備配置、建設業における部屋あるいは家具の配置、新聞記事の配置、電子機器の部品配置などは良く知られている問題である。

これらの問題を計算機処理させるために、最適化技法などを用いた種々のアプローチがなされてきたが、計算機によって得た解と専門家によるものとの間に質的な差があるということが良く言われてきた。

レイアウト問題のより良い解決のためには、次の諸機能が要請されると考える。

(r 1) 配置物の持つ複雑な形状やその意味の表現と利用機能

(r 2) 配置物をレイアウトする際の意味的拘束条件の表現や満足化機能

(r 3) 組合せ問題に対する効率的解探索の機能

さらに、C A D開発一般について言えることであるが、C A Dの目的が設計対象製品の合成の効率化や自動化である以上、対象製品のイノベーションに対して常に適合した処理機能を持つことが望まれる。このことは、C A Dシステム自身の開発過程やその後の運用保守過程において、プログラム機能の改良・変更・再利用などが容易に可能でなくてはならない¹。すなわち、次が要請される。

(r 4) 複雑なC A Dシステムが効率的に開発保守できる。

従来の手続き型言語によるC A D開発においては、プログラムの機能設定やモジュール分割を事前に確定するということが一般的であり、事後でのプログラム変更には多大の保守労力を要するというのが一般的である。

設計領域での古くからの典型的問題のひとつであったレイアウト問題に対してこれら(r 1)～(r 4)の要請を満足しうる新しいプログラミングパラダイムが存在するとすれば、それは他の多くの設計問題の解決に対しても有効性を持つ可能性は高く、新しいC A Dシステム実現へのひとつの基礎を与えてくれるものと期待できる。

新世代コンピュータ技術開発機構(I C O T)で開発されたオブジェクト指向核言語E S P²は次の特徴を持っている。

(c 1) オブジェクト指向機能による対象物の諸属性や事物間の関係の表現力の高さ

(c 2) オブジェクト指向表現の持つ知識のモジュール化記述力の高さ

(c 3) 継承機能やメッセージ交信機能による事物間の関係定義の柔軟さ

(c 4) 論理型言語 Prolog を内包したことによるバックトラッキング機能による解探索能力の高さ

(c 5) 並列処理模擬機能を有することによる並列問題解決方式の実験の容易性

機能 (c 1) は (r 1)(r 2) に、(c 2)(c 3) は (r 4) に、(c 4)(c 5) は (r 3) に対して各々有効であると期待される。

本報告では、E S P 言語を用いて開発した計算機室への機器レイアウト・パイロットモデルの紹介と、その段階的開発過程において E S P の果たした役割りを述べると共に、本事例を足場としたオブジェクト指向データベースに対する考察をおこなう。

2. レイアウト問題解決へのアプローチ概要

2. 1 問題説明

計算機室内に、計算機システムの構成機器である、中央処理装置 (C P U)、コンソール・ディスプレイ (C D)、カード・リーダー (C R)、磁気ディスク装置 (D K U)、漢字プリンタ (K P R)、ライン・プリンタ (L P) などを配置・配線する。制約条件として下記を考える。

(c 1) 機器は室内に配置し、柱等の障害物を避ける。本体同士は重なってはならないが、保守エリア同士は重なってもよい。

(c 2) 類似の機器は近くに置く。

(c 3) 中央処理装置や磁気ディスク装置、あるいは磁気テープ装置は部屋の入り口から離れた奥の方に置く。逆に漢字プリンタやライン・プリンタ類は出入口の近くに置く。カード・リーダーはコンソール・ディスプレイの近くに置く。

(c 4) 配置した機器の前面線をできるだけ揃えて、整頓する。

(c 5) 機器の前面パネルをコンソール・ディスプレイ装置の方に向けて、それらの稼働状況がコンソール・ディスプレイ装置部にいる計算機オペレータから見えるようにする。

(c 6) 機器のケーブル穴は、フリーアクセス (床面に敷きつめた正方形のパネル) 一個の上に位置させ、一つを取り外せば床下のケーブルと接続できるようにする。

2. 2 問題解決アプローチ

本研究では階層化並列問題解決アプローチをとった。

(s 1) 入力データとして部屋の諸属性(形状, 出入口, 柱, 床面タイル配置など)及び配置すべき機器のリスト, 機器間の配線データを与える。

(s 2) 上位階層で問題分割を行なう。グループ化ルールを用いた機器のグループ化の後、部屋を各グループに対応したゾーンに分割し、その形状を適切なものに設定する。機器間のケーブル接続データよりグループ間の配線を分析し、これがゾーン境界を通る点に仮想コネクタを定義する。全体の配置配線問題は各ゾーン内での機器及び仮想コネクタの配置及びそれらの配線問題からなる部分問題に分割される。

(s 3) 下位階層で配置制約条件を考慮して部分問題を並列に解く。最初に配置問題を解き、得られるケーブル端子や仮想コネクタの位置を前提として配線問題を解く。異なる部分問題の間に機器の距離や方向に関する制約が存在するために、部分問題(特に配置)間での協調が必要となる。協調の方式として本実験システムでは二方式を用いた。

(c 0 1) 他の部分問題の解決を待って後、制約を満たす配置を決める。

(c 0 2) 他の部分問題の解に応じた制約満足化戦略をあらかじめ各部分問題に持たせておき、待ち無しで部分問題間の制約を満足させる(種の事前常識の導入)。

(s 4) 最後に仮端子間を配線し全体の解を構成する。

以上の問題解決で使用する主要な知識として下記をあらかじめ保有しておき利用する。

- (k 1) 機器グループ化や部屋のゾーン分割及びグループのゾーン割り当て時の制約
- (k 2) 機器の諸属性や配置順序及び配置時の制約条件
- (k 3) 機器の配置候補地の生成法(既配置機器を基準として定義する)
- (k 4) 配置不可となった場合の既配置機器のスライド操作
- (k 5) 配線機能
- (k 6) 配置問題解決時に必要な各種の計算幾何学演算機能(図形演算)

3. レイアウト問題解決パイロットモデルのESPによる実現

3. 1 ESPオブジェクトの構成

本システムにおけるESPオブジェクトの構成を図3. 1に示す。

- (1) 問題定義オブジェクト： 与えられたレイアウト問題を表現する。配置する部屋の平面図、配置機器の型紙及び配線データ等を定義している。
- (2) 並列問題解決オブジェクト： 問題解決過程を制御するマネージャである。
- (3) 上位配置処理オブジェクト： 与えられた全ての機器をグループ分けし、部屋を各グループごとのゾーンに分割する。
- (4) 上位配線処理オブジェクト： 上位配置処理オブジェクトの処理結果を基に、機器グループ間の配線計画を行ない、ゾーン境界線と交わる点に仮想コネクタを設けて、配線問題をゾーン毎の部分配線問題に分割する。又、全ゾーンでの配線終了後に仮想コネクタ間を配線し、全体の配線を完成させる。
- (5) 並列配置プロセス制御オブジェクト： 並列動作する各下位配置配線オブジェクトに対して、プロセス間通信や同期処理などを行なう。
- (6) 並列配線プロセス制御オブジェクト： 並列動作する各下位配置配線オブジェクトに対して、プロセスのスケジューリング等を行なう。
- (7) 下位配置配線オブジェクト： 各ゾーン内での機器配置や機器間配線を行うオブジェクトである。互いに並列に動作する。
- (8) 配置配線状況記憶オブジェクト： 配置状況及び配線状況を表現する。その時点までに決定したゾーンの位置・形状、機器の配置位置及び配線位置等を記憶する。
- (9) 上位階層用ルール・オブジェクト： 望ましいゾーン面積を計算し、それぞれのゾーンを部屋内に生成するためのルールを保持している。
- (10) 上位階層用制約評価オブジェクト： 生成されたゾーンが、与えられた上位階層用制約条件を満たしているかどうかを評価する。
- (11) 仮想コネクタ間配線オブジェクト： 上位配線処理で仮想コネクタ同士を最終的に配線する。
- (12) 下位階層用ルール・オブジェクト： 各ゾーン内で現時点までの配置状況を認識し、未配置機器の配置候補地を生成するためのルールを保持している。ルールは機器の種類毎（CR、DKU等）に用意しており、条件部で状況認識を行ない、処理部で候補地を生成する形式となっている。
- (13) 下位階層用制約評価オブジェクト： 生成された配置候補地が、与えられた下位階層用制約条件を満たしているかどうかを評価する。

- (14) スライド調整オブジェクト：既に配置された機器に対して、許容範囲内での移動操作を行ない、機器配置スペースを確保する。
- (15) 形状処理オブジェクト：形状演算を行なう。
- (16) 機器・仮想コネクタ間配線オブジェクト：ゾーン内の機器間、機器・仮想コネクタ間及び分電盤等との配線を行なう。配線の重なりを判定したり障害物を避けたりしながら信号及び電源ケーブルを配線し、配置配線状況記憶オブジェクトに記録する。

3. 2 ESPオブジェクトの動作

(1) ユーザに処理を依頼された並列問題解決オブジェクトは、問題定義オブジェクトからレイアウトに必要な諸データを取得する。先ず上位配置処理オブジェクトにゾーン生成処理を依頼し、次に並列配置プロセス制御オブジェクトに各下位配置配線オブジェクトによる配置処理の並列制御を依頼する。

(2) 上位配置処理オブジェクトでは、先ず全ての配置機器のグループ分けを行ない、グループごとのゾーンを上位階層用配置ルールに用いて生成する。この際、形状処理オブジェクトに処理を依頼することにより、ゾーン面積推定値を拘束としつつ、各ゾーンの絶対位置と形状を求め、これを配置配線状況記憶オブジェクトに記憶する。上位階層用制約評価オブジェクトは、生成された各ゾーンを上位階層用制約条件に照らし合わせて評価し、上位階層用制約条件を充たしている案の中で正方形に近いゾーンを採用する。

以上で上位配置処理は終了し、再び並列問題解決オブジェクトに制御が戻り、次に上位配線処理オブジェクトに起動をかける。

(3) 上位配線処理オブジェクトは問題定義オブジェクトから取得した機器間配線データと上位配置処理の結果とから、ゾーンをまたがる配線を調べ仮想コネクタを導入することによって配線問題を分割する。結果として機器間の配線ネットごとに仮想コネクタや線分のオブジェクトが作成される。但し、詳細配線位置は未決定である。再び並列問題解決オブジェクトに制御が戻り、上位配置・配線計画は終了する。

(4) 並列問題解決オブジェクトでは、他の機器配置の基準となるコンソール・ディスプレイ(CD)と中央制御装置(CPU)の、対話形式での配置を行う。

次に並列問題解決オブジェクトから、並列配置プロセス制御オブジェクトを起動する。

(5) 並列配置プロセス制御オブジェクトでは、ゾーンに対応した下位配置配線オブジェ

クトに順次起動をかけてゆく。起動をかけられた各下位配置配線オブジェクトは、下位配置処理の準備ができた時点で、並列プロセス制御オブジェクトにストリーム機能を使って、各プロセス識別子を含んだ準備完了メッセージを返す。

並列配置プロセス制御オブジェクトは、すべての下位配置配線オブジェクトの実行準備完了まで同期をとる。すべてのプロセスが準備完了になったならば、ストリーム機能を使って全プロセスすなわち全下位配置配線オブジェクトに対して、配置実行開始のメッセージを送る。

(6) 並列プロセス制御オブジェクトに処理を依頼された各下位配置配線オブジェクトは、事前に定められた配置順序に従い、ゾーン内に一つずつ機器を配置してゆく。

下位配置配線オブジェクトは、現在の配置状況に応じた機器の配置候補地（一般に複数個）を、下位階層用ルール・オブジェクトを用いて生成する。次に下位階層用制約評価オブジェクトに配置候補地を評価するよう依頼する。

(7) 下位階層用制約評価オブジェクトは、生成された候補地を下位階層用制約条件に照らしあわせて評価する。下位階層用制約条件には、ゾーン内制約（2.1節 c1,c2,c4,c6）とゾーン間制約（同 c2,c4,c5）が存在する。

下位階層用制約評価オブジェクトは、各々の制約条件に対して、配置状況記憶オブジェクトを参照し、形状処理オブジェクトを使用して、候補地を評価する。評価の結果、制約条件を満たすものがあればその配置案を採用し、配置配線状況オブジェクトを更新する。無い場合にはスライド調整オブジェクトに配置機器のスペース確保を依頼する。

(8) スライド調整オブジェクトは、配置配線状況記憶オブジェクトを参照して許容範囲内での機器移動を試み、該当機器の配置を行なう。配置できたならば、配置状況記憶オブジェクトを変更して終了する。

各ゾーンでの配置が終了すると、並列問題解決オブジェクトに制御が戻り、並列配線プロセス制御オブジェクトを起動し、配線問題の処理に入る。

(9) 並列配線プロセス制御オブジェクトでは、最初に各ゾーンの仮想コネクタ位置を決める。続いて各配線プロセスを並列実行させる。各下位配置配線オブジェクトは、ゾーン内の配線を機器・仮想コネクタ間配線オブジェクトに依頼し、配線や障害物の重なりをチェックしながら信号ケーブル線と電源ケーブル線の配線を作成する。全配線プロセスのゾーン内配線が終ると並列問題解決オブジェクトに制御が戻り、再び上位配線処理オブジェ

クトに起動をかける。

(10) 上位配線処理オブジェクトでは、仮想コネクタ間のゾーン間配線を行ない、配線を終了する。

以上のように本パイロット・モデルでは、レイアウト問題は並列処理によって解かれる。

3. 3 実験結果の一例

(1) 配置図：全機器の配置が完了した状況を図3.2上部に示す。ゾーン内での機器配置制約、ゾーン間での機器のCD方式への向きや、LP、KPとCDの距離、LPとFDの距離等の制約を満足した配置となっている。

(2) 配線図：各ゾーン内で並列配線処理を実施し、その後仮想コネクタ間配線によってゾーン間配線を完了した状況を図3.2下部に示す。実線は信号ケーブルを、破線は電源ケーブルを示している。

(3) 結果の考察：配置配線結果は機器間の意味的配置制約条件を満足したものになっている。配置に関して評価すると、DISKやMTなど外部ユーザのアクセス頻度の低い機器グループは出入口から遠い所に置かれ、KPRやLPなどアクセス頻度の高いものは出入口の近くに置かれている。各機器は運転員の着座するCDの方向を向いている。機器の前面線が揃って通路が確保されており、運転員がアクセスしやすくなっている。スペース余裕のある場所では機器の保守エリア同士も重ならぬように配置し、スペース余裕の無い場所では保守エリア同士を重ねるが本体と保守エリアの重なりは防止して、保守に支障をきたさぬよう柔軟な対処をしている。配線についても、相互の重なり防止、ケーブル端子と配線の重なり防止が配慮されている。さらに、類似系統のケーブルを束ねて配線することにより見易い配線を与えている。

4. CADソフトウェア開発におけるESPの意義

4. 1 開発経過

パイロットモデルは次の4ステップを経て段階的に開発した。

(1) step1 (小規模な部屋への機器配置問題解決)：ESPオブジェクト体系を図4.1(a)に示す。

(2) step 2 (比較的大規模な部屋への機器配置) : 機器グループを部屋のゾーンへ割り当てることで問題を部分問題に分割する上位階層と、機器の配置を行なう下位階層から成る階層型問題解決。ESPオブジェクト体系を図4. 1 (b)に示す。問題の規模を拡張するため、問題定義オブジェクトを変更した。上位階層、下位階層と処理を階層化するため問題解決オブジェクトの制御機能を変更した。上位階層処理として部屋のゾーン分割、機器グループの割り付け等で原問題の部分問題化を担当する上位階層処理オブジェクト、上位階層用ルールオブジェクト、上位階層用制約評価オブジェクトを新規に開発した。またstep 1の配置処理オブジェクトに代るものとして、各部分問題の配置処理を制御する下位階層処理オブジェクトを開発した。配置状況記憶機能として新たに部屋の中にゾーンを作るため、状況記憶オブジェクトを変更した。その他はstep 1で開発したオブジェクトをそのまま利用した。下位階層用ルールオブジェクトはstep 1の配置候補地ルールオブジェクト、下位階層用制約評価オブジェクトはstep 1の配置候補地制約評価オブジェクトである。

(3) step 3 (並列プロセス機能を使用して下位階層処理を並列に行なう並列配置問題解決) : 説明は略す。

(4) step 4 (機器間の配線処理も並列に行なう並列配置・配線問題解決) : 説明は略す。

4. 2 段階的開発におけるESPの貢献

step 1～step 4への段階的開発はESP言語の持つモジュラリティ、オブジェクト継承機能等の特徴により効果的に実施できた。ここでは変更の一例を説明する。

(1) 問題定義の拡張 : パイロットモデルが対象としている問題自体が、step 2での部屋規模の拡大に伴う機器、柱等の追加、step 4での配線機能の拡張と、計2回変更されている。本パイロットモデルでは、部屋、機器、柱、出入口のように問題を構成している物をひとつのESPオブジェクトとして表現し、オブジェクトの集合で定義している。従って上記2回の変更に対し、基本的にはオブジェクトの追加で対応できた。step 2での拡張に際してのESPオブジェクトの追加イメージを図4. 2 (a)に示す。機器や柱等のインスタンスオブジェクトの追加を行なった。

(2) ゾーン概念の追加 : step 2での階層型問題解決方式への移行により、問題を

部分問題へ分割することから、部屋の中のゾーンという概念を状況記憶オブジェクトの一部として導入することになった。ゾーンには部屋、機器、柱等のオブジェクトと同様の形状構造データを記憶する機能が必要であった。そこで図4. 2 (b)に示すように、構造データ記憶機能を持っているオブジェクトを継承するようにゾーンクラスを定義した。このようにクラスオブジェクトの継承機能を活用することで、既に開発していた構造データ記憶機能を使用することができ、この機能を実現するプログラムを作成する必要はなかった。

5. オブジェクト指向のデータベースに関する議論の現状と

本パイロットモデルの関連に関する考察

5. 1 オブジェクト指向のデータベースに関する現在の議論

オブジェクト指向のデータベース（以下OOD）はホットなテーマとして現在議論が活発である。その理由として次のことが考えられる。一つは、OODの定義が明確でないため、早くその枠組みをはっきりさせたいという願望である。もう一つは、データベースが扱おうとしているデータの形態が複雑になってきており、従来からの静的なデータ構造の表現方法だけでは扱いが困難であり、新しい方法が必要になっていることである。

データベースの分野でオブジェクト指向の考え方が議論されるようになった経緯は次のようであると考えられる。プログラム言語の世界では、1970年代からsmalltalkに代表されるデータと手続きを一体としたオブジェクト指向の言語が着実に発達してきた。¹⁾ これらの言語の基本的な要素は(1)クラスとインスタンス(2)データと手続きを一体とした情報隠蔽(3)継承、オーバーライディング、レートバインディング²⁾(4)メッセージパッシング、オーバーローディング³⁾、である。一方、データベースの世界では1980年代よりマルチメディアデータベースへの関心が急速に高まっていた。それは、従来の文字・数値を扱うデータベースから、図形、画像、動画、音声等の多種類のデータを統合管理するシステムのニーズが増大してきたためである。^{4) 5) 6) 7)} 統合管理するのに重要な点として次の二点が主な関心事となった。

(1) 多種類のデータをまとめて管理できる方法⁸⁾ : たとえば、文書の各ページには、種々のデータが含まれるため、ページ単位の管理が必要な場合には何らかの新しい管理方

法が必要である。その解決策として、オブジェクトという統一的な単位で管理できないかと考え、オブジェクト指向に注目することとなった。

(2) 多種類の個々のデータを一貫した方法で管理できる方法^{*)}： 従来のデータベースのように文字・数値だけの場合には扱いも単純であるが、多種類のデータを扱うと、処理が複雑になる。データの表示を考えても、データの種類に従って必要な手続きを実行させねばならない。データ毎に異なる手続き要する点を何とか解決できないかと考えたことが、手続きまで一体化するオブジェクト指向に注目することとなった。

上記の(2)ではオブジェクト指向言語の研究成果が直接利用できる可能性がある。上記(1)については、別な観点からの検討が必要であり、複合オブジェクトと呼ばれる考え方が提案されている。^{*)} オブジェクト指向言語の基本的な要素以外にデータを管理するというデータベースの立場から見たとき必要と考えられる基本的な要素は以下のようものである。^{*)} ^{*)} ^{*)}

- (1) 複合オブジェクト： 複数オブジェクトの合成による新オブジェクト作成機能^{*)}
- (2) 大量データ高速処理： 大量データから必要なデータを高速に読み出す仕組み
- (3) 永続性の処理： 実行結果データ(オブジェクト)の保存機能^{*)}
- (4) データの一貫性： 更新時のオブジェクト間データの一貫性管理
- (5) 同時制御： 複数ユーザからの同時更新に対する処理機能
- (6) 機密保護： ユーザ毎のアクセス許可データ(オブジェクト)の制限機能
- (7) データ構造記述： オブジェクト内に記述するデータの構造記述機能

以上から、OODの主要な要素あるいは特徴は下記である。

- (1) クラスとインスタンス
- (2) データと手続きを一体とした情報隠蔽
- (3) 継承、オーバーライディング、レートバインディング
- (4) メッセージパッシング、オーバーローディング
- (5) 複合オブジェクト
- (6) 大量データ高速処理
- (7) 永続性処理
- (8) データ一貫性
- (9) 同時制御

(10) 機密保護

上記の中には同時実現が困難な要素もある。情報隠蔽と大量データ操作、メッセージパッシングと大量データ操作である。データベースでは大量データから目的のデータを読み出すために通常インデックスを使用する。しかしこの方法はオブジェクトの中を見ながら、目的のオブジェクトを探し出すという方法であり、情報隠蔽の考えとは合いにくい。メッセージパッシングは各オブジェクトにメッセージを送って必要な処理を行なってもらう方法である。大量オブジェクトの中から必要な内容をもつオブジェクトを探し出す場合は難しい。大量オブジェクトを扱うためにはその実現方式の研究も必要である。またOODに対する形式化の研究も今後の課題である。⁸

以上OODに関する概要を述べたが、本稿で取り上げたCADシステムの事例に基づき、OODの基本的要素の必要性を次に考察する。

5. 2 オブジェクト指向のデータベースの基本的要素と本事例との関連

- (1) クラスとインスタンス： インスタンスは同種データの作成と利用に有効である。メモリエリアの事前確保は不要であり、本事例でも形状物の定義などに十分活用できた。
- (2) データと手続きを一体とした情報隠蔽： 本事例ではプログラムの段階的開発を行なった。このような開発ではデータや手続きの変更は不可避である。モジュラリティの高いオブジェクト指向概念はプログラム変更時の他への影響を少なくし大変効果的であった。
- (3) 継承、オーバーライディング、レートバイディング： 同じ形状物、たとえば、部屋内のゾーンや機器についての共通処理を上位のメソッドとして定義する等、効果的に使用できた。
- (4) メッセージパッシング、オーバーローディング： 全ての処理はメッセージパッシングを基本にして実現している。プログラムのモジュラリティが高まり、変更は容易であったが、反面問題個有のデータ構造を実現できない点や、多種のオブジェクトのメッセージ定義をプログラマ側で把握する必要性のある点は問題であると考えられる。
- (5) 複合オブジェクト、オブジェクトアイデンティティ： 前面縁をそろえた機器群の群単位での移動機能などを必要に応じて実現し、利用している。このような群についての複合オブジェクト定義と操作がより自然に行なえると便利である。
- (6) 大量データ高速処理： インスタンスは、レイアウト処理の中間生成物も含め大量

に発生する。本事例は部屋への機器のレイアウト問題であるが、他の問題（たとえばLSIレイアウト等）では極端に量が増える可能性があり、高速処理の必要性が高い。

（7）永続性の処理： 本事例では、ESP言語のOSであるSIMPOS機能を利用して、中間状態や最終状態のインスタンスを保存して再演算を行なえるようにしている。永続性処理機能は必要性の高いものといえる。

（8）データ一貫性： 本実験では、特に必要性は感じられなかった。

（9）同時制御： 本事例では並列協調操作（ゾーン間で並列処理を行なうが、相手の配置に合わせて自分の配置を決める処理）があるため、同時制御の問題が発生した。今回の実験では、単純な先処理優先方式でこの問題を回避したが、一般的には必要性のある機能である。

（10）機密保護： 本実験では必要性は感じられなかった。

（11）データ構造記述： 本実験では幾何構造を線分の結合で表わしている。処理で扱う単位構造は複数の線分で表現することになるが、この単位構造を複数オブジェクトで記述している。単位構造をごく自然に一つのオブジェクトのデータとして表現できるデータ構造の記述方法が望まれる。

上記のうち、（1）～（4）は、オブジェクト指向プログラミングで議論されている内容であり、（5）～（11）は主としてOODで議論されている内容である。これらの内容は互いに関連性が強い。全体を統合したシステム環境の開発が必要であると考えられる。²⁹

6. おわりに

ICOTで開発されたオブジェクト指向論理型言語ESPの上に実現したレイアウト問題解決パイロットモデルを紹介し、最近話題となっているオブジェクト指向のデータベースの基本要素との関連性について考察した。事例で示したようにCADプログラム開発や保守の面で、オブジェクト指向言語は有効な諸側面を持っている。反面、本格的CADの実現には性能面でのさらなる進歩が望まれる。一方、複合オブジェクト処理や永続性など、OOD領域で話題となっている事柄はCADについても重要であると考えられる。オブジェクト指向言語とデータベースがシステムとして統合化して行くことを期待したい。

謝辞

本研究は第五世代コンピュータプロジェクトの一環として(財)新世代コンピュータ技術開発機構 (ICOT) からの委託により行ったものである。パイロットモデル開発に際して御指導いただいた ICOT 生駒憲治第5室長およびオブジェクト指向のDBに関して議論していただいているDOO・WGの方々に感謝します。

参考文献

- 1) 渡辺： 計画/設計型知識システムの事例分析，計測自動制御学会講習会エキスパートシステム，方法論と応用テキスト，67/82 (1987)
- 2) T. Chikayama: ESP as Preliminary Kernel Language of Fifth Generation Computers, ICOT Research Center TECHNICAL REPORT, TR-005 (1983)
- 3) T. Andrews, et al: Combining Language and Database Advances in an object-oriented Development Environment, Proc. of the 1987 ACM SIGMOD Conf. on OOPSLA, 430/440 (1987)
- 4) A. Goldberg and D. Robinson: Smalltalk-80, Language and its Implementation, Addison-Wesley (1983)
- 5) D. Woelk, et al: An Object-Oriented Approach to Multimedia Database, Proc. of the 1983 ACM SIGMOD Conf, 311/325 (1983)
- 6) G. Altair: O, an Object-Oriented Data Model, Proc. of the 1988 ACM SIGMOD Conf, 424/433 (1988)
- 7) 田中： オブジェクト指向データベース・システム—その背景と概念，bit, VOL.20, NO.6, 83/90 (1988)
- 8) 横田： オブジェクトの形式化とスキーム，信学技報，DE研究会, (1988)
- 9) Y. Tanaka and T. Imataki: IntellingentPad, 「アドバンスデータベースシステム」シンポジウム, 11/20 (1988)
- 10) 近藤・辻・高志・花塚：エキスパートシステム構築ツールES/X90 (7)：-

知識ベース管理機能, 情報処理学会第35回全国大会講演論文集(II), 1745/1746
(1987)

- 11) 近藤・大町・小口・高志: オブジェクト指向データの特徴と記憶・利用方式, 信学技報, DE研究会, VOL.87, NO.98, 1/8(1987)
- 12) 増永: オブジェクト指向マルチメディアデータモデル, 信学技報, DE研究会, VOL.87, NO.98, 25/32(1987)
- 13) 久野: 型階層をもつオブジェクト指向言語, 情報処理学会, VOL.29, No4, 318/324(1988)
- 14) 佐藤: オブジェクト指向アプローチによるマルチメディアDBの研究動向, 「マルチメディア通信と分散処理」シンポジウム, 21/30(1988)
- 15) 加藤: パターン情報処理指向のマルチメディアデータベースシステム, 信学技報, DE研究会, VOL.87, NO.98, 17/24(1987)
- 16) 特集: マルチメディアデータベース, 情報処理学会, VOL.28, NO.4(1987)
- 17) 特集: オブジェクト指向プログラミング, 情報処理学会, VOL.29, NO.4(1989)
- 18) 田中: Post Scriptと関係データベースによる、マルチメディア・データベース・オブジェクトの格納と操作, 情報処理学会, データベース・システム研究会, 1/7(1988)
- 19) 小島: オブジェクト指向データベース(＋マルチメディアデータベース, 日本ソフトウェア科学会サマー・チュートリアル「最近のデータベース理論」資料, 115/132(1988)
- 20) G.Copeland and D.Maier: Making Smalltalk a Database System, Proc. of the 1984 ACM SIGMOD Conf, 316/325(1984)

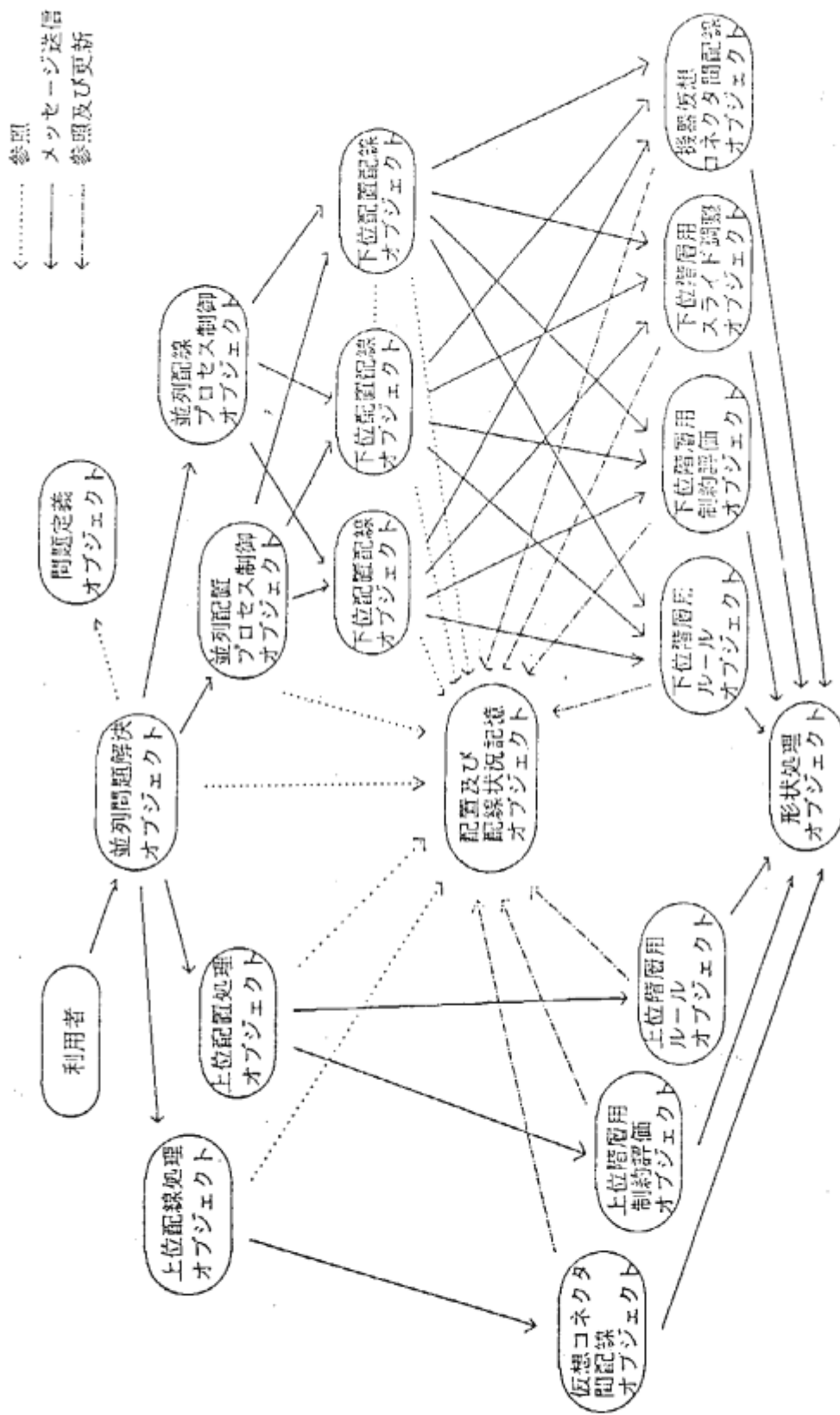
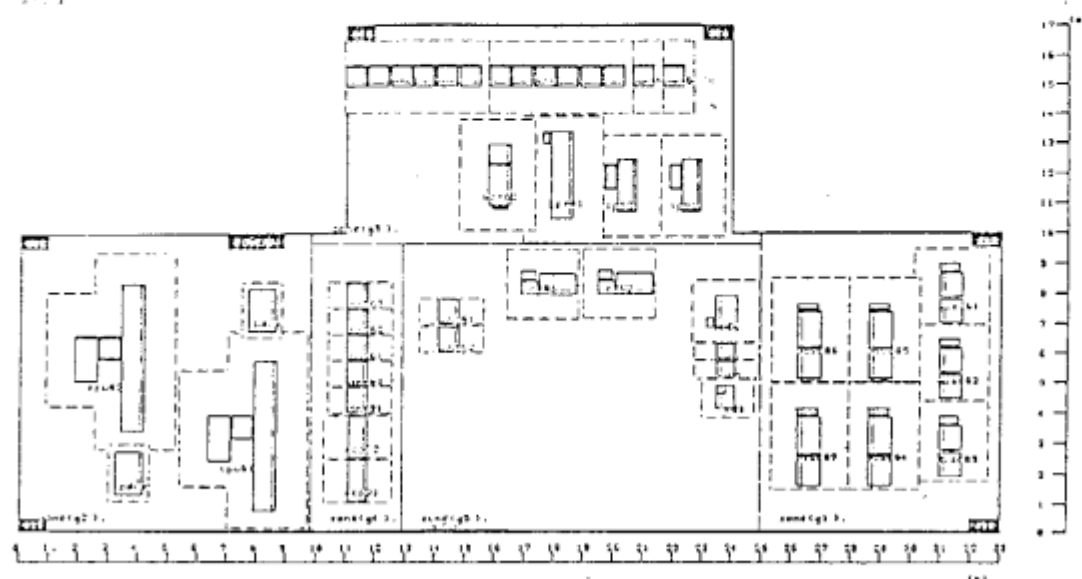


図 3. 1 ESPオブジェクト体系図

Computer Room Layout Expert

Fig. 1



Computer Room Layout Expert

Fig. 2

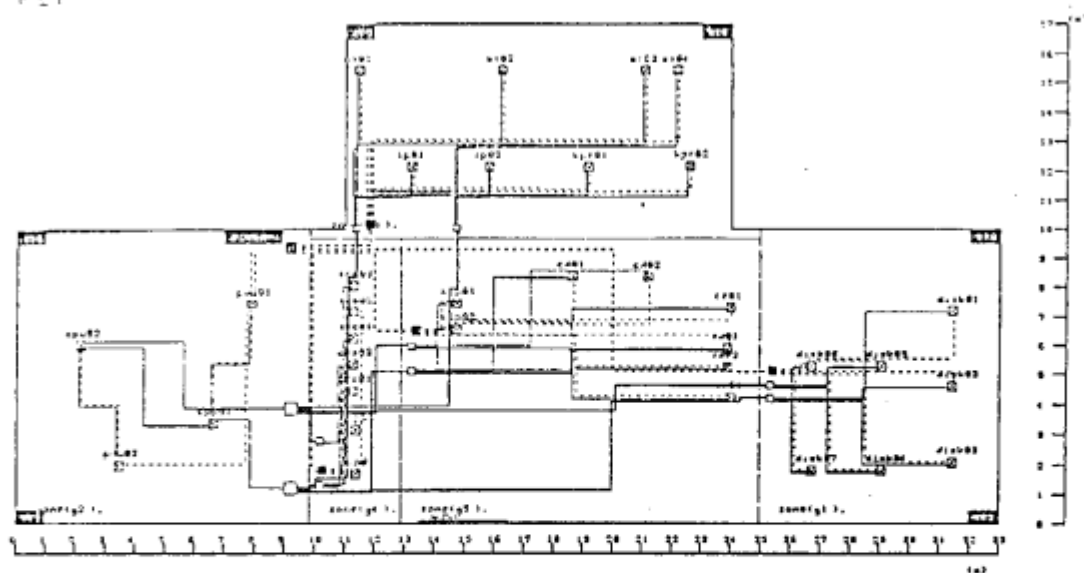
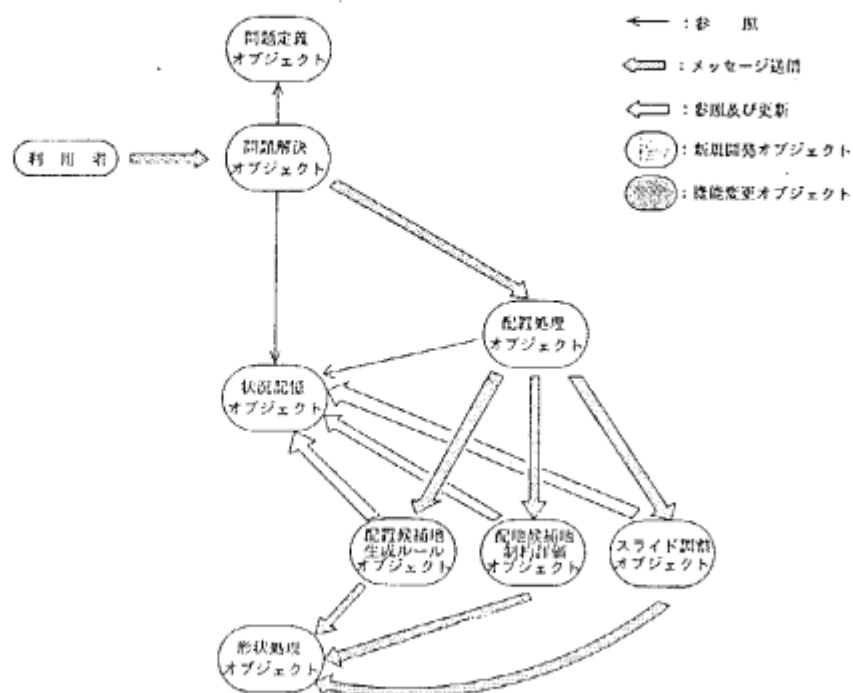
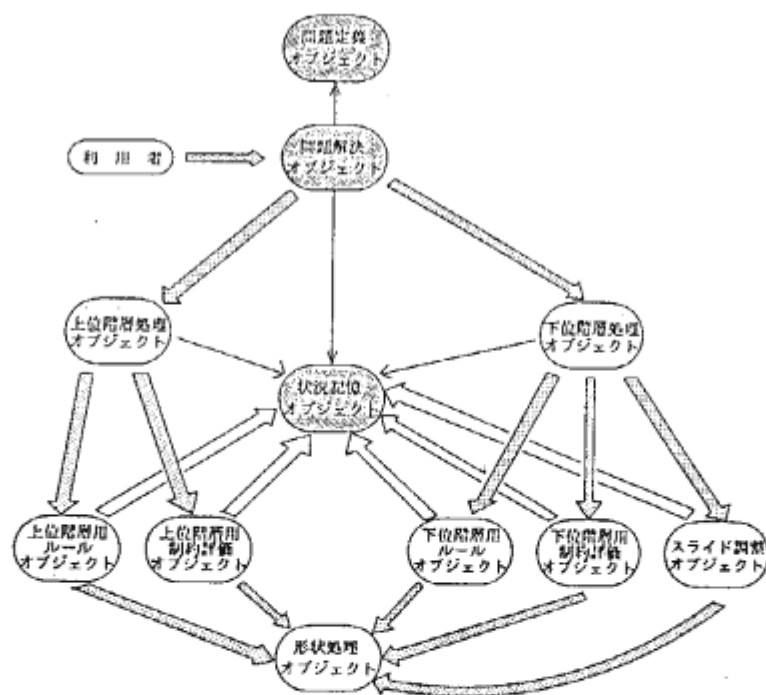


図 3. 2 レイアウト結果

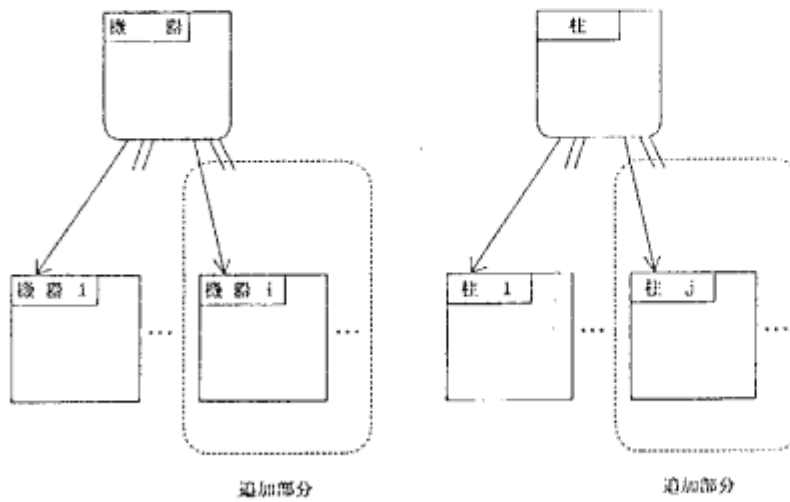
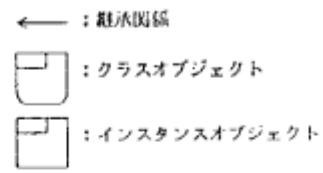


(a) step 1

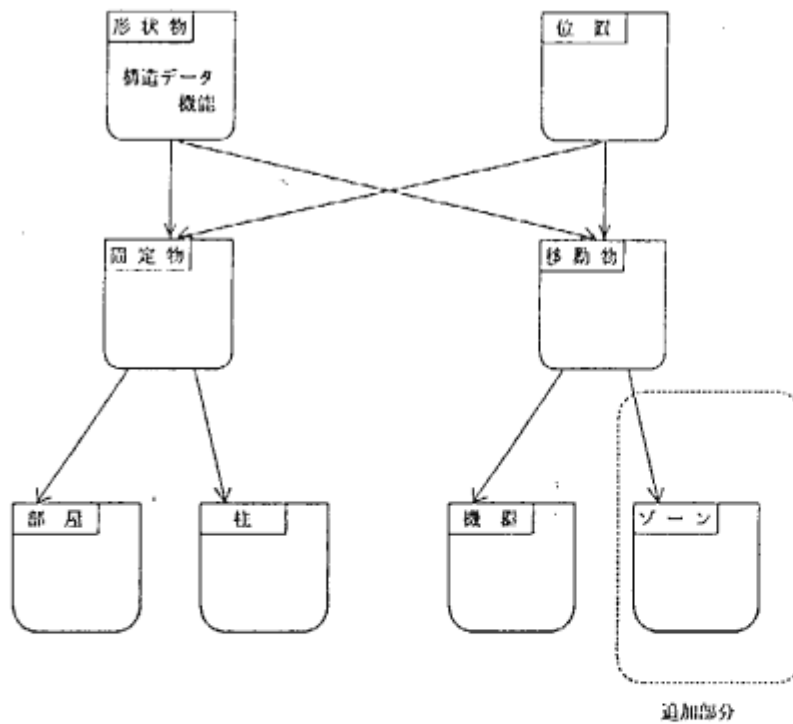


(b) step 2

図4.1 ESPオブジェクト体系の変更例



(a) 問題定義の拡張 (step 2)



(b) ゾーン概念の追加 (step 2)

図4.2 ESPオブジェクトの変更例