

〔非公開〕

TR-C-0014

概念図作成支援システムの試作

佐藤 亮一
RYOICHI SATO

1988. 8. 31.

A T R 通信システム研究所

概念図作成支援システムの試作

佐藤 亮一

ATR通信システム研究所

1. はじめに

図は言語や数式と並ぶ重要なコミュニケーションの道具であり、今日、情報量の飛躍的増大・情報相互の関係複雑化等が進む中で、視認性に優れた図の重要性はますます増大している。しかしながら、これまでの図についての研究は十分とは言えず、またコミュニケーションに用いられている種々のシステムも、図は言語ほど十分な取り組みが行われていない。

本研究はこうした状況に鑑み、コミュニケーションにおける図の役割を再認識し、図が意味を媒介するメカニズムを解明することを目的としている。特に図の言語的規則性を明らかにし、図の認識・生成系モデルを計算機上に実現することが目標である。このための一つのステップとしてこれまで概念図作成支援システムについて検討を行ってきた。本稿ではこれを中心に報告する。

2. 図に関する考察

概念図作成支援システムについて論じる前に、図に関してその位置付け・機能・特質・分類について以下に考察を試みる（大部分は[1]を参照した）。

2-1. 図の位置付け

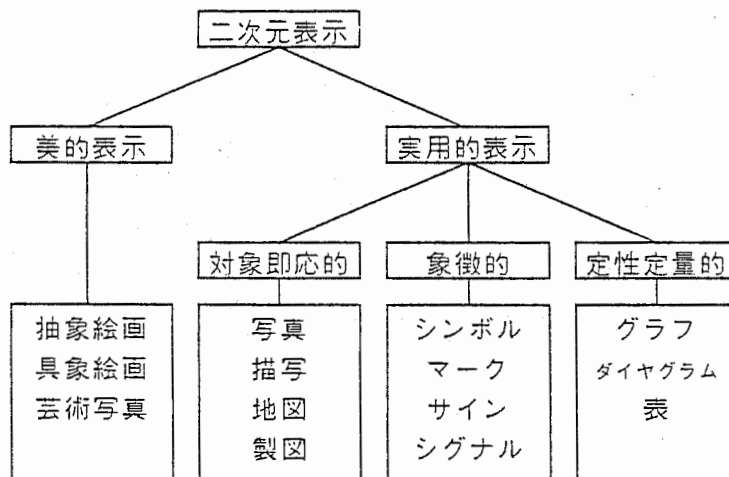
図は第3の言葉である。

- 第1の言葉……文章
- 第2の言葉……数式
- 第3の言葉……図

これらのうち、図を文章や数式と大きく隔てているのは、表現形式が一次元的か二次元的かの違いによる。

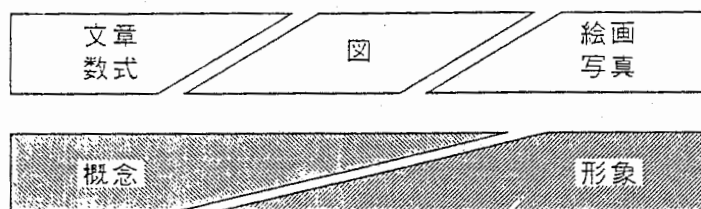
- 一次元表示……文章, 数式
 - ・ 走査的読み取りを行う（部分的読み取り→全体の把握）
- 二次元表示……図, 絵
 - ・ 全体的読み取りを行う（全体の把握→部分的読み取り）

即ち、図などの二次元表示は『全体の把握のしやすさ』によって特徴付けられる。これら二次元表示は[fig. 2-1]に示すように分類される。



[fig. 2-1] 二次元表示の分類

媒介する情報の内容でみると、文章・数式は概念表現に、絵画・写真は形象表現に向き、図は両者の中間である([fig.2-2])。



[fig.2-2]図の性格

2-2. 図の機能

図は情報を媒介する外部表現の一手段である。その基本的機能は『情報伝達』と『思考整理』である。情報には送り手と受け手が存在し、『情報伝達』は両者が異なる場合、『思考整理』は両者が同一の場合、と見ることができる。

2-3. 図の特質

(1) 二次元情報である

図は面に対し読み書きが行われる。これに対し言語は一次元的に読み書き聞き話すが行われる。

(2) 記述的(視覚的)である

図はその性質上書かれて初めて意味を持つ。言語のように音声による表現は不可能である。

(3) 概形に大まかな意味を含む

図はその概形から意味内容をある程度判断することができる。言語は記述する際のレイアウトに内容を含ませるのは困難であるが、発音される場合の声の調子で大まかな内容を表現することは可能である。

(4) 一目で大まかな意味が判る

図は二次元的に記述され、全体と部分の関係が直感できるが、言語は頭の中に全体と部分の関係を整理しながら理解をすすめなければならない。

(5) 関係表現に向く

図は全体の構造が概観できるため、全体と部分、あるいは部分同士の関係が捉えやすい。この点、言語に比べて大幅な思考の省略が可能である。

(6) 大量の情報を凝縮させることができる

大量のデータは図化することによって相互関係を捉え易くすることができる。これにより大きな視点から全体を概観・把握することが可能となる。

(7) 情報伝達と思考整理に用いられる

言語は専ら情報伝達や記録に用いられるが、図は思考整理にも積極的に活用される。

(8) 要素記号と関係記号から構成される

図は意味要素を表す要素記号と、それら相互の関係を表す関係記号から成る。これらは言語における単語にほぼ相当する。

(9) 記号配列には規則がある

言語には単語の厳密な配列規則を定める文法があり、図にも記号の配列規則があるが、後者は言語のそれに比べ厳密性は低い。

(10) 関係表現の方法により図の形式が異なる

図の構造は要素間関係をどのような構造で表現するかによって決まる。

(11)表現・解釈の自由度が大きい

言語による表現や解釈は文法に一定の規制を受けるが、図の場合は規則が厳密でないために表現や解釈の規制が緩い。

(12)言語によって補完される

図の意図を明確にするため、図には言語による標題や注が付けられる。また図の要素記号や関係記号そのものにもしばしば言語が用いられる。

(13)詳細記述には向かない

図は全体把握に適し、詳細な記述には言語や他図の助けを必要とする。

(14)高度な抽象概念記述には向かない

高度な抽象概念（『假定』『矛盾』『思想』『意味』等）の表現は言語において勝っており、図による表現は非常に困難である。

(15)美的・認知的・記号的という三種の評価基準を持つ

図は①見た目の美しさ（きれいさ・美的基準）、②関係の捉えやすさ（見やすさ・認知的基準）、③意図の捉えやすさ（判りやすさ・記号的基準）、により評価される。

2-4. 図の分類

図は形態から連結系、配列系、領域系、座標系、形象系の5つに分類できる。これらの違いは特に関係記号の用いかに顕著に現れる。

● 連結系の図

- ・要素記号と関係記号の区別が明確なもの
- ・要素間の関係明示を本質的な図の意図とする
- ・テンポラリな要素フィールドの設定

● 配列系の図

- ・明示的な関係記号を使わず、空間的配置でその代わりに果たす
- ・関係記号の相互関係が空間的配置によって縛られる
- ・テンポラリな関係フィールドの設定

● 領域系の図

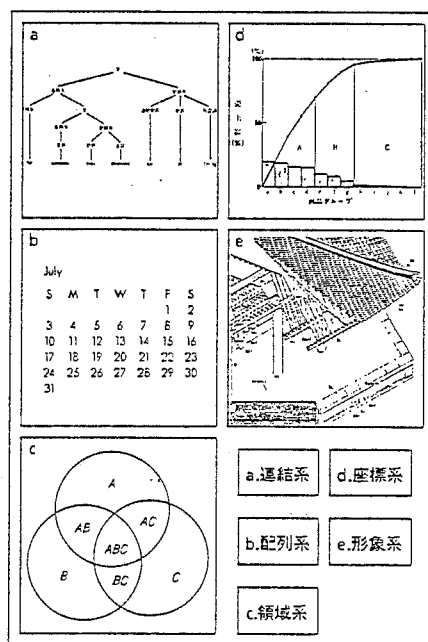
- ・要素記号が拡張され、関係記号の役割を兼ねたもの
- ・要素提示を主とし、要素間の相互関係は比較的単純
- ・要素フィールドの設定

● 座標系の図

- ・関係記号を座標上に配置したもの
- ・位置的に意味付けされた空間に関係記号を配置したもの
- ・関係フィールドの設定

● 形象系の図

- ・実態に即した図、スケッチ
- ・基本的には投影図



[fig.2-3]形態による図の分類

2-5. 本研究が対象とする図

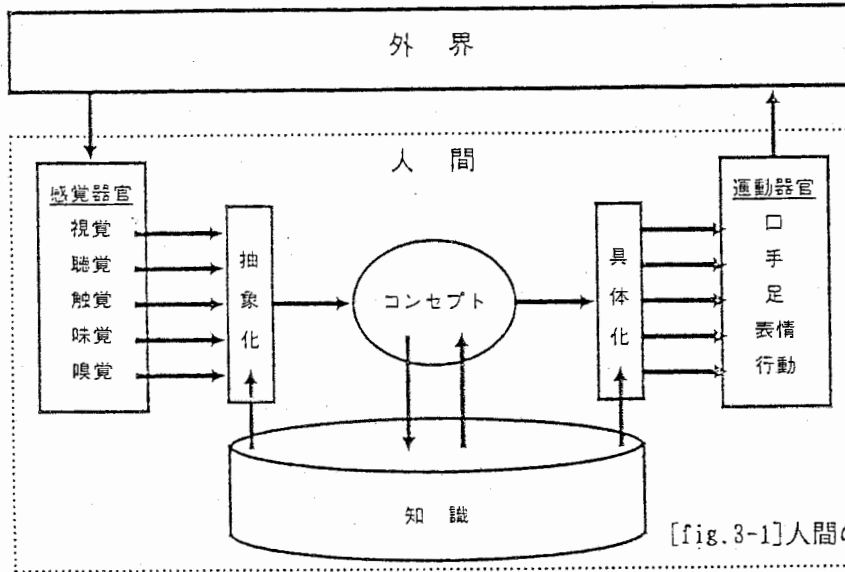
本研究では定性的な図を対象としており、特に“定性的な概念を正確に伝達することを目的として作成された図”を概念図と呼んでいる。これに属する図は、形態の分類に従えば連結系・配列系・領域系に属する図である。この中でもとりわけ連結系の図は要素記号と関係記号の区別が明確であり、本研究は連結系の図を中心に検討を行った。

3. 概念図とその作成支援

図の研究を行うに当たり、まず人間の情報処理モデルを考え、これを図に適用するアプローチを取った。このモデルは大きく認識系と生成系に分かれるが、特に生成系を取りあげ、概念図作成支援を行う上での問題点、実現法について述べる。また、図の構造は要素間関係の表現方法と密接に関係することから、要素間関係について考察を行う。

3-1. 情報処理モデル

人間の情報処理は[fig.3-1]のようなモデルで表現することができる。このモデルは人間の持つ概念(=コンセプト)を中心に外界からの抽象化過程、外界への具体化過程、及び知識から成っている。この図の『概念』は「人間の行動・思考の中心となっている内的情報」と言い表すことができる。

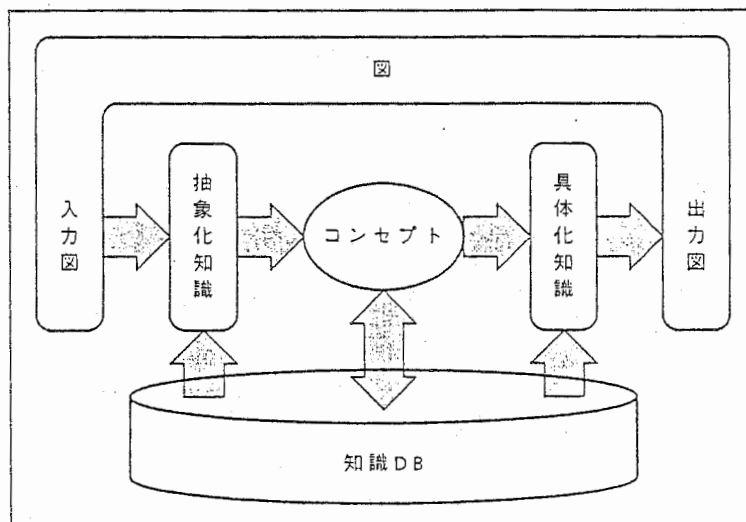


[fig.3-1]人間のコミュニケーションモデル

人間の情報処理は概念を中心に次のように行われる。

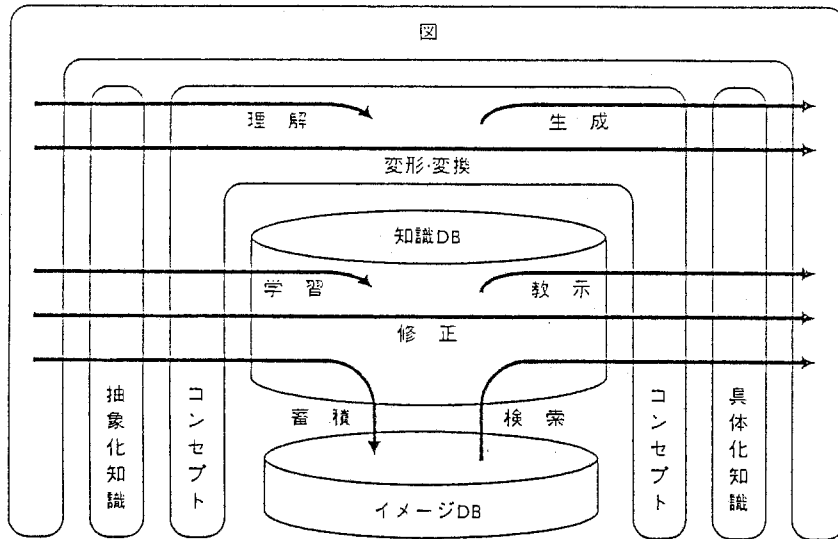
- ①外界の情報は感覚器官により取り込まれ、抽象化され、概念が形作られる
- ②概念は具体化され、運動器官により外界に表現される
- ③抽象化や具体化の過程は知識によって制御される
- ④概念は外界との抽象化・具体化の繰り返し(フィードバック)で成長する
- ⑤形作られた概念は知識に蓄えられ、また知識により修正される

この仕組みを図に応用すると[fig.3-2]のように描ける。ここでは、概念(図中では“コンセプト”)は図の意味記述、抽象化知識は図の認識系、具体化知識は図の生成系と考えることができる。



[fig.3-2]図の処理モデル

このモデルを計算機に取り込むことにより、知的図処理システムが実現できる。[fig.3-3]は上記のモデルにイメージDBを加え、知的図処理機能を処理の流れごとに描き表したものである。



[fig.3-3]知的図処理システム

このような知的図処理システムは、①明確な意図表現を目的とした、②定性的概念を表現する図、に馴染みやすいと考えられ、これらを特に概念図と呼ぶことにする。本研究では概念図を作成支援するシステム（[fig.3-3]では『生成』を支援するシステム）について検討と試作を行った。

3-2. 概念図作成支援の問題点

図の重要性は

- ①情報媒体として文章や数式と補い合う関係にある
- ②大量の情報をうまく圧縮できる
- ③複雑な関係を視覚的に簡潔に表現できる

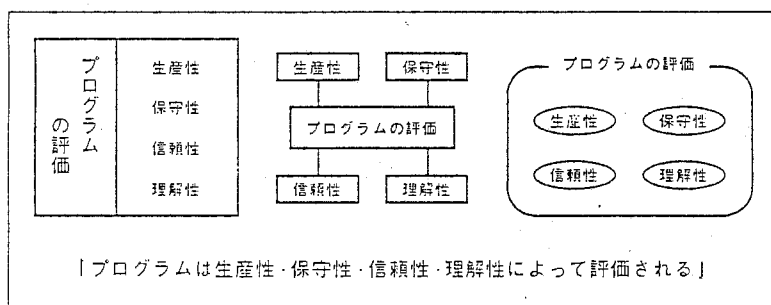
などにより明らかであるが、問題もある。情報媒体は

- a. 表記の的確さ
- b. 見やすさ
- c. 理解しやすさ
- d. 表記しやすさ
- e. 編集しやすさ

の5つにより評価できるとされるが[6]、図においては特にdとeが問題となる。これは図が二次元表現であり、描かれて初めて情報を媒介できることからくる本質的な問題である。概念図作成支援システムは計算機支援によって、図の持つ本質的な問題をカバーしようとする試みである。

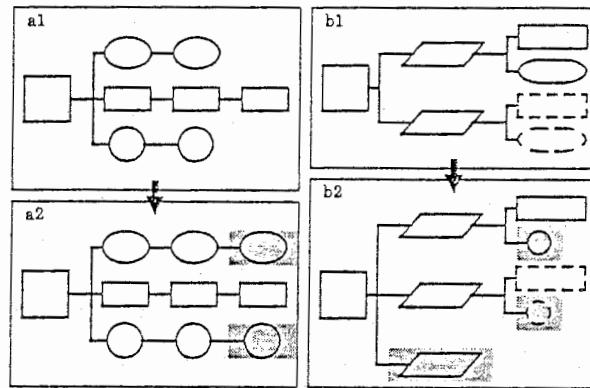
概念図作成支援を行う上で解決すべき課題となるものに以下のようなものがある。

- 意味と図は一意に対応しない（[fig.3-4]）
 - ・一つの意味からいくらかでも形の異なる図が描ける



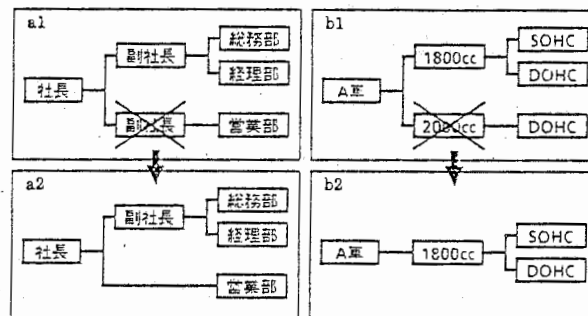
[fig.3-4]同じ意味を表現する異形図

- 挿入される要素の形は周囲の状況に影響を受ける ([fig.3-5])
 - a1→a2では挿入された要素は直列的な形状の統一が行われている (楕円と円の要素)
 - b1→b2では並列的な形状の統一が行われている (平行四辺形の要素)



[fig.3-5]要素の挿入と修正

- 図には“同じような”操作が多い ([fig.3-5])
 - [fig.3-5]b1→b2で楕円の要素が二つとも円になっている
 - しかしながら、実線・破線という属性は変更されない
- 要素の削除は要素間関係により処理が異なる ([fig.3-6])
 - a1,b1はともに図の概形は同じ
 - 同じ位置の要素を削除しても内容により結果はa2,b2のように異なる



[fig.3-6]要素の削除

3-3. 概念図作成支援システム

概念図作成支援の目的は、①作図・修正の手間の省力化、②新しいMMIの創造、ということである。このためには従来ワープロ等が提供していたような図形編集機能 ([fig.3-7]) では不十分である。

基本図形	水平線、垂直線、斜線、矩形、多角系、折線、 円、円弧、扇形、楕円、楕円弧、楕円扇形、 自由曲線、点、テキスト
属性	線種、線幅、線色、線端の形、 矢印端、矢印の形、点の形 点の大きさ、面塗り(シェード、パターン)
基本 コマンド	選択、拡大/縮小、表示のみの拡大/縮小 変形、移動、転記、消去 合成、回転、登録、グリッド

[fig.3-7]既製の図形処理システムの主な機能

概念図の作成支援は、周囲の状況から処理を類推し、簡単な指示（例えば「この要素を削除する」「ここに要素を挿入する」等）により図が作成・編集できなければならない。図を描く際に指示しなければならない事項は

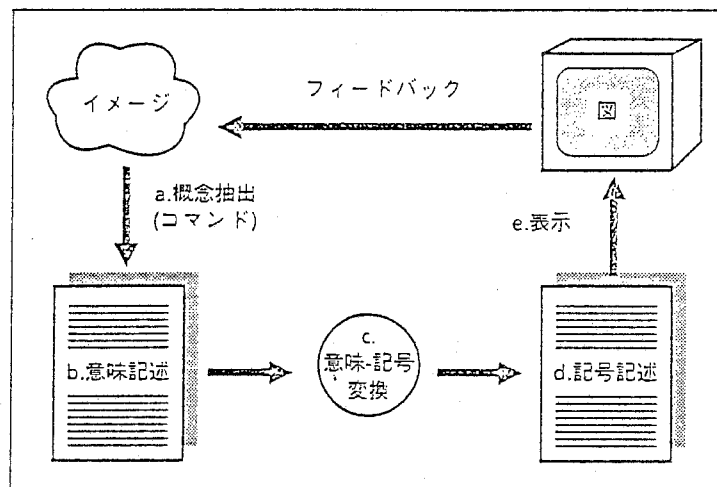
- a. 対象に関するもの（情報そのものを記述すること）
- b. 言語系に関するもの（形式に沿った記述をすること）
- c. 修辞に関するもの（美しく見せる記述をすること）

の3つがあるが、aのみの入力での図の作成ができることが最も望ましい。このためには図形と意味を対応させて扱うことが必須である。

図の構成は幾つかの段階で捉えることができる。

- 1. 図全体（図の意図や目的）
- 2. 図形集合（単位要素、要素間関係）
- 3. 図形（円、矩形など）
- 4. 図素（点、線など）

概念図作成支援システムでは上記の2の段階、即ち意味を持つ図形の集まりを基本単位とし、相互の意味的つながりを考慮した記述・処理を行う。概念図作成支援システムの概要は[fig.3-8]のように書き表せる。意味記述は図の意味に関する記述、記号記述は図の形状に関する記述である。人間の持つ概念やイメージは形状情報の指示ではなく、意味情報の指示としてシステムに伝えられ、システムはこれをもとに意味記述を行う。そしてこれを記号記述に変換し、図として表示する。人間は表示された図をもとに、フィードバックをかける。



[fig.3-8]概念図作成支援システムの構成

3-4. 図の関係表現

図示するということは、事象とその相互関係を、要素記号と関係記号で置き換えてゆく作業であり、記号の配置には図の形式ごとに約束がある。特に要素間関係をどのような構造で表すかによって図の形は全く異なったものとなる。2つの要素間関係を分類すると[fig.3-9]のようになる。方向性を持たない関係というのは、『分類項目』とか『共通属性を持つ要素同士』など要素相互に同等の重みを持つものであり、方向性を持つ関係というのは、『前後関係』『上下関係』など要素間に序列が生まれるような関係である。

定性的関係	方向性無し	—	並列的關係	要素相互が同等の重みを持つ関係 排他的カテゴリー形成や属性を共有する要素
	方向性有り	外在的方向性	順序的關係	要素相互に定性的順序が定義できる関係 時間的、空間的、あるいは観念的順序を示す要素
		内在的方向性	包含的關係	一方の要素が他方の要素を包含する関係 全体と部分、抽象化と具体化などを表す要素
定量的関係	相対量に基づくもの		相対量的關係	要素相互の相対量による定量的關係 大小比較が可能な要素
	絶対量に基づくもの		絶対量的關係	要素相互の絶対量による定量的關係 倍率比較が可能な要素

[fig.3-9]関係の分類

また図の形式ごとに、これらの関係を表す適否を大まかに評価すると[fig.3-10]のようになる。

		連結系	配列系	領域系	座標系
定性	並列的	○	◎	◎	△
	順序的	◎	◎	○	○
	包含的	◎	○	◎	×
定量	相対量的	△	△	△	◎
	絶対量的	×	×	△	◎

(◎:非常に向く、○:向く、△:あまり向かない、
×:向かない)

[fig.3-10]図の表現能力

この表から連結系、配列系、領域系の図は関係表現能力において近く、また定性的表現、つまり概念図向きであることがわかる。また、これら3種の図は意味を図化する際の適用分野が接近しており、多くの場合相互に変換が可能である。しかし関係表現方法は[fig.3-11]に見られるように図の形式によりそれぞれ特徴的である。

		連結系	配列系	領域系
定性的関係	並列的關係	同じ形の要素記号、同じ要素に接続する同階層要素	要素の相互配置 同じ形の要素記号	閉曲線で領域を仕切る
	順序的關係	要素間接続	要素の相互配置	閉曲線領域の相互配置
	包含的關係	要素間接続 図に枠を付け入れ子にする	要素の相互配置 図に枠を付け入れ子にする	閉曲線で仕切られる領域を 入れ子にする
定量的関係	要素記号・関係記号の大きさ、要素記号相互の距離	要素記号・関係記号の大きさ、要素記号相互の距離	閉曲線領域の面積	

[fig.3-11]図の関係表現方法

4. 概念図作成支援システム

以下では概念図作成支援システムの記述、処理、変換について述べる。

4-1. 記述

図の記述は次のような点を基本としている。①図の担う意味を記述する『意味記述』と、図形情報を記述する『記号記述』をそれぞれ独立に行うこととし、前者を図の約束ごとに従って『変換』することにより後者を生成・編集する。②図の成り立ちに従い、『意味記述』『記号記述』ともに内部で『要素』の記述と『関係』の記述を書き分ける。③記述にはフレームを用いる。

意味記述は図の要素とそれら相互の関係を記述したものであり、図の生成や意味的操作を支援する際のデータとなる。また、記号記述には図形情報が記述され、図の生成・編集における表示用図形データとなる。

4-1.1. フレームによる記述

意味記述を行うフレームを一般に『意味フレーム(MF)』と呼ぶことにし、特に要素の記述を行うフレームを『要素意味フレーム(EMF)』、関係の記述を行うものを『関係意味フレーム(RMF)』と呼ぶことにする。

また、記号記述を行うフレームを一般に『記号フレーム(SF)』と呼ぶことにし、特に要素の記述を行うフレームを『要素記号フレーム(ESF)』、関係の記述を行うものを『関係記号フレーム(RSF)』と呼ぶことにする。

これらをまとめると次のようになる ([fig. 4-1])

	意味フレーム (MF)	記号フレーム (SF)
要素記述……	要素意味フレーム(EMF)	要素記号フレーム(ESF)
関係記述……	関係意味フレーム(RMF)	関係記号フレーム(RSF)

[fig. 4-1] フレームの分類

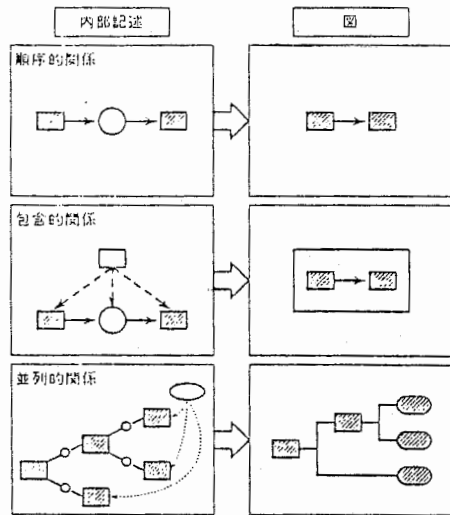
4-1.2. 意味フレーム

概念図は要素とそれら相互の関係を視覚化したものである。要素の表現はどのような形態の図も文字や記号を用いており大差がないが、関係の表現法は [fig. 3-11] に見られるようにそれぞれ特徴的である。つまり概念図の構造は関係の内容と表現の方法に強く依存する。従って意味記述においては既に述べた3種の要素間関係、即ち、順序的關係・包含的關係・並列的關係、に従ってフレーム記述を行う。

意味フレームは次のような構成を取る。

- 一般意味フレーム(MF)
 - ・ 要素意味フレーム(EMF)
 - ・ 関係意味フレーム(RMF)
- 継承意味フレーム(IMF)

これらのフレームを用いて基本的な要素間関係を表すと [fig. 4-2] のようになる。図の左列は内部記述のポイント関係を示し、矩形はEMF, 円はRMF, 楕円はIMFを表す。右列は図として表示されるイメージである。

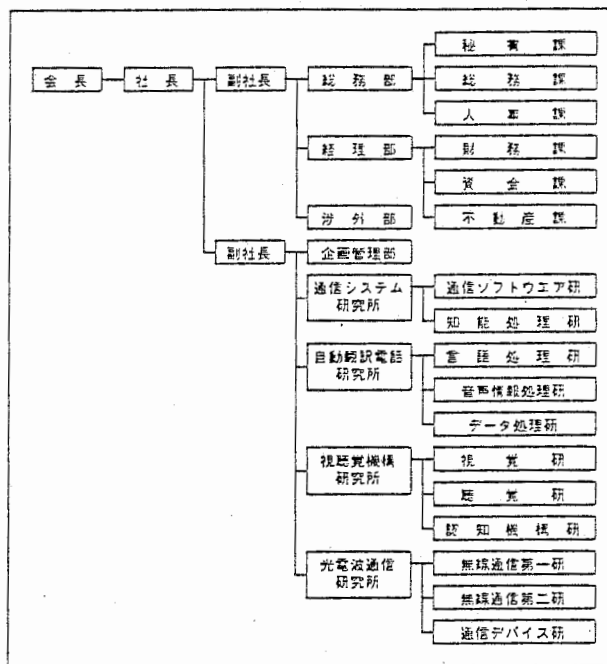


[fig.4-2]意味フレームによる3種の関係表示

• 順序的關係の記述

順序的關係は要素の相互関係に向きを付けることが可能で、2つの要素間をポインタで接続することにより表現することができる。フレームでこれを記述するには要素間関係の記述がしやすいように2つのEMF間にRMFを挟んで順序的關係を示すポインタで接続する。こうすることにより、特に連結系の図で要素と記号の対応が取りやすくなり、要素間関係に依存する編集処理も行いやすくなる。

[fig.4-3]は一般的な組織図の例であり、矩形が要素、矩形間を結ぶ線分が関係を表している。この図が組織の『指揮命令系統』あるいは『権限』という関係に基づいて描かれたものであれば、矩形間を結ぶ線分は全て順序的關係と見なすことができる。

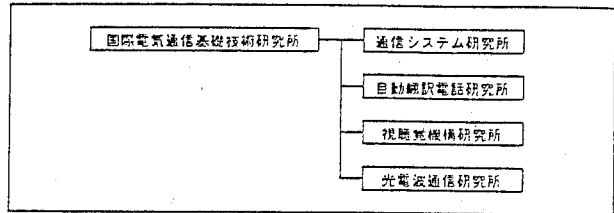
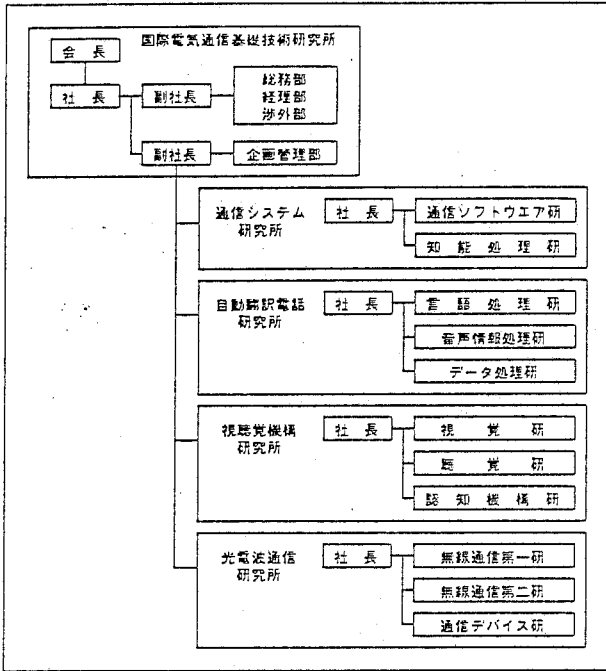


[fig.4-3]ATRの組織(A)

● 包含的関係の記述

包含的関係はある親要素と、それを構成し完全に親要素に含まれる子要素との関係である。この関係は親子要素の相互をポイントで接続することにより表現することができる。フレームによる表現は、親要素から包含する子要素全てに包含関係を示すポイントを張る。

[fig.4-3]の見方を変え、これが『構成』という関係に基づいて描かれたものとする、『部』と『課』の関係、『研究所』とその下の『研究室』との関係は包含的関係で記述され、その他は順序的關係で記述される。この場合、順序的關係と包含的関係の図化の仕方は同じである。これに対し、[fig.4-4]は順序的關係と包含的関係を異なる表現方法で図化した例であり、この図の場合、組織とその内部構造との関係は包含的関係として、また組織間との関係は順序的關係として記述される。また[fig.4-4](C)は[fig.4-4](B)と同じ意味記述から作られ、図化する際に包含的関係の要素を表示せずにラベルで置き換えた例である。このように順序的關係と包含的関係の区別は、図の構造を描き分ける重要な情報となる。



[fig.4-4] ATRの組織(B,C)

● 並列的關係の記述

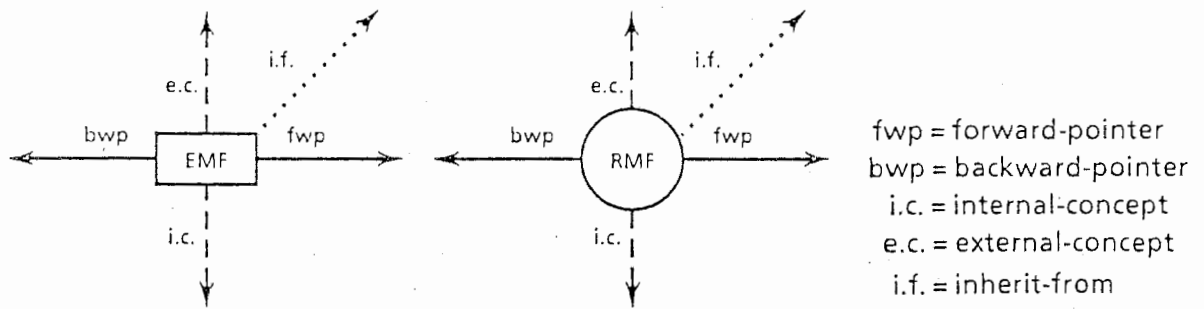
並列的關係は属性の共有という形で記述することができる。したがってこの関係は属性の継承関係によって表現することができる。同じ場所から属性の継承を行う対象同士はその属性に関してグループを形成する。例えば『順序的關係における同階層要素』とか『ある要素に包含されている要素』などといったグループの他、全く任意に要素のグループ化を行い、並列的關係を作ることもできる。この際、継承する属性はIMFに記述し、この属性に関して並列的關係を持つEMF, RMFにポイントを張る。

[fig.4-3]では同じ階層にある要素の枠位置が縦に揃えられ、枠の大きさもこの階層ごとに揃えられている。よってこの図では各々の要素に対して階層レベルという属性を与えることができ、同じ階層レベルの要素は1つのグループとして並列的關係を持つということが出来る。

● 一般意味フレーム(EMF, RMF)のロット

EMFとRMFは同じ構造、同じロットを持つ。MFは3種の関係に対応する次のようなポイントを持つ。

1. 順序的關係……順方向(fwp)、逆方向(bwp)
2. 包含的関係……包含(int_concept)、被包含(ext_concept)
3. 並列的關係……継承元(inherit_from)



[fig.4-5]EMF,RMFのポインタ

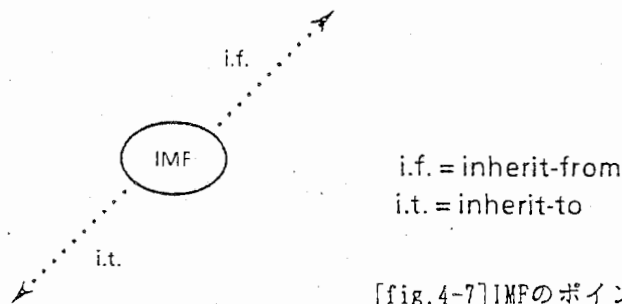
この他にフレーム名を示す“is_a”、フレームの種類を示す“frame_tag”、属性を記述する“property”、記号フレームとの対応をとる“sf_pointer”などのスロットにより、一般意味フレームが構成される。尚、継承元を示すポインタ(inherit_from)はpropertyスロットの中の1つの属性毎に設定されるポインタである。

スロット名	内 容	説 明
is_a	フレームの名前	他のフレームと区別しうるユニーク名を記述
frame_tag	フレームの種類	要素意味記述なら“EMF”、関係意味記述なら“RMF”と記述
sf_pointer	対応する記号フレーム	表示されている図形と意味の対応を取る
forward_pointer	順序的關係記述(順方向)	順序的關係をもつ順方向フレームを記述
backward_pointer	” (逆方向)	順序的關係をもつ逆方向フレームを記述
internal_concept	包含的關係記述(順方向)	包含するフレームを記述する
external_concept	包含的關係記述(逆方向)	包含されるフレームを記述する
property	属性記述	「属性名」「属性値」「継承元」からなるリストを記述

[fig.4-6]EMF,RMFのスロット

● 継承意味フレーム(IMF)のスロット

IMFは並列的關係の記述に關係し、これを継承關係によって記述する。このためこのフレームは継承元と継承先を示すポインタを持つ。



[fig.4-7]IMFのポインタ

この他にフレーム名を示す“is_a”、フレームの種類を示す“frame_tag”、属性を記述する“property”などのスロットにより、継承意味フレームは成る。尚、継承元と継承先を示すポインタ(inherit_from, inherit_to)はpropertyスロットの中の1つの属性毎に設定されるポインタである。

スロット名	内 容	説 明
is_a	名前	他のフレームと区別しうるユニーク名を記述
frame_tag	フレームの種類	“IMF”と記述
property	属性記述	「属性名」「属性値」「継承元」「継承先」からなるリストを記述
import	属性継承元	デフォルトの属性継承元

[fig.4-8]IMFのスロット

4-1.3. 記号フレーム

記号フレームは意味フレームと対応し、図の形に関する情報が記述される。これは表示のためのデータとして用いられ、表示用図形データベースにはほぼ等しい。記号フレームは次のような構成を取る。

- 一般記号フレーム(SF)
 - 要素記号フレーム(ESF)
 - 関係記号フレーム(RSF)
 - コンポーネント(COMPONENT)
- プリミティブ(PRIMITIVE)
- 一般記号フレームのスロット

ESF, RSF, COMPONENTは同じ形をとるが、ESFは要素記号に関する記述、RSFは関係記号に関する記述が行われる。またCOMPONENTはPRIMITIVEを組み合わせるための記述である。ESF, RSF, COMPONENTはCOMPONENTおよびPRIMITIVEから構成される。一般記号フレームは相互接続と構成を記述するために次のようなポイントを持つ。

1. 接続関係……順方向(fwp)、逆方向(bwp)
2. 構成関係……包含(component)、被包含(parent)

その他に、一般記号フレームと同様is_a, frame_tag, property, mf_pointerなどのスロットを持つ。

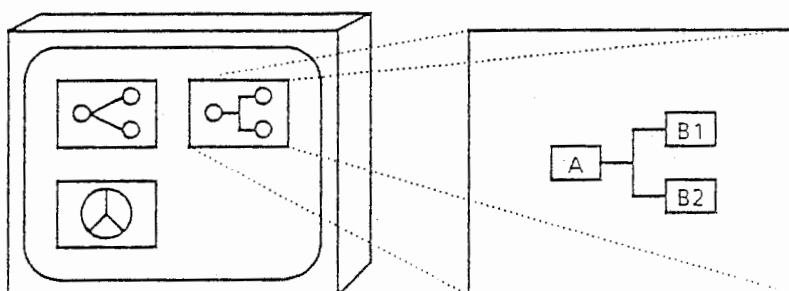
• プリミティブ

PRIMITIVEは基本図形を定義するフレームで、線分・円・矩形など、図形に応じて必要なスロットを持つ。例えば線分プリミティブでは、始点・終点の座標値、線のタイプ、線の太さ、終端の形(矢印など)を定義するスロットが用意される。

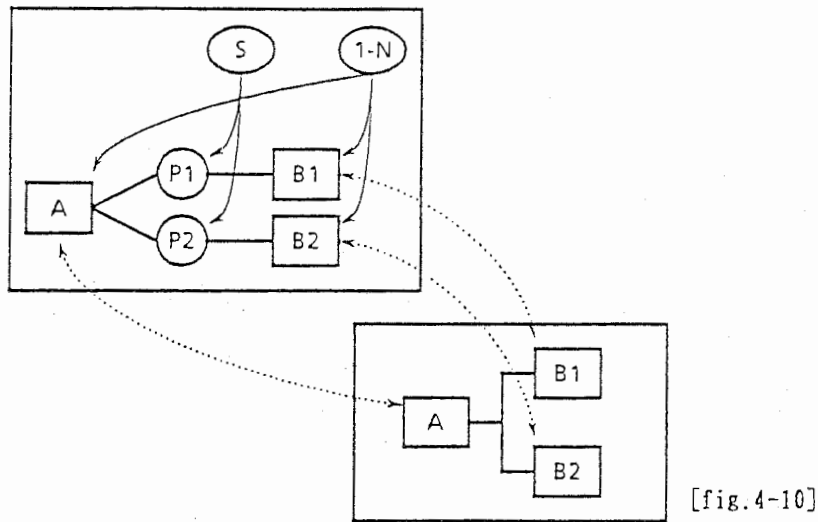
4-2. 処理

4-2.1. 操作概要

- ▶ 最初にこれから作成しようとする図の『雛型アイコン』を選ぶ([fig. 4-9]左)
 - 画面上には図の雛型を示すアイコンが並んでいる
 - 『雛型を選ぶ』ということは、『作図したい図の形式を選ぶ』ということである
 - 雛型は、図の形式に依存する作図規則を個々に持つ
- ▶ 雛型アイコンを選ぶと、このウィンドウがオープンされ、編集可能状態になる([fig. 4-9]右)
 - ウィンドウには雛型の図が表示されている
 - 雛型の図は相対応する意味フレーム([fig. 4-10]左上)と記号フレーム([fig. 4-10]右下)から成る
 - 図2の表示上必要なデータは記号フレームに記述されている
 - このとき意味フレームは枠組みのみが用意されている(実質的な意味は記述されていない
ただし、最小限の情報は継承意味フレームにより与えられる)
 - 継承意味フレームの初期値は階層構成図の場合以下のようなになる
([fig. 4-10]の楕円が継承意味フレーム, NODE_TYPE, CONNECTION_TYPEについては後記)
 - 属性名(属性値): NODE_TYPE(1-N) 継承対象: ALL*EMF
(ノードタイプ"1-N"で継承対象は全ての要素意味フレーム)
 - 属性名(属性値): CONNECTION_TYPE(SUCCESSION) 継承対象: ALL*RMF
(接続タイプは"連続"で継承対象は全ての関係意味フレーム)



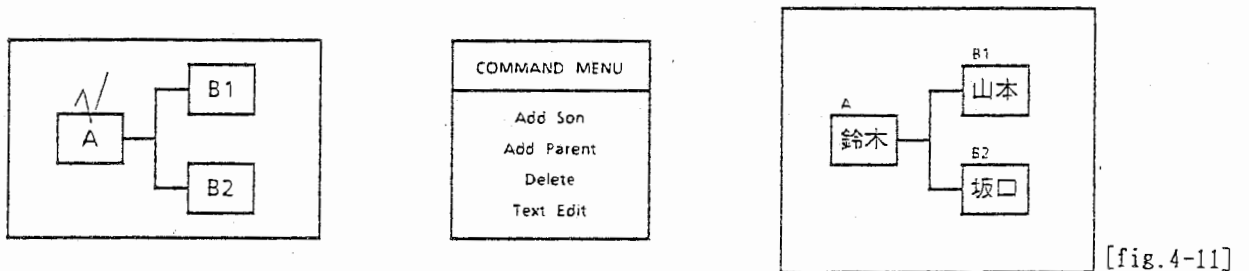
[fig. 4-9]



[fig.4-10]

▶Text_Editする([fig.4-11])

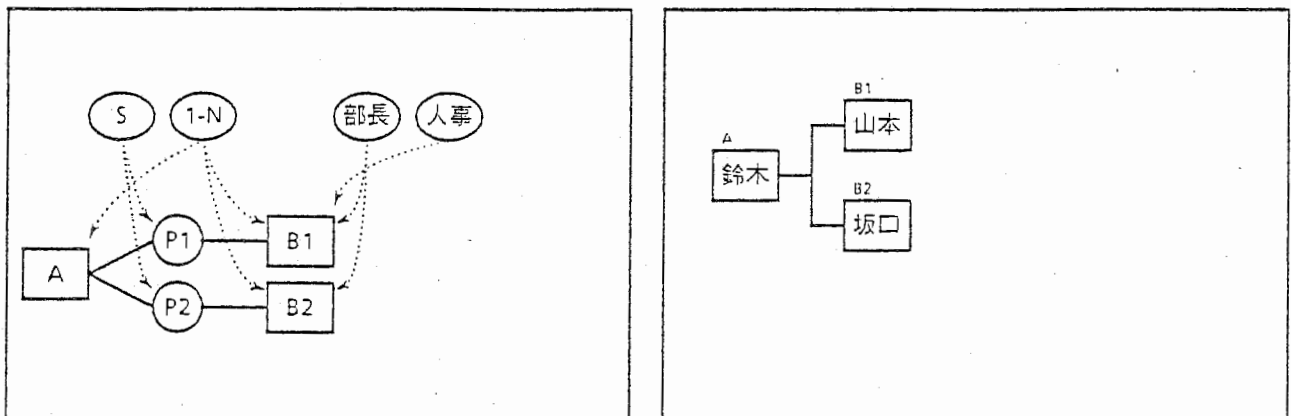
- 要素Aを“鈴木”に変える
 - 要素Aを選択⇒コマンドメニューが出る：“Text_Edit”を選択⇒“鈴木”とキーインし
⇒表示が変わる
- 同様に要素B₁,B₂もそれぞれ“山本”,“坂口”に変える
- 内部記述はEMFのSUBJECTスロットに書き込まれる
 - 要素A：ラベル(鈴木)
 - 要素B₁：ラベル(山本)
 - 要素B₂：ラベル(坂口)



[fig.4-11]

▶要素B₁をChange_Propertyする([fig.4-12])

- “役職”“部署”という属性名を生成し、属性値と継承パターンを与える
 - 要素B₁を選択⇒コマンドメニューが出る：“Change_Property”を選択
⇒プロパティメニューから“Create”コマンドを選択
⇒生成したい属性名と、属性値、継承範囲を記述する
 - ・〔役職(部長),.CL*EMF〕……同じクラスの要素に部長という役職名を与える
 - ・〔部署(人事),.FW*EMF〕……下位の要素全てに部署=人事を与える(FW=FORWARD)



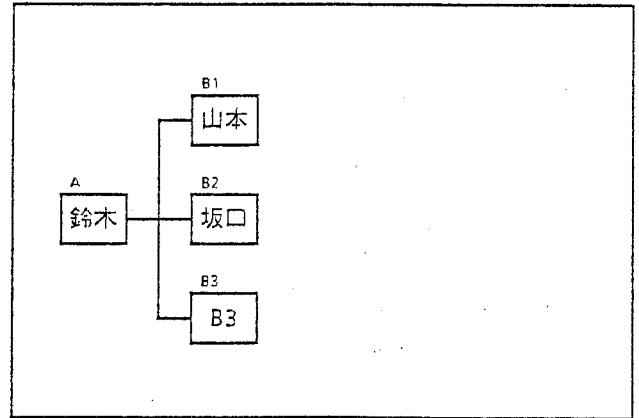
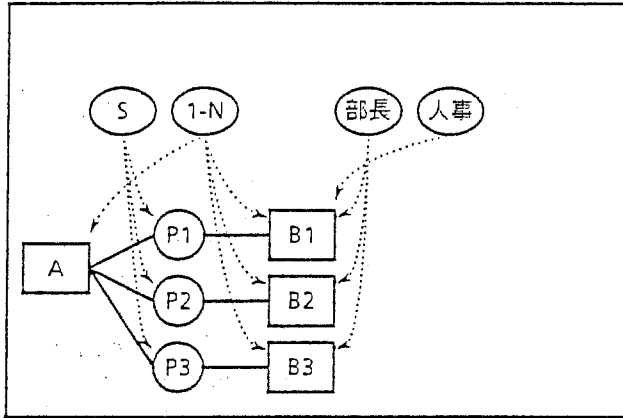
[fig.4-12]

▶要素A に対し "Add_Son"する ([fig.4-13])

○図に要素B₃が挿入され、Aと接続される。全体の形は対称形を保つように整えられる

○意味フレームでは次のような処理が行われる

- まず要素意味フレーム "B₃" と関係意味フレーム "P₃" が生成され、A→P₃→B₃ とポインタを張る
- 継承意味フレームからは、新しく生成されたフレームにポインタを張る
 - 要素 "B₃" NODE_TYPE=1-N, 役職=部長
 - 関係 "P₃" CONNECTION_TYPE=SUCCESSTION



[fig.4-13]

▶表示属性を変える (Change_Property コマンド) ([fig.4-14])

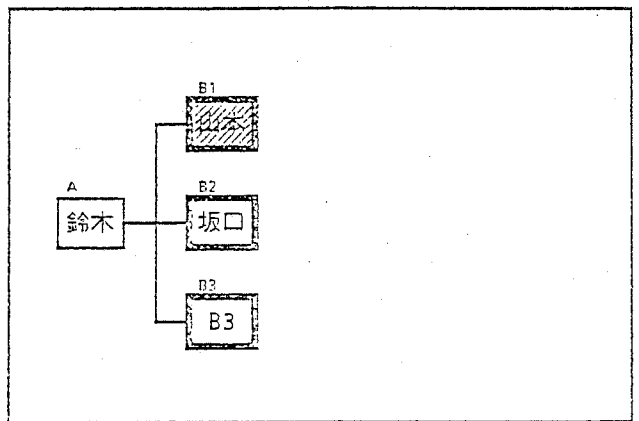
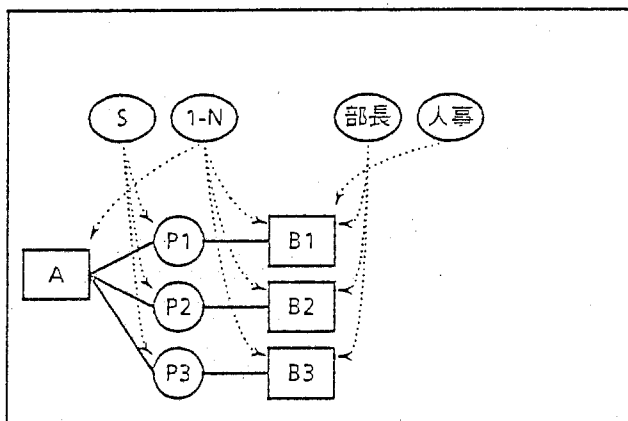
○部長を太枠で表示する

- Change_Property コマンドにより部長に対応する表示属性を記述する

• [役職(部長), es_line_width(4), CL*EMF]

○人事を斜線のハッチで表示する

• [部署(人事), es_pattern(RIGHT_UP), FW*EMF]



[fig.4-14]

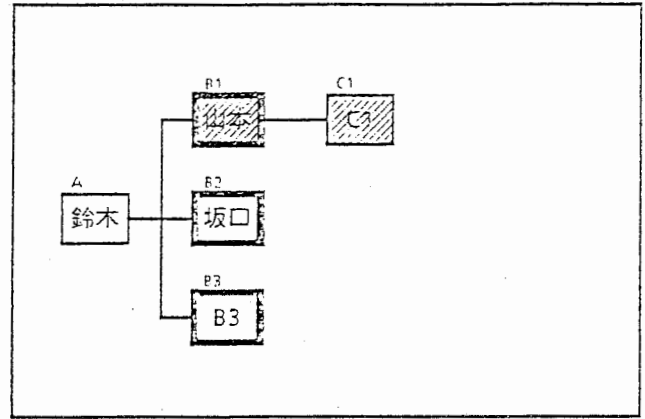
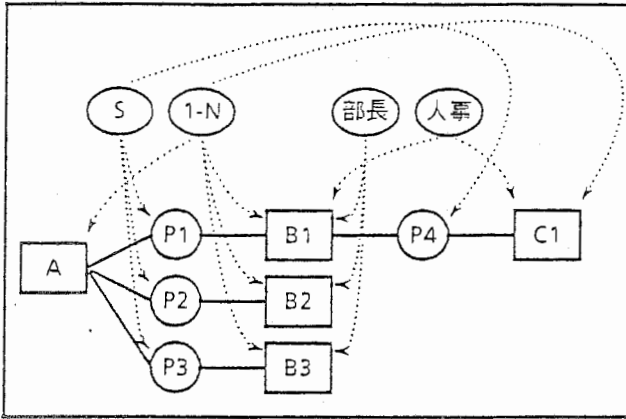
▶要素 "B₁" に対し "Add_Son"する ([fig.4-15])

○図に要素C₁が挿入され、B₁と接続される

○意味フレームでは次のような処理が行われる

- 要素意味フレーム "C₁" と関係意味フレーム "P₄" が生成され、B₁→P₄→C₁ とポインタを張る
- 継承意味フレームからは、新しく生成されたフレームにポインタを張る
 - 要素 "C₁" NODE_TYPE=1-N, 部署=人事
 - 関係 "P₄" CONNECTION_TYPE=SUCCESSTION

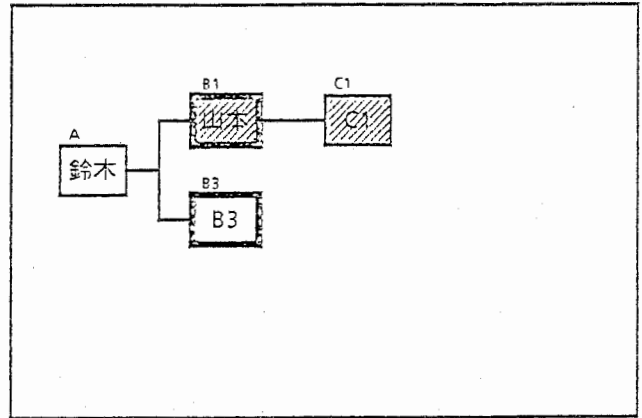
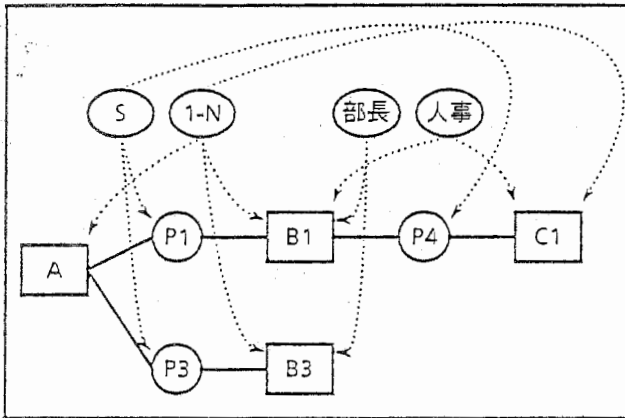
○表示はC₁が部署=人事であることから、ハッチングがかけられる(枠は太くならない)



[fig.4-15]

▶要素“B₂”をDeleteする([fig.4-16])

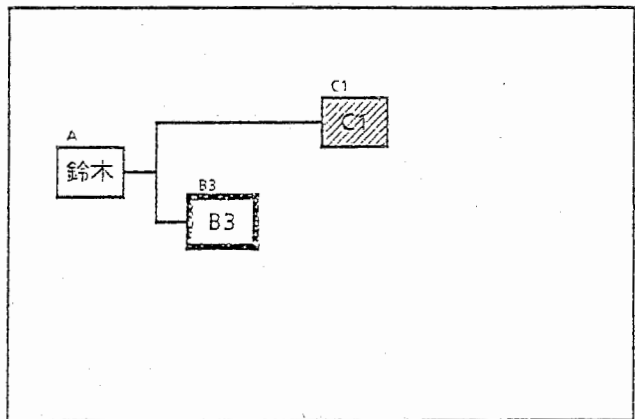
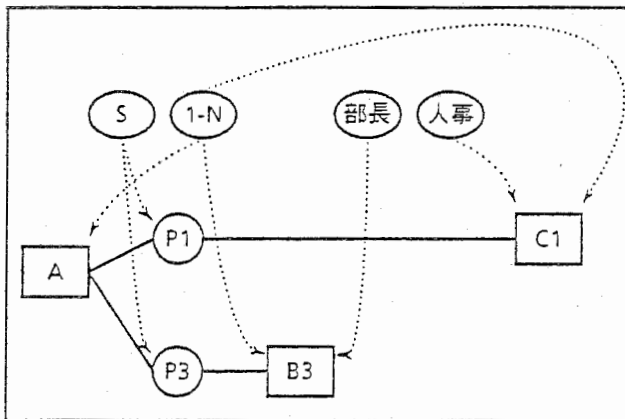
- 要素意味フレーム“B₂”と関係意味フレーム“P₂”が消去され、不要なポインタも削除される
- 図は対称形を保つように整えられる



[fig.4-16]

▶要素“B₁”をDeleteする([fig.4-17])

- 要素意味フレーム“B₁”と関係意味フレーム“P₄”が消去され、不要なポインタも削除される
- 関係意味フレームの接続タイプが“SUCCESSION” (連続)なので、“P₁”と“C₁”がポインタで接続される
- 要素“C₁”の階層は変化しない



[fig.4-17]

4-2.2. ノードの分類

フレームによる概念図の記述は、図の構成要素を個々に切り離し、ポインタで相互の関係付けを行うものであり、この構造はデータを有向グラフとして捉えることを可能にする。有向グラフは、順序的關係や包含的關係をEMFやRMFの接続で直接的に表現し、並列的關係はIMFによる継承關係で間接的に表現する。有向グラフの取る形は、包含的關係では完全な木構造でなければならず、並列的關係を表現する継承構造はループが許されない。これに対し、順序的關係を表す有向グラフには特に制限が無く、一般的事象間關係を表す能力が高い半面、編集操作においては有向グラフの形を規定するためのより多くの指示が必要になる。このため、以下では順序的關係を中心に有向グラフの基本的処理（要素の挿入や削除等）を制御する方法を考察する。

順序的關係は、要素（EMF）をノード、關係（RMF）をアークと見ると有向グラフに置き換えて考えることができる。各ノードはポインタの方向により、入・出力が定義でき、入力を0,1,n(2入力以上)、出力も0,1,n(2出力以上)と分けると、それらの組み合わせでノードを9種に分類できる[fig.4-18]。

便宜上、1入力n出力ノードを“1-n型”と呼ぶ。組織図におけるノードは一般に1-n型であり、最高責任者は0-n型となる。ノードの属性にこのようなノードタイプを与えておくと、ノードの挿入・削除等でグラフの形が変わる場合、組織図中にn入力ノードが作られるような条件分岐を自動的に除外することができる。

入力 \ 出力	0	1	n
0			
1			
n			

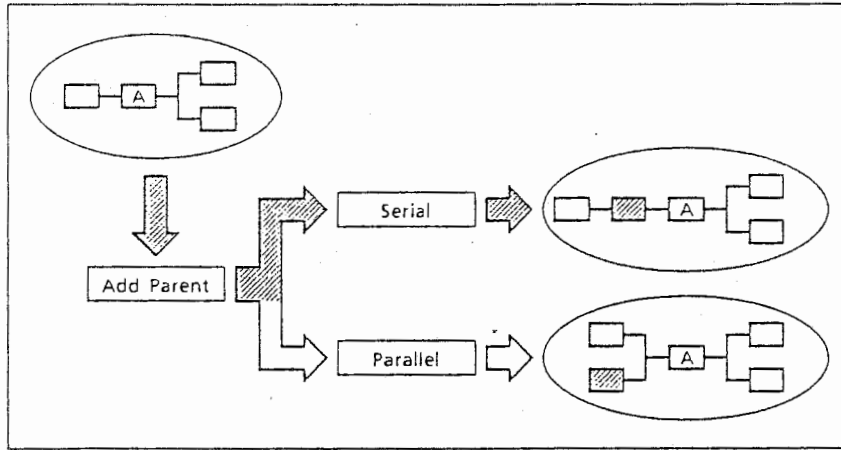
[fig.4-18] ノードの分類

4-2.3. 要素のAdd

あるノードとの関係指示により、新しいノードを挿入する方法を考えると、ノードの挿入と同時にアークを張ることができる。有向グラフでは、ノードの上位あるいは下位に新しいノードを挿入する場合が考えられ、前者を“Add Parent”、後者を“Add Son”と呼ぶことにする。それぞれはさらに6通りのパターンに分けられ、合計12通りの場合が考えられる[fig.4-19]。図から明らかのように、孤立したノードではAdd_SonまたはAdd_Parentにより、その結果期待されるグラフの構造が一意に決まる。その他の場合は構造を一意に決定することができず、新たに挿入されるノードの接続パターンを指定しなければならない。ノードタイプはこの選択の幅を狭めることができ、例えば[fig.4-20]のような図の場合、要素AにAdd Parentするには一般に可能性が2つあって、直列に要素を挿入するか、並列に要素を挿入するかを指定する必要がある。しかしながら、これが組織図で全てのノードのタイプが1-n型に指定されていれば[fig.4-20]の右下図の可能性は否定され、Add Parentというコマンドのみで一意に右上図の結果を得ることができる。

Add Son			Add Parent		
出力	Add 前	結果	入力	Add 前	結果
0		1	0		7
1		2	1		8
		3			9
n		4	n		10
		5			11
		6			12

[fig.4-19] ノードのAdd



[fig.4-20]Add Parentの具体例

4-2.4.要素のDelete

あるノードが削除されたとき、他のノードに及ぼす影響を次のように考える。①上位ノードが削除される
とき、下位ノードは上位ノードへの依存度に応じ消える場合と残る場合がある、②直線的につながる3つ以
上のノードの中間のノードが削除され、かつ1つ以上の下位ノードが残るとき、前後の関係（接続）が切れ
る場合と切れない場合がある、③上位ノードは下位ノードが削除されても影響を受けない。このとき、0出
力ノードは③によりノード削除後の状態が一意に決まるのに対し、1以上の出力を持つノードでは①②によ
り削除後の状態が一意に決まらない。これはノード自身に依存するものではなく、ノード相互の接続のしか
たに依存するものである。接続の種類は次の3種に整理することができる[fig.4-21]。

1. 依存的接続（D接続：(D=Dependence)）

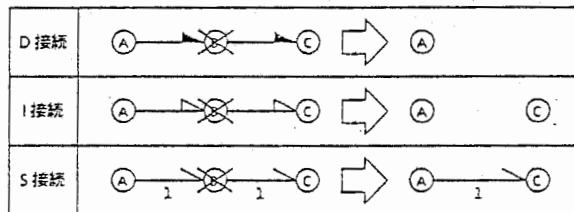
上位ノードが削除されると下位ノードも削除される

2. 独立的接続（I接続：(I=Independence)）

上位ノードが削除されても下位ノードは残る、ただし接続は断たれる

3. 連続的接続（S接続：(S=Succession)）

S接続にはラベルが付けられ、中間のノードが削除されても同じラベルの接続は保たれる



[fig.4-21]要素間接続の分類

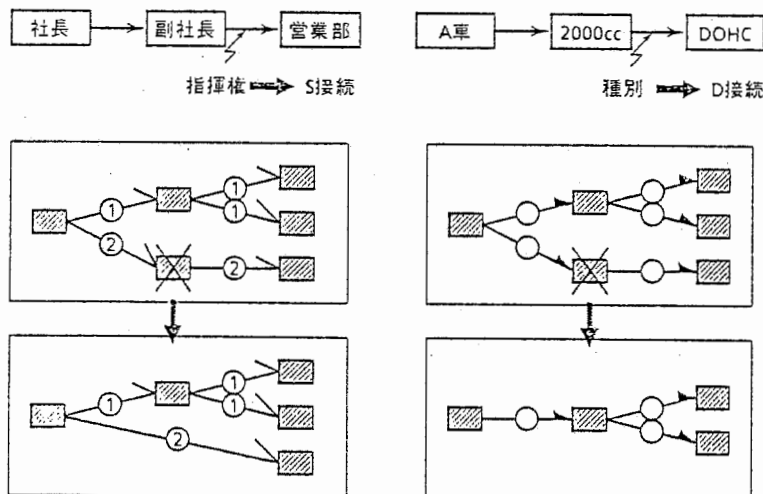
これらの接続状態をノードに記述することにより、Delete後の状態を一意に決めることができる [fig.4-22]。

型	Delete 前	内部記述例	結果	型	Delete 前	内部記述例	結果
0-0			nil	1-n			
1-0			○				
n-0			○				
0-1			○				
1-1			○ ○	n-n			
			○ ○				
n-1			○ ○				
			○ ○				
			○ ○				
			○ ○				
0-n			○ ○				
			nil				

内部記述例の凡例
 D接続 → 任意の接続 (但しS接続のラベル以外)
 I接続 →
 S接続 →

[fig.4-22]ノードのDelete

例えば[fig.3-6]は[fig.4-23]のように要素間接続を記述しておくことにより要素削除後に結果が一意に決まる。組織図では『社長』、『副社長』、『営業部』が連続的接続で接続されるため『副社長』が削除されても『社長』から『営業部』への接続が保たれることになる。また車のエンジンの種別を表す図では『2000cc』と『DOHC』が依存的関係で接続されるため『2000cc』が削除されるのに伴い『DOHC』も削除される。



[fig.4-23]Deleteを制御する内部記述

順序的關係ではEMFがノード、RMFがアークに対応し、接続方法の違いはRMFの属性として記述できる。これにより、EMFの削除結果はRMFを参照することにより一意に決定することが可能となる。なお包含的關係は本質的に全てD接続と見なすことができる。また並列的關係は継承關係を用いて表されるが、これはS接続に基づいて処理される。

4-3. 変換

概念図作成支援システムは意味フレームを変換して記号フレーム（即ち図形データベース）を得る。この変換部分は様々な形態の図に容易に対応できるようにルールで記述しておく。ルールの基本的機能は次の3点である。

1. 要素間関係をどのような構造で表すか決定する
 2. 要素の属性をどのような記号で表すか決定する
 3. きれい・見やすい・判りやすいという作図ノウハウを蓄積・利用する
- ルール全体の構成は以下ようになる。

- ・ デフォルトパラメータ記述部
 - ・ ルールセット定義部
 - ・ 制御部
 - ・ ルール記述部

4-3.1. デフォルトパラメータ記述部

デフォルトパラメータ記述部は変換する際の記号フレームのデフォルト値を与える。パラメータは図の形状を決める変数で、主なパラメータおよびデフォルトは以下のようなものである。

(es_shape	= 'RECTANGLE)	; 要素記号の形
(es_line_width	= 1)	; 要素記号の線幅
(es_line_type	= 'SOLID_LINE)	; 要素記号の線タイプ
(es_width	= 'TEXT_WIDTH)	; 要素記号の幅
(es_height	= 'TEXT_HEIGHT)	; 要素記号の高さ
(es_direction_1	= 'HORIZONTAL_R)	; 一次方向の要素記号配置
(es_direction_2	= 'VERTICAL_D)	; 二次方向の要素記号配置
(es_text	= nil)	; 要素記号に付加する文字列
(es_text_direction	= 'HORIZONTAL)	; 組み方向
(es_character_size	=)	; 文字の大きさ
(es_text_position	= '(CC CC))	; 枠と文字列の位置揃え
(rs_line_width	= 1)	; 関係記号の線幅
(rs_line_type	= 'SOLID_LINE)	; 関係記号の線タイプ
(rs_forward_connection	= 'LC)	; 要素記号への順方向接続位置
(rs_backward_connection	= 'RC)	; 要素記号への逆方向接続位置
(rs_forward_end	= 'NORMAL)	; 関係記号の終端形
(rs_backward_end	= 'NORMAL)	; 関係記号の始端形
(rs_x_interval	=)	; 要素記号の横方向マージン
(rs_y_interval	=)	; 要素記号の縦方向マージン

4-3.2. ルールセット定義部

ルールセット定義部はルールの集まりと優先順位を定義する。ルールセットはルール及びルールセットを含むことができ、記述された順にルールの適否を調べ、最初に適合したルールを実行する。

例: (define rule_set_10
 (rule_set_9, rule_1, rule_3, rule_5))

4-3.3. 制御部

制御部は意味フレームのスロット内容に応じて処理の流れを制御する。一般形は

```
((if (MFのスロット値比較)) ..... 条件比較部
 (then (rule_set or rule)) ..... then実行部
 (else (rule_set or rule))) ..... else実行部
```

となる。条件比較部では意味フレームの内容を調べ、実行部でルールを適用してゆく。

```
例: ((if (state_flag .EQ. 'Add_Son_1))
 (then (rule_set_1))
 (else if (state_flag .EQ. 'Delete))
 .....
 (end if))
```

4-3.4. ルール記述部

ルール記述部は意味フレームの属性記述を調べ、これに応じて表示する記号の形、位置、接続を決定してゆく。構成は次の5部より成る。

- ID定義部 rule_1
 - property比較部 ((property(status) .EQ. 'CHIEF) .AND.)
 - SF shape決定部 ((comp_shape 'RECTANGLE).....)
 - positioning部 (RELPOS S₀ S₁)
 - connectioning部 (CONNECT S₀ S₁)
- ID定義部：ここでルール識別のためのユニークなルール名を定義する。
- property比較部：意味フレームのpropertyを調べる。一般形は

```
(( (属性名) .演算子. (値) )
```

である。演算子は .EQ. .NE. .LT. .LE. .GT. .GE. 等で、式全体を .AND. .OR. の論理演算子でつなぐことができる。式全体を評価してその結果が真ならば、制御を次のSF shape 決定部に渡す。

```
例: ((node_type .EQ. '1-n) .AND.
 (status .EQ. 'CHIEF))
```

- SF shape決定部：記号の形状を決定する。記号の形状はコンポーネント変数を用いて値と組みにして次のように記述する。

```
一般形: ((コンポーネント変数1 値1)
 (コンポーネント変数2 値2).....)
```

コンポーネント変数は個々の要素記号形状を決定する変数で、項目は"es-"で始まるパラメータと同じで、変数名はこれを"comp-"で置き換えたものになっている。デフォルトパラメータの値を参照する場合は先頭にコロンを付けてパラメータ名を書く。

```
例: ((comp_shape 'RECTANGLE)
 (comp_line_width :es_line_width)
 .....)
```

- positioning部：要素記号の位置を与える。位置指定の方法は絶対位置指定と相対位置指定の方法があり、それぞれ次のように記述する。

○絶対位置指定

一般形: (ABSPOS (位置決めを行うEMF) (x変数 絶対値) (y変数 絶対値))

```
例: (ABSPOS S0 (comp_x 0) (comp_y 0))
```

○相対位置指定

一般形: (RELPOS (位置決めを行うEMF) (位置を参照するEMF)

(x変数 方向性 相対値) (x変数 方向性 相対値))

例： (RELPOS S₀ S₁ (comp_x :es_direction_1 :rs_x_interval)
(comp_y :es_direction_1 :rs_y_interval))

• connectioning部：要素記号間の接続を与える。記号の形状はコンポーネント変数を用いて値と組みにして次のように記述する。

一般形： (CONNECT S₀ S₁ (コンポーネント変数1 値1)
(コンポーネント変数2 値2)……)

コンポーネント変数は個々の要素記号形状を決定する変数で、項目は“rs-”で始まるパラメータと同じで、変数名はこれを“comp-”で置き換えたものになっている。デフォルトパラメータの値を参照する場合は先頭にコロンを付けてパラメータ名を書く。

例： (CONNECT S₀ S₁ (comp_shape :rs_shape)
(comp_line_width :rs_line_width)
(comp_line_type :rs_line_type)
(comp_forward_connection :rs_forward_connection)
(comp_backward_connection :rs_backward_conection)
(comp_forward_end :rs_forward_end)
(comp_backward_end :rs_backward_end))

5.まとめ

概念図作成支援システムについて、その記述と処理および意味から記号への変換について述べた。記述では、①意味と記号の分離記述、②要素と関係の区別、③特に関係を3種に分類する、などにより図の成り立ちや意味的な処理に向けた記述が行えることを示した。処理については要素をノード、関係をアークと見なしてノードタイプや接続タイプを分類定義することが有意義であることを述べた。また変換についてはルールの書式を、①属性比較、②要素記号決定、③要素位置決定、④関係記号決定、に分けて行うことについて述べた。

これらの記述・処理・変換という方法をとることにより次のようなメリットが期待できる。

1. 図の入力・編集が意味の操作により行える
2. 図の意味記述が他のシステムのデータとして使える
3. 図の意味と記号の対応が常にとれており、マンマシンインタフェースとしての有用性が高い
4. ルール選択により異なる形態の図に変換することができる
5. 同じデータを異なる角度から見ることができる
6. 作図知識を独立に記述できる
7. 図に対する修辭的指示を減らすことができる

今後の課題としては、

1. 一般の図に適用できる、より普遍的な変換ルールの研究
2. きれい・みやすい・わかりやすい、といった作図知識の蓄積・利用の研究
3. 『よい図』の評価法の研究
4. 図を操作するのに優れたマンマシンインタフェースの研究
5. 意図からの図の作成・編集・検索などの研究
6. 動きを取り入れた図示の研究

などがあげられる。

6.参考文献

- [1] 出原、吉田、渥美(1986): 図の体系(日科技連)
- [2] 佐藤、田中、他(1987): 図の意味的処理へのアプローチ(昭和62年信学全大No.1432)
- [3] 佐藤、田中、他(1987): 図のコンセプト記述法および図形情報記述との結合(第35回情処全大1P-8)
- [4] 佐藤、田中、他(1987): 図の意味的処理を指向する記述と処理について(第36回情処全大6N-9)
- [5] 佐藤、田中、他(1988): 概念図作成支援のための図の意味記述(情処学会、88-A1-59-15)
- [6] 出原(1977): 人間と図形言語, 63-117(吉川編: コンピュータグラフィック論, 日科技連, 所収)
- [7] 概念図作成支援システムの試作に関する東洋情報システム(株)との打ち合わせ資料(非公開)