

ダイアグラム・プロセッサの構築

— 傾斜連結図を用いる思考支援
システムへの応用を中心として —

三末 雅男

ダイアグラム・プロセッサの構築

領域連結図を用いる思考支援システムへの応用を中心として

三末 和男

概要

「ダイアグラム・プロセッサ (DP)」とは図を利用して人間の思考過程を計算機により支援するシステムである。そこで利用される図とは要素間の関係を可視化したものであり、情報処理の上流工程において未整理な断片情報を処理するために有用と考えられる。DP を利用することにより図の編集、作成が効率化され、上流工程における情報処理の改善が期待できる。すなわち、DP を構築することの意義は、「情報処理上流工程のための基盤システムの提供」にあると言える。

本研究では、情報の可視化技術を思考支援に応用することで DP を構築する。ここでの課題は、(1) DP に必要とされる基礎技術 (情報可視化技術) の開発、(2) 基礎技術の統合による試用可能なシステムの実現、(3) システムの評価である。さらに、本研究では図を扱う上で最も重要と考えられる描画機能 (図の可視化機能) の高度化に焦点を合わせ、強力な表現力を備えた「領域連結複合系」の図を描画対象とする。領域連結複合系の図に関しては自動描画技術がこれまで全く開発されていなかったため、新しく開発する必要がある。

本研究の貢献は、領域連結複合系の図の自動描画技術を開発し、それを基に高度な機能を備えた DP を実現し、その有効性を評価したことにある。

自動描画技術としては、領域連結複合系の図の配置技術、装飾技術、表示技術を開発した。配置技術としては、領域連結複合系の図を「複合グラフ」として定式化し、図の美的基準を満足する自動配置アルゴリズムを開発した。装飾技術としては、図の見方に関するユーザの要求を分析し、図の視覚属性を自動的に決定する方法「図ドレッシング」を開発した。表示技術としては、限られたサイズの画面において図の全体と詳細を見せる問題を議論し、新しい表示方式「多視点遠近画法」を開発した。

開発した基礎技術を統合し、経験的な発想技法である KJ 法を参考にして、「D-ABDUCTOR」と呼ばれる DP を実現した。その特徴としては、(a) 図的思考過程の対話型支援システムであること、(b) 強力な視覚表現形式の図の自動描画機能を備えた高機能図インタフェースに重点を置いていること、(c) 直接操作環境とアニメーション環境によりユーザの手間と認知的負担を軽減していること、(d) 通信機能やマルチメディア機能などにより図的思考の新しい可能性を目指していること、などがあげられる。

実現した DP の評価としては、ユーザへのアンケート調査および評価実験を行った。使用感や満足度を問うアンケート調査の結果、思考作業に関しては総合的に 5 段階の 4 という良い評価を得た。評価実験では作業効率に対する機能の客観的かつ定量的な有効性を示した。新しく開発した評価法により実験を行なったところ、手作業に比べて約 6 倍、また汎用の作図システムに比べても約 5 倍効率が良いという結果が得られた。

目次

第 1 章	序論	1
1.1	本研究の意義	1
1.2	本研究の位置付けと課題	2
1.3	本研究の成果	3
1.4	論文の構成	3
第 2 章	ダイアグラム・プロセッサの構想	5
2.1	思考支援システムとしてのダイアグラム・プロセッサ	5
2.1.1	思考支援の分類	5
2.1.2	ダイアグラム・プロセッサの支援レベル	6
2.2	関係情報の可視化技術の応用	7
2.2.1	図の基本系と領域連結複合系	7
2.2.2	可視化技術の必要性	9
2.2.3	ダイアグラム・プロセッサのための可視化技術	12
2.3	ダイアグラム・プロセッサの構成	13
2.3.1	図を構成する情報	13
2.3.2	可視化パイプラインに基づくダイアグラム・プロセッサの構成	15
2.3.3	ダイアグラム・プロセッサの高機能化	15
2.4	技術的課題	17
第 3 章	描画に関する従来研究	19
3.1	図の配置技術	19
3.1.1	連結系	19
3.1.2	領域系	20
3.2	図の装飾技術	20
3.3	図の表示技術	21
3.4	システム化技術	21
3.4.1	視覚情報のパイプライン	21
3.4.2	ユーザ・インタフェース技術	22
3.5	図を扱う対話型システム	22

第4章 図の配置技術	24
4.1 描画対象	25
4.2 図的可読性	26
4.3 自動描画処理	27
4.3.1 複合階層化	27
4.3.2 正規化	31
4.3.3 順序決定	33
4.3.4 座標決定	38
4.4 出力例	40
第5章 図の装飾技術	43
5.1 図素と属性	44
5.2 視覚属性操作の形式化	45
5.2.1 要求への統一的な対応	45
5.2.2 図の動的な変化への対応	46
5.2.3 組合せ要求への対応	47
5.2.4 重要度と重要度関数	47
5.3 図ドレッシングの実現法	47
5.3.1 図ドレッシングの構成	48
5.3.2 重要度関数	49
5.3.3 自動図ドレッサ	50
5.3.4 例の一般性について	51
5.4 図ドレッシングの適用例	51
5.4.1 多様な要求への対応例	53
5.4.2 組合せ要求への対応例	55
5.4.3 図の動的な変化への対応例	55
第6章 図の表示技術	58
6.1 表示方式への要求	59
6.2 多視点遠近画法	61
6.2.1 魚眼表示方式	61
6.2.2 直交魚眼表示方式	62
6.2.3 二様表示方式	62
6.2.4 多視点遠近画法と写像の性質	64
6.3 複合グラフの表示のための多視点遠近画法	64
6.3.1 多視点遠近画法による表示例	64
6.3.2 複合グラフへの適合性	68

第 7 章	システムへの統合	70
7.1	D-ABDUCTOR 開発の考え方	71
7.2	D-ABDUCTOR の機能	71
7.2.1	計算機上での手作業の模擬機能	73
7.2.2	計算機による手作業の代行機能	74
7.2.3	計算機の不利を克服する機能	75
7.2.4	進んだ計算機環境の活用機能	75
7.3	KJ 法的作業の実施例	77
7.4	試用に対するアンケート評価	77
7.4.1	アンケートの方法	79
7.4.2	集計結果	82
第 8 章	システムの評価	85
8.1	評価法	86
8.1.1	これまでの評価法	86
8.1.2	評価法開発の考え方	86
8.1.3	LCR 法	87
8.1.4	LCR 法のバリエーション	89
8.2	実験法	91
8.2.1	作業	91
8.2.2	モード	92
8.2.3	図のクラス	92
8.2.4	ユニット	93
8.2.5	実験プログラム	93
8.2.6	被験者	94
8.2.7	作業数	94
8.2.8	データの収集と解析の仕方	94
8.3	実験結果	94
8.3.1	モード間の比較	95
8.3.2	クラス間の比較	96
8.3.3	作業数間の比較	97
8.3.4	ユニット設計の効果	97
8.3.5	他のシステムとの比較	98
8.3.6	被験者の感想	99
8.4	考察	102
8.4.1	評価法としての LCR 法の有効性	102
8.4.2	同種システム一般に対する示唆	102

第 9 章 結論	104
9.1 本研究の貢献	104
9.1.1 図の配置技術	104
9.1.2 図の装飾技術	105
9.1.3 図の表示技術	105
9.1.4 システム統合化	106
9.1.5 システムの評価	106
9.2 今後の課題	108
9.2.1 描画技術における課題	108
9.2.2 システムにおける課題	109
9.2.3 評価における課題	111
謝 辞	113
付 録 A 汎用プラットフォーム化	114
A.1 D-ABDUCTOR の拡張機能	114
A.2 アプリケーションの例	115
A.2.1 構造アナライザ	115
A.2.2 ディレクトリ・ブラウザ	117
A.2.3 ネットワーク・モニタ	117
A.2.4 グラフィカル・ハイパーテキスト	118
A.2.5 グラフィカル・アウトライン・プロセッサ	120
参考文献	120

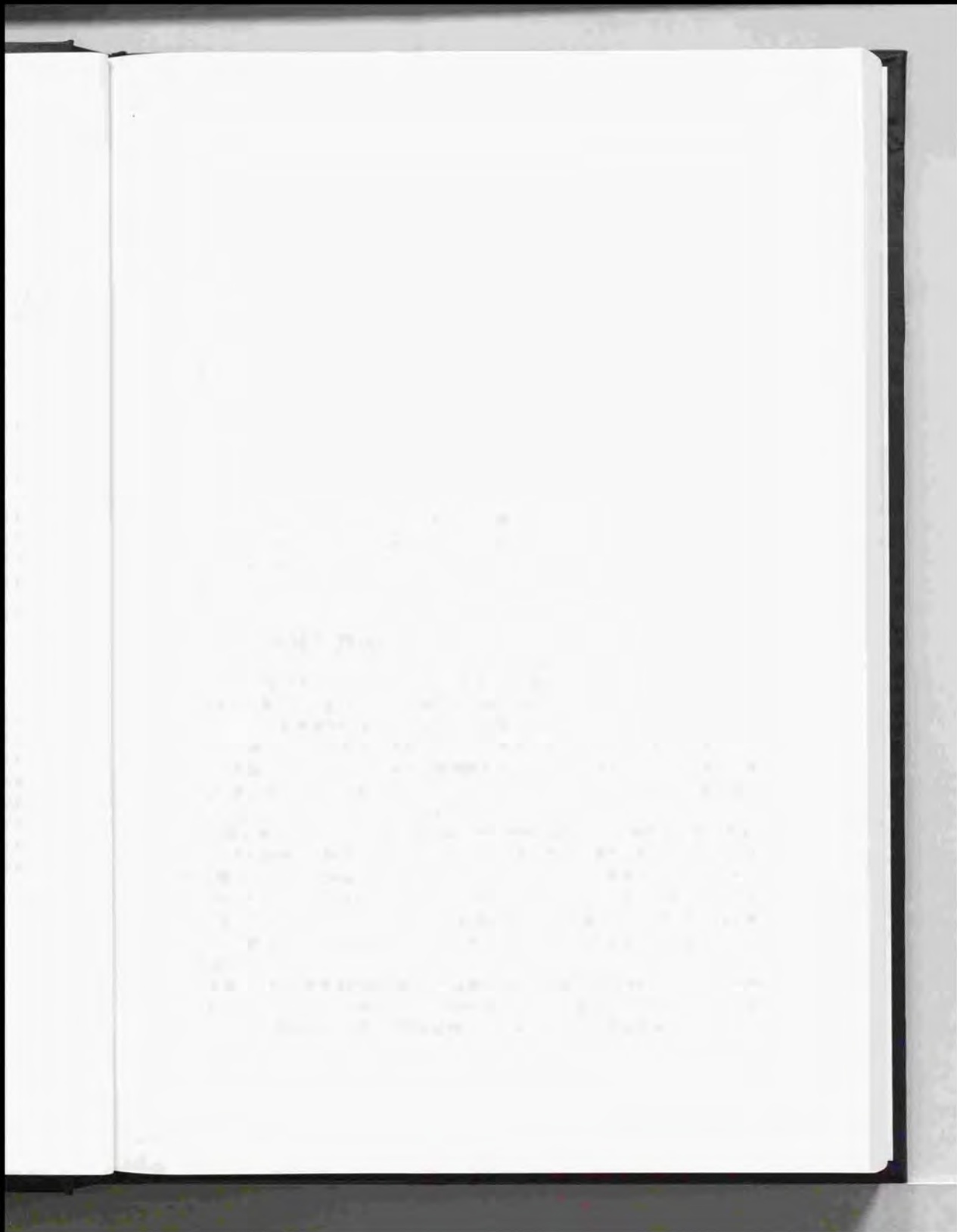
目 次

2.1	図の基本的な視覚表現形式 (基本系)	8
2.2	領域連結複合系の図の例	9
2.3	良いレイアウトと悪いレイアウト	11
2.4	図を構成する4レベルの情報	14
2.5	可視化パイプラインに基づくダイアグラム・プロセッサの構成図	14
4.1	複合グラフの描画例	28
4.2	自動描画処理の流れ	28
4.3	複合階層と複合階層値	29
4.4	正規化の例	32
4.5	複合グラフ G	34
4.6	複合グラフ G (図 4.5) の頂点 v に関する順序決定用局所グラフ G_D	35
4.7	順序決定問題 $P1$	35
4.8	複合グラフ G (図 4.5) の頂点 v に関する座標決定用局所グラフ G_P	39
4.9	複合グラフの自動描画プログラムへの入力データ	41
4.10	複合グラフの自動描画例	42
5.1	図ドレッシングの構成	48
5.2	複合グラフの自動描画例	52
5.3	多様な要求への対応例	54
5.4	組合せ要求への対応例	56
5.5	図の動的な変化への対応例	57
6.1	二様表示方式で用いる写像における拡大領域	64
6.2	縮小 (相似) 表示した複合グラフの例	65
6.3	魚眼表示方式による表示例	66
6.4	直交魚眼表示方式による表示例	67
6.5	二様表示方式による表示例	68
7.1	図的発想支援システム D-ABDUCTOR の構成	73
7.2	D-ABDUCTOR の画面例	74
7.3	KJ 法的作業の画面例	78

7.4	D-ABDUCTOR による図解の表示例	79
7.5	KJ 法的作業で利用するテキストの例	80
8.1	LCR 法で記述される作業の例	87
8.2	頂点のラベルの例 — 図 8.1(a) の各ラベルの全体を表示したもの	89
8.3	漢字 LCR 法による作業の完成図の例 (1)	90
8.4	漢字 LCR 法による作業の完成図の例 (2)	90
A.1	構造アナライザ	116
A.2	ディレクトリ・ブラウザ	117
A.3	ネットワーク・モニタ	118
A.4	グラフィカル・ハイパーテキスト GDS-info	119
A.5	グラフィカル・アウトライン・プロセッサ GOL	120

表 目 次

2.1	ダイアグラム・プロセッサの高機能化の次元	16
4.1	採用した描画規約, 描画規則, 優先関係	27
5.1	要求と視覚属性操作	45
5.2	図の属性による対象集合および属性関数の分類	46
6.1	表示方式の分類	60
6.2	写像の性質	60
6.3	多視点遠近画法の各方式と写像の性質	65
6.4	階層描画された複合グラフに必要な写像の性質	69
7.1	KJ 法の各ステップにおける作業内容とそこで行なわれる図的操作	72
7.2	D-ABDUCTOR の試用評価アンケート A-D の集計結果	83
7.3	D-ABDUCTOR の効果・使用感に関するユーザのコメント	84
8.1	実験で比較される五つのモード	92
8.2	各クラスの図の要素数	92
8.3	実験のプログラム	93
8.4	作業時間のモード別平均 (秒)	95
8.5	作業者ごとの作業時間と作業効率の比較	96
8.6	習熟の程度	97
8.7	D-ABDUCTOR と他のシステムとの比較	98
8.8	SuperPaint と D-ABDUCTOR(モード 1, 2) での作業内容の比較	100
8.9	被験者のコメント	101



第 1 章

序論

本論文では、「ダイアグラム・プロセッサ (DP)」の構築に関して、「領域連結複合系図表現 (または領域連結図)」を用いた思考支援システムへの応用を中心にして述べる。ここで、ダイアグラム (または図) は要素間の関係を可視化したもの、DP はダイアグラムを利用して人間の思考過程を計算機により支援するシステムである。本章では、まず本研究の意義を説明し、次に本研究の位置付けと課題を明確にし、その課題に対する本研究の成果について述べる。最後に論文の構成を説明する。なお、本章での用語の説明は必要最小限にとどめ、第 2 章でより詳しく説明する。

1.1 本研究の意義

従来の情報処理は言語化あるいは形式化された情報を処理の対象とすることが多く、数値計算、記号推論などの自動化や対話的支援を行うことを中心に発展してきた。しかし、本来情報処理というものは、人間の頭の中から始まる一連の処理の流れとして捉えられるべきものであろう。その意味において、上流工程の支援が下流工程の支援に比して不十分であり、情報処理のネックとなってきたといえよう。例えば知識工学における知識獲得、ソフトウェア工学における要求分析、問題解決におけるブレインストーミングなどである。

情報処理の上流では、しばしば、断片的な情報が頭の中で未整理な状態にある。そのような状態の情報を全てテキストのような逐次的で厳密性が高いメディアで表現し操作することは困難なことが多い。それに対して、断片情報を要素とし、それらの間の関係を全体的かつ直截に表現し操作できる図の方が上流の工程においてはより便利なが場合が多い。このことは、経験的に開発され発展してきた KJ 法 [Kaw86] のような創造技法や発想技法において図表現を用いるものが数多く存在することにも現れている。

本論文で扱う領域連結複合系は、2 種類の関係を連結線と領域の包含により同時に表現する図のクラスであり、図の視覚的形式として一般的なものである。先にあげた KJ 法で利用される図も領域連結複合系に含まれる。領域連結複合系の図は思

考過程で求められる複雑な関係の表現に適した高い表現力を備えていると言えるが、その反面、編集や作成が困難であるという問題があり、そのため計算機による支援が望まれる。

DPは、直接的に上流工程の情報処理を行うものではないが、上流工程で有用と考えられる図(特に本研究では領域連結複合系の図)の編集、作成を支援するものである。DPを利用することにより図の編集、作成が効率化され、上流工程における情報処理の改善が期待できる。すなわち、DPを構築することの意義は、「情報処理上流工程のための基盤システムの提供」にあると言える。

1.2 本研究の位置付けと課題

本研究の応用領域は「思考支援」であり、そこで用いる情報技術は「情報の可視化技術」である。以下、思考支援における位置付けと、情報の可視化技術における位置付けを説明する。

1. 思考支援における位置付け。

人間の思考過程を支援する計算機システムの開発を目標とする研究を、Young [You89]は、計算機の果す役割によって、「生成」レベル、「枠組パラダイム」レベル、「秘書」レベルの3レベルに分類している。秘書レベルの支援システムは計算機が文房具など従来の道具から発展した道具として利用されるものであり、ワード・プロセッサ(WP)が代表的である。DPはテキストを処理するWPに対するものとして図を処理するシステムであり、WPと同様に秘書レベルとして位置付けられる。DPの構築は秘書レベルにおいて図を対象とする新しいアプローチと言える。

2. 情報の可視化技術における位置付け。

情報の可視化技術という研究分野には、ビジュアル・プログラミングやアルゴリズム・アニメーションといったプログラムの可視化や、サイエンティフィック・ビジュアリゼーションといった数値の可視化などがあるが、本研究における対象は関係情報、つまり頂点とそれらの間の関係として表現できる(あるいは、グラフとして抽象化できる)ような情報であり、グラフの自動描画や図の自動描画が主要な研究分野である。ただし、DPでは図を見るだけでなく編集も行なうので、可視化だけでなく操作も考慮した対話的な可視化技術が必要である。このような分野において、本研究で開発する技術は「領域連結複合系」に属する図を対象とするという点で新しいと言える。

本研究では、情報の可視化技術を思考支援に応用することでDPを構築する。技術的な詳細は第2章で述べるが、先に述べたような位置付けにおいて、DPの構築に関する課題は次の通りである。

- DP に必要とされる要素技術 (情報の可視化技術) の開発.
- 要素技術の統合による試用可能なシステムの実現.
- システムの評価.

1.3 本研究の成果

図を利用した思考支援という問題は、人間の視覚、認識、認知、思考、心理といった、まだ十分に解明されていない大変難しい問題に密接に関係している。したがって、科学的な研究対象として本質的な解明がなされるためには、今後多大な研究努力が必要であることは明らかである。現段階における本研究の成果としての筆者の主張は次のとおりである。

1. 図の自動描画技術の開発

従来、領域系、連結系の図の自動描画技術は存在したが、それらの複合系の図は扱われていなかった。本研究において領域連結複合系の図の自動描画技術を新たに開発した。この技術は目指す DP を実現するために必要な中核的基盤技術である。

2. ダイアグラム・プロセッサの開発

領域連結複合系図表現に基づく DP を開発した。その DP は、紙やペンなどの文房具に代わる、より強力な図の処理能力を提供することを特徴とし、領域連結複合系の図を扱えること、さらに作業を効率化するための高度な機能 (自動描画機能など) を備える点で新しい。

3. システムの評価

ユーザへのアンケート調査および新しく開発した評価法による評価実験を行った。アンケートからは、ユーザに受け入れられるかどうかを知る手がかりとなるユーザの使用感や満足度を得た。また、評価実験では作業効率に対する機能の客観的かつ定量的な有効性を示した。

1.4 論文の構成

続く第2章では、DP を思考支援システムとして位置付け、その構想について述べる。DP に対する可視化技術の応用について述べ、DP の構成および備えるべき機能を説明した上で、本研究における技術的な課題を述べる。

第3章では描画に関する従来研究について説明する。図の自動描画のための基礎技術である「配置技術」、「装飾技術」、「表示技術」に関連する研究、および「シ

システム化」に関わる研究について述べる。さらに実際に実現されている図を扱うシステムのうち関連の高いものも紹介する。

第4章では、「領域連結複合系」の図の配置技術について述べる。まず、領域連結複合系の図をグラフの拡張である「複合グラフ」で定式化する。そして、開発した複合グラフの自動描画アルゴリズムについて説明する。

第5章では、領域連結複合系の図の装飾技術について述べる。まず、図の見方に関するユーザの要求を整理する。その要求に対する図の装飾によるアプローチ、「図ドレッシング」について説明する。

第6章では、領域連結複合系の図の表示技術について述べる。まず、計算機で扱う表示対象に対して画面が小さすぎるという問題を説明する。そして、その問題に対する表示技術によるアプローチ、「多視点遠近画法」について説明する。

第7章では、開発した基礎技術の対話型システムへの統合について述べる。併せて開発したユーザインタフェース機能についても説明する。さらに、ユーザに各機能の使用感や満足度を問うアンケートを実施した結果を示す。

第8章では、開発したシステムの定量的な評価について述べる。まず、システムを定量的に評価するための方法について説明し、それに従って行った評価実験について述べる。また、現在使用されている汎用の作図システムとの比較も行なう。

第9章では、結論として本研究の貢献を整理すると共に研究としての今後の展開および課題について述べる。

第 2 章

ダイアグラム・プロセッサの構想

本章では、本研究の新規性、位置付けおよび技術的な課題を明確にするために、ダイアグラム・プロセッサ (DP) の構想について述べる。まず、思考支援システムとしての DP について説明し、つづいて、そのような DP に対する関係情報の可視化技術、つまり図の自動描画技術の応用について述べる。そして、実際に可視化技術を利用する DP の構成について説明し、DP の備えるべき機能を整理する。その上で、本研究における技術的な課題を述べる。なお、本章で述べる図をメディアとする思考支援の考え方は文献 [Yos77, IYA86, You89, Sug88a, Sug93a] などに基礎を置いている。

2.1 思考支援システムとしてのダイアグラム・プロセッサ

本節では、一般的な観点から「思考支援」の分類について説明し、思考支援システムとしての DP の位置付けを明らかにする。

「思考支援」の定義は人により多少の差異があるが、ここでは、いわゆる「発想」や「創造」と呼ばれる知的作業の上流で行なわれる、情報の (1) 収集、(2) 整理・組織化、(3) 表現、(4) 評価からなる情報処理を計算機支援することを思考支援と呼ぶことにする [Sug91]。このような人間の知的作業のいわゆる「上流工程」の計算機支援を目指す「思考支援」や「発想支援」に関する研究は近年盛んになってきている [Kun93]。

2.1.1 思考支援の分類

人間の思考過程に関連する研究はまずその目標によって大きく二つに分けられる。ひとつは人間の思考過程の科学的な解明であり、もうひとつは人間の思考過程を支援する計算機システムを開発することである。それぞれの研究成果は互いに活用できると期待されるが、目標としては大きく異なるものであることを認識しておく必要がある [You89, p. 262]。本研究では後者を目標とするが、前者を目標とする研

究としては、さまざまな経験的創造技法や計算機技術を始め大脳生理学、心理学、認知科学などを踏まえた研究が進められている [NA94a].

思考過程を支援する計算機システムの開発を目標とする研究は、計算機の果たす役割によって大きく三つに分けることができる [Hor92]. まず、計算機に人間の代役としてアイデアの生成をさせるかどうかによって二分できる。生成をさせないとしたら、計算機は道具という位置付けとなるが、それは文房具など従来からある道具を計算機内に構築することで発展したものとするか、それとも思考を支援するための新しい道具とするかによって分けられる。第1章でも簡単に説明したが、Youngはこのような分類を、「秘書」レベル、「枠組パラダイム」レベル、「生成」レベルのように名付け、思考支援のレベル分けとしている [You89].

「秘書」レベル: このレベルでは、計算機は文房具など従来の道具から発展した道具、すなわちアイデアの獲得、記録、表現のための便利な道具として利用される。計算機が思考過程におけるいわゆる雑用を軽減してくれることでユーザが思考活動に専念できるという意味で思考支援と言える。

「枠組パラダイム」レベル: このレベルでは、計算機は人間の思考に適した枠組やパラダイムを導入した新しい道具として利用される。計算機はユーザの思考目的に適した枠組を提供する。その枠組はアイデアの組織化を助けるだけでなく、思考展開の過程においてガイドの役目も果たす。

「生成」レベル: このレベルでは、計算機は人間の代わりに新しいアイデアを生成する。つまり、ユーザの入力の合成や連想により新しいアイデアを生成し提示する。生成されたアイデアを採用するかどうかの最終的な判断は人間が行なっても良い。

これらの三つのレベルは階層をなしていて、後ろのものほど支援の程度や洗練度が高い、すなわち上位のレベルと考えられる。しかしながら、支援が適切かどうかは単に支援レベルで決まるものではなく、問題に対する要求、ユーザの好みやスタイル、そして代償となるコストに対する経済的評価によって決定されるべきである。たとえば、Young自身が指摘しているように、上位レベルの支援が必ずしも効果的な支援であるとは限らない。また、下位レベルの支援が技術的に容易とは限らないということも重要である。つまり、支援のレベルは研究の意義や困難さという観点でのレベルとは独立である。

2.1.2 ダイアグラム・プロセッサの支援レベル

DPは「図」を編集・作成するためのツールである。ここではDPを思考支援システムの一つとして位置付けるが、先に述べたように、DPはYoungのレベル分けではWPと同じく秘書レベルの思考支援システムと言える。すなわち、計算機が文房具など従来の道具から発展した道具として間接的に思考作業を支援するもので、

WP がテキストに対する便利な道具であるのに対して、DP は図に対する便利な道具であると言える。

秘書レベルの支援は直接的な支援ではない。事実 WP が提供するものは文章を書くための様々な便利で高度な機能である。そのような機能により、たとえば、編集が容易にまた効率的になり推敲がしやすくなる、文章の再利用が容易になり生産効率が上がる、また誰でも練習なしで綺麗な書体が得られる。しかしながら、WP は文章そのものを生成する訳ではない。また、WP を使うことで必ずしも知的生産物として良質の文章が得られるわけでもない。

WP を思考支援システムとして捉えたときに、紙やペンなどの従来の道具よりも思考にとって本当に役立つかどうか、あるいは知的生産物の質が高いかどうかの科学的な証明はたいへん難しい問題だと思われる。究極的にはそのような検証が求められるかもしれないが、WP の普及している状況によって WP の有用性は十分に示されているとも考えられる。WP の普及は基本的には作業の快適さや効率を指向した高度な機能にあると考えられるため、WP や DP のような秘書レベルのシステムにおいては、まずは思考そのものよりも思考過程における作業の快適さや効率をどの程度改善するかを評価の対象とすべきである。

2.2 関係情報の可視化技術の応用

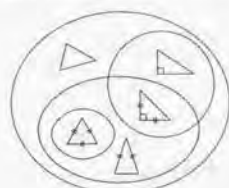
本研究では DP を一つの思考支援システムとして位置付ける。そして、関係情報の可視化技術、言い換えると図の自動描画技術の応用として、そのようなシステムの構築を目指す。では、なぜ図の自動描画技術が必要とされるのか、どのような自動描画技術が DP に必要とされるのか、本節では、まず DP の処理対象である図について説明し、図の性質および思考過程での図の使い方から自動描画技術が必要な理由を述べる。そして、DP における可視化技術の役割および DP にとって望まれる可視化技術について説明する。

2.2.1 図の基本系と領域連結複合系

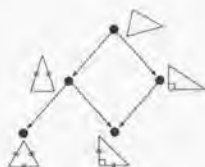
点、線、面などの図形を文字と共に平面上に配置した「図」は、古くから情報の表現、伝達、記録あるいは思考の道具として数多く用いられている。

現在、たとえば、工学の分野でも工程図、システム構成図、フローチャート、関連樹木図、論理回路図、PERT 図、ER ダイアグラム、組織図など、様々な図が利用されている。これらの多くは、基本的には要素を表す頂点とそれらの関係を表す辺からなるグラフを、点などの図形とそれらをつなぐ連結線により視覚的に表現したものである。つまり、グラフのような数学的概念を、意味単位(図素: 図を構成する基本要素)と構文法(図素の配置規則)からなる表現形式に基づいて可視化したものといえる。

図はその視覚的な表現形式に着目するといくつかの基本系に分類することができる。出原らの分類 [IYA86] では下に説明する領域系、連結系、配列系、座標系の4種類の基本系に分類される(図2.1参照)。これらは、集合、グラフ、行列、座標といった数学的概念にそれぞれ対応するものといえる。



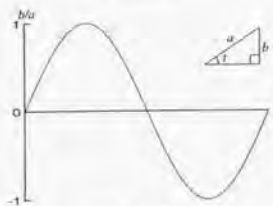
(a) 領域系



(b) 連結系

Triangle	○	○	○	○	○
Isosceles			○	○	○
Regular				○	
Right		○			○

(c) 配列系



(d) 座標系

図2.1: 図の基本的な視覚表現形式(基本系)

領域系: 領域を閉曲線で囲って表す図で、集合を表すベン図がその代表である(図2.1(a))。閉曲線がもつ平面分割機能を利用し、任意の形状の閉曲線を主要な意味単位とし、配置規則として閉曲線相互の包含、交差、分離、隣接関係がそれぞれ意味をもつ。

連結系: 頂点と呼ばれる点などの図形と、それらの間を連結する線とによって表される図である(図2.1(b))。線がもつ連結機能を利用した「辺(エッジ、アーク)」と呼ばれる線分と、「頂点(ノード)」と呼ばれる点(または図形や文字)を意味単位とし、配置規則として頂点間を連結する辺の有無が意味をもち、頂点と頂点との間の関係を表す。

配列系: いわゆる行列の形で表される図で、一般に「表」と呼ばれるものである(図2.1(c))。線がもつ平面分割機能を利用して、平面を横に分割する行線と縦に分割する列線を主要な意味単位とし、配置規則としては、行と列との交差関係が意味をもち、交差する行項目と列項目との関係を表す。

座標系: 座標によって位置や数量を表す図で、関数のグラフ、ヒストグラム、パイチャートや地図などが含まれる(図2.1(d)). 点があつて位置決め機能と、線があつて量表示、方向表示、運動軌跡表示などの機能を利用し、座標軸を示す軸線、位置を示す点、数量や方向や運動を表す線分を主要な意味単位とし、配置規則として座標軸が示す場における点や線の大小関係、位置関係が意味をもつ、

基本系はこのように分類されるが、現実には、扱うべき情報の大規模化、複雑化に伴って、基本的な表現形式だけでは情報をモデル化し人間の思考を助けることが難しくなっている。そこで、複雑な情報を表現できるように図の表現力を強化するために、基本系を二つ以上組み合わせた複合系の図も多く利用されている。

複合系のうち、領域系と連結系を組み合わせた複合系を領域連結複合系と呼ぶ。図2.2は領域連結複合系の図の例を示したものである。(a)はHarelによって提案された Higraph [Har88] の例で、デジタル腕時計の状態遷移を表現したものであり、領域系と連結系を組合せることで複雑な状態遷移を簡潔に表現することを可能にしている。(b)は発想法として広く知られている KJ 法で利用される図の例である。

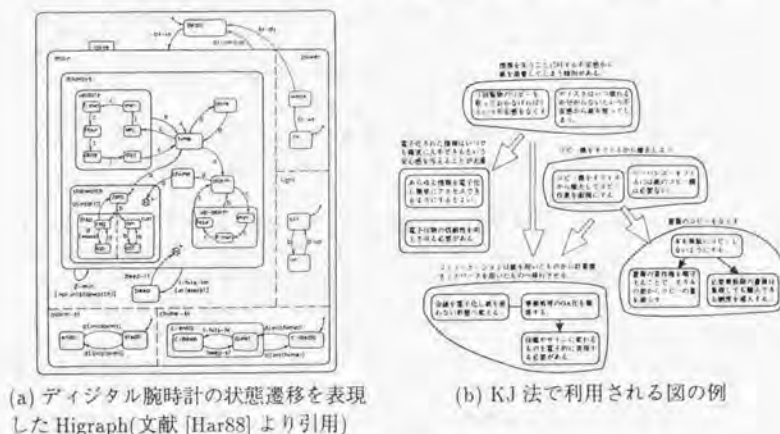


図2.2: 領域連結複合系の図の例

2.2.2 可視化技術の必要性

ここでは、可視化技術の必要性、言い換えると上で説明したような図の自動描画のための機能の必要性について説明する。

可視化技術が必要とされるのには大きく分けて二つの理由がある。

第1は、図の編集を容易にするためである。図の編集は不便で面倒、少し大きく複雑になると人手に負えないという問題がある。領域連結複合系の図は領域系と連結系の複合系であるために、特に作成・編集が困難と言える。

第2は、図に関する様々な操作の自動化を可能にするためである。計算機によって図の自動変形や自動変換などの機能を提供するためには、その結果を図として自動的に描けなければならない。つまり、操作の自動化を可能にするためには自動描画機能が不可欠と言える。

以下、これらの理由についてもう少し詳しく説明する。まず第1の理由に関して、図の編集における問題点を述べ、つづいて第2の理由に関して、自動化が求められる操作の例として思考を助成する編集操作について述べる。そして、その後、DPのための可視化技術について説明する。

図の編集における問題点

図の編集における問題点は、図の編集というものが不便で面倒で、図が少し大きく複雑になると人手におえないということである。言い替えると、図の編集に対して人間の描画技術や能力が不十分であり、計算機の助けが求められるということである。

図の編集とテキストの編集との決定的な違いは、テキストが基本的には1次元であるのに対して、図は2次元的な広がりを持っていることである。1次元的な文字の並びに対する編集操作は追加、削除、挿入、置換などいずれも比較的単純なのに対して、2次元的な図素の広がりをもつ図はそのような操作が単純ではない。

さらに図の編集を面倒なものにしているのがレイアウトの重要性である。たとえば、図2.3において、(a)の図では三角形の概念の階層関係がひとめで分かるのに対して、(b)の図では何を表わしているのかすぐには分からない。このように、レイアウトの良い図は、それが表現する意味を見る者に的確に伝え理解を助けるが、良くない図は逆に混乱や誤解を招く可能性がある。つまり、図の編集においては常に良いレイアウトを行なう必要がある。紙の上でテキストを修正する場合には、挿入を表わす記号などと共に欄外に修正文を書いたりするが、図を編集する場合には、レイアウトを気にしないそのような修正はたとえ行なっても有効ではない。

上流工程で図を利用する場合には、頭の中の情報を図として表現することを目指して行なわれる。頭の中の情報は初めは未整理で曖昧であるため、思考を進めながら整理や組織化あるいは精緻化を行ない図を完成させていくことになる。ただし、思考の展開過程は試行錯誤を伴うため、図を修正する作業も避けられない。つまり、思考過程において行なわれる図の編集では思考の進展とともに図が繰り返し描き替えられる。つまり、2次元的な広がりやレイアウトの重要性が、図の編集を不便で面倒なものにしているが、このように不便で面倒な図の描き替えを、思考過程で図を利用する場合には、さらに何度も繰り返し行なう必要がある。

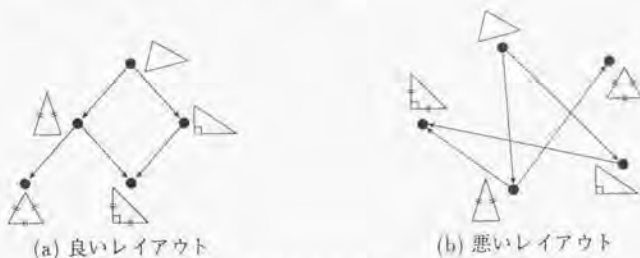


図 2.3: 良いレイアウトと悪いレイアウト

思考を助成する編集操作

図の編集作業では、基本的には一般的な作図と同様に、図素の付加、削除、移動といった修正操作が行なわれる。さらに思考過程では、単なる修正ではなく思考を助成するための次のような操作も積極的に行なわれる。

変形操作: 意味単位および構文法から導き出される「意味を変えない変形規則」に従う操作。たとえば連結系の図であれば、頂点の位置の変更や、辺の形状の変更は意味を変えない変形操作である。

同じ意味を表す図でも辺がむやみに交差しているよりも交差が少ない方が分かりやすいことが多い。頂点が規則的に並んでいるとさらに分かりやすいかもしれない。レイアウトの良い図、言い替えると美しく見やすい図は、それが表現する意味を見る者に的確に伝え理解を助けるが、良くない図は逆に混乱や誤解を招く可能性もある。このように、意味の理解のしやすさや、特定の意味の強調などに結び付くという点で、図の変形操作は、数式や論理式の変形、あるいは文章の修辞術にたとえることができる。その主要な効果は隠れていた構造や意味を顕在化することであり、それによって、意味の発見や理解あるいは思考の展開に結び付く可能性がある。

翻訳操作: 表現形式を変換する操作。基本的な表現形式はそれぞれ、意味を表す上で共通する機能と異なる機能を備えている。そのため、共通する機能で表されている意味を保存したまま、表現形式を変換することができる。

問題の解決においては表現の方法が思考の目的に合っているかどうかが重要である。図の表現形式は「考え方」を規定してしまう性質があるため、思考過程においては、思考の目的に合った表現形式をうまく選ぶ必要がある。たとえば、図 2.1(a) は三角形の概念の階層関係を表したものであるが、表現形式の異なる (b) や (c) の図でも同じ階層関係が表現されているといえる。ただし、(c) で直接表現されているのは階層関係ではなく各種の三角形が備えている性質

であり、階層関係はその性質から間接的に読みとられる。つまり(c)では階層関係の背後にある性質が強調されていて、この図を用いる場合には性質との対応を意識せざるを得ない。図の表現形式を変換する翻訳操作は、情報が未整理で曖昧な段階で適切な表現を模索する、あるいは観点や考え方を考えるために効果的である。

思考過程での図の編集作業においては、これらの操作が自在に行なえることが重要である。さらに計算機による支援システムでは、図に対するこれらのような様々な操作を自動的に行なう機能を提供してくれるとありがたい。

2.2.3 ダイアグラム・プロセッサのための可視化技術

図の編集作業における図の自動描画技術の必要性を二つの観点から述べたが、自動描画の機能を備えていない作図システム(あるいは、いわゆる「お絵描きソフト」)は従来から数多く存在した。ただし、そのような作図システムは、WPにたとえりて挿入機能(インサート・モード)を備えていないWPである。

挿入機能を備えていないWPは大変不便である。文章中にたった1文字追加するためにも、一々そのための場所を確保するためにそれより後の部分をコピー・ペーストあるいは移動しなければならない。もし移動に対して行末の文字を次の行へ送るという機能も無ければ、段落の終りまで各行ごとにそのような操作を繰り返さなければならないのである。挿入機能はWPにとっては、ごく基本的であるが重要な機能であり、もし備えていなければ、WPはこれほどまで普及していなかったに違いない。

図の編集において新しい図素の挿入に対してその場所を確保することは、図素が追加された新しい図を再描画することに他ならない。つまり、WPにおけるごく基本的な機能である挿入機能に相当するものをDPにおいても備えるためには図の自動描画機能が必要である。もちろん、図素の挿入の結果が図素同士が不必要に重なった「悪いレイアウト」の図で構わなければ再描画のための機能は必要ないが、悪いレイアウトが良くない理由は既に説明した。また、先に述べたように様々な自動機能を提供するためにも自動描画機能が必要であるが、それ以前に、自動描画機能はDPをごく一般的なWPレベルにするために欠かすことのできない基礎的かつ重要な機能と言える。

DPのための自動描画機能は人間に見せるための図を描くことが目的であり、良いレイアウトの図を作成することが重要である。どのような図でも構わなければ図の自動描画は簡単であるが、良いレイアウトの図を描くことは一般に難しい問題である。従来、このような描画機能のための「美しい図を自動的に描く」技術はほとんど開発されていなかった。また、たとえば、グラフを連結系の図として描く自動描画は、回路の自動レイアウトなどと類似の問題であるが、目的、最適基準、制約条件などにおいて全く異なるため、人間に見せるための自動描画機能には回路の自動レイアウト技術は適さない[Sug93a]。したがって、自動描画技術(すなわち、関

係情報の可視化技術)を開発することがDPの構築において最も基礎的で重要といえる。

2.3 ダイアグラム・プロセッサの構成

図を視覚的メディアとした計算機と人間の情報交換の一般的な枠組として、計算機システムが対象とする応用情報から画面上の図までをいくつかのレベルに分け、それらの間の情報の流れを表現した、可視化パイプラインと呼ばれるモデルがいくつか提案されている [Kam88, Lin93, SW93]。このような可視化パイプライン・モデルを利用することで、情報の可視化のための機能を系統的に分割することが可能となり、システム実現に必要な技術を要素レベルへ細分化して検討できる。

本節では、目指すDPを可視化パイプライン・モデルに基づいて構成し、DPの高機能化について検討する。

2.3.1 図を構成する情報

我々は一つの図から多様な情報を読みとることができる。そのような情報は一般にいくつかのレベルの情報に分けて考えることができる。すなわち、図はいくつかの異なるレベルの情報によって構成されているといえる。ここでは、次のような4レベルに分けて考える。

応用情報: いわゆる「図の意味」。表現形態は特に限定されない。

論理情報: 応用情報のもつ論理的な構造を抽出したもの。応用情報の各概念に対応する論理的オブジェクトとそれらの間の論理的関係から構成される。

幾何情報: 論理情報に、色、形、サイズのような視覚属性と、位置や経路のような位置属性を与えたもの。論理情報の各構成要素に対応するものを幾何的オブジェクト (または図素) と呼ぶ。

表示情報: 実際に紙や画面の上に描かれた図。幾何的オブジェクトが紙や画面の上の実際の座標系に描かれるが、そのとき、対応する応用情報を表すテキストや画像などのラベルも共に描かれる。これによって我々は図を見ることができる。

図2.4はプログラムを構成する関数間の呼び出し関係を表現した図を例にして、図の4レベルの情報について説明したものである。ただし、上流工程においては、応用情報は人間の頭の中にある情報であり、プログラムなどとは異なり未整理で曖昧であることが多い。

```

int a(int x)
{
    b(x); c(x);
}

int b(int y)
{
    d(y); e(y);
}

```

(a) 応用情報

テキストで表現されたソースプログラム。

$G=(V, E)$
 $V=\{a, b, c, d, e\}$
 $E=\{(a,b), (a,c), (b,d), (b,e)\}$

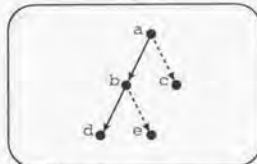
(b) 論理情報

関数を頂点 (論理的オブジェクト) で、呼び出し関係を辺 (論理的关系) で表現した有向グラフ。

a: (10, 20)	(a,b): solid
b: (5, 10)	(a,c): dotted
c: (15, 10)	(b,d): solid
d: (0, 0)	(b,e): dotted
e: (10, 0)	

(c) 幾何情報

有向グラフの頂点を表す点や、辺を表す線分、およびそれらの色、サイズ、位置または経路など。



(d) 表示情報

有向グラフを実際に画面に描いたもの。関数を表す点の側に関数名がラベルとして描かれている。

図 2.4: 図を構成する 4 レベルの情報
プログラムを構成する関数間の呼び出し関係を表現した図による例。

ダイアグラム・プロセッサ

ユーザ



図 2.5: 可視化パイプラインに基づくダイアグラム・プロセッサの構成図

2.3.2 可視化パイプラインに基づくダイアグラム・プロセッサの構成

図2.5は先に示した4レベルの情報からなるパイプライン・モデルを用いてDPの機能的な構成を表現したものである。図中の矢印は、右向きがユーザに図を提示するための可視化プロセスを、左向きはユーザが図の情報を変更する操作プロセスを表している。それぞれのプロセスは次のような機能により構成される。

配置機能: 論理情報を幾何情報に変換するための部分機能。論理的オブジェクトおよび論理的関係の各々に対応する幾何的オブジェクトの位置属性を決める。

装飾機能: 論理情報を幾何情報に変換するための部分機能。論理的オブジェクトおよび論理的関係の各々に対応する幾何的オブジェクトの視覚属性を決める。

表示機能: 幾何情報を表示情報へ変換するための機能。幾何的オブジェクトを実際の画面上(あるいは他の出力装置)に表示する。

修正機能: 論理情報および幾何情報をユーザが修正するための機能。ユーザの操作に従って論理的オブジェクト、論理的関係、幾何的オブジェクトの変更(追加や削除を含む)を行なう。

2.3.3 ダイアグラム・プロセッサの高機能化

DPは上で説明した機能も含めて様々な機能を持ちうるので、高機能化の次元を多数考えることができる。表2.1は代表的な高機能化の次元を一般的に示したものである。以下、それぞれの次元について説明する。

対象図

表現力のより高い図を対象として扱えるようにすることで、応用情報の忠実な表現、応用範囲の拡大、などが可能になる。図の表現力の高め方としては、視覚表現形式の拡張とラベルのメディアの拡張がある。表現形式の拡張は領域系や連結系のような基本系から領域連結複合系のような複合系への拡張が考えられ、メディアの拡張はテキストから、静止画、さらには音声、動画といったマルチメディアへの拡張が考えられる。

描画処理

図の描画処理(または可視化処理)を高度にし、手作業による図の作成や変形などの面倒な作業からユーザを解放することで、図の編集作業を容易に、また効率的にできると考えられる。描画処理の高度化には、配置機能、装飾機能、表示機能の高機能化がある。配置機能の高機能化としては、ユーザの責任によるマニュアルでの配置だけでなく、論理情報と幾何情報との整合性を保つ自動修正、さらには論理

表 2.1: ダイアグラム・プロセッサの高機能化の次元

次元		高機能化の向き	
対象図	表現形式	基本系	→ 複合系
	メディア	テキスト	→ マルチメディア
描画処理	配置機能	マニュアル	→ 自動修正, 自動配置
	装飾機能	マニュアル	→ 自動装飾, 動的装飾
	表示機能	拡大縮小	→ 視点・画面考慮
対話処理	修正機能	コマンド言語	→ 直接操作
	応答機能	図のみの表示	→ 予告, 過程, 結果
その他	履歴機能	undo	→ 履歴保存, 再現, 逆行
	通信機能	無	→ プログラム間, WS 間
	⋮		⋮

情報から幾何的オブジェクトの位置属性を決定する自動配置などが考えられる。装飾機能の高機能化としては、ユーザがマニュアルで幾何的オブジェクトの視覚属性を個別に変更するのではなく、ユーザの編集操作に合わせて自動的かつ動的に視覚属性を変更することが考えられる。表示機能に関しては、一様な拡大縮小だけでなく、図に対するユーザの視点を基に表示情報に要求される質や量、または表示される画面のサイズや解像度を考慮して表示することなどが望まれる。

対話処理

計算機とユーザの対話のための処理機能を高度にすることも重要である。対話処理は、ユーザの入力操作に対する修正機能とユーザへの情報出力のための応答機能からなり、修正機能の高度化により作業の単純化、効率化が図れ、応答機能の高度化によりユーザの誤りの回避、認知的負担の軽減が期待できる。修正機能については、言語ベースのコマンドによる操作だけでなくマウスなどのポインティング・デバイスを用いた直接操作 [Shn83], あるいは例やジェスチャーを利用するさらに進んだ操作環境など [Koh91, Mye92, Zha93] がすでに提案あるいは実現されている。応答機能については、各操作時点での対象図を静的に表示するだけでなく、ユーザの操作に対して処理を予告する、処理や変化の過程を見せる、結果を直ちにフィードバックするなど、より豊富で動的な応答を行なうことも重要である。

その他

上記以外にも高機能化の次元としては他にも様々なものが考えられる。たとえば、履歴機能として、操作の単純な取消 (undo) 機能だけでなく、操作の履歴を保

存しかつ再現する機能を用意することは有効である。それにより、ユーザは作業のどの時点にでも戻ってそこから作業を再開できるようになる。さらに、思考過程の分析、それに基づくシステムの改良など研究上の効果も期待できる。また、通信機能によるプログラム間やワークステーション間の情報交換も有効である。ワークステーション間の通信により、DP が個人用だけでなく複数ユーザで共有できる環境になり協調作業などが可能になる。またプログラム間の通信により、他のシステムやツールと連携した使い方が可能になり行なえる作業の幅が広がると考えられる。

2.4 技術的課題

本節では、先に説明した DP の構想において、高機能化を実現するために必要な技術のうち本研究で開発を目指すものについて説明する。

本研究では図を扱う上で最も重要と考えられる描画機能の高度化に焦点を合わせる。描画処理の対象としては、強力な視覚表現形式として領域連結複合系を対象とする。領域連結複合系を対象として選択した理由は次のとおりである。

- 領域系と連結系はどちらも論理的な関係を視覚的に表現するのに適している。それらの複合系である領域連結複合系では異なる関係を別の形式で表現できるため、複雑な関係の表現に一層適している。
- 領域連結複合系の図を実際に思考過程で用いる手法として創造的思考の「KJ法」[Kaw86]や論理的思考の「存在グラフ」[Koj79]などがある。これらは上流工程における領域連結複合系図表現の有用性を裏づけていると考えられる。
- 座標系や配列系の図は形態がほぼ決まっていて描画が比較的容易である。それに対して、領域系や連結系の図の描画は難しく、それらの複合系である領域連結複合系においては描画が一層困難であるため計算機による支援が望まれる。

図の自動描画技術は領域連結複合系に関してはこれまで全く開発されていなかった。そのため本研究において新しく開発する必要がある。具体的にはまず領域連結複合系の図に関して配置技術、装飾技術、表示技術などの基礎技術を開発する。さらにそれらの基礎技術を統合してシステムを実現し、評価することも重要である。それらの各段階における目標は次のとおりである。

配置技術

領域連結複合系の図の自動配置技術を開発する。領域連結複合系の図を美しく見やすい図として描くための配置技術はこれまで全く研究されていなかった。そこで、まず領域連結複合系の図の論理情報をグラフを拡張して定式化する。また、図の美しさ見やすさの基準も形式的に定める。そして、その基準を満足するグラフの自動描画アルゴリズムを開発する。グラフ描画に関する描画規則の多くは最適解を求め

ることがNP完全であることが分かっているが、発見的解法を用いることで対話型システムの使用に耐えるような高速なアルゴリズムを目指す。

装飾技術

領域連結複合系の図の自動装飾技術を開発する。図の「見方」に関するユーザの要求を視覚属性に関して整理を行ない、視覚属性の決定のために使われる情報および決定の仕方を形式的に表現することを試みる。そして、それを基に図の「見方」に対して視覚属性を自動的に決定する方法を開発する。

表示技術

領域連結複合系の図の表示技術を開発する。幾何情報は大きさや解像度などの制限がない仮想的な2次元平面に描かれた図と見なすことができる。そのような図を有限の大きさや解像度しか持たない画面上にいかに効率的に写すかが問題である。そこで、限られたサイズの画面において図の重要な性質の一つである一覧性をいかにして達成するかを議論し、そのために有効な表示方式を開発する。まずユーザの立場から表示方式への要求を明確にし、それらの要求を満たす表示方式を開発する。

システム統合化

個別に開発した要素技術に対話型システムとして統合する。実際に対話型システムを実現するためには上で述べた基礎技術に加えて、論理情報や幾何情報の編集機能なども必要であり、それらを含むユーザ・インタフェース技術なども併せて開発しなければならない。その際には、メディア、対話処理、その他の機能の高度化についても考慮する必要がある。開発に際しては次の項目に重点をおく。

- 図を利用する思考過程の対話型支援システムとする。
- 図の自動描画技術を利用した高機能図インタフェースを備える。
- ユーザの手間と認知的負担を軽減できる環境を用意する。
- 計算機の進んだ能力を活用することで新しい可能性を目指す。

システムの評価

本研究で開発しようとしている技術の対象は人間であるため、客観的には人間に与えた効果を評価する必要がある。しかしながら、このような評価はこれまで不十分であったため、その方法も手探りの状態である。そこで、本研究においては思考支援システムの評価法に関して新たに枠組を考案する。そして、試作したシステムに関して複数の方法で定量的あるいは定性的のような異なる観点から評価を試みる。

第 3 章

描画に関する従来研究

本章では図の描画に関連する従来の研究について説明する。DP を構築するに当たって筆者らが重点をおく、配置技術、装飾技術、表示技術の各基礎技術に関連する研究、およびそれらのシステム化に関わる研究について述べる。さらに実際に実現されている図を扱うシステムのうち関連の高いものも取り上げる。

3.1 図の配置技術

図の配置技術に関しては、グラフの自動描画法として多くの研究がなされている [BET93, Sug93a]。本節ではそのような数多くの研究の中から代表的なものを選んで簡単に説明する。

3.1.1 連結系

グラフの辺を線で描く自動描画法はすべて連結系の図の配置技術と見なすことができる。そのような自動描画法は対象となるグラフによって様々に分類することができるが、ここでは大きく無向グラフと有向グラフに分けて説明する。

無向グラフの自動描画アルゴリズムとして代表的なものに Eades [Ead84] が開発した「スプリング・アルゴリズム」がある。グラフをその辺がスプリングで構成される仮想的な物理モデルとみなしてその安定状態を見つけることでグラフの配置を求めるものである。グラフの対称性を表現しやすいことを特徴とする。スプリングの割り当て方や力の定義を変更した改良版もいくつか提案されている [Kam88, FE91, SK94]。さらに最近では、スプリングに加えて磁場を導入することで有向グラフあるいは無向辺と有向辺が混在したグラフ (混在グラフ) [Mis89] へも適用可能な「マグネティック・スプリング・アルゴリズム」[MS94b, SM95b, SM95a] も開発されている。

有向グラフの自動描画アルゴリズムとして最も利用されているものの一つが Sugiyama ら [STT81] によって開発された「Sugiyama アルゴリズム」である。このアルゴリズムは有向グラフのすべての辺の向きを揃え交差が少なくなるように頂

点を階層的に配置するものである。グラフが表す構造のもつ「流れ」が表現しやすいことが特徴である。Sugiyama アルゴリズムの変形としては、アルゴリズムを構成するサブステップを別の手法で置き換えるとともに辺の描画に曲線を利用したものの [GNV88] や、分割統治法を用いて大規模な有向グラフの高速処理を可能にしたものの [Mes88] がある。第4章で説明するアルゴリズムは Sugiyama アルゴリズムを領域連結複合系へと拡張したものである。

3.1.2 領域系

グラフを領域系の図として自動配置する研究は連結系に比べるとごく僅かしかない。代表的なものとしては、Mäkinen [Mak90] によって、ハイパーグラフ¹の一つの表現法として「サブセット標準²」と呼ばれる形式での自動描画法が議論されている。また Eades ら [ELL93] によって有向木の表現の仕方の一つとして「包含規約」、つまり頂点を長方形で、親子の関係を長方形の包含関係で描く方法が提案されている。

3.2 図の装飾技術

装飾技術が扱う視覚属性に関しては Bertin [Ber82] によって、サイズ、濃淡、きめ、色、方向、形、などが取り上げられ、機能や特性の分析が行なわれている。

視覚属性の決定に利用する基準を与えるものとしては、Furnas [Fur86] によって提案された関心度関数 (DOI 関数) が重要である。関心度関数は対象の特定の部分を見ることに対する人間の要求の度合を、視点からの距離と対象の潜在的な重要度から求められる「関心度」と呼ばれる量として与えることができる。この関数は大量の情報を狭い画面において効率的に表示することを目的として魚眼レンズのアナロジーに基づいて開発されたものであるが、一般化された考え方であるため視覚属性の決定を始め様々な応用が可能である。Furnas によって与えられた関心度以外にも、木構造も対象に対してはフラクタルの概念に基づいた値を用いる方法などが提案されている [KI92]。第5章で説明する「重要度」は Furnas の関心度の変形として捉えることができる。ただし、情報のレベルに合わせて応用情報における重要度と論理情報における重要度を分けて扱っていること、および必ずしも重要度がユーザの視点に依存するものではないことが特徴である [Mis94]。

このような値を視覚属性の決定に利用した例としては、Sarkar [SB91] らによってグラフの自動描画において頂点のサイズや可視性をユーザの視点に基づいて自動的に決定する研究がなされている。また、小池ら [KI92] は、そのような値を頂点のサイズや輝度に対応させることを示唆している。

¹ 辺が通常の2項関係から n 項関係へ拡張されたグラフ

² ベン図のように領域系の図として描く形式。連結系の図として描く形式は「エッジ標準」と呼ばれる。

Furnas によって提案された関心度は、一般化された考え方であるためグラフィカル・インタフェースに関する多くの研究に影響を与えている。関心度を図の装飾技術に利用する研究はむしろ少なく、次節で説明する表示技術に関する研究の方が盛んである。

3.3 図の表示技術

Furnas の関心度関数に先駆けて、Spence ら [SA82] はデータベース検索を目的とし、全体的な概略情報と部分的な詳細情報を組み合わせる手法 bifocal display を提案した。これは1次的に配列された情報を木構造によっていくつかの詳しさのレベルに分けて管理し、視点部分をより詳しいレベルで、かつその前後(周辺)をあまり詳しくないレベルで同時に表示するものである。

Furnas 以後、魚眼視に基づく表示方式の研究が盛んになり、筆者らの提案した多視点遠近画法 [MS89a, MS90, SM90, MS91b, MS91a, MEL95] 以外にも、Hollands ら [HCMM89] による地下鉄の路線図を対象にした魚眼視とスクローリングの性能比較実験や、Leung [Leu89] による地図を対象にした表示方式の考察などが行なわれた。また Mackinlay ら [MRC91] により Perspective Wall が開発された。これらは Furnas によって [Fur86] において示された関心度をもとにして表示対象をフィルタリングする方式とは大きく異なり、まさに特殊なレンズを通して対象を歪めて表示するものである。Leung ら [LA93] は後者の方式、すなわち「歪方式」を利用する写像をそれらの性質によって分類している。

第6章で説明する多視点遠近画法も「歪方式」の一つである。特に、多視点遠近画法の一つ二様表示方式は bifocal display を2次元平面上の対象へと拡張したものとして捉えることができる。しかしながら、複数の視点を指定可能な点も bifocal display にはない特徴である。

大きな対象を狭い画面上に効率的に表示することを目的とするが、魚眼視とは異なるアプローチとして大岩ら [OKK90] は対象の縮小全体像と部分拡大像を色を変えて重ね合わせる方式を開発した。

3.4 システム化技術

3.4.1 視覚情報のパイプライン

視覚的対話型システムが扱う情報の流れを表現したパイプライン・モデルがいくつか提案されている。Kamada [Kam88] はテキストデータを可視化して図として表示するための、またはその逆として平面上に描かれた図からテキストデータを合成するための情報の流れをパイプラインとしてモデル化している。Lin [Lin93] は Kamada のパイプラインを基にオブジェクト指向の概念を取り入れ一般的な DUI(図

的ユーザ・インタフェース)のスキーマを提案している。Scottら [SW93]はKamiadaのパイプラインの可視化プロセスを細分化しより詳細にしたモデルを提案している。第2章で示した筆者らのパイプライン・モデルはScottらのものに似ているが、DPが応用情報を持たないため、前半のモデリングのプロセスが存在しない。

3.4.2 ユーザ・インタフェース技術

ユーザ・インタフェース技術として最も重要な考え方の一つはShneiderman [Shn83]によって提案された直接操作であろう。現在では直接操作環境はごく一般的に広く利用されている。さらに直接操作を越えるものとして、操作の実際の利用分布を利用するもの [Koh91]、デモを利用するもの [Mye92]、ジェスチャーを利用するもの [Zha93]などが提案されている。また操作の進歩と共に情報の視覚的な提示法も3Dやアニメーションを利用するものにまで発展している [RCM93]。第7章で説明するD-ABDUCTORも標準的な直接操作インタフェースを備えている。その直接操作環境に加えて、ユーザの操作に対して間接的に起こる(自動配置機能が働くことでユーザの予想とは関係なく起こり得る)図の変化を動画を利用することでユーザに変化の様子を分かりやすく示すアニメーション環境を統合していることが特徴である。

そのようなユーザ・インタフェース開発のための技術は一般に大きく、ツールキットとユーザ・インタフェース開発システム(UIDS: User-interface Development System)に分けられる [Mye89]。ツールキットはユーザ・インタフェースを構築するために使われるプログラムの部品群を関数や手続きあるいはデータ形で集めたライブラリである。それに対して、UIDSはインタフェースの様々な側面をプログラマが構築し管理することを支援する統合化ツールである。第7章で説明するD-ABDUCTORの開発には、Xウィンドウ・システム [Jon89]のアプリケーション開発用ツールキットであるXView [Raa91]を利用した。XViewは画面デザインや機能を規定した一つの規格であるOPEN LOOK [Sun90b, Sun90a]に準じたユーザ・インタフェースを効率的に開発するためのツールキットである。

3.5 図を扱う対話型システム

図を扱う対話型システムは既に数多く存在する。たとえば、MacDraw [Cla91]、SuperPaint [Sil91]、Illustrator [Ado93]、xfig³などは、ドローイングエディタと呼ばれる汎用の作図システムである。これらは既に市販されていたり、フリーソフトウェアとして流通していたりして実用に供されているものが多い。これらのシステムはいずれも基本的にはペンや定規に代わるツールを計算機上で提供するだけで、扱え

³Xウィンドウシステムの上の作図ツール。作者: Supoj Sutanthavibul, University of Texas at Austin (supoj@sally.utexas.edu)

る図の種類は限定しない。つまり、どのような種類の図でも描くことができ、図を用いた思考の紙とペンに代わる操作環境として利用可能である。

ドローイングエディタの中でも、Key3 [MNN90]、DOGS [KSH94]、IntelliDraw [Sil92]、Igif⁴、などは上で述べた MacDraw などとは異なり図の論理情報を管理できる。それらは汎用で特定の種類の図を生成することを目指していないため、完全な自動配置機能は備えていない。しかしながら、ユーザが幾何情報を変更した場合に論理情報に矛盾しないように連動して幾何情報を自動修正する機能は備えている。これらは論理情報というよりも幾何的な制約条件が管理されていると考えることもできる。

自動配置機能を備えたシステムとしてはグラフエディタと呼ばれるものが数多く開発されている [Sco95]。たとえば、EDGE [PT90]、GraphEd [Him90, Him95]、DiagramServer [BBL92]、DaVinci [FW94]、SPREMB [EFK85] などは、いずれもあるクラスのグラフの自動描画機能を備えている。しかしながら、扱える図の種類としてはドローイングツールのように汎用ではなくグラフを論理構造とする連結系の図専用である。

思考支援を目的としたシステムも既に数多く開発されていて、その中には図あるいは図と見なすことのできる表現形式を利用するシステムも少なくない。AA1 [Hor94] は概念形成の支援を目的としたシステムで、概念を構成するキーワードを平面上に配置した座標系の図を利用する。論理情報であるユーザの入力するキーワード間の近さに基づいて図が自動的に描かれ、描き替えを繰り返すことで思考が進められる。AA1 と同様な座標系の図を利用するシステムに CAT1 [SHO94] や SCO/SCI [SHO93] がある。KJ エディタ [OKK90, KKO92] や ISOP [Cra93, Hir93] は、KJ 法で使われる領域連結複合系の図を編集できるシステムで、KJ 法におけるカードやグループ間の隣接関係や包含関係を論理情報として管理する。自動配置機能を備えていないので機能的には Key3 などと同じ分類に入るが、ドローイングエディタという観点からは描ける図の種類に制限があり汎用的ではない。CONSIST [Shi93] は KJ 法的なカード操作を基本とした知識整理支援を目的としたシステムでカード間の論理情報を計算機が管理し、ユーザが「視点」として指定した論理的な関係を領域連結複合系の図として提示できる。第7章で説明する D-ABDUCTOR は KJ 法を参考にして、領域連結複合系の図を編集できるという点では KJ エディタ、ISOP、CONSIST などと同類である。ただし、図を人にとって美しく見やすい図として自動的に描く機能を備えていることが特徴である。

⁴X ウィンドウシステムの上の作図ツール、作者:William Chia-Wei Cheng (william@sc. UCLA.edu)

第 4 章

図の配置技術

図の可視化プロセスにおいて論理情報から幾何情報を自動生成するためには、幾何的オブジェクトの配置、すなわち位置属性を自動的に決定する必要がある。この場合の、配置決定の目的は情報構造やモデル構造などの概念の表現や伝達が中心であり、LSI 配線などで考慮される物理的な制約よりも、人間にとっての図の美しさ見やすさ(可読性)が重要である。

第 3 章でも述べたように、このような観点からの研究は、連結系の図に関しては有向グラフと無向グラフ(特に平面グラフ)の自動描画法として広く行われている[BETT93]。しかしながら、領域系の図に関しては数少なく、さらに領域連結複合系の図に関しては行なわれていなかったと言えよう。そこで筆者らは領域連結複合系の図の論理情報を有向グラフの拡張である「複合グラフ」として定式化するとともに、可読性を描画規約と描画規則として定め、その可読性の基準に基づく複合グラフの自動描画法¹を開発した。

本章では、領域連結複合系の図の自動配置技術として、「複合グラフ」の自動描画法について述べる。有向グラフに対する自動描画アルゴリズムとしては、Sugiyama アルゴリズム(あるいは STT アルゴリズム)[STT81, Sug85, SM86, Sug87a, Sug87b]が有名であり広く利用されている。Sugiyama アルゴリズムの特徴の一つは、有向グラフの可読性を高めるためにグラフの頂点を階層的に配置することにある。ここで示す描画法は Sugiyama アルゴリズムの複合グラフへの拡張であり、基本的な戦略などを踏襲し、頂点の階層的な配置を行なうことを特徴とする[MS88a, MS88b, MS89b]。

以下、4.1 節で、まず描画対象である複合グラフの定義を行なう。4.2 節では、図の可読性の基準である描画規約と描画規則について説明する。4.3 節では、4 フェーズから構成される処理の順序に沿ってアルゴリズムを具体的に説明する。4.4 節では、実現したプログラムによる描画例を示す。

¹第 1 章では、「描画」を「配置」、「装飾」、「表示」を含めた広い意味で用いたが、通常は配置技術を指して「描画法」と呼ぶことが多い。本章ではこれに倣って狭い意味で「描画」を用いる。

4.1 描画対象

領域連結複合系の図の論理的構造を, グラフの拡張である「複合グラフ」として定式化する.

定義 4.1 (有向グラフ)

有向グラフ D を次のように定義する.

$$\begin{aligned} D &= (V, E, \partial^+, \partial^-) \\ V &: \text{頂点集合} \\ E &: \text{辺集合} \\ \partial^+, \partial^- &: E \rightarrow V \text{ (} E \text{ の結合関数)} \end{aligned}$$

辺 $e \in E$ は始点 $\partial^+(e)$ から終点 $\partial^-(e)$ に向いているとする.

定義 4.2 (複合グラフ)

複合グラフ G を次のように定義する.

$$\begin{aligned} G &= (V, E, \partial^+, \partial^-, \tau) \\ V &: \text{頂点集合} \\ E &: \text{辺集合} \\ \partial^+, \partial^- &: E \rightarrow V \text{ (} E \text{ の結合関数)} \\ \tau &: E \rightarrow T \text{ (} E \text{ の型関数)}. \end{aligned}$$

関数 τ は辺に型を与える. ここでは $T = (\text{整数})$ とする. $\tau(e) = 0$ のとき e を包含辺と呼び, $\tau(e) \neq 0$ のとき隣接辺と呼ぶ.

定義 4.3 (包含辺集合, 隣接辺集合)

包含辺, 隣接辺の集合をそれぞれ E_I, E_A で表す. つまり,

$$\begin{aligned} E_I &\stackrel{\text{def}}{=} \{e \in E \mid \tau(e) = 0\} \\ E_A &\stackrel{\text{def}}{=} \{e \in E \mid \tau(e) \neq 0\} \end{aligned}$$

とする.

複合グラフに関連して, 以下, いくつかの用語および記法の定義を行う. また, 描画問題の簡単化のために複合グラフに二つの制約を設ける.

定義 4.4 (包含グラフ)

複合グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ に対し, その隣接辺すべてを解放除去してできる有向グラフ $G_I = (V, E_I, \partial^+|_{E_I}, \partial^-|_{E_I})$ を G の包含グラフと呼ぶ. ただし, $\partial^+|_{E_I}, \partial^-|_{E_I}$ はそれぞれ関数 ∂^+, ∂^- の定義域を E_I への制限したものである.

制約 1

包含グラフ G_I は木²である。

定義 4.5

制約 1 を満たすとき G_I を包含木と呼ぶ。包含木 $G_I = (V, E_I, \partial^+|E_I, \partial^-|E_I)$ において、ルートを r で表す。2 頂点間の接続する頂点と辺の系列を路、路に含まれる辺の数を路の長さと呼ぶ。頂点 $v \in V$ からルート r までの路の長さを頂点 v の深さと呼び $\text{depth}(v)$ で表す。頂点 $v \in V$ の親を $\text{mo}(v)$ 、祖先を $\text{An}(v)$ 、子を $\text{Ch}(v)$ 、子孫を $\text{De}(v)$ で表す。親は頂点であり、祖先、子、子孫は頂点の集合である。祖先、子孫は自分も含むとする。深さ d (ただし、 $d \leq \text{depth}(v)$) を指定すると祖先は唯一定まるので特に $\text{an}_d(v)$ で表す。

制約 2

祖先や子孫との間に隣接辺は存在しない。つまり、任意の隣接辺 $e \in E_A$ に対して、

$$(4.1) \quad \partial^+(e) \notin \text{An}(\partial^-(e)) \cup \text{De}(\partial^-(e)).$$

制約 1 と制約 2 を両方満たす複合グラフを描画対象とする。以下、特に断らない限り複合グラフは両方の制約を満たすとする。

4.2 図的可読性

可読性、すなわち図の見やすさとして図的可読性と概念的可読性を考えることができる。図的可読性とは図のレイアウトだけに関係する可読性で、概念的可読性は図にする以前の概念整理に関係する可読性である。ここでは、図的可読性だけを扱う。図的可読性に影響する要因としては、(1) 描画規約、(2) 描画規則、(3) 規則間の優先関係、(4) 描画アルゴリズム、がある [Sug88b]。本節では (1)–(3) について、次節で (4) について述べる。

複合グラフの描画において、筆者らが採用した規約、規則、優先関係を表 4.1 に示す。規約 C1–C4 は、必ず満たされなければならない条件であり、形状や配置に関する。規則 R1–R5 は、見難さの要因を排除するための、また高い可読性を得るための目標である。描画規則は相互に競合し得るので規則間に優先関係を導入した。規約、規則、優先関係はすべて経験則として選択されたものである。

階層表現 (C4) が本描画法の最も特徴的な規約である。これは、階層表現が図の可読性を著しく高めるという考え方で採用された。その定式化については 4.3.1 節で詳しく述べる。

² G_I は森としてもよい。そのときは森のすべてのルートを包含するダミー頂点の一つ用意することで木として同様に扱うことができる。

表 4.1: 採用した描画規約, 描画規則, 優先関係

描 画 規 約	描画のスタイルに関する基本的約束
	C1 頂点は2辺が水平な長方形の領域として描く
	C2 包含辺は領域の包含関係で表現する
	C3 隣接辺は直線または折れ線として描く
	C4 階層表現をとる
描 画 規 則	描画に際して考慮する見やすさの基準
	R1 隣接辺の交差数の最小化
	R2 隣接辺と頂点の交差数の最小化
	R3 長い隣接辺の直線化
	R4 隣接頂点の近接化
	R5 隣接辺の対称配置
描画規則間の優先関係	
$R1 > R2 > R3 > R4 > R5$	

描画規約に基づいて描いた複合グラフの例を図 4.1 (a) に, 同じ複合グラフを描画規則も考慮して描いた例を (b) に示す. (b) の方が, 隣接辺の交差が少なく対称性も高く, 可読性の向上がみられる.

4.3 自動描画処理

自動描画処理の入力は複合グラフ, 出力は頂点の位置とサイズ, 隣接辺の位置と形状である. 処理は図 4.2 に示される様に, 複合階層化, 正規化, 順序決定, 座標決定の四つのフェーズからなり, この順に行われる.

各フェーズとも外向グラフの自動描画法 [STT81] における概念やアルゴリズムの拡張になっている. 包含木を基にした再帰的処理の各段階において, 頂点毎に適当な外向グラフを構成し, 外向グラフのための手法を利用する. ただし, 包含辺と隣接辺の複合性のため全体の処理はかなり複雑になる.

以下, 各フェーズについて処理の順に説明するが, 各フェーズごとに, まず問題の定式化を行い問題の性質を明かにし, そして実際に用いた方法について述べる.

4.3.1 複合階層化

描画規約 C4 に定められた階層表現のために, 各頂点がある条件の下で階層へ割り当てなければならない.

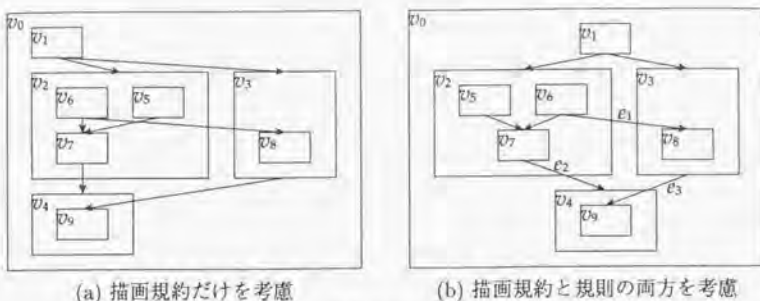


図 4.1: 複合グラフの描画例

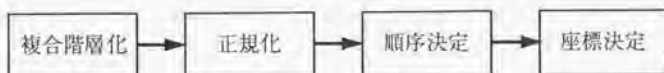


図 4.2: 自動描画処理の流れ

有向グラフの場合には、階層は水平な複数の直線である。これを隣接階層と呼び、上から順に $1, 2, \dots$ と隣接階層値と呼ぶ整数を与える。有向グラフの場合の様に隣接辺だけによる階層化、つまり隣接階層化は、隣接辺の始点と終点の間に定義される順序関係を満足するように各頂点に隣接階層値を割り当てる。これはポロジカル・ソート [Wir76] で、アルゴリズムは単純である。

複合グラフの場合には、描画規約 C2 に従って頂点間の包含関係を表すため、階層は図 4.3 に示す水平でネストになった帯状の層となる。これを複合階層と呼び、各階層に複合階層値と呼ぶユニークな数列を与える。ルート r が割り当てられる複合階層は複合階層値 (1) が与えられ、複合階層値 $(1, \dots, a, b)$ は、 $(1, \dots, a)$ が与えられた複合階層に直接含まれる複合階層の内、上から b 番目の複合階層に与えられる。各頂点に複合階層値を割り当てることを複合階層化と呼ぶ。以後、単に「階層 (化)」と記した場合は複合階層 (化) を表すとする。

階層化において満たすべき条件は描画規約の C2 と C4 に関するものである。C2 により、頂点 v には $mo(v)$ が属する階層に直接含まれる階層の階層値が割り当てられなければならない。C4 により、隣接辺が常に下向きになる (始点が終点より上に位置する) よう階層値が割り当てられなければならない。以上の条件の下で割当可能な複合グラフを複合階層化可能な複合グラフと呼び、割当の完了した複合グラフを複合階層グラフと呼ぶ。ここで扱う複合グラフは全て複合階層化可能であるとする。

以下、複合階層化可能な複合グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ についての議論とする。

まず、隣接辺に関する頂点間の順序関係を定義する。この順序関係を頂点の上下関係と見なすことで隣接辺は常に下向きに描かれる。この順序関係は深さの等しい

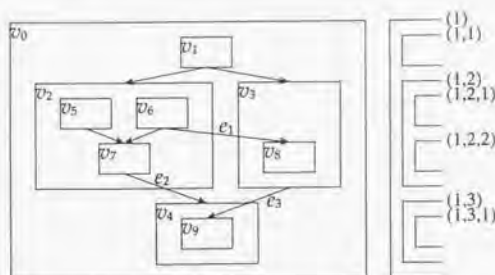


図 4.3: 複合階層と複合階層値

頂点間においてだけ定義される。

定義 4.6 (順序関係)

隣接辺 $e \in E_A$ に対して, 包含木における $\partial^+(e)$ から $\partial^-(e)$ への路上の頂点の系列が,

$$(4.2) \quad \partial^+(e) = p_m, p_{m-1}, \dots, p_1, t, q_1, q_2, \dots, q_n = \partial^-(e)$$

であるとする。ただし, t は系列上で深さ最小の頂点とする。このとき, e に関する頂点間の順序関係 ($<_e, \leq_e$) を次のように定める。

$$(4.3) \quad m > n \text{ のとき } p_n <_e q_n, p_i \leq_e q_i (i = 1, \dots, n-1)$$

$$(4.4) \quad m \leq n \text{ のとき } p_m <_e q_m, p_i \leq_e q_i (i = 1, \dots, m-1)$$

たとえば, 図 4.3 において頂点 v_2, v_3, v_4 の間には隣接辺はないが, 隣接辺 e_2, e_3 により, $v_2 \leq_{e_1} v_3, v_2 <_{e_2} v_4, v_3 <_{e_3} v_4$ の順序関係が定められる。

複合階層化は, 等しい複合階層値を持つ頂点の子集合に対する定義 4.6 の順序関係を用いた隣接階層化を根から葉へ再帰的に行うことでなされる。

たとえば, 図 4.3 において, まずルートの複合階層値を (1) とし, その子の集合 $\{v_1, v_2, v_3, v_4\}$ に対して隣接階層化を行なうと, v_1, v_2, v_3, v_4 の隣接階層値はそれぞれ 1, 2, 2, 3, 複合階層値はそれぞれ (1,1), (1,2), (1,2), (1,3) となる。次は複合階層値 (1,1) を持つ頂点集合 $\{v_1\}$ に子がないので, (1,2) を持つ頂点集合 $\{v_2, v_3\}$ の子集合 $\{v_5, v_6, v_7, v_8\}$ の隣接階層化を行う..., というように再帰的に処理を行う。

アルゴリズム 4.1 (隣接階層化)

隣接階層化アルゴリズムは関数 LA として下の様に表される。LA は, 複合グラフ G と, 親の複合階層値が等しい頂点の集合 W を引数とする。LA は定義 4.6 の順序関係を保存するよう, 頂点 $w \in W$ に対して隣接階層値 $al(w)$ を割り当て, 関数の値として総階層数 $d(W)$ を返す。(関数 LA は G が複合階層化可能なときだけ停止する。)

```

function LA( $G$ : 複合グラフ,  $W$ : 頂点集合): integer;
begin
   $\forall v \in W$  に対して,  $al(v) = 0$  とする;
   $\forall v \in \{x \in W | (\neg(y <_e x) \text{ for } \forall y \in W, \forall e \in E)\}$  に対して,
     $al(v) = 1$  とする;
  repeat
    for ( $\forall v \in \{x \in W | al(x) \geq 1\}$ ) do
      for ( $\forall w \in W, \forall e \in E$ ) do begin
        if ( $(v <_e w)$  and ( $al(v) \geq al(w)$ )) then
           $al(w) = al(v) + 1$  とする;
        if ( $(v \leq_e w)$  and ( $al(v) > al(w)$ )) then
           $al(w) = al(v)$  とする;
      end
    until (隣接階層値  $al$  が変化しなくなるまで);
     $LA := \max_{w \in W}(al(w))$             $\{= tl(W)\}$ 
  end
end

```

複合グラフ G が複合階層化可能なら, 関数 LA の repeat-until の部分は高々 $|W|$ 回の繰り返しなので計算時間は $O(|W|^3 \times |E|)$ でおさえられる. ただし, あらかじめ順序関係 ($<_e, \leq_e$) によるトポロジカル・ソートを行い, その順に隣接階層値を決定していけば, ソート時間 $+O(|W| \times |E|)$ で計算できることが分かっている.

アルゴリズム 4.2 (複合階層化)

複合階層化アルゴリズムは手続き LLA として次のように表される. 手続き LLA は, 複合グラフ G と, 親の複合階層値が等しい頂点の集合 W を引数とする. 再帰的に関数 LA を適用し, 頂点 $w \in De(W)$ に対して複合階層値 $cl(w)$ を割り当てる.

```

procedure LLA( $G$ : 複合グラフ;  $W$ : 頂点集合);
begin
  level := LA( $G, W$ );
  for  $i := 1$  to level do begin
     $\forall w \in \{x \in W | al(x) = i\}$  に対して,
       $cl(w) = \text{append}(cl(mo(w)), al(w))$  とする;
     $X := Ch(\{x \in W | al(x) = i\})$ ;
    if ( $X \neq \emptyset$ ) then LLA( $G, X$ )
  end
end;

```

手続き LLA を, 引数 $G, \{r\}$ に適用することで, 全頂点の複合階層値が求まる.

ただし $cl(mo(\tau))$ は空列, $append$ は第1引数の列の後ろに第2引数の値を追加した列を返す関数とする.

4.3.2 正規化

ここでは, 頂点の配置により隣接辺の配線が一意に決定できる「正規複合階層グラフ」と呼ばれる複合階層グラフのクラスを定義する. そして, 複合階層グラフを正規複合階層グラフへ変換(つまり「正規化」)する方法について説明する. この「正規化」により, 頂点の配置を決定することで隣接辺の配線も自動的に定まることになる.

定義 4.7 (正規隣接辺)

複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ において, 隣接辺 $e \in E_A$ が式 (4.5) および式 (4.6) で表される条件の両方を満たすとき正規隣接辺と呼ぶ.

$$(4.5) \quad cl(mo(\partial^+(e))) = cl(mo(\partial^-(e)))$$

$$(4.6) \quad al(\partial^+(e)) = al(\partial^-(e)) - 1$$

正規隣接辺は同じ深さで一つ下の階層に属する頂点と接続し, 他の階層を飛び越えない. したがって, 隣接辺を直線分で描く場合, 隣接辺同士の交差数は階層内の頂点の順序だけに依存する.

定義 4.8 (正規複合階層グラフ)

全ての隣接辺が正規隣接辺である複合階層グラフを正規複合階層グラフと呼ぶ.

定義 4.9 (正規後者)

複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ において, 頂点 $v \in V$ の正規後者とは, 式 (4.7)–(4.9) で表される条件のいずれかを満たす頂点 $w \in V$ である.

$$(4.7) \quad mo(v) = w \text{ and } al(v) = \#(\text{Ch}(\{u \in V | cl(u) = cl(w)\}))$$

$$(4.8) \quad w \in \text{Ch}(v) \text{ and } al(w) = 1$$

$$(4.9) \quad cl(mo(v)) = cl(mo(w)) \text{ and } al(v) = al(w) - 1$$

たとえば図 4.4 においては, 式 (4.7) を満たす例として d_2 に対して d_3 (最下層の子に対する親), 式 (4.8) を満たす例として d_4 に対する d_5 , d_5 に対する d_6 (親に対する最上層の子), 式 (4.9) を満たす例として v_3 に対する d_1 (正規隣接辺の始点に対する終点) が正規後者である.

定義 4.10 (複合路)

複合グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ において, ある頂点から別の頂点へ, 隣接辺の向きには従い包含辺の向きとは無関係に, 接続する頂点, 隣接辺, あるいは包含辺をたどることによって得られる頂点, 隣接辺, あるいは包含辺の系列を複合路と呼ぶ.

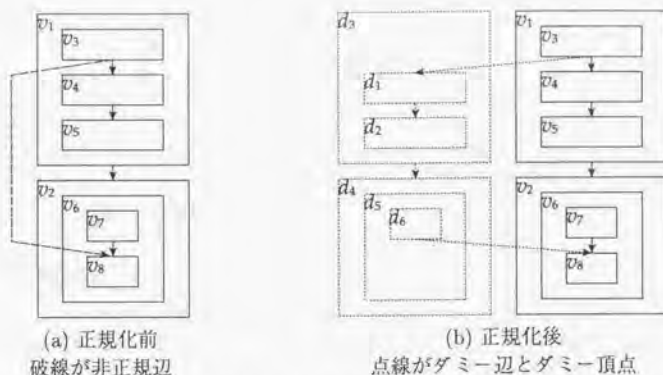


図 4.4: 正規化の例

定義 4.11 (正規複合路)

系列上の任意の頂点の直後の頂点が、存在すれば、正規後者であるような複合路を正規複合路と呼ぶ。

正規化とは、図 4.4 の例に示すように、与えられた複合階層グラフの全ての非正規辺をそれぞれ正規複合路で置き換えることである。正規複合路を構成する際に必要な全ての頂点、隣接辺、あるいは包含辺はダミーとして加えられ、以後必要があれば区別して処理される。ここで構成された正規複合路は描画時には一つの隣接辺として描かれる。

正規化は隣接辺だけに限る処理であるので、次に示すように、隣接辺毎に正規性のチェックおよび正規複合路の構成を繰り返すことで行われる。

アルゴリズム 4.3 (正規化)

正規化アルゴリズムは手続き PR として次のように表される。PR は複合グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ を引数とし、 G を正規化する。

```

procedure PR( $G$ );
begin
  for ( $\forall e \in E_A$ ) do
    if ( $e$  が非正規辺) then begin
       $e$  を正規複合路でおきかえ、
      必要なダミー頂点、隣接辺、包含辺を
      それぞれ  $V'$ ,  $E'_A$ ,  $E'_I$  とする;
       $V := V + V'$ ;
       $E_A := (E_A - \{e\}) + E'_A$ ;
    end
  end

```



```

 $E_I := E_I + E_I^1;$ 
 $\partial^+, \partial^-$  の修正
end
end;

```

複合階層の総数は高々 $|V|$ なので, 非正規辺 e の正規複合路での置き換え時間は $O(|V|)$ 以下である. したがって, 全ての隣接辺に対して同じ処理を繰り返しても計算時間は $O(|V| \times |E_A|)$ でおさえられる.

4.3.3 順序決定

このフェーズでは複合階層化により各階層に割り当てられた頂点の階層内での順序を決定する. その際に描画規則 R1, R2 を考慮する. つまり, 階層間での隣接辺の交差と, 階層内での隣接辺と頂点の交差が最少になるような頂点の順序を求めることが目的となる.

順序決定は, 包含木をルートからリーフへ向かって再帰的にたどりながら, 頂点毎にその子の集合から順序決定用局所グラフと呼ばれる有向グラフを構成し, それを対象に処理を進める.

定義 4.12 (順序決定用局所グラフ)

複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ において, 頂点 $v \in V$ に関する順序決定用局所グラフ G_O を次のように定義する³. ただし, v の割り当てられた階層における頂点の順序はすでに決定しているとし, その階層内で, v の左側に位置する頂点の集合を V_L , 右側に位置する頂点の集合を V_R で表す.

$$(4.10) \quad G_O = (V_O, E_O, \partial_O^+, \partial_O^-)$$

$$(4.11) \quad V_O = \text{Ch}(v) \cup \{v_L, v_R\}$$

$$(4.12) \quad E_O = E_O^1 \cup E_O^2 \cup E_O^3 \cup E_O^4$$

$$(4.13) \quad E_O^1 = \{e \in E_A \mid \partial^+(e), \partial^-(e) \in V_O\}$$

$$(4.14) \quad E_O^2 = \{e \in E_A \mid \partial^+(e) \in \text{De}(x) \text{ and } \partial^-(e) \in \text{De}(y) \\ \text{for } \forall x, y \in V_O, x \neq y\}$$

$$(4.15) \quad E_O^3 = \{e \in E_A \mid (\partial^+(e) \in \text{De}(V_L) \text{ and } \partial^-(e) \in \text{De}(V_O)) \\ \text{or } (\partial^+(e) \in \text{De}(V_O) \text{ and } \partial^-(e) \in \text{De}(V_L))\}$$

$$(4.16) \quad E_O^4 = \{e \in E_A \mid (\partial^+(e) \in \text{De}(V_R) \text{ and } \partial^-(e) \in \text{De}(V_O)) \\ \text{or } (\partial^+(e) \in \text{De}(V_O) \text{ and } \partial^-(e) \in \text{De}(V_R))\}$$

³ $G_O, V_O, E_O, \partial_O^+, \partial_O^-, V_L, V_R$ などは全て頂点 v に依存するので, $G_{O[v]}, V_{O[v]}, E_{O[v]}, \partial_{O[v]}^+, \partial_{O[v]}^-, V_{L[v]}, V_{R[v]}$ のように記述の方が厳密である. しかしながら, ここでは記法が複雑になり過ぎるのを避けるために $[v]$ を省略する.

$$(4.17) \quad \partial_O^+(e) = \begin{cases} \partial^+(e) & \text{if } e \in E_O^1 \\ x & \text{if } e \in E_O^2 \\ v_L & \text{if } e \in E_O^3 \\ v_R & \text{if } e \in E_O^4 \end{cases}$$

$$(4.18) \quad \partial_O^-(e) = \begin{cases} \partial^-(e) & \text{if } e \in E_O^1 \\ y & \text{if } e \in E_O^2 \\ z & \text{if } e \in E_O^3 \cup E_O^4 \end{cases}$$

ただし, $d = \text{depth}(v) + 1$ とすると,

$$(4.19) \quad x = \text{an}_d(\partial^+(e))$$

$$(4.20) \quad y = \text{an}_d(\partial^-(e))$$

$$(4.21) \quad z = \begin{cases} \text{an}_d(\partial^+(e)) & \text{if } \text{an}_d(\partial^+(e)) \in V_O \\ \text{an}_d(\partial^-(e)) & \text{if } \text{an}_d(\partial^-(e)) \in V_O \end{cases}$$

順序決定用局所グラフは $\text{Ch}(v)$, V_L , V_R をすべて点と見なし, $\text{Ch}(v)$, V_L , V_R の各子孫と接続する隣接辺のうち自己閉路となるものを取り除いて作った有向グラフである. 例として, 図 4.5 に示される複合グラフ G の頂点 v に関する局所グラフ G_O を図 4.6 に示す.

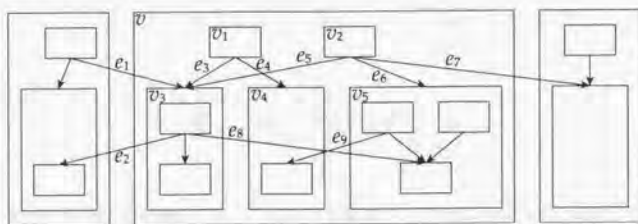


図 4.5: 複合グラフ G

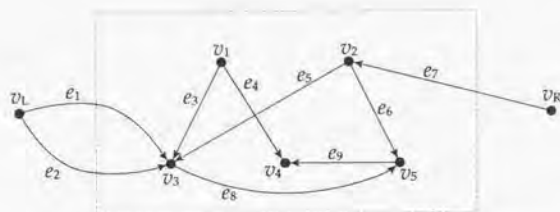
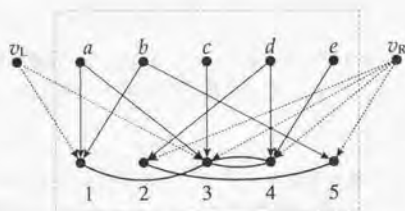
局所グラフ G_O において, 解決すべき問題を $\mathcal{P}1$ とする. これは図 4.7 に示されるようなもので, 次のように定式化できる.

$G_O = (V_O, E_O, \partial_O^+, \partial_O^-)$ において,

$$(4.22) \quad V_s = \{w \in V_O \mid \text{AL}(w) = s\}$$

$$(4.23) \quad E_{s,t} = \{e \in E_O \mid \partial_O^+(e) \in V_s \cup \{v_L, v_R\} \text{ and } \partial_O^-(e) \in V_t\}$$

$$(4.24) \quad E_s = \{e \in E_O \mid \partial_O^+(e), \partial_O^-(e) \in V_s\}$$

図 4.6: 複合グラフ G (図 4.5) の頂点 v に関する順序決定用局所グラフ G_O 

上段の頂点集合を固定して、下記条件を満足する下段の頂点の順序を求める。

- (1) 実線、点線のなす交差を最小化する。
- (2) 太線と頂点との交差 (太線が飛び越す頂点の数) を最小化する。
- (3) (1) を (2) より優先する。

図 4.7: 順序決定問題 $P1$

とする. 第 s 階層 V_s の頂点の順序を全単射関数

$$(4.25) \quad r_s: V_s \rightarrow \{1, \dots, |V_s|\}$$

で表す. また関数 r'_s を次のように定義する.

$$(4.26) \quad r'_s: V_s \cup \{v_L, v_R\} \rightarrow \{0, 1, \dots, |V_s| + 1\}$$

$$(4.27) \quad r'_s(w) = \begin{cases} 0 & \text{if } w = v_L \\ |V_s| + 1 & \text{if } w = v_R \\ r_s(w) & \text{otherwise.} \end{cases}$$

ただし, 以上において $1 \leq s, t \leq TL(V_O)$ とする. このとき G_O における辺の全交差数を f_1 , 辺と頂点の交差数を f_2 とすると

$$(4.28) \quad f_1 = \sum_{s=1}^{TL(V_O)-1} |\{ \langle e_1, e_2 \rangle \in E_{s,s+1} \times E_{s,s+1} \mid P(e_1, e_2) \}|$$

$$(4.29) \quad P(e_1, e_2) = ((r'_s(\partial_O^+(e_1)) < r'_s(\partial_O^+(e_2))) \text{ and } ((r'_{s+1}(\partial_O^-(e_1)) < r'_{s+1}(\partial_O^-(e_2)))$$

$$(4.30) \quad f_2 = \sum_{s=1}^{TL(V_O)} \sum_{e \in E_s} \{ |r_s(\partial_O^+(e)) - r_s(\partial_O^-(e))| - 1 \}$$

となる. つまり, 描画規則 R1 は f_1 を, R2 は f_2 を最小化する全単射関数 r_s ($1 \leq s \leq TL(V_O)$) を求めることであり, 両方を考慮する場合

$$(4.31) \quad c_1 f_1 + (1 - c_1) f_2$$

を最小化する r_s を求めることになる. ここで c_1 は加重係数 ($0 < c_1 \leq 1$) である.

規則 R2 は各階層毎に分割するとグラフの線形配置問題 [Eve79] と等価である. 有向グラフに対する線形配置問題とは, 与えられた有向グラフ $D = (V, E, \partial^+, \partial^-)$ において頂点を数直線上の $1, 2, \dots, |V|$ の位置にそれぞれ一つずつ配置し, 頂点の位置を全単射関数 $r: V \rightarrow \{1, 2, \dots, |V|\}$ で表す場合に, 辺の総長

$$(4.32) \quad \sum_{e \in E} |r(\partial^+(e)) - r(\partial^-(e))|$$

を最小にする関数 r を求めるものである. また, 規則 R1 と R2 はどちらも NP-完全問題である.

規則 R1 の解決には, 多段グラフの辺の交差数減少化のために開発された発見的解法, 重心法 [STT81, Mak88] を用いる. また, 規則 R2 の解決にも, 線形配置問題の辺の長さを2段グラフの辺の交差に置き換えて重心法を適用する方法を開発した.

定義 4.13

線形配置問題の対象となる有向グラフ $D = (V, E, \partial^+, \partial^-)$ から構成される2段グラフ D_B を次のように定義する.

$$(4.33) \quad D_B = (V_B, E_B, \partial_B^+, \partial_B^-)$$

$$(4.34) \quad V_B = V \cup E$$

$$(4.35) \quad E_B = E \times \{0, 1\}$$

$$(4.36) \quad \partial_B^+(< e, i >) = e \text{ for } \forall e \in E, \forall i \in \{0, 1\}$$

$$(4.37) \quad \partial_B^-(< e, 0 >) = \partial^+(e)$$

$$(4.38) \quad \partial_B^-(< e, 1 >) = \partial^-(e)$$

つまり, 2段グラフ D_B は, 有向グラフ D の各辺を新しい頂点で2つに分割し, 常に新しい頂点が始点になるよう辺の向きを定めたものである.

この様に構成した2段グラフ D_B において, E の要素を上段に, V の要素を下段に配置し, 重心法により辺の交差を減少させる. その結果得られた下段の頂点の順序を線形配置問題の解とする.

実際には $P1$ は規則 $R1$ と $R2$ の複合した問題であるので, 優先関係 $R1 > R2$ の下で同時に考慮しなければならない. そこで各階層毎に重心法を修正した次のような一連の処理を行う. 第 s 階層について下向きの場合,

1. 第 $s-1$ 階層の要素に左から $1, 2, \dots, |V_{s-1}|$ の重さを与える.
2. 第 s 階層の頂点を辺 $E_{s-1,s}$ で結合する第 $s-1$ 階層の頂点の重心 [STT81] で並べ替え, 順序 r_s を求める. 順序 r_s から第 $s-1$ 階層の各頂点の重心も求める.
3. 有向グラフ $D_s = (V_s, E_s, \partial_s^+, \partial_s^-)$ から2段グラフ $D_s = (V_B, E_B, \partial_B^+, \partial_B^-)$ を構成する.
4. V_B の下段の頂点の順序を r_s とし, 上段の頂点の重心を求め, 適当に第 $s-1$ 階層中に挿入する.
5. 第 s 階層の頂点を辺 $E_B \cup E_{s-1,s}$ で結合する $V_B \cup V_{s-1}$ の頂点の重心で並べ替え, 順序 r_s を求める.

以上の処理を重心法と同様に, 下向きと上向きを交互に数回繰り返す. 1と2で描画規則 $R1$ に従って順序 r_s が求められ, その r_s をもとに3, 4, および5で $R2$ に関して修正がなされるという意味でこの方法は $R1 > R2$ の優先関係があるとする.

アルゴリズム 4.4

順序決定アルゴリズムは手続き ORDER として次のように表される. ORDER は, 正規複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ と, 頂点集合 $W \subseteq V$ を引数とし, $\text{De}(W)$ の各階層内での順序を決定する.

```

procedure ORDER( $G, W$ );
begin
  for ( $\forall v \in W$ ) do begin
    局所グラフ  $G_O$  を構成する;
     $G_O$  について  $\mathcal{P}1$  を解き,  $V_O$  の全要素の順序を決定する;
    ORDER( $G, \text{Ch}(v)$ )
  end
end;

```

計算時間は階層数や各階層への頂点の分布状況に強く依存するが, 最悪の場合でも $O(|V|^3)$ で計算できる.

4.3.4 座標決定

このフェーズでは, 階層および階層内における順序の決定した各頂点の座標値を決定する. 垂直座標値は各頂点の複合階層値から容易に決定できる. 水平座標値決定の際に描画規則 R3, R4, R5 を考慮する.

座標決定は, 順序決定とは逆に包含木をリーフからルートへ再帰的にたどりながら, 頂点毎にその子の集合から座標決定用局所グラフと呼ばれる有向グラフを構成し, それを対象に処理を進める.

定義 4.14 (座標決定用局所グラフ)

複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ において, 頂点 $v \in V$ に関する座標決定用局所グラフ G_P を次のように構成する⁴.

$$(4.39) \quad G_P = (V_P, E_P, \partial_P^+, \partial_P^-)$$

$$(4.40) \quad V_P = \text{Ch}(v)$$

$$(4.41) \quad E_P = \{e \in E \mid \partial^+(e), \partial^-(e) \in V_P\}$$

$$(4.42) \quad \partial_P^+ = \partial^+|E_P$$

$$(4.43) \quad \partial_P^- = \partial^-|E_P$$

⁴ $G_P, V_P, E_P, \partial_P^+, \partial_P^-$ などは全て頂点 v に依存するので, $G_{P[v]}, V_{P[v]}, E_{P[v]}, \partial_{P[v]}^+, \partial_{P[v]}^-$ のように記述する方が厳密である. しかしながら, ここでは記法が複雑になり過ぎるのを避けるために $[v]$ を省略する.

座標決定には領域外の情報は用いないので、座標決定用局所グラフは頂点に直接含まれる頂点とそれに接続する隣接辺だけで構成する。順序決定用局所グラフと同様に $\text{Ch}(v)$ をすべて点と見なし $\text{Ch}(v)$ の各子孫と接続する隣接辺から作られたものだが、 V_L , V_R は考慮にいれず、また、自己閉路だけでなく、同階層内の頂点を接続する辺も取り除く。例として、図 4.5 に示される複合グラフ G の頂点 v に関する局所グラフ G_P を図 4.8 に示す。

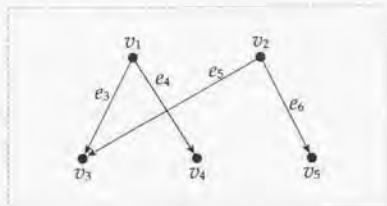


図 4.8: 複合グラフ G (図 4.5) の頂点 v に関する座標決定用局所グラフ G_P

局所グラフ G_P において、解決すべき問題を P_2 と呼ぶ。これは文献 [STT81] に従い、次のように定式化できる。

局所グラフ $G_P = (V_P, E_P, \partial_P^+, \partial_P^-)$ において、各頂点の中心の x 座標を関数

$$(4.44) \quad x: V_P \rightarrow$$

で表す。そして、 $w \in V_P$ に対して、 E_L , E_U , $g_L(w)$, $g_U(w)$ を次のように定義する。

$$(4.45) \quad E_L = \{e \in E_P | \partial_P^+(e) = w\}$$

$$(4.46) \quad E_U = \{e \in E_P | \partial_P^-(e) = w\}$$

$$(4.47) \quad g_L(w) = \begin{cases} x(w) - \frac{1}{|E_L|} \sum_{e \in E_L} x(\partial_P^-(e)) & \text{if } |E_L| \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$(4.48) \quad g_U(w) = \begin{cases} x(w) - \frac{1}{|E_U|} \sum_{e \in E_U} x(\partial_P^+(e)) & \text{if } |E_U| \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

このとき、隣接頂点の近接性 f_4 、隣接辺の対象性 f_5 は次のように表される。

$$(4.49) \quad f_4 = \sum_{e \in E_P} \{x(\partial_P^+(e)) - x(\partial_P^-(e))\}^2$$

$$(4.50) \quad f_5 = \sum_{w \in V_P} \{g_U(w)^2 + g_L(w)^2\}$$

規則 R4 は f_4 を, R5 は f_5 を最小化する関数 x を求めることである。ただし, 描画規則 R1, R2, および R3 から次の制約条件が要求される。規則 R1 と R2 に関し, $\forall v, w \in V_g$ に対して, $r_s(v) < r_s(w)$ のとき $x(v) + \alpha < x(w)$, ただし $1 \leq s \leq TL(V_P)$ であり, α は v と w の幅と間隔から決まる。規則 R3 に関し, $e \in E_P$ に対して, $\partial_P^+(e)$ と $\partial_P^-(e)$ が共にダミー頂点のとき, $x(\partial_P^+(e)) = x(\partial_P^-(e))$ 。

問題 P2 は以上の制約条件の下で,

$$(4.51) \quad c_2 f_4 + (1 - c_2) f_5$$

を最小化する関数 x を求めることである。ここで c_2 は加重係数 ($0 < c_2 \leq 1$) である。

問題 P2 の解決には, 計算コストを下げるために, 有向階層グラフの自動描画のために開発された発見的解法である優先度法 [STT81] を利用する。

優先度法とは, ある規則に従って頂点に優先度を定め, 優先度の高い順に座標を改善することを, 各階層ごとに何度か繰り返すことで頂点の座標を決定する方法である。このとき, 順序決定のフェーズで決定した順序を保存することで R1, R2, R3, R4, R5 が成立する。優先度法において, まず隣接辺の直線性 (R3) を達成するように優先度を設定できるという意味で R3 > R4, R5 である。さらに, 近接性 (R4) を高めるための頂点の座標の改善を主処理とし, それを接続次数の大きい順に行い, なるべく多くの隣接辺の対称性 (R5) を目指すことで R4 > R5 であるとする。

アルゴリズム 4.5

座標決定アルゴリズムは手続き PLACE として次のように表される。PLACE は, 正規複合階層グラフ $G = (V, E, \partial^+, \partial^-, \tau)$ と, 頂点集合 $W \subseteq V$ を引数とし, $De(W)$ の各頂点の座標を決定する。

```

procedure PLACE( $G; W$ );
begin
  for ( $\forall v \in W$ ) do begin
    if ( $Ch(v) \neq \emptyset$ ) PLACE( $G, Ch(v)$ );
    局所グラフ  $G_P$  を構成する;
     $G_P$  について P2 を解き,  $V_P$  の
    全要素の  $v$  に対する相対座標を決定する
  end
end;
```

計算時間は順序決定と同様に階層数などに強く依存するが, 最悪の場合でも $O(|V|^2)$ で計算できる。

4.4 出力例

本章で述べた方法に従って, 複合グラフの自動描画プログラムを実現した。そのプログラムは図 4.9 に示すようなテキストによって頂点間の隣接辺と包含辺だけが定

義された複合グラフを入力とし、図4.10に示すような図を計算機の画面上に出力する。図4.10は、頂点数61(そのうちグループ数22)、隣接辺数35、包含辺数60の複合グラフであり、SPARCstation2でレイアウトを計算するのに約0.2秒要した。

なお、第7章で説明する対話型システムは、この自動描画プログラムを中核機能として備えていて、そこでは図4.9に示したようなテキストの入力は特に必要とせずマウスを用いた直接操作によって対話的に隣接辺や包含辺を定義することも可能である。

%I _00 : _26 _54 _60 _61	%A _13 : _23
%I _40 : _01 _16	%A _15 : _11
%I _41 : _08 _09	%A _16 : _14
%I _42 : _03 _07	%A _17 : _35
%I _43 : _02 _05 _25 _33	%A _19 : _04 _18
%I _44 : _12 _32	%A _20 : _10 _14
%I _45 : _10 _14 _20	%A _21 : _38
%I _46 : _22 _27 _28	%A _24 : _11
%I _47 : _11 _15 _24	%A _25 : _20
%I _48 : _30 _37	%A _27 : _22
%I _49 : _34 _39	%A _28 : _30
%I _50 : _04 _17 _18 _19	%A _32 : _12 _38
%I _51 : _13 _23 _36	%A _33 : _25
%I _52 : _31 _35	%A _35 : _31
%I _54 : _06 _29 _44	%A _36 : _23
%I _55 : _41 _47	%A _37 : _27
%I _56 : _46 _48	%A _40 : _49
%I _57 : _40 _49	%A _41 : _47
%I _58 : _42 _50 _52	%A _43 : _54
%I _59 : _43 _45	%A _44 : _29
%I _60 : _51 _57 _59	%A _50 : _42
%I _61 : _21 _38 _55 _56 _58	%A _52 : _42 _26
%A _01 : _16	%A _57 : _51
%A _02 : _33	%A _58 : _55
%A _05 : _25	%A _59 : _51
%A _06 : _29	%A _60 : _61
%A _09 : _24	

図4.9: 複合グラフの自動描画プログラムへの入力データ

%Iで始まる行が包含辺の定義、%Aで始まる行が隣接辺の定義である。アンダースコア(_)で始まる数は頂点の名前である(名前としては英数字や漢字も利用できる)。各行とも、コロン(:)の左側が始点、右側が終点を表し、始点が同じ辺は1行に複数定義することができる。

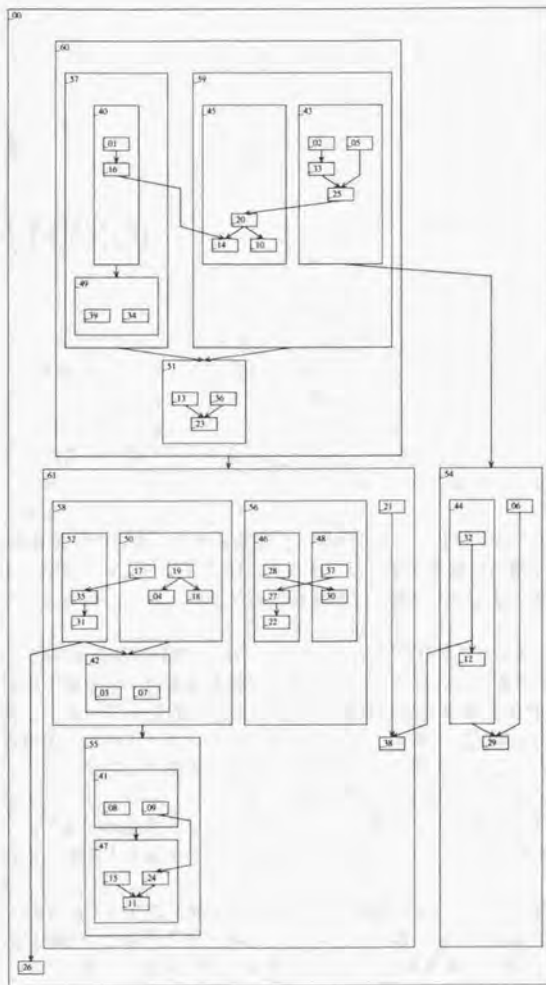


図 4.10: 複合グラフの自動描画例

図 4.9 を入力データとして自動的に描いたもの、頂点数 61 (そのうちグループ数 22)、隣接辺数 35、包含辺数 60 の複合グラフである。SPARCstation 2 によるレイアウトの計算時間が約 0.2 秒であった。

第5章

図の装飾技術

図の可視化プロセスにおいて論理情報から幾何情報を自動生成するためには、第4章で述べた位置属性の決定と共に、幾何的オブジェクトの色、形、サイズなど、すなわち視覚属性を決定する必要がある。図を編集していく過程においては、図の論理的な構造の変更や図素(幾何的オブジェクト)¹の配置の変更だけでなく、視覚属性の変更に対する要求も数多くまた繰り返しある。一例をあげると、図素の表す意味内容に応じた色分け、重要な部分の拡大強調、図の一部を縮小あるいは省略する概略化などである。

これらの視覚属性の変更に対する要求は、図の「見方」に対するユーザの要求として捉えることができる。しかしながら、「見方」に対する要求は一般に多様であり、個々の要求ごとに個別に対応することは非効率的で大変なため、統一的な対応が求められる。

本章では、図の自動装飾技術の一つとして、ユーザの多様な「見方」に統一的に対応できる図の「見かけ」の変更方式を提案する。この方式は、図の「見方」を図素の「重要度」で統一的に表現し、「見かけ」の変更を図素の視覚属性の変更操作、すなわち「視覚属性操作」によって行うものである。本章で提案する方式を、ユーザの「見方」に応じて図に化粧を施すという意味から「図ドレッシング」と呼ぶことにする。図ドレッシングは、(1) 要求への統一的な対応ができることに加え、(2) 図が編集によって動的に変化しても「見かけ」を自動的に「見方」に合わせる、(3) 複数の基本的な「見方」を組み合わせた要求にも対応できる、という特徴を備えている [Mis94]。

以下、5.1節では、まず図に関する用語として図素と属性について説明する。5.2節では視覚属性操作を一般化して、図ドレッシングの基となる形式的な表現を与え、とともに、図の「見方」を統一的に表現するための「重要度」を導入する。5.3節では複合グラフを例に図ドレッシングの実現法を説明をする。5.4節では実際の画

¹本稿ではしばしば「図素」を「幾何的オブジェクト」と呼ぶ。一般的には「図素」が用いられるが、論理情報を構成する「論理的オブジェクト」との対応付けのために「幾何的オブジェクト」も用いている。

面出力を使って図ドレッシングの適用例を示す。

5.1 図素と属性

本章で取り扱う図に関する用語を説明する。

図素 図の構成単位は、点、線、面であり、これらを図素と呼ぶ [IYA86]。図は平面上に図素を配置したものである。

視覚属性 図素はいくつかの視覚属性 (または視覚変数 [Ber82]) をもち、視覚属性を変更することによって図は見かけ上様々に変化する。視覚属性は次のように論理的、質的、量的なものに分けられる。

視覚属性	論理的: 可視性
	質的 : きめ、模様、色、形
	量的 : サイズ、濃淡

構造属性 図素はそれぞれ独立に存在するのではなく、全体としてある論理的な構造をもっている。したがって、図素は構造の構成要素としての側面から構造属性を持つ。一般に図の構造には、グラフ、行列、座標などが用いられる [IYA86]。本章ではグラフを構造とする図を対象とし、それ以外の構造については5.3節で簡単に説明する。

意味属性 実際の図においては、図素はしばしば特別な意味を表現している。したがって、図素はそれが表現するものとして意味属性を持つ。たとえば、鉄道路線図などでは点が駅を、線が路線を表現し、点や線の傍に記された駅名や路線名のようなテキスト、またはアイコンや画像などによって意味属性が表される。

視点属性 計算機の画面上に表示された図はユーザによって見られる対象である。ユーザは図の一部分に着目して見ることが多い。そこで、ユーザに着目されている図素を視点と呼ぶ。各図素は視点かどうか、視点からどれくらいの距離にあるかという視点属性を持つ。

上で説明した四つの属性のうち、視覚属性が視覚属性操作で変更される対象である。他の三つの属性に基づいて、図ドレッシングでは視覚属性が変更される。

5.2 視覚属性操作の形式化

視覚属性操作に求められる三つの特徴、(1) 要求への統一的な対応、(2) 図の動的な変化への対応、(3) 組合せ要求への対応、を説明しながら、それらを備える視覚属性操作の形式的な表現を構成する。

5.2.1 要求への統一的な対応

表5.1にユーザの要求と要求に対する操作の例をあげた、要求は視覚属性操作で対応可能なものという観点で集められている。要求間の境界を破線で表し、境界線を操作の欄まで引いていないのは、要求が抽象的でそれぞれの境界が曖昧だからである。表にあげたような操作を要求ごとに用意した場合、ユーザは自分の要求に応じて操作を選択しなければならず、さらに各操作の適用方法が異なればそれぞれについて修得する必要がある。

表5.1: 要求と視覚属性操作

要求	視覚属性操作 (複合操作)
分類	頂点に階層*1ごとに異なるきめ/模様/色/形を与える。 頂点の色/形を頂点に付加された画像に応じて変える。
適合	頂点のサイズを文字数に応じて変える。 頂点のサイズ/濃淡で頂点の意味属性を表す。
可視化	頂点を視点に関連するものほど大きく/濃くする。 視点とその近傍の図素を大きく/濃くする。
強調	視点から遠い図素を小さく/淡くする。 次数が一定以下/以上の頂点を小さく/大きくする。
詳細/概略化	一定階層*1以下/以上の頂点を見せる/見せない。 意味属性値*2が一定以下/以上の頂点を見せる/見せない。

*1 木や階層グラフなど階層的グラフ構造をもつ図が対象の場合。

*2 図素が量的な意味属性をもつ場合。

そこで、多様な要求に統一的に対応できるように、視覚属性操作の一般的な表現を構成する。例にあげた操作はそれぞれ異なるが、それらの基本的な違いは視覚属性の変更対象である図素の集合と各図素に対する視覚属性値の決め方である。したがって、図素 x の視覚属性 A を属性値 u に変更する基本操作を

$$(5.1) \quad o_A(x, u)$$

で表わすと、表5.1にあげたような操作は一般的に基本操作の集合 O として次のように記述できる。

$$(5.2) \quad O = \{o_A(x, a(x)) | x \in X\}$$

これを複合操作と呼ぶ。ここで、 X は操作の対象となる図素の集合 (対象集合)、 a は X から視覚属性 A の値への関数 (属性関数) である。

5.2.2 図の動的な変化への対応

図は編集過程において、新しい頂点が追加されて (階層グラフ構造の場合) 階層がずれたり、またはユーザが視点を変更したりして、動的に変化する。しかしながら、視覚属性操作が、直接指定した図素に対してのみ適用される操作であるなら、図が変化するたびに繰り返し適用しなければならない。たとえば、要求が階層ごとの頂点の色分けなら、頂点の階層が変化することに色を変える操作を適用しなければならない。

複合操作において図の動的な変化に対応するためには、「この頂点を赤にする」ではなく「第1階層に含まれる頂点を赤にする」のように間接的に対象を指定する必要がある。つまり、式 (5.2) において対象集合 X を具体的な図素で直接指定することと、属性関数 a を図素 X から属性値への直接的な関数として定義することをやめ、要求を表現する図素に対して間接的に指定または定義すべきである。そこで、「階層」のように間接的に対象を指定するために使われる属性を表5.2に整理した。これは、表5.1にあげた操作における対象集合と属性関数に対応する表現を図素の属性によって分類したものである。

表5.2: 図の属性による対象集合および属性関数の分類

属性	対象集合	属性関数
視点属性	視点とその近傍の図素 視点から遠い図素	視点に関連するものほど大きく濃く
構造属性	一定階層以下/以上の頂点 次数が一定以下/以上の頂点	階層ごとに異なるきめ/模様/色/形
意味属性	意味属性が一定以下/以上の頂点	文字数に応じたサイズ 頂点に付加された画像に応じた色/形 意味属性に応じたサイズ/濃淡

表5.2にあげた属性の一つを、図素 x に対してで表わし、式 (5.3) のように、複合操作 O を I を利用して間接的に表現すると、複合操作が図の動的な変化に対応可能になる。

$$(5.3) \quad O = \{o_A(x, a(I(x))) | x \in \{y | P(I(y))\}\}$$

ここで、 P は対象集合を決定するための属性に関する条件(対象条件)であり、属性関数 a は属性値から視覚属性 A の値への関数である。

5.2.3 組合せ要求への対応

ユーザの図の「見方」は必ずしも単純ではなく、時には、「図の全体を概略的に表示し、かつ着目している頂点を大きく拡大する」というように複数の「見方」を組み合わせたような要求もありうる。ところが、表5.1にあげたような操作では複数の「見方」に対する要求を組み合わせて同時に満たすことは困難である。

そこで、組合せ要求に対応するために、一つの属性ではなく複数の属性を同時に利用して対象集合や視覚属性値を決める。つまり、 m 種類の属性を同時に利用する場合、各属性値を数値で表現したものを $I_j(x)$ ($1 \leq j \leq m$) とし、それらを結合関数 h で式(5.4)のように組み合わせたものをあらためて $I(x)$ とすると、式(5.3)と同様の複合操作 O で組合せ要求に対応可能になる。

$$(5.4) \quad I(x) = h(I_1(x), I_2(x), \dots, I_m(x))$$

利用する属性は大きく分類すると構造属性、意味属性、視点属性の三つであるが、たとえば、構造属性でも階層と次数のように異なるものがあり、それらが同時に使われ得るので m は4以上の可能性もある。

5.2.4 重要度と重要度関数

表5.1にあげた操作の例では具体的な例を示すために各属性の直接的な表現を用いた。しかし、実際にユーザが対象集合や属性関数を表現する際には、属性を具体的に表現するよりも、「重要な部分」とか「重要さの程度」といった抽象的な表現を用いることが多い。つまり、ユーザは各属性を意識の上で区別することはせず、「重要さ」という統一的な尺度を無意識的に用いていると思われる。そこで、一つ以上の属性を組合せて統一的に表現した数値 $I(x)$ を重要度と呼び、各図素に対して重要度を与える関数 I を重要度関数と呼ぶ。また、属性の一つを表現した数値 $I_j(x)$ を基本重要度と呼び、各図素に対して基本重要度を与える関数 I_j を基本重要度関数(または単に基本関数)と呼ぶ。

5.3 図ドレッシングの実現法

前節で構成した形式的な表現に基づいて実現する視覚属性操作法が図ドレッシングである。本節では、まず図ドレッシングの構成を説明する。そして、4.1節で定義

した複合グラフに対して、図ドレッシングを構成する重要度関数と自動図ドレッサの具体的な実現例を説明する。さらに、複合グラフ以外の構造を持つ図に対しても同様に実現可能なことを示す。

5.3.1 図ドレッシングの構成

図5.1に図ドレッシングの構成を示す。図データは、属性や配置情報をもつ図素の集合として表される。図の表示に際しては構造属性、意味属性、視点属性はあらかじめ与えられていなければならない。視覚属性は図ドレッシングによって、配置情報は自動描画機能によって与えられるので未定でよい(視覚属性に関して、デフォルト値は与えられているとする)。

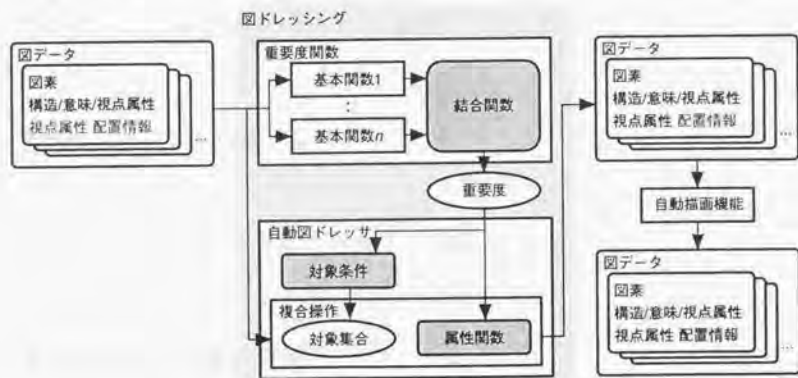


図5.1: 図ドレッシングの構成

図ドレッシングは、各図素ごとに重要度を与える重要度関数と、その重要度を使って複合操作を自動的に行なう自動図ドレッサによって構成される。重要度関数では、 m 個の基本関数によって各図素ごとに m 個の基本重要度を求め、それらを結合関数によって結合して重要度とする。自動図ドレッサでは、重要度を使って対象条件に従って対象集合を決定し、属性関数で各図素に与える属性値を決定しながら複合操作を行う。

図ドレッシングのあと自動描画機能により配置情報が与えられて、画面上に表示することが可能になる。

斜線を付けた結合関数、対象条件、属性関数はユーザの要求に応じて修正される。

5.3.2 重要度関数

複合グラフを例にして重要度関数の定義の仕方の一例を示す。ここで定義する重要度関数は複合グラフの各頂点に重要度を与える。まず構造属性、意味属性、視点属性それぞれに対して基本関数を一つずつ定義して、つぎにそれらの結合関数を定義する。

構造属性に対する基本関数の例

集合を表すベン図に見られるような領域の包含関係で概念間の関係を表す場合、包含のネストの外側に比べて内側ほど下位の概念を表すことが多い。そのような場合、ここでは上位の概念に高い重要度を与えることにして、構造属性に対する基本関数 I_s を包含木 G_I における頂点の深さ $depth$ を用いて下のように定義する。深い位置にある(下位の概念を表す)頂点の重要度を小さくするために符号を変えている。

$$(5.5) \quad I_s(x) = -depth(x)$$

意味属性に対する基本関数の例

表示領域の観点から効率的に図を表示するためには各頂点のサイズを常に同じにするよりも状況に応じて変えるほうが望ましい。そこで、意味属性に対する基本関数 I_m をその頂点を含む文字数で下のように定義する。

$$(5.6) \quad I_m(x) = (\text{頂点 } x \text{ の含む文字数})$$

視点属性に対する基本関数の例

視点として着目された頂点に対して、その頂点を含むまたは含まれる、またはその頂点と隣接関係にあるような頂点が最も近い頂点と考えられる。そこで、視点属性に対する基本関数 I_f を頂点 y_1, y_2, \dots, y_n が視点である場合に複合グラフ上の距離 D を用いて下のように定義する。どの視点からも遠い頂点ほど重要度を小さくするために符号を変えている。

$$(5.7) \quad I_f(x; y_1, \dots, y_n) = -\min_{1 \leq i \leq n} D(x, y_i)$$

結合関数の例

基本関数の結合は1次結合とする。各係数が属性の組合せ方を調節するためのパラメータとなる。また、属性の影響を公平に扱えるように基本重要度が $[0, 1]$ の値になるように正規化しておいて結合する。つまり、重要度関数 I を上で定義した3種類の基本関数の一次結合として次のように定義する。

$$(5.8) \quad I(x; y_1, \dots, y_n) = w_s \cdot I_s^*(x) + w_m \cdot I_m^*(x) + w_f \cdot I_f^*(x; y_1, \dots, y_n)$$

ここで, $w_s + w_m + w_f = 1$, $w_s \geq 0$, $w_m \geq 0$, $w_f \geq 0$ とする。ただし, I_s^* , I_m^* , I_f^* はそれぞれ, I_s , I_m , I_f の値域を $[0, 1]$ に正規化したものである。

5.3.3 自動図ドレッサ

自動図ドレッサによって自動的に複合操作を行うためには対象条件と属性関数を形式的に定義する必要がある。先に定義した重要度が $[0, 1]$ の実数値であることから, 対象条件は実数から真偽値への関数, 属性関数は実数から視覚属性値への関数として形式的に表現できる。ここでは, 複合グラフに対する自動図ドレッサの例を三つ示す。

可視性操作

第1の例は可視性 (visibility) を操作するものである。ある閾値以下の重要度をもつ頂点を見えなくするもので, 複合操作の一般形で表現すると下のようになる。ここで, t_1 は閾値である。

$$(5.9) \quad O = \{o_{\text{visibility}}(x, \text{Invisible}) | x \in \{v \in V | I(v) \leq t_1\}\}$$

離散型サイズ操作

第2の例はサイズ (size) を操作するものである。サイズを与える属性関数に離散単調増加関数を用いる。複合操作の一般形で表現すると下のようになる。

$$(5.10) \quad O = \{o_{\text{size}}(x, f(I(x))) | x \in V\}$$

そして属性関数 f を下のよう定義する。ここで, p は小さいサイズ, q は大きいサイズとする。また, t_2 は閾値である。

$$(5.11) \quad f(i) = \begin{cases} p & \text{if } i \leq t_2 \\ q & \text{otherwise} \end{cases}$$

連続型サイズ操作

第3の例もサイズを操作するものである。先に説明した離散型との違いは属性関数に連続単調増加関数を用いる点にある。複合操作は式 (5.10) と同じで, 属性関数 f を下のよう定義する。ここで, s は標準のサイズとする。定数 r はサイズへの重要度の影響の仕方を調節するパラメータである。

$$(5.12) \quad f(i) = r \cdot i^T \cdot s$$

5.3.4 例の一般性について

上で示した重要度関数と自動図ドレッサの例は、対象を複合グラフに、視覚属性を可視性やサイズに限定したため、図ドレッシングの実現法の説明としては部分的なものである。本節では、そのように限定されない一般的な場合においても同様に実現可能であることを示す。

重要度関数の一般性

図の構造および構造上の距離が形式的に与えられれば、重要度関数は複合グラフの場合と同様にして定義可能である。5.1節で述べた通り、視覚属性操作の対象となる図は構造を持つことを前提としていることと、グラフ構造以外の、行列や座標などにも距離を定義することは可能である [Mis90a] ことから、一般の多くの図に対しても、同様にして重要度関数を定義できる。

また、実現例では三つの基本重要度だけを用いたが、一般的にはそれ以上の基本重要度が必要となる。その場合でも結合関数を修正するだけで適応できる。

自動図ドレッサの一般性

図ドレッサは視覚属性に依存するため、視覚属性値の決定および変更処理について視覚属性ごとに検討する必要がある。しかし、本質的には重要度から視覚属性値への属性関数の構築の仕方が問題である。量的視覚属性である濃淡については濃淡を連続的な数値で表すことで、連続型サイズ操作の場合と同様に属性関数を構築できる。また、質的視覚属性に対しては各属性値に離散的な数値を割り当て、離散型サイズ操作で用いたような離散関数を用いることで属性値関数を構築できる。つまり、可視性とサイズ以外の視覚属性についても同様に自動図ドレッサの構築が可能である。

5.4 図ドレッシングの適用例

前節で説明した図ドレッシングを実際に複合グラフに適用した例を用いて、多様な要求への対応、組合せ要求への対応、図の動的な変化への対応という三つの問題が解決できたことを示す。図5.2は複合グラフを第4章で示した自動描画法によって描いたものである。そして以下に示す図は図5.2を元の図として図ドレッシングを適用したものである。ただし図の比較において節点の対応を分かりやすくするために頂点のテキストや画像を番号に置き換えた。

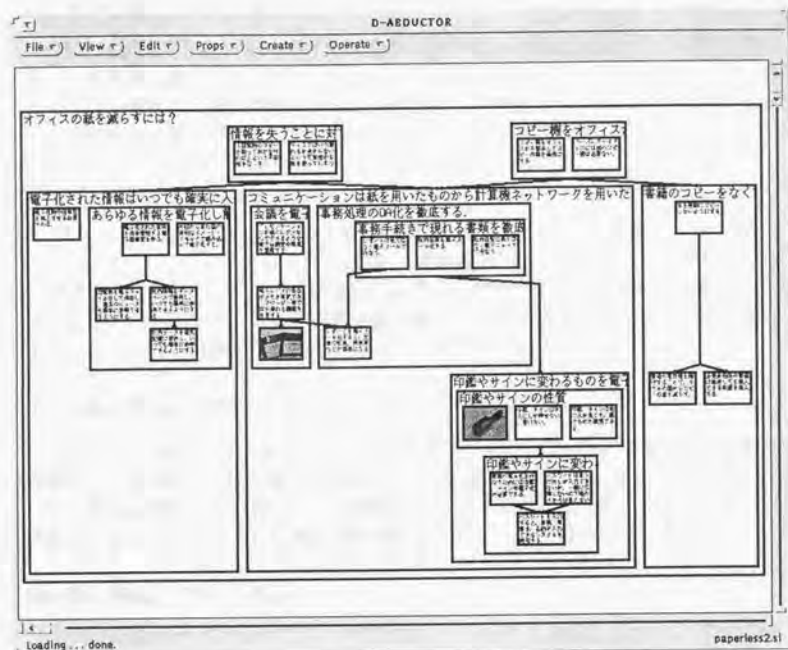


図 5.2: 複合グラフの自動描画例

(図ドレッシング未使用)

5.4.1 多様な要求への対応例

図 5.3 によって、表 5.1 にあげたような多様な要求に対応できることを示す。以下、各図ごとに、対応している要求と重要度関数のパラメータや利用している自動図ドレッサについて具体的に説明する。

包含関係の深さによる分類

図 5.3(a) はネストになった包含関係の深さごとに頂点をサイズで分類した例である。重要度関数は式 (5.8) の定義において、 $w_s = 1$, $w_f = w_m = 0$ として構造重要度だけを利用し、自動図ドレッサは連続型サイズ操作でとした。深い位置にある頂点ほど小さくなっているのが概略化とみなすこともできる。

意味の可視化/テキストへの適合

図 5.3(b) は各頂点の意味属性を可視化した例である。重要度関数は式 (5.8) の定義において、 $w_m = 1$, $w_f = w_s = 0$ として意味重要度だけを利用し、自動図ドレッサは図 5.3(a) と同じにした。式 (5.6) では頂点の文字数で意味重要度を定義したが、例では頂点の中のテキスト等を番号に置き換えたので、番号が表す数値を意味重要度として用いた。実際には、文字を多く含む頂点ほど大きく描かれる。

視点近傍の強調/詳細化

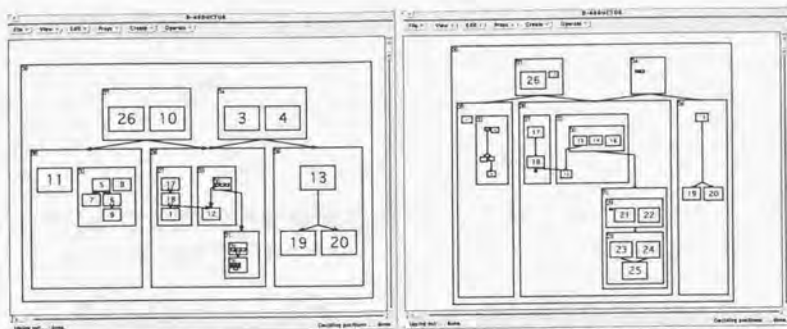
図 5.3(c) は視点となる頂点とその近傍を拡大することで、視点付近を強調または詳細化した例である。重要度関数は式 (5.8) の定義において、 $w_f = 1$, $w_s = w_m = 0$ として視点重要度だけを利用し、自動図ドレッサは図 5.3(a) と同じにした。例では頂点 6 に視点があり、それに構造的に近い頂点ほど大きく描かれている。

全体図の概略化 (全体を表示)

図 5.3(d) は全体図を概略化した例である。重要度関数は図 (a) と同じく構造重要度だけを利用し、自動図ドレッサは離散型サイズ操作で $t_2 = 0.45$ とした。ネストの深さ (一番外側を 0 とする) が 3 以上の頂点が詳細部分として縮小されている。すべての図素が表示されているので図の構造が完全に把握できる。

全体図の概略化 (詳細を省略)

図 5.3(e) も (d) と同じく全体図を概略化した例である。重要度関数は図 (a) と同じく構造重要度だけを利用し、自動図ドレッサは可視性操作で $t_1 = 0.45$ とした。離散型サイズ操作の例 (図 5.3(d)) と異なって、ネストの深さが 3 以上の頂点は表示されずに省略されている。



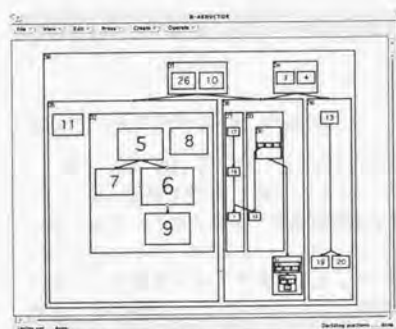
(a) 包含関係の深さによる分類

$$w_s = 1, w_m = w_f = 0$$

連続型サイズ操作: $r = 1.3$

(b) 意味の可視化/テキストへの適合

$$w_s = 0, w_m = 1, w_f = 0$$

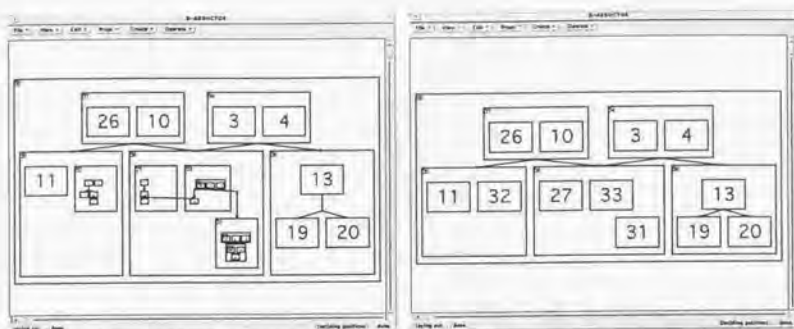
連続型サイズ操作: $r = 1.3$ 

(c) 視点近傍の強調/詳細化

$$w_s = w_m = 0, w_f = 1, \text{視点: } 6$$

連続型サイズ操作: $r = 1.3$

図 5.3: 多様な要求への対応例



(d) 全体図の概略化 (全体を表示)

$$w_s = 1, w_m = w_f = 0$$

離散型サイズ操作: $t_2 = 0.45$

(e) 全体図の概略化 (詳細を省略)

$$w_s = 1, w_m = w_f = 0$$

可視性操作: $t_1 = 0.45$

図 5.3: 多様な要求への対応例 (つづき)

5.4.2 組合せ要求への対応例

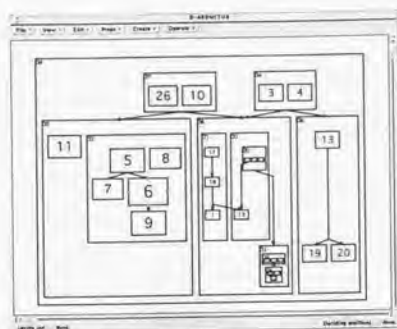
図 5.4 によって、二つ以上の基本的な「見方」を組み合わせることで同時に満たすことが可能なことを示す。前節と同様に、対応している要求と重要度関数のパラメータや利用している自動図ドレッサについて具体的に説明する。

概略化と視点近傍の強調／詳細化

図 5.4 は、先に示した、図 5.3(a) 包含関係の深さによる分類 (すでに述べたように、これは概略化として捉えることもできる) と図 5.3(c) 視点近傍の強調／詳細化を組み合わせた例である。重要度関数は式 (5.8) の定義において図 5.3(a) と (c) の中間になるように、 $w_s = w_f = 0.5$, $w_m = 0$ として構造重要度と視点重要度の両方を利用し、自動図ドレッサは図 5.3(a) や (c) と同じにした。番号が 6 の頂点に視点があり、階層の深い部分が縮小されて概略化されながらも、視点とその近傍は拡大されている。

5.4.3 図の動的な変化への対応例

構造の変化と視点の変化を例に図の動的な変化に対応できることを示す。例において、ユーザは構造または視点を変化させる操作を行なっているだけで、図ドレッシングに関しては何も操作していない。



概略化と視点近傍の強調/詳細化

 $w_n = 0.5, w_m = 0, w_f = 0.5$, 視点:6
連続型サイズ操作: $r = 1.3$

図 5.4: 組合せ要求への対応例

構造の変化

図 5.5(a) は図 5.3(d) に示す状態で図の構造が変化するような編集操作を施した例である。ユーザは編集操作として頂点 28(頂点 23, 24, 25 を包含する頂点) を別の位置に移動(マウスでドラッグ)しただけである。

頂点 28 は、図 5.3(d) では深さ 3 の位置にあり、それに包含される三つの頂点と共に縮小されていたが、図 5.5(a) では深さ 1 の位置に移動したため、包含される頂点も共に拡大された。

視点の変化

図 5.5(b) は図 5.3(c) に示す状態で、視点を変更した例である。ユーザは視点変更として、頂点 10 と 22 を新しい視点として指定(頂点 10 をマウスの左ボタンでクリック、頂点 22 を中ボタンでクリック)しただけである。

図 5.3(c) では頂点 6 とその近傍が拡大されていたのに対して、図 5.5(b) では頂点 10 と 22 およびその近傍が拡大されている。



(a) 構造の変化

$$w_s = 1, w_m = w_f = 0$$

離散型サイズ操作: $t_2 = 0.45$

(b) 視点の変化

$$w_s = w_m = 0, w_f = 1, \text{視点: } 10, 22$$

連続型サイズ操作: $r = 1.3$

図 5.5: 図の動的な変化への対応例

第 6 章

図の表示技術

図の自動配置および自動装飾によって得られる幾何情報は大きさや解像度などの制限がない仮想的な 2 次元平面に描かれた図と見なすことができる。そうして得られた図をユーザに実際に見せるためにはワークステーションの画面上に表示する必要がある。その際、仮想的な平面上の図が小さいものなら問題はないが、実際には大規模な図も多く使われるため作業空間のサイズが問題になる [Sug89, Sug90]。すなわち、図は一覧性が特に重要視されるにも関わらず、ワークステーションの画面の限られたサイズと解像度では大きな図の全体を詳細に表示することが困難という問題である。

このことは一般的な問題として次のように言うこともできる。つまり、図解、図面、テキストなどを対象とする視覚的対話型システムにおいて、表示対象の全体を詳細に表示するには画面のサイズや解像度が十分でないとき、詳細情報を見たいという要求と全体構造や視点位置を把握するための大局的情報を見たいという要求を同時に満たす問題である。限られた領域内で全体的な情報と詳細な情報をいかに組み合わせるかという問題をここでは「全体視・詳細視問題」と呼ぶことにする。

本章では、全体視・詳細視問題に対して、新しい表示方式「多視点遠近画法」[MS89a, MS90, MS91b, MS91a, MELS95] を提案する。これは、写像によって平面上の図を変形させることで、全体の概略と部分的な詳細を 1 枚の図として同時に表示するものである。詳細性を必要とする視点の近くを拡大してより近くに見せ、概略だけでも十分な視点から離れた部分を縮小してより遠くに見せることから、「遠近画法」と呼ばれる。

以下、6.1 節で、まずヒューマン・インタフェースの観点から表示方式への要求を整理する。6.2 節では、その要求に基づいて全体視・詳細視問題を解決するための新しい表示方式「多視点遠近画法」を提案する。3 種類の多視点遠近画法、「魚眼表示方式」、「直交魚眼表示方式」、「二様表示方式」を定式化し、それらの性質を整理する。6.3 節では、対象を複合グラフに絞り、実際の適用例を示すと共に、「二様表示方式」が複合グラフに対して有効であることを示す。

6.1 表示方式への要求

図解、図面、テキストなど計算機で取り扱われる様々な表示対象は、ある写像により表示像に写され、表示装置に表示される。この一連の方式を表示方式と呼ぶ。視覚的な対話型システムで作業する人間が抱き得る表示方式への要求として次のものが考えられる。

(1) 詳細性

表示対象の(必要な部分の)詳細を見ることができること。すなわち、文字や記号の解説、線種やフォントの識別などが可能なこと。

(2) 全体性

全体構造の把握、視点の周囲や位置の把握、遠く離れた部分の参照などのために、表示対象の全体が表示されること。

(3) 同時性

表示の切り換えなどで作業を中断したり、表示していない情報の記憶を人間に強いいために、必要な全情報が同時に表示されること。

(4) 表示像の単一性

同一対象物に対する複数像間での同一性の判定を人間に強いいために、一時には単一像だけが表示されること。

表示方式としてはこれら四つの要求を全て満たすことが望ましい。また全体視・詳細視問題のためには全体性と詳細性は必ず満たされなければならない。

現在多く採用されている表示方式には、スクリーンエディタのように常に詳細図を表示する「詳細図方式」、拡大率の異なる図を画面を切り換えて表示する「画面切替方式」、同様な複数の図をマルチウィンドウなどで同時に表示する「多画面方式」などがある。これらを、全体性と詳細性を満たすことを前提として、他の二つの要求を満たすかどうかで分類したものが表6.1である。表中の「多視点遠近画法」は本章で提案する表示方式である。詳細図方式は全体性を満たさないため表には現れない。

表6.1の示す通り、画面切替方式と多画面方式は同時性または表示像の単一性を満たしていない。その理由はそれらが相似写像を採用しているからである。相似写像では、限られた表示領域において詳細性と全体性の両方を一つの表示像で満たすことができず、両方を満たすためには同時性が表示像の単一性が失われる。

どのような写像を用いるべきかは、表示対象や利用目的に密接に関連している。表6.2に示したような写像の性質の内から、表示対象や利用目的に対して適切なものを選択する必要がある。このことは上記の四つの要求の他に、次のような要求が存在することを意味している。

表 6.1: 表示方式の分類

		表示像の単一性	
		不満足	満足
時 性	不満足		画面切替方式*1
	満足	多画面方式 ・ スーパーインポーズ型*2 ・ タイリング型*3 ・ オーバラップ型*4	多視点遠近画法 ・ 魚眼表示方式 ・ 直交魚眼表示方式 ・ 二様表示方式

*1 ワードプロセッサ [Fujitsu] など

*2 KJ エディタ [OKK90]

*3 SuperPaint [Sil91](Macintosh ソフトウェア) など

*4 xdvi(X ウィンドウシステムのアプリケーション) など

(5) 写像の適性

表示対象の性質を損なわずに見ることができるために、写像の性質が適切であること。たとえば、形状を重視する機械部品などの図面では相似性が、分布地図などでは正積性が、鉄道の路線図では同相性が必要である。

次節で提案する「多視点遠近画法」は、上記の五つの要求を満たすものとして考えられている。ただし、写像の適性が表示対象や利用目的に依存するため、それらに対する表示方式の適切さの考察が重要である。

表 6.2: 写像の性質

相似性	形状、角を保存
正距性	2点間の距離の比を保存
正積性	領域の面積比を保存
正方位性	ある点に対する他の点の方位を保存
直交性	直交する2直線の直交性を保存
直線性	直線の直線性を保存
同相性	位相を保存
拡大率	拡大率に関する性質(一定、連続など)

6.2 多視点遠近画法

多視点遠近画は、全体の概略と部分的な詳細を1枚の図として同時に表示する方式である。多視点遠近画法の図はある写像によって図を変形することで得られる。その写像は、図全体を一定の大きさに保つと同時に視点部分を局所的に拡大する性質をもつ。そのため、その写像によって写された図は拡大率、縮小率が図全体に渡って一様ではない。ただし、最低限、図の位相を保存するために、その写像は対象となる図を含む範囲で局所的に同相であるとする。

以下、実際に写像を定義することで多視点遠近画法を3種類定式化する。ただし、いずれにおいても元の点 (x, y) が (x', y') に写されるとする。

6.2.1 魚眼表示方式

無限の領域を有限の領域に写すために関数 \tan^{-1} を用いて写像を構成した。この写像によると、視点が一つの場合に魚眼レンズを通して見たような図が得られるので魚眼表示方式と呼ばれる(図6.3(a), (b)参照)。ただし、魚眼レンズとは異なり、魚眼表示方式では複数の視点を同時に拡大することも可能である。魚眼表示方式で用いる写像は次のように記述される:

$$(6.1) \quad x' = \frac{1}{n} \sum_{i=0}^{n-1} \phi_i$$

$$(6.2) \quad y' = \frac{1}{n} \sum_{i=0}^{n-1} \psi_i$$

$$(6.3) \quad \phi_i = \begin{cases} 0 & \text{if } x = p_i \text{ and } y = q_i \\ l_i \cdot \cos \theta_i & \text{otherwise} \end{cases}$$

$$(6.4) \quad \psi_i = \begin{cases} 0 & \text{if } x = p_i \text{ and } y = q_i \\ l_i \cdot \sin \theta_i & \text{otherwise} \end{cases}$$

$$(6.5) \quad \theta_i = \tan^{-1} \frac{y - q_i}{x - p_i}$$

$$(6.6) \quad l_i = \frac{2r}{\pi} \cdot \tan^{-1} \frac{\sqrt{(x - p_i)^2 + (y - q_i)^2}}{s_i}$$

($i = 0, 1, \dots, n-1$).

(p_i, q_i) は元の図における各視点の座標である。式(6.5)は視点 (p_i, q_i) に対する方位角 θ_i を求め、式(6.6)は視点 (p_i, q_i) からの新しい距離 l_i を定数 s_i と関数 \tan^{-1} で決定する。写された図の全体を半径 r の円内に入れるために定数 $2r/\pi$ を掛けている。式

(6.3) と (6.4) が各視点一つに対する写像で、視点に対する方位角を保存し、視点からの距離が l_i の位置に点を写す。各視点ごとの関数 (6.3), (6.4) の相加平均により、式 (6.1) および (6.2) のように n 視点の写像が与えられる (図 6.3 参照)。

6.2.2 直交魚眼表示方式

魚眼表示方式では直線が曲線に写され図の形状が大きく歪むことが大きな特徴である。その魚眼表示方式を修正し x 座標と y 座標の写像を独立にして、 x 軸または y 軸に平行な直線の直線性とそれらの直交性を保存するようにしたもの⁶、直交魚眼表示方式である。直交魚眼表示方式で用いる写像は次のように記述される：

$$(6.7) \quad x' = \frac{1}{n} \sum_{i=0}^{n-1} \phi_i$$

$$(6.8) \quad y' = \frac{1}{n} \sum_{i=0}^{n-1} \psi_i$$

$$(6.9) \quad \phi_i = \frac{r}{\pi} \cdot \tan^{-1} \frac{x - p_i}{s_i}$$

$$(6.10) \quad \psi_i = \frac{r}{\pi} \cdot \tan^{-1} \frac{y - q_i}{s_i} \\ (i = 0, 1, \dots, n-1).$$

(p_i, q_i) は元の図における各視点の座標である。式 (6.9), (6.10) が各視点一つに対する写像で、各座標軸ごとに視点 (p_i, q_i) からの新しい距離を定数 s_i と関数 \tan^{-1} で決定する。写された図の全体を 1 辺の長さが r の正方形内に入れるために定数 r/π を掛けている。各視点ごとの関数 (6.9) および (6.10) のそれぞれの相加平均により、式 (6.7), (6.8) のように n 視点の写像が与えられる (図 6.4 参照)。

魚眼表示方式では無限の領域を一定の大きさの円内に、また直交魚眼表示方式では無限の領域を一定の大きさの正方形内に写している。このことから、これらの表示方式では対象のサイズが大きくなるほど、全体の形状が円または正方形に近付くため、元の形状が保存されない傾向にある。そして、「図の右上の方」とか「中心のやや左側」といった図全体に対する視点の位置把握が困難になる。また、無限の領域を有限の領域に写すことで、視点から遠い部分ほど縮小される (視点から無限に離れた部分は無限に縮小される) ため、理論的には全てのものが表示されるが、実際上は周辺がつぶれて見えなくなる。

6.2.3 二様表示方式

魚眼表示方式や直交魚眼表示方式の欠点を克服するために、外形形状を保存し、周辺部が無限に縮小されることのない表示方式を構成した。この表示方式は魚眼表示

方式などとは異なり有限の長方形領域を相似の長方形領域に写すもので、各座標軸ごとに視点領域が一樣に拡大され、その他の領域が一樣に縮小されることから二様表示方式と呼ばれる。視点領域に対しては、両軸方向の拡大率が等しいので相似拡大となる。

図6.1を用いてより具体的に説明する。最大の長方形が図の外枠に対応し、太線で表された四つの長方形が視点領域であるとする、////の部分が一様軸方向に一樣に拡大され、\\\\の部分が一様軸方向に一樣に拡大される。残りの部分は図の外枠の形状を変えないように各軸ごとに一樣に縮小される。

二様表示方式の用いる写像は次のように記述される：

$$(6.11) \quad x' = \begin{cases} b_0 + \lambda(x - b_i) + \sigma \sum_{k=1}^i (b_k - a_k) + \lambda \sum_{k=1}^i (a_k - b_{k-1}) \\ \quad \text{if } b_i < x \leq a_{i+1} \quad (i = 0, 1, \dots, m) \\ b_0 + \sigma(x - a_i) + \sigma \sum_{k=1}^{i-1} (b_k - a_k) + \lambda \sum_{k=1}^i (a_k - b_{k-1}) \\ \quad \text{if } a_i < x \leq b_i \quad (i = 1, 2, \dots, m). \end{cases}$$

$$(6.12) \quad y' = \begin{cases} d_0 + \mu(y - d_j) + \sigma \sum_{l=1}^j (d_l - c_l) + \mu \sum_{l=1}^j (c_l - d_{l-1}) \\ \quad \text{if } d_j < y \leq c_{j+1} \quad (j = 0, 1, \dots, n) \\ d_0 + \sigma(y - c_j) + \sigma \sum_{l=1}^{j-1} (d_l - c_l) + \mu \sum_{l=1}^j (c_l - d_{l-1}) \\ \quad \text{if } c_j < y \leq d_j \quad (j = 1, 2, \dots, n). \end{cases}$$

ここで、 m と n はそれぞれ x 軸方向と y 軸方向の拡大領域の数である。図6.1に示されるように、図全体における拡大領域の数と x 軸方向、 y 軸方向の拡大領域の数は一般に異なる。 x 軸方向の拡大領域と y 軸方向の拡大領域はそれぞれ $\bigcup_{i=1}^m [a_i, b_i]$, $\bigcup_{j=1}^n [c_j, d_j]$ となり、それ以外の領域が縮小される。 (b_0, d_0) と (b_{m+1}, d_{n+1}) は、それぞれ図の外枠にあたる長方形の左下の頂点と右上の頂点である。 σ は視点領域の拡大率である。 λ と μ は視点領域以外のそれぞれ x 軸方向と y 軸方向の拡大率で、 σ から次のように定められる。

$$(6.13) \quad \lambda = \frac{(a_{m+1} - b_0) - \sigma \sum_{k=1}^m (b_k - a_k)}{\sum_{k=1}^{m+1} (a_k - b_{k-1})}$$

$$(6.14) \quad \mu = \frac{(c_{n+1} - d_0) - \sigma \sum_{l=1}^n (d_l - c_l)}{\sum_{l=1}^{n+1} (c_l - d_{l-1})}$$

ここに示した二様表示方式では、視点領域が全て一樣に拡大されるが、拡大領域が x 座標、 y 座標共に重なり合わない場合には拡大領域ごとに異なる拡大率で拡大することも可能である。その他、二様表示方式のように外形形状を保存し、かつ魚眼表示方式のように拡大率が連続的に変化するものも考えられるが、写像はかなり複雑になる。

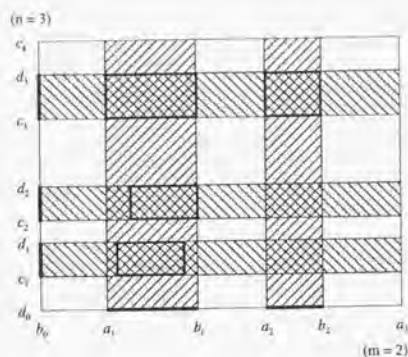


図 6.1: 二様表示方式で用いる写像における拡大領域

6.2.4 多視点遠近画法と写像の性質

多視点遠近画法の各方式が用いる写像の性質は表 6.3 のようになる。表中の○は写像がその性質を満たすことを示し、×は満たさないことを示す。また、△はその性質を条件または制限つきで満たすことを示し、右肩に付与されたアルファベットが各性質を満たす条件または制限を示す。アルファベットが二つ付与されたものはどちらかの条件の下で満たされる。

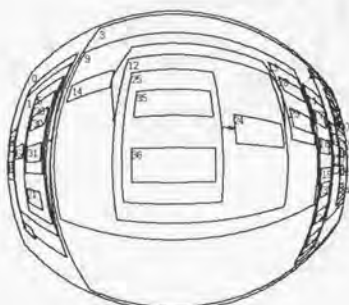
6.3 複合グラフの表示のための多視点遠近画法

先に 6.1 節で述べたように、表示方式への要求の内、写像の適性は、表示対象や利用目的に依存する。ここでは、表示対象を複合グラフに限定し、図の一覧性達成を目的として、多視点遠近画法の適用を考える。

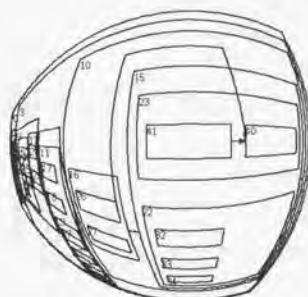
6.3.1 多視点遠近画法による表示例

図 6.2 に示した複合グラフを魚眼表示方式を用いて表示した例を図 6.3 に示す。(a) と (b) はどちらも視点数が 1 の例である。(c) は (a) における視点と (b) における視点の両方を同時に視点にしたもので視点数が 2 である。式 (6.1) (6.2) から判るように、(c) の各点の座標は、対応する (a) の点の座標と (b) の点の座標の相加平均、この場合は (a) の点と (b) の点の中点、になっている。

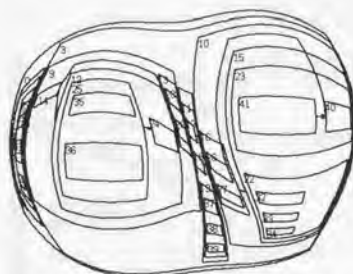
直交魚眼表示方式を用いて同じ複合グラフを表示した例を図 6.4 に示す。(a)、(b)、(c) の各々の関係は図 6.3 と同様である。



(a) 視点:36 (視点数 $n = 1$)



(b) 視点:41 (視点数 $n = 1$)



(c) 視点:36, 41 (視点数 $n = 2$)

図 6.3: 魚眼表示方式による表示例

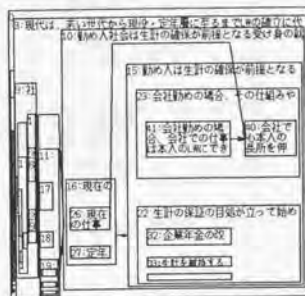
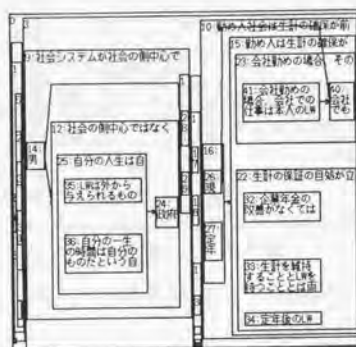
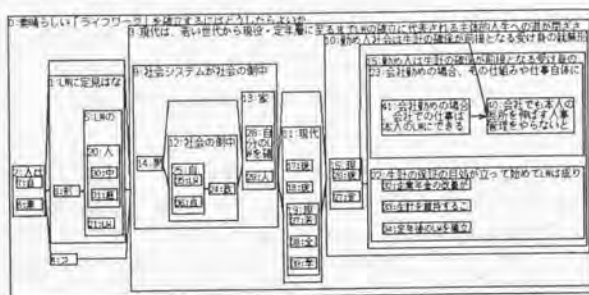
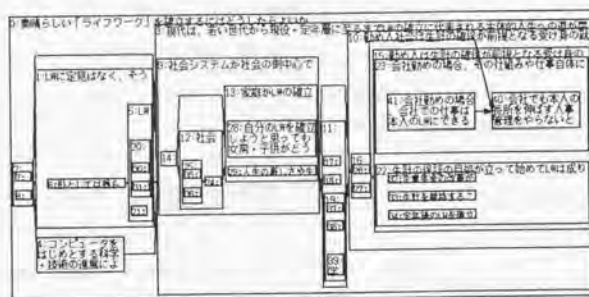
(a) 視点:36 (視点数 $n = 1$)(b) 視点:41 (視点数 $n = 1$)(c) 視点:36, 41 (視点数 $n = 2$)

図 6.4: 直交魚眼表示方式による表示例

二様表示方式を用いて同じ複合グラフを表示した例を図6.5に示す。(a)は視点数が1, (b)は視点数が3の例である。図6.2と比較すると、外枠の形状が変化せず視点領域が相似拡大されていることが判る。



(a) 視点:23 (視点数 $n=1$)



(b) 視点:4, 23, 28 (視点数 $n=3$)

図6.5: 二様表示方式による表示例

6.3.2 複合グラフへの適合性

ある特定の表示対象に対する多視点遠近画法の適合性を検証するためには、表示方式への要求のうち写像の適性の検証が必要となる。その他の要求は満たされているので考慮不要である。

表示対象がグラフに基づく図である場合、写像の性質として同相性は必要条件となる。さらに、グラフが何らかの描画規約に基づいて平面上に描かれている場合、

その描画規約に整合する写像の性質が満たされるべきである。表4.1の描画規約に対応して必要とされる写像の性質は下記ようになる。

- C1: x 軸または y 軸に平行な直線の直線性, 直交性が満たされる。
- C2: 対象を含む範囲で同相性が満たされる。
- C3: 直線は, 直線または折れ線に写る。
- C4: x 軸または y 軸に平行な直線の直線性と, x 軸方向と y 軸方向の方向性が満たされる。

これを写像の性質について整理すると表6.4のようなになる。表6.3と表6.4を比較して, 多視点遠近画法の複合グラフへの適合性を判断する。同相性については全ての方式が満足している。しかし, 魚眼表示方式は直交性と直線性を満たさないのが不適当である。直交魚眼表示方式は x 軸または y 軸に平行な直線については直交性と直線性を満たすが, x 軸と y 軸の両方に平行でない直線の直線性は依然満たされない。二様表示方式は上記条件を全て満たす。

表6.4: 階層描画された複合グラフに必要な写像の性質

正方位性	x 軸方向と y 軸方向の方向性を保存
直交性	x 軸または y 軸に平行な直線について満たされる
直線性	x 軸または y 軸に平行な直線について満たされ, それ以外の直線は折れ線に写っても良い
同相性	対象を含む範囲で同相

以上より, 階層的に描かれた複合グラフを表示対象とする場合, 二様表示方式は表示方式への五つの要求, 写像の適性, 詳細性, 全体性, 同時性, 表示像の単一性, を全て満足する有効な表示方式であるといえる。また, 表6.3に示される通り, 二様表示方式は, 条件付きながら最も多くの写像の性質を満たしている。そして, 局所的ではあるが, 視点領域で写像の性質として望ましい相似性が満たされている。これらのことから, 複合グラフ以外の表示対象に対する二様表示方式の有効性も十分に期待できる。

第 7 章

システムへの統合

第 4 章から第 6 章で、領域連結複合系の図の可視化プロセスを構成する要素技術の各々について説明した。これらの要素技術を DP において実際に活用するあるいはその有効性を検証するためには、さらに論理情報や幾何情報の編集機能やそれらを包括するユーザインタフェース機能などを組み合わせて、一つの対話型システムとして統合することが重要である。

対話型システムの応用領域としては、発想法として広く知られている KJ 法 [Kaw86] を参考にした。KJ 法の作業を図的な観点から捉えようと、図を描き替えながら思考を展開する過程としてとらえることができる。さらにそこで使われる図は領域連結複合系の図である。筆者らは、従来机上で紙のカードを用いて行なわれていた KJ 法の作業を計算機上でより効率良く行なえるような DP の開発を目指した。

本章では、領域連結複合系図表現に基づく DP である、図的発想支援システム D-ABDUCTOR [Mis90b, MS93, Mis93a, MS94d, MS94a, SM96] の開発について述べる。D-ABDUCTOR の特徴は次のようにまとめることができる。

- 「ラベルづくり」から「図解化」を経て「叙述化」に至る「狭義の KJ 法」と呼ばれる過程を「図を描き替えながら思考を展開する過程 (図的思考展開過程)」としてとらえ、対話型支援システムとして実現している。
- 図の高度な編集を可能にし、操作を容易にする高機能グラフィック・インタフェース (GUI) の提供に最も重点を置いている。
- 直接操作環境とアニメーション環境 [Mis93a] を実現し、その環境のもとで自動描画機能を利用することにより、図の編集作業におけるユーザの手間と認知的負担を大幅に軽減している。また、自動描画機能の利用の仕方をユーザが選択できる機能も用意している。
- 上記機能のほか、通信機能 [Mis93b] やマルチメディア機能などの計算機的能力を活用することにより、グループによる思考支援や画像データを用いた思考支援など新しい可能性を実現している。

以下、7.1節ではD-ABDUCTORの開発の基本的な考え方を簡単に述べ、7.2節では今回実現したD-ABDUCTORが提供する機能について詳しく説明する。7.3節では実施例を用いてD-ABDUCTORの利用法を示し、7.4節ではユーザの実作業に対する試用報告に基づいてD-ABDUCTORを評価する。

7.1 D-ABDUCTOR 開発の考え方

KJ法は、研究、教育、ビジネス、産業などの広範な分野における定性的データに基づいた問題解決への系統的接近法として有名である。KJ法は問題提起から結果を味わうまでの「一仕事」の全体を対象としているが、このうち「狭義のKJ法」と呼ばれる、本質追求の過程がKJ法の核心をなしている。本稿では、狭義のKJ法をあつかい、以後単に「KJ法」と呼ぶ。

KJ法1ラウンドは大きく分けると、(1)ラベルづくり、(2)グループ編成、(3)A型図解化、(4)B型叙述化、の4ステップからなるが、これらのステップにおける手順は小道具を用いた手作業の体系として提示されている。筆者らはまずKJ法を図的思考展開過程としてその特徴を分析し、研究課題を抽出するとともに、KJ法の手順を図言語とその操作の観点から考察した[Sug89, SM90]。表7.1に各ステップにおける作業内容と図的操作を示す。

筆者らは、KJ法を参考とした図的発想支援システムD-ABDUCTORの開発に際し、次のような考え方のもとに研究を進めた。ただし、実際の作業は下記の順序通りには進行せず、むしろ、(2)-(4)は繰り返し並行して行われた。

- (1) 表7.1に示したようなKJ法の手作業の模擬機能をD-ABDUCTORのための必須機能として実現する。
- (2) 手作業のうち手間がかかり非本質的な作業を計算機的能力を使って代行できる部分がないかを考察し、そのような機能を実現する。
- (3) 計算機上で行うことで手作業の場合に比べて不利になることはないかを考え、それを克服するための機能を実現する。
- (4) 計算機環境において情報処理技術を活用することで手作業では行えなかった新たな可能性について考察し、そのような機能を実現する。

7.2 D-ABDUCTOR の機能

図7.1は本システムの構成図である。本システムはUNIXワークステーション上でXウィンドウ・システムのアプリケーションとして開発された。スタンド・アローンのワークステーションでも使えるが、ネットワークでつながれたワークステーションによりグループの協同思考作業にも利用できる。

表 7.1: KJ 法の各ステップにおける作業内容とそこで行なわれる図的操作

ステップ		作業内容	図的操作
(1)	ラベルづくり	アイデアや情報を収集してカードに書く。	カードの作成、削除、テキスト、図、絵の記入。
(2) グループ 編成	(2-1) ラベル並び	カードを(ランダムに)並べる。	カードの移動。
	(2-2) ラベル集め	関係のあるカードを近くに集めてグループを作る。	カードの移動。
	(2-3) 表札作り	各グループに表札(別のカード)をつけ、まとめて1枚のカードとして扱う。	カードの作成、テキストの記入、グループをカード(表札)にまとめる。
(3) A型 図解化	(3-1) 空間配置	グループを展開しながらカードを適切に配置する。	カードの移動、グループの移動、カード(表札)をグループに展開する。
	(3-2) 図解化	グループの輪郭線やカード間の関係線を描く。	輪郭線の描画、関係線の描画。
(4)	B型叙述化	図解の内容を文章化する、または、口頭で発表する。	図解のテキストへの変換。

以下、本システムが提供する機能それぞれについて前章で述べた開発の考え方の順に沿って説明する。

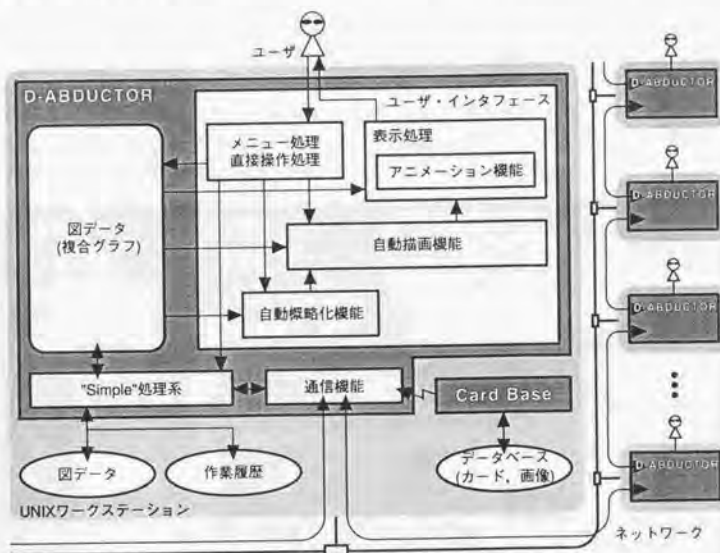


図 7.1: 図的発想支援システム D-ABDUCTOR の構成

7.2.1 計算機上での手作業の模擬機能

複合グラフ KJ 法で用いられる図解は、図的な分類 [IYA86] に従うと、領域による包含関係と関係線による隣接関係が表現できる領域連結複合系の図である。本システムでは領域連結複合系の図を表現できるようにグラフを拡張したデータを扱う。それは複合グラフと呼ばれ、数学的には複数種類の辺を持つグラフとして定義される [MS89b, SM91] (第4章)。本システムでは、カードと関係線をそれぞれノードとエッジと呼ぶ。

操作環境 マウスによる直接操作、メニューによるコマンドの選択、ダイアログ・ボックスを通じたシステムとの対話など、簡単で分かりやすい操作環境を提供している。特に、作業の中心となる、ノードの移動、エッジの作成、グループ (他のノードを包含するノード) へのメンバー (ノード) の追加、削除など図の編集操作はすべ

てマウスによる直接操作だけで行なえる。図7.2にメニューやダイアログ・ボックスを含むD-ABDUCTORの画面例を示す。

視覚レスポンス 各操作に対して豊富な視覚レスポンスを提供している。たとえば、図の変更を引き起こすような操作や設定変更を行なった場合には直ちにその変化を視覚的に知ることができる。また、ラバーバンドを用いてノードやエッジを選択するときの選択候補や、ノードの移動によってグループを変更するときの新グループの候補、エッジを作成するときの終ノードの候補など、すべてハンドルを動的に付加消去することで視覚的に示される。

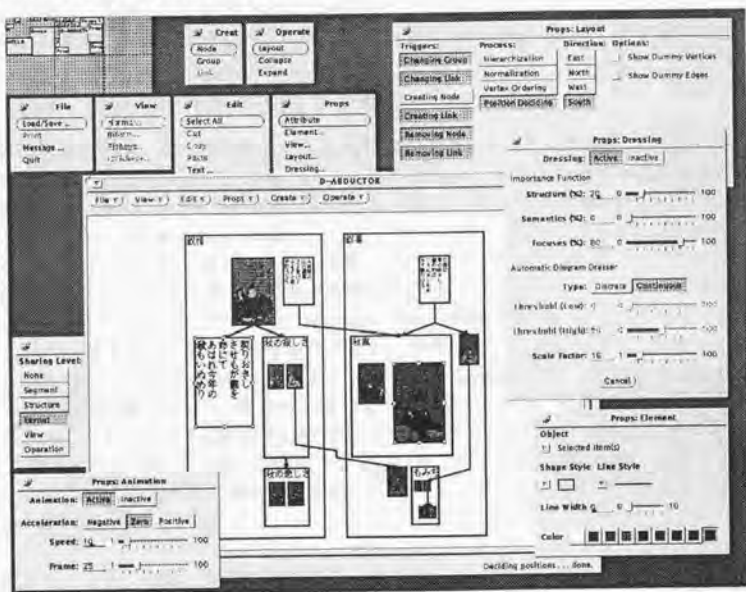


図7.2: D-ABDUCTORの画面例

主ウィンドウとともにメニューとダイアログ・ボックスを表示している。

7.2.2 計算機による手作業の代行機能

自動描画機能 手作業による図の再配置や書き替えは面倒な作業である。そこで、本システムでは認知的基準に基づいて複合グラフを見やすく美しい図として描く描画アルゴリズム [MS89b, SM91](第4章)を実装した自動描画機能を提供している。

自動描画機能は、図の清書機能としていつでもメニューから選んで実行できる。図7.3は7.3節で説明するKJ法的作業の画面例であるが、このうち(d)が自動描画機能の適用前、(e)が適用後である。この自動描画機能は清書機能だけでなく、次に説明するインクリメンタル自動レイアウト機能や自動概略化機能など本システムが提供する多くの機能の基礎をなしている。

インクリメンタル自動レイアウト機能 図の編集においてユーザが最小限の指定を行なうだけで図が適切に描き替えられるとありがたい。本システムではノード、グループ、エッジの追加、削除、変更などの編集操作のうち適当なものをユーザが自動描画のトリガーとして設定することを許している。それにより、図を編集するごとに、直ちに図が描き替えられるインクリメンタル自動レイアウトが可能である。

7.2.3 計算機の不利を克服する機能

自動概略化機能 KJ法が行なわれる机の上や黒板に張り付けた模造紙に比べて一般に計算機の画面は小さすぎる。従来からの対策としてはスクロール機能などが利用されているが、それだけでは一覧性という図の重要な性質が失われてしまう。

自動概略化機能は、図の構造、ノードの意味、そしてユーザの視点に基づいて与えられる重要度に従ってノードの可視性やサイズを変更する[Mis90a]。これにより、重要なノードだけが大きく残された概略図を得ることができる。ユーザは重要度の与え方などをパラメータで調節できるので好みの概略化を行える。たとえば、重要度がユーザの視点に強く依存するようにし、重要なノードが拡大され、他のノードは存在が分かる程度に縮小されるようパラメータを設定すると、視点とその近傍が大きく詳細に描かれるが、視点から遠いノードは縮小されて、全体としては表示領域が節約できる。図7.4は重要度に応じてノードのサイズを変えて表示した例である。

ノードの拡大縮小に対してテキストも拡大縮小するかまたは固定サイズとするかはユーザが選択できる。縮小にはグレースケールを用いているので、単色での縮小より認識性がよい。

アニメーション機能 図の清書機能や、インクリメンタル自動レイアウト機能を用いる場合、計算機による描き替えが図を一瞬に急激に変化させるためユーザのメンタル・マップ[ELMS91, MELS95]を破壊して作業の効率を低下させる可能性がある。本システムではアニメーション機能により図の変化を徐々に見せることでメンタル・マップの保存を助けている。

7.2.4 進んだ計算機環境の活用機能

言語 Simple D-ABDUCTOR が扱う図のデータ、ユーザの操作、コマンドをすべて記述できる言語"Simple"を用意した[Mis93a]。これにより図をテキスト・ファイル

に保存でき、また、そのファイルを読み込んで作業を継続できる。D-ABDUCTORは言語 Simple のインタープリタとして機能するので、言語 Simple のステートメントを与えることで対話的な操作なしで D-ABDUCTOR を制御できる。これにより、次に説明する作業履歴の記録・再生や通信機能などが実現できた。

作業履歴の記録・再生 作業履歴を保存することで、前に戻って考え直すことが可能になる。また、思考履歴として分析することで、発想過程の研究にも利用できる。そこで、ユーザの作業をすべて言語 Simple のステートメントに直し、ファイルに書き出す機能を用意した。このファイルを見ることでどのような作業がどのような順で行なわれたか知ることができる。さらに、D-ABDUCTOR でそのファイルを読み込むことで、作業過程を再生することもできる。

他のワークステーションとの通信 ネットワークで接続された他のワークステーションと情報を共有する機能を提供する [Mis93b] ことで、個人だけでなくグループでの、また同室だけでなく遠隔での協同の思考作業が可能になる。D-ABDUCTOR では、言語 Simple のステートメントに直したユーザの作業を、ネットワークでつながった他のワークステーション上の D-ABDUCTOR のプロセスに送ることができる。受け取ったプロセスはそれを直ちに解釈、実行する。この機能により複数のワークステーション間でコマンドや操作を共有することが可能である。

他のツールとの通信 カードデータベース Card Base [Mis93a] を提供し、本システムと連携可能にした。Card Base は、キーワードを用いた論理式に基づいてテキストや画像を検索できるデータベース管理システムで、テキストや画像をもつノードを作成する言語 Simple のステートメントを D-ABDUCTOR に送る。

また、D-ABDUCTOR はグループによる発想作業を総合的に支援するグループ発想支援システム GrPS [KWM⁺93] において収束的思考を支援するサブシステムとしても機能する。その際、「ドラッグ・アンド・ドロップ」と呼ばれる簡単な直接操作によって他のツールからテキストや画像などの情報を受け取る機能を備えている。

マルチメディア機能 人間が思考に用いるメディアは多様であるため発想支援システムでもマルチメディア化が期待される。それにより、静止画はもちろん、紙のカードでは取り扱えなかった動画や音声もラベルとして利用可能になる。

本システムでは、マルチメディア化へ向けての第一歩として、画像の管理を行う Card Base を用意し、テキストだけでなく画像もノードに付加できるようにしている。さらには音声や動画の利用も技術的には可能であるが、図を利用した思考支援システムという観点からは、いくつかの問題が考えられる [MS94e]。

7.3 KJ 法的作業の実施例

開発したシステムを使って「D-ABDUCTOR のグループウェア化手法」というテーマで KJ 法的作業を実施した例を用いて、前章で説明した機能がどのように利用されるかを説明する。以下、各ステップで行なった操作を具体的に説明し作業途中の画面例 (図 7.3) も一緒に示す。なお、この作業は文献 [Mis93b] を作成するために実際に行われたものである。

(1) ラベルづくり テーマに関してアイデアを思い付くごとに、ノードを一つ作成しテキスト (または画像) として入力する。あらかじめ図 7.5(a) のようなラベルのデータをテキスト・エディタで作成しておいて一度に読み込ませることもできる。いずれのやり方でも、ノードはまず図 7.3(a) のように格子状に配置される。

(2) グループ編成 全体が数個のグループになるまで以下の (2-1) から (2-3) を繰り返す。(2-1) 各ノードをそれぞれが見えるように移動 (マウスでドラッグする) する。(2-2) ノードを移動して関係あるもの同士を集める (図 7.3(b))。(2-3) 近くに集まったノードを選択してグループを作成 (メニューからグループ作成を選択) し、ラベルを入力する (図 7.3(c))。作成したグループを一つのノードにまとめる (ノードのメニューから Collapse を選択)。

(3) A 型図解化 グループを展開 (ノードのメニューから Expand を選択) しながら、エッジの作成 (始ノードを選択し、ハンドルから終ノードへマウスをドラッグする) を続ける (図 7.3(d))。自動描画機能を利用すると直ちに図 7.3(e) のように清書できる。最終的に A 型図解化によって得られた図解 (最終ノード数 59) を表示した例を図 7.4 に示す。自動概略化機能により、限られたサイズの画面において図解全体を表示しながらも、一部のノードを文字が十分に読めるまでに拡大できている。

(4) B 型叙述化 完成した図解をテキストに変換 (メニューからセーブを選択しアウトライン・テキストの形式で保存) すると、図 7.5(b) に示したテキストが得られる。テキストには、図解におけるノードの配置や包含関係を反映し、章節項のようにネストした番号が付けられていて、以後はそれを基にしてテキスト・エディタでも編集できる。

7.4 試用に対するアンケート評価

発想支援システムが扱おうとしているアイデアというものは質や量に関して尺度が明確になっていないため、システムが発想にどう役立つかを客観的に評価することは現状では困難であり、評価法自体が別の研究課題とも考えられる。そこでまず筆者らは、本システムを試用中のユーザーの使用感や満足度を把握するために、各

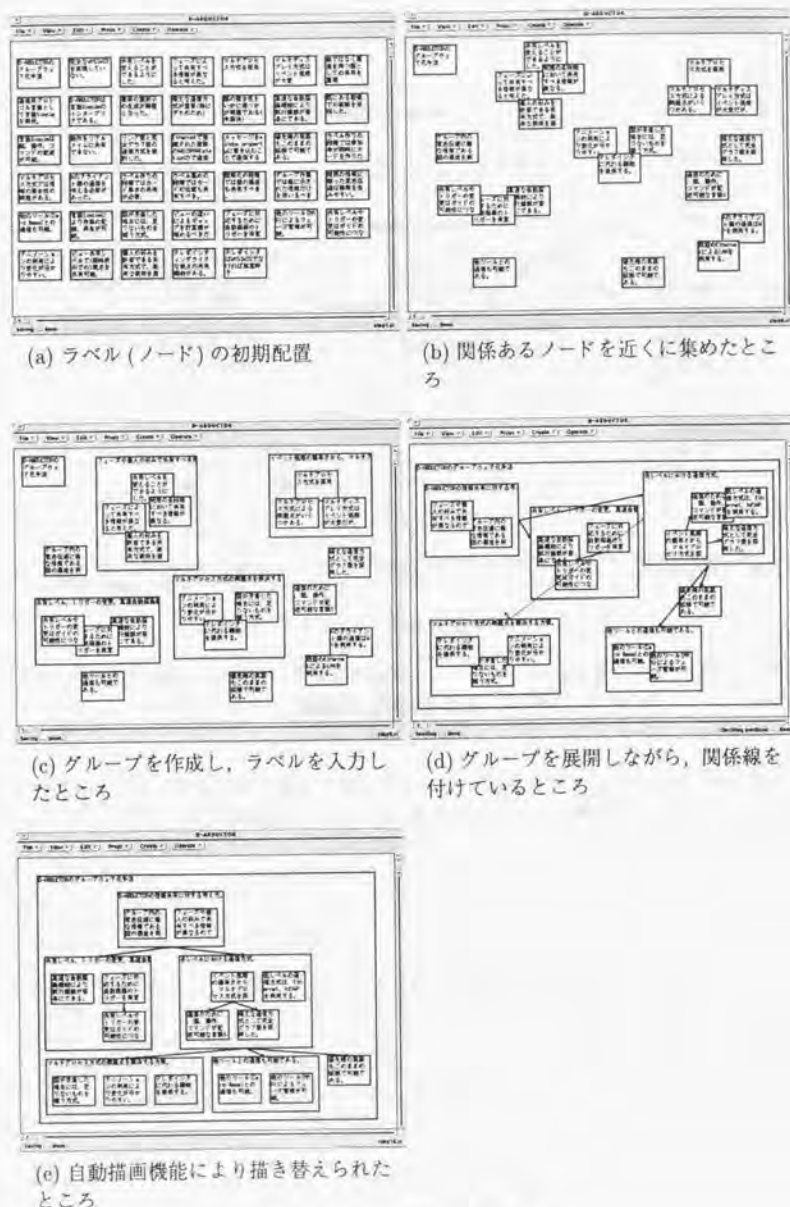


図 7.3: KJ 法的作業の画面例

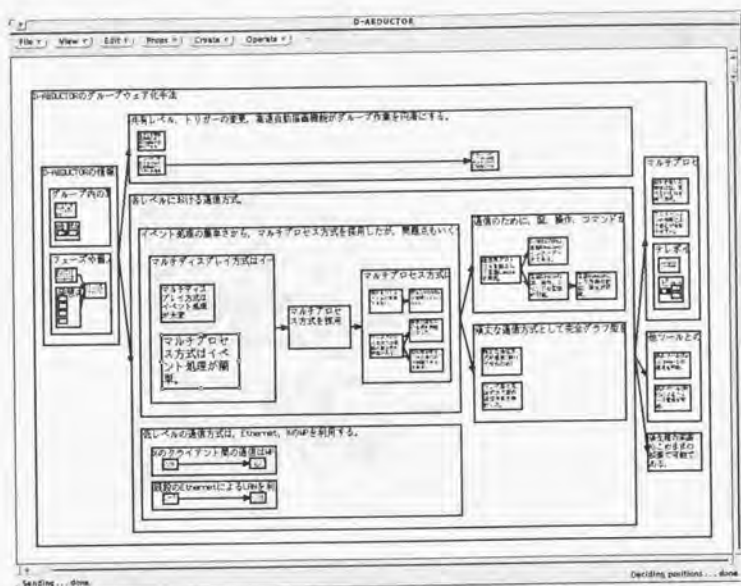


図 7.4: D-ABDUCTOR による図解の表示例

「D-ABDUCTOR のグループウェア化」というテーマで KJ 法的作業を行なって得られた図解 (自動概略化機能を使用, 最終ノード数 59)

機能やシステム全体の使い勝手に関しアンケート方式でユーザから試用報告を求め、D-ABDUCTOR の機能的評価や総合的評価を行なうことを試みた。本節では、そのアンケートの方法と取りまとめ結果を示す。システムの定量的な評価については第 8 章で述べる。

7.4.1 アンケートの方法

報告者 (アンケートに回答したユーザ) は、(株) 富士通研究所内 5 名および富士通 (株) 内 3 名、大学 2 名の計 10 名である。報告者は、日常的にワークステーションを使って仕事をしていて、ウィンドウ・システムには十分に慣れているレベルの人達である。報告者による D-ABDUCTOR の試用はこちらからの依頼ではなく、報告者側からの希望によるものである。報告者が D-ABDUCTOR を利用して行なった



(a) テキスト・ファイルとして作成したラベル



(b) 図解から変換され、番号づけされたテキスト

図 7.5: KJ 法的作業で利用するテキストの例

作業もこちらから指定したものではなく、各報告者は自分の実作業に対して自由に D-ABDUCTOR を利用した。

アンケートは全て電子メールで行なった。はじめに各ユーザに自由形式で利用状況の報告を求め、報告を受けた作業を分類し各ユーザにフィードバックするとともに、報告者のうちまだ実作業に利用していなかった(株)富士通研究所の1名および大学の1名を除く8名に評価のための詳細なアンケートを依頼した。アンケートの質問項目は次の通りである。

- A. 作業名/作業内容の概要
- B. 作業数/ワークステーション数
- C. 図解サイズ(ノード数/隣接エッジ数)
- D. 下記項目に対する5段階評価(5:とても役立つ、優れている、..., 1:全く役立たない、劣っている)とその理由
 - 1. 複合グラフ(複合グラフの表現能力)
 - 2. 操作性(直接操作環境、メニュー、ダイアログボックス)
 - 3. 視覚レスポンス(操作に対して動的にノードやエッジにつくハンドル)
 - 4. 自動描画機能(図の描き換え機能)
 - 5. インクリメンタル自動レイアウト機能¹(操作に対して自動的に図を描き変える機能)
 - 6. 自動概略化機能(視点を選択すると自動的にノードを拡大縮小する機能)
 - 7. アニメーション機能(図の変化を動画で見せる機能)
 - 8. 言語 Simple(図のデータや操作を記述する機能)
 - 9. 作業履歴の記録・再生(作業履歴を言語 Simple で記録・再生出来る機能)
 - 10. 通信機能(グループウェア機能)
 - 11. 総合(あなたの作業に対する D-ABDUCTOR の総合的な評価)
- E. 特に良い点/悪い点
- F. 全体的なコメント

¹アンケート実施時には「インクリメンタル編集機能」と呼ばれていた。

7.4.2 集計結果

報告された16種類の作業を大きく五つのカテゴリに分類した。表7.2はそのカテゴリに沿って、アンケートの質問A-Dの数量データを集計したものである。作業者数、ワークステーション数、図解サイズは報告されたもののうちそれぞれ最大のものあげた。質問Dの5段階評価は各カテゴリごとの平均と作業全体の総平均を求めた。(未使用のため評価できない場合は「—」で表した。)

利用された作業の種類

D-ABDUCTORは、もともとKJ法のようなアイデア発想・整理を想定して開発されたシステムであるが、報告者はそれ以外の様々な種類の作業に利用していた。このことは開発者が想定していたよりも用途の広がりの可能性がもっと大きいことを示している。

項目別評価

表2の項目(1)-(7)の評価に着目する(項目(8),(9),(10)はまだ利用者が少ないので考察の対象からはずす)。総平均で点が高い(4以上の)項目は、(1),(4),(6),(7)であった。これらに高い評価を与えた理由としては次のものが挙げられた((6)に関しては具体的なコメントは得られなかった)。

- (1) KJ法ライクな図解をつくるのに十分役立つ。
「流れ」と「まとまり」の両方を表現できる。
- (4) 図解作成作業が楽である。
時間短縮・思考の円滑化に効果的である。
- (7) 図の変化がとても分かりやすい。
作業の円滑化につながる。
見ていて楽しい。

項目(2),(3),(5)は総平均が4未満であった。特に(2)と(5)は2がそれぞれ6件と5件あり、その理由として次のような不満や改良要求があった。

- (2) 日本語が入力しにくい。
表示の極小状態で難がある。
COPY, PASTE, UNDOなどの機能が欲しい。
- (5) 変更の度に描き替えを待たなければならない。
動いて欲しくない時に描き替えが起こった。

表 7.2: D-ABDUCTOR の試用評価アンケート A-D の集計結果

作業	ユーザ	WS	図解サイズ	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
(I) KJ 法等によるアイデア整理 (平均)														
グループウェア利用の検討	12	3	392 / 324	4.40	3.80	3.00	4.40	4.20	4.40	4.60	3.00	3.67	5.00	4.00
ドキュメント執筆環境の検討	4	1	40 / 20	5	4	2	4	4	5	5	—	3	5	4
グループウェア執筆環境の検討	1	1	78 / 0	5	5	4	5	5	5	5	—	5	—	4
システムの問題点の整理	1	1	60 / 23	3	4	5	4	5	4	4	—	—	4	4
論文のアイデア整理	1	1	37 / 14	4	2	3	4	2	3	4	3	3	—	4
(II) 文章構成などの検討 (平均)														
OJT 報告書の題材・文章構成の検討	1	1	23 / 24	4.50	3.00	4.00	4.25	3.25	4.33	4.75	2.50	3.00	—	3.75
講義レポートの作成	1	1	15 / 50	5	3	5	4	4	2	—	5	—	—	4
プレゼンテーション手順の検討	1	1	27 / 8	4	2	3	4	2	3	4	2	—	—	3
ビデオ・プログラムの設計	1	1	123 / 70	5	5	4	5	5	5	5	—	—	—	4
(III) 構造の設計 (平均)														
データベースのスキーマ設計	1	1	28 / 7	4.50	2.50	4.50	4.00	3.00	5.00	5.00	2.00	—	—	3.50
プログラム・モジュールの機能配分	1	1	10 / 10	5	3	5	4	4	5	5	2	—	—	3
(IV) 図解の作成 (平均)														
プログラム・モジュールの関連図作成	1	1	22 / 33	4	2	4	4	2	—	5.00	—	—	—	4
OMT 法でのオブジェクトの二次元配置	2	1	19 / 29	—	3	4	5	5	—	5	—	—	—	4
(V) 図解による考察 (平均)														
プログラムの性能チューニング	1	1	50 / 69	3.50	3.33	4.00	4.00	4.33	4.00	5.00	4.00	—	—	3.33
自動生成した決定木の可視化	1	1	979 / 978	3	2	3	3	4	4	5	4	—	—	4
代数構造の図解化による学習	1	1	108 / 86	4	4	4	5	5	3	5	—	—	—	2
総平均				4.29	3.19	3.94	4.25	3.75	4.33	4.81	3.00	3.50	5.00	3.75

表 7.3: D-ABDUCTOR の効果・使用感に関するユーザのコメント

効果	前年の約 1/2 程度の時間で報告書の骨組みができたように思う。
	発表の手順表を簡単に作ることができ準備時間の短縮に役立った。
使用感	紙やホワイトボードでの作業と比較すると色/線種による効果が大きい、レイアウトの変更により発想の転換が図れた。
	表現できる図に制限はあるが KJ 法ライクな絵を作るには十分。
感想	全体のイメージが即座に把握できるので思考の助けになる。
	多少の制限はあるがかなり手軽に考えをまとめることができる。
	作業内容を整理して表示してくれるので考えがまとめやすかった。
	自由度が低いことが逆に思考を制限しているのではないかと思われる。
	KJ 法を普段から行っていないと性能を十分に生かし切れないかもしれない。

総合評価

総合評価(項目(11))は、4が13件、3が2件、2が1件とほとんどが4の評価であった。特に筆者らが本システムの開発にあたり想定した作業に最も近いと思われるカテゴリ(1)の「KJ法等によるアイデア整理」の作業ではすべて4の評価を得ている。2や3を与えた作業は想定とはかなり異なった作業に用いられており、特に2を与えた「代数構造の図解化による学習」はあらかじめ定まった数学的な構造を対象とした作業であった。以上より今回実施したアンケートによる D-ABDUCTOR の総合評価は大体4と考えてよいであろう。

報告者の全体的なコメントのうち総合的な観点から発想作業への効果や使用感を述べたものを表7.3に紹介する。元の文章をそのまま残すように努めたが、冗長な部分は削除し文体は多少修正した。

ユーザの要望

報告者のコメントの約3割は機能に対する要望であり、それらは下のようにまとめることができた。今後はこのようなユーザの意見を考慮してシステムを改良していく必要がある。

- 複合グラフのクラス拡張
- 自動描画スタイルの多様化
- 図の編集コマンドの充実化
- テキスト入力・表示機能の改良
- 図のマルチウィンドウ表示のサポート

第 8 章

システムの評価

前章では、図を用いて思考過程を支援する DP である D-ABDUCTOR について述べたが、D-ABDUCTOR に限らず、思考支援システムに期待される効果としてまず第一に考えられるのは思考作業のプロダクト（つまり、アイデア）の量の増加や質の向上であろう。しかしながら、その効果の定量的な評価は、現状ではアイデアというものの測度が明確になっていないため、たいへん困難な問題である。そこで筆者らは、思考支援システムの評価という問題に対して次のような段階的アプローチを考えた。

- (1) 要素技術レベル: システムを構成する各機能ごとに、アルゴリズムの理論的な速さやプログラムの実行速度など、技術的観点から評価を行なう。
- (2) 操作レベル: 思考作業（たとえば KJ 法）に含まれる、カードをさがす、集める、並べる、といった「操作」をどう支援できるかを評価する。
- (3) 思考レベル: 思考作業に含まれる操作だけでなく「思考」をもどう支援できるかを評価する。

これまで行なわれた、D-ABDUCTOR の評価としては、要素技術レベルにおいて、中核機能である自動レイアウト機能の基礎をなすアルゴリズムの各処理フェーズにおける計算量 [MS89b] や、実現したプログラムの実際の処理時間 [SM91, MS94d] が示されている（第 4 章）。また、定量的ではないが、ユーザに各機能の使用感や満足度を問うアンケートを実施した結果、思考作業に関して総合的には 5 段階の 4 という良い評価が得られている（[MS94e]、第 7 章）。

本章では、D-ABDUCTOR の操作レベルにおける定量的な評価に焦点をあてて [MS94c, MS95b, MS95a, MS96]、筆者らは、操作レベルの時間的な観点からの定量的評価を行なうために、まず評価実験のための厳密な作業を設計した。その作業は、(1) 時や人によって作業内容が変わらない、(2) 実際的な一連の作業全体を基にしている、(3) 作業を中断なく連続的に進められる、(4) 必要な思考の量や質が可変である、といった特徴を持つ。さらに特徴 (4) によって、思考をあまり必要としない単

純な作業と思考をより多く必要とする作業との時間差を計測することで、評価法を操作レベルから思考レベルへ発展させることも期待できる。筆者らは、このように設計された作業を基に D-ABDUCTOR で使える機能をいろいろと制限した五つのモード (計算機を使わない手作業も含む) において実験を行ない作業時間を計測、比較した。

以下、8.1 節では、まず筆者らの考案した評価法について説明する。8.2 節では D-ABDUCTOR に関して行なった評価実験の方法について説明し、8.3 節でその結果を示す。8.4 節で評価に対する考察を述べる。

8.1 評価法

本章では、KJ 法支援システムに関してこれまで行なわれた評価法を検討するとともに、筆者らの開発した評価法について説明する。

8.1.1 これまでの評価法

KJ 法支援を目的とした代表的なシステムとしては、KJ エディタ [KKO92]、郡元 [MHN94]、ISOP [Cra93] などがあり、それぞれ時間的な観点から評価がなされている。小山らは、カードを抜ける、カードに文字を書き込むといった、それぞれ小さな単位の基本的な作業において KJ エディタを使う場合と使わない場合とで作業時間を比較している [KKO92]。宗森らは、大勢の学生に、郡元を使う場合と使わない場合とで実際の KJ 法を行なわせて比較をしている [MHN94]。また、KJ エディタと ISOP に関して山口らは、完成図を参照しながらそれを基に KJ 法の手順をそれぞれのシステムを使う場合と使わない場合とで比較をしている [YKH93]。

それぞれ実験法は異なるが、いずれも同一の被験者が比較すべき条件 (つまりシステムを使うか使わないか) を変えて作業する方法を基本にしている。さらに、それぞれにおいて、作業を同一条件にするために、テーマに慣れないようにテーマを変える (宗森ら)、作業を操作的な小単位にする (小山ら)、完成図を参照することで作業から考える部分を除く (山口ら)、といった工夫がなされている。しかしながら、テーマを変えるという方法は、作業ごとに難易度が異なるという問題点があり、作業単位を小さくする方法や、完成図を参照する方法にも、一連の作業全体の評価ができない、作業が中断される、といった問題点がある。

8.1.2 評価法開発の考え方

筆者らは評価実験のためにより厳密な作業を設計した。設計にあたっては次に説明する、再現性、全体性、連続性、拡張性を考慮に入れたが、その作業は上で述べた問題点を解決するものでもある。

再現性: いつ誰が行っても作業内容が同じであること。KJ法では、同じカードの集合から開始しても、人あるいは時によって異なるグループ編成や関係線付けを行なう可能性がある。作業を同一条件に揃えるためには、このようなばらつきをなくす必要がある。

全体性: 一連の思考作業の全体であること。KJ法のような実際的な一連の作業への有効性を判断するためには、思考作業を構成する作業断片による評価でなく作業の全体を対象にする必要がある。

連続性: 作業(被験者)が作業を連続的に進められること。実際的な一連の作業への有効性を判断するためには、実験のための指示を見聞きするような動作に作業が中断されることなく、作業を連続的なものにしておくことも重要である。

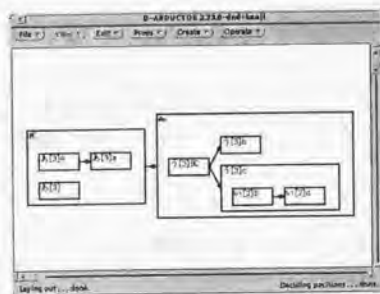
拡張性: 操作レベルから思考レベルの評価に向けて拡張できること。操作レベルの評価には深い思考を必要としない作業が適している。操作レベルの評価を基礎に置き、作業に必要な思考の量を徐々に増やした実験が行なえれば、思考レベルの評価に向けて段階的なアプローチがとれる。

8.1.3 LCR 法

LCR法 (Labeled Construction-Rule method; ラベル化作図規則法) は、評価実験に用いる作業を記述するための方法であり、その作業の目的はKJ法で用いられるような頂点間の包含関係と隣接関係が表される図を作成することである。作業の例を図8.1に示す: (a) はあらかじめ被験者に与えられる図, (b) は被験者が作成すべき図である。



(a) あらかじめ被験者に与えられる図



(b) 被験者が作成すべき図

図8.1: LCR法で記述される作業の例

LCR法では作業を記述するために、ラベル文法(頂点に記されるラベルの文法),

グループ規則 (グループの作成に関する規則)、関係線規則 (関係線の引き方に関する規則)、ラベル規則 (グループのラベル付けに関する規則)、を用いる。

ここで最も基礎となる「単純 LCR 法」を例に LCR 法を具体的に説明する。単純 LCR 法のラベル文法と三つの規則は次のように与えられる。

ラベル文法: 各ラベルは、1 行以上からなり、一つの行は平仮名 1 文字の後ろに角括弧で括られた数字とアルファベットが続いたものか、または#の後ろにアルファベットが続いたものからなる、アルファベットは省略できる。

ラベル文法を形式的に記述すると次のような生成規則で表される。出発記号は *label* であり、*CR* は改行、*KANA* は平仮名 1 文字、*ALPH* はアルファベット 1 文字、*DIGIT* は数字 1 文字、 ϵ は空文字列を表す終端記号である。

```

label ::= line | label CR line
line  ::= node-label [ member ] edge-label | # edge-label
node-label ::= KANA
edge-label ::=  $\epsilon$  | edge-label ALPH
member ::= DIGIT | member DIGIT

```

グループ規則: ラベルの 1 行目が同じ平仮名で始まる頂点を括弧内の数だけ集めて同じグループとする。ラベルが#で始まる頂点はどのグループにも含まれない。

関係線規則: ラベルの 1 行目が同じアルファベット (たとえば、*A* と *a*) を持つノード間に大文字 (*A*) を持つ頂点から小文字 (*a*) を持つ頂点の向きに関係線を引く。

ラベル規則: グループに含まれる頂点のラベルの 2 行目以降をグループのラベルとする。

図 8.1 に示した作業の例は、単純 LCR 法で記述されたものである。図 8.1(a) ではラベルの先頭行しか表示されていないが、実際は図 8.2 に示すようなラベルを持っている。

一般的に、LCR 法の特徴は下のようによまとめることができる。

- 作図のための規則を曖昧さなく与えることができる。規則が明確になっていることで、いつ誰がどのような手順をとっても完成図は一定の範囲で同じになる (再現性)。
- あらかじめ用意されるラベル付き頂点に規則を適用するだけで図を完成させることができる。これにより作業対象 (図) 以外のものを参照することなく (連続性)、一連の作業全体を通して行なうことができる (全体性)。

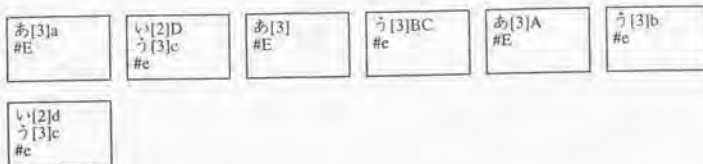


図 8.2: 頂点のラベルの例 — 図 8.1(a) の各ラベルの全体を表示したもの

- ラベル文法と規則を変化させることで、ラベルのボタンマッチングだけで行なえる作業から、深く考えなければ行なえない作業まで、多様なバリエーションを作ることができる。また考える対象を、ラベルの意味のような言語的なものだけでなく、図の構造など図形的なものへも拡張できる(拡張性)。
- 図を作成するための手順を規定しないので、手順の比較実験にも利用できる。たとえば、KJ 法に忠実な手順と、それ以外の手順との効率の比較も可能である。

8.1.4 LCR 法のバリエーション

LCR 法は様々なバリエーションを作ることができるため、評価を操作レベルから思考レベルへと進めるために有効であると考えられる。ここでは LCR 法のバリエーションの例とそれを利用することで期待できる評価について述べる。

単純 LCR 法

先に具体例としてあげた単純 LCR 法の特徴を KJ 法と対比させると、作図の過程において、グループを作る、ラベルを付ける、関係線を引く、といった KJ 法では深く考える必要があり、作業者ごとにやり方や作業時間のばらつきが生じやすい部分が、平仮名とアルファベットのボタンマッチングや文字列の複写といった、形式的で単純な作業に置き換えられている。つまり、深く考える部分がなく、作業の大部分がカードの移動や線の描画といった操作であるため、操作レベルの評価に適している。

8.2 節以降で説明する実験では、操作レベルの定量的な評価を目的とし、この単純 LCR 法を用いた。

漢字 LCR 法

単純 LCR 法ではグループを仮名で表したが、漢字 LCR 法ではグループを漢字で表す。漢字は仮名に比べて形の複雑なものが多く、1文字でも意味を持つことが特徴と言える。そのような特徴を利用して、グループの規則を修正することができる。たとえば、図 8.3 の例のように同じ部品（または部首¹）をもつ漢字をグループにするという規則にすることができる。同じ平仮名を集めるのに比べると難しくそうであるが漢字の意味まで知っている必要はない。さらに、図 8.4 の例のように共通する概念（動物、植物など）をもつ漢字をグループにするという規則も可能である。この作業は漢字の意味を考えなければ形を見ただけでは行なえない。このように漢字を用いることで、作業を単純なボタンマッチングで行なえるものから背景知識と思考を必要とする作業へと拡張できる。

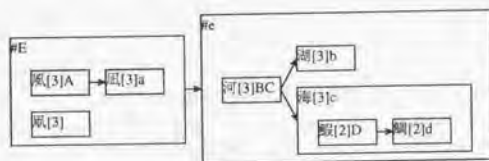


図 8.3: 漢字 LCR 法による作業の完成図の例 (1)

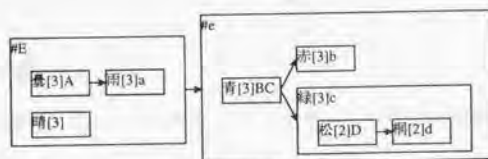


図 8.4: 漢字 LCR 法による作業の完成図の例 (2)

同類のバリエーションとして、漢字の代わりに、単語や熟語さらには文章を利用することも考えられる。その場合、日本語でなく外国語も利用可能である。

¹ 部品ではなく部首とするとより正確な知識が必要とされる。たとえば、図 8.3 において「風」と「風」は「凡の部」に含まれるのに対して、「風」は「風の部」に含まれるが、これを形だけから判断するのは不可能であろう。

構造的 LCR 法

KJ 法的な作業では、図のある部分構造を参照する、全体構造を把握するなど図の構造を考慮することも重要である。ところが、単純 LCR 法は三つの規則がすべてラベル情報だけしか利用していないため、図の構造よりもむしろラベル情報に基づくノード間の2項関係(包含関係と隣接関係)を考慮するだけで作業を進めることができる。それに対して、構造的 LCR 法は規則が図の構造をも利用する。たとえば、単純 LCR 法の関係線規則に下の規則*が追加されたものが構造的 LCR 法の一例である。

* 同形の構造を持つグループは同じ配置とする。

この場合、作業を行なう上で、図の構造に着目する、同じ構造の部分を見比べることなどが必要になる。つまり、構造的 LCR 法を用いることで図の構造を考慮する必要のある作業を対象とした評価が可能になる。

漢字 LCR 法や構造的 LCR 法で記述された作業は、単純 LCR 法に比べて、漢字の意味を考えると図の構造を考慮するといった思考を余計に必要とするはずである。単純 LCR 法での実験結果を基準にして、そのような思考がどの程度余分な時間を必要とするか、そしてその余分な時間が思考支援システムを使うか使わないかによって変化するかどうかを調べるのが思考レベルの評価につながると考えられる。

ただし、LCR 法のこのような拡張は、思考レベルの評価に少し踏み込めるとしても、思考レベル全体の評価に対応できるものではないことも指摘しておかなければならない。思考レベルの本質に対する評価は最も重要であるが大変難しい問題であり、それに迫るためにはさらに新たな評価法の研究を進めていかねばならない。

8.2 実験法

開発した評価法を用いて D-ABDUCTOR に関して行なった実験法について説明する。

8.2.1 作業

単純 LCR 法によって記述された作業を KJ 法に準じた手順で行なう。つまり、まずグループ編成 (BU) を行なう:グループを作りそれらをコラプスする(束ねて1枚のカードとして扱う)ことを繰り返す。グループをすべて作り終えたなら、次に、図解化 (TD) を行なう:関係線を引しながらグループをエキスパンドする(束ねたカードを解く)。グループのラベル付けは、D-ABDUCTOR を使って作業する場合には、自動的にラベル規則に従って行なわれる。

8.2.2 モード

D-ABDUCTOR の持つ基本的な直接操作による編集機能と自動レイアウト機能を中心とする特徴的な機能について評価するため、利用できる D-ABDUCTOR の機能を表 8.1 に示すように制限した五つのモードで作業を行なう。なお、モード 1-3a の計算機環境としては、SPARCstation IPX(メインメモリ 16Mbyte)を X ウィンドウシステムのターミナルとして利用し D-ABDUCTOR はリモートの SPARCstation 2(メインメモリ 64Mbyte)で動かした。

表 8.1: 実験で比較される五つのモード

モード 0	計算機を使用しないで紙の上でカードを利用して作業する。
モード 1	D-ABDUCTOR を利用する。自動レイアウト機能は利用できない。
モード 2	モード 1 に加えて、自動レイアウト機能が利用できる。
モード 3	モード 2 に加えて、インクリメンタル自動レイアウト機能が利用できる。
モード 3a	モード 3 に加えて、アニメーション機能が使われる。

8.2.3 図のクラス

作業対象となる図はその規模によって 8 クラス用意し、時間の都合でクラス 1, 3, 5 だけを利用した。各クラスの図の要素数(頂点数および関係線数)は表 8.2 に示す通りである。実験では被験者が図に慣れてしまうのを防ぐために、各クラスにつき五つの図を用意し作業ごとにランダムに選んで用いた。

表 8.2: 各クラスの図の要素数

	頂点	リーフ	グループ	関係線
クラス 1	21	16	5	13
クラス 3	25	18	7	13
クラス 5	32	24	8	19

註)「リーフ」は別の頂点を包含しない頂点で、LCR 法で記述された作業ではあらかじめ用意される。「グループ」は別の頂点を包含する頂点で、作業過程で作成される。

8.2.4 ユニット

実験は同一クラスの図に対して、次の8回の作業を1ユニットとして行なう。

モード 3a → 3 → 2 → 1 → 1 → 2 → 3 → 3a

つまり作業をモード 3a → 3 → 2 → 1 の向きとその逆のモード 1 → 2 → 3 → 3a の2通りの向きで交互に繰り返す。これは、作業者が習熟することで徐々に作業が速くなる(一方向だけを繰り返すと、最初のモードよりも最後のモードの方が習熟が進んだ状態でのデータが得られる)ことを想定したものである。この習熟による影響を取り除くために2通りの向きを繰り返し、作業時間を平均することにした。

8.2.5 実験プログラム

各作業者に対して、表8.3のようなプログラムで実験を行なった(ただし、プログラムのAMとPMは厳密に正午で分けてはいない)。実験は1人につき3日行ない、3日を最大8日に収めた。このプログラムに先だって、実験の目的や進め方に関する知識をそろえるために、A4サイズ7ページの実験マニュアルを被験者に配布しておいた。

表8.3: 実験のプログラム

日	時	クラス	作業
第1日	AM	—	オリエンテーション
		0, 2	作業のデモを見る
		0	前半1/2ユニット(練習)
		1	1ユニット
	PM	1	1ユニット
第2日	AM	3	1ユニット
	PM	3	モード0(手作業)
		3	1ユニット
第3日	AM	5	1ユニット
	PM	5	1ユニット

8.2.6 被験者

被験者は計5名、全員男性で、日常的に何らかのウィンドウ・システムを使って仕事をしていて、キーボードやマウスの操作には慣れている。ただし、D-ABDUCTORに関する経験度には変化を持たせてあり、開発者1名(KM:32歳;本実験の設計者でもある)、日常的にD-ABDUCTORを使っている熟練者1名(KN:27歳)、使用経験者2名(KS:48歳, TK:26歳)、使用未経験者1名(TS:27歳)という構成である。

8.2.7 作業数

以上のユニットおよびプログラムを整理すると被験者1名あたりの作業は次のようになる。

- 計算機を使わない作業: 1回
- 計算機を使った作業: 48回
- モードによる内訳: 12回×4モード
- クラスによる内訳: 16回×3クラス

8.2.8 データの収集と解析の仕方

D-ABDUCTORを利用した作業(モード1, 2, 3, 3a)ではデータの収集はすべてD-ABDUCTORの作業履歴保存機能を利用した。生成された履歴から作業時間を自動抽出し、さらに表形式に出力するいくつかのプログラムにより、8.3節章で示すデータを自動的に得た。ただし、モード0については作業の様子を記録したビデオの時間情報を利用した。

8.3 実験結果

作業開始から図の完成までに要した時間を作業時間とする。ただし、モード0の場合には条件を同一にするために筆記時間を差し引いたものを作業時間とする。また、手作業(モード0)での作業時間を1.0とした時間比を作業効率とする。

モード別の作業時間の平均を表8.4に示す。表中、c*は全クラスの平均、c1, c3, c5はそれぞれクラス1, 3, 5での平均である。c1/16, c3/18, c5/24は各クラスにおける平均作業時間をリーフ数(それぞれ16, 18, 24)で割ったもの、つまり1リーフあたりの平均作業時間である。BUおよびTDは、作業を前半(グループ編成)と後半(図解化)に分けてそれぞれに要した時間を平均したものである。AMおよびPMは1日の前半と後半に行なわれた作業を分けて作業時間を平均したものであり、FHおよびSHは、1ユニットの前半と後半に行なわれた作業を分けて作業時間を平均したものである。

表 8.4: 作業時間のモード別平均 (秒)

	モード 0	モード 1	モード 2	モード 3	モード 3a
c*	—	468.30	218.60	198.59	206.88
c1	—	428.45	187.92	170.82	184.42
c3	1327.60	447.14	199.40	180.64	181.99
c5	—	529.32	268.49	244.30	254.23
c1/16	—	26.78	11.75	10.68	11.53
c3/18	73.76	24.84	11.08	10.04	10.11
c5/24	—	22.01	11.19	10.18	10.59
c3(BU)	245.80	159.62	101.67	99.16	101.94
c3(TD)	1081.80	287.51	97.72	81.48	80.04
c*(AM)	—	486.64	223.68	200.51	214.13
c*(PM)	—	449.96	213.52	196.67	199.63
c*(FH)	—	472.81	221.30	200.40	225.59
c*(SH)	—	463.79	215.90	196.77	188.16

註) c* : クラス 1,3,5 の平均; cn : クラス n の平均;
 BU : グループ編成段階; TD : 図解化段階;
 AM : 1 日分の作業の前半; PM : 1 日分の作業の後半;
 FH : 1 ユニット前半 (3a → 1); SH : 1 ユニット後半 (1 → 3a)

8.3.1 モード間の比較

表 8.4 においてモード別の作業時間を比較する。クラス 3 において (モード 0 の作業はクラス 3 だけで行なったためクラス 3 において比較する), モード 0 の平均 1327.60 秒に対してモード 1 は平均 447.14 秒である。つまり D-ABDUCTOR を使えば、使わない場合の 0.337 (約 1/3) 倍の時間で作業ができたことになる。また全クラスにおいて、モード 1 の平均 468.30 秒に対してモード 2 は平均 218.60 秒である。つまり、自動レイアウト機能を使うと、使わない場合の 0.467 (約 1/2) 倍の時間で作業ができたことになる。

以上から、D-ABDUCTOR で自動レイアウト機能を使うと、手作業の場合の約 1/6 倍 (モード 3 では 0.136 倍) の時間で作業ができたことになる。それに対して、モード 2, 3, 3a 間には顕著な差は認められない。

8.3.2 クラス間の比較

表8.4においてクラス別の作業時間(c1, c3, c5の行)を比較すると、モード3aを例外として、どのモードにおいても規模が大きくなるほど作業時間が増加している。この作業時間をリーフ(あらかじめ与えられる頂点)数で割って1リーフあたりについて換算した時間がc1/16, c3/18, c5/24の各行に示される。1リーフあたりに要する作業時間は同一モードにおいてほぼ一定といえる。ただし、今回の実験では、クラス1, 3, 5という順で実験を進めたため習熟の影響を受けている可能性があることから厳密な結論は下せない。

表8.5: 作業者ごとの作業時間と作業効率の比較

作業時間 (秒)					
	モード 0	モード 1	モード 2	モード 3	モード 3a
KM	949.00	○219.18	○107.63	○102.27	○96.30
KN	○814.00	323.24	122.48	122.36	140.19
KS	1181.00	431.87	270.79	253.45	246.33
TK	●2143.00	●790.33	●332.51	●275.69	●290.16
TS	1551.00	471.07	163.57	149.43	136.97
最大/最小	2.63	3.61	3.09	2.70	3.01
標準偏差	478.20	192.82	87.68	70.50	73.43
相関係数	—	0.949	0.784	0.709	0.698

作業効率					
	モード 0	モード 1	モード 2	モード 3	モード 3a
KM	1.0000	○0.2310	0.1134	0.1078	0.1015
KN	1.0000	●0.3971	0.1505	0.1503	0.1722
KS	1.0000	0.3657	●0.2293	●0.2146	●0.2086
TK	1.0000	0.3688	0.1552	0.1286	0.1354
TS	1.0000	0.3037	○0.1055	○0.0963	○0.0883

○最小(最速), ●最大(最遅)

8.3.3 作業者間の比較

表8.5はモード別に、作業者(被験者)ごとの平均作業時間と作業効率、作業時間の最大最小の比、各モードにおける作業時間の標準偏差およびモード0の作業時間との相関係数を示している。表中の白丸(o)は各モードにおける最小値を、黒丸(●)は最大値を表している。

作業者によって作業時間に関差がある。その幅(最大/最小)は2.63倍から3.61倍で、平均3.01倍であり、そのばらつき方を標準偏差で見ると、モード0, 1, 2, 3の順で減少している。ただし、モード3aはモード3より少し大きい。

作業は開発者(KM)と熟練者(KN)が圧倒的に速い。しかし、作業効率で比較すると、作業時間とは最高、最低の作業者が同じではない。特にモード2, 3, 3aでは未経験者であるTSが最高効率であるのが特徴的である。D-ABDUCTORの効果は初心者でもすぐに得られることを示唆していると言えよう。

手作業(モード0)の速さと、D-ABDUCTORを利用した際の速さにはモード0との相関係数が示すように多少なりとも相関がある。ただし、モード1, 2, 3, 3aと利用できる機能が増えるにつれて相関係数は減少している。つまり機能が増えるほど手作業とは違ったスタイルの作業になっていると考えられる。

表8.4(c3の行)ではモード0, 1, 2, 3と順に速くなっているものの、モード3aだけはモード3より遅くなっている。これについて作業者ごとに細かく見ると、モード3よりモード3aが遅いのは5名中2名(KNとTK)だけで、その2名がモード3aの平均作業時間を長くしている。KNは熟練者であることから、モード3aで使われるアニメーション機能の効果には慣れよりも相性のような別の要因が影響しているのかも知れない。

表8.6: 習熟の程度

	モード1	モード2	モード3	モード3a	平均	標準偏差
PM/AM	0.9246	0.9546	0.9808	0.9323	0.9481	0.02187
SH/FH	0.9809	0.9756	0.9819	0.8341	0.9431	0.06299

8.3.4 ユニット設計の効果

第8.2.4節で述べたように、作業者の習熟による影響を取り除けるように1ユニット内の作業の順序を決めた。実際に習熟の影響を取り除けたかどうかを、1日の前半(AM)と後半(PM)に分けたデータと、1ユニットの前半(FH)と後半(SH)に分けたデータをもとに検証する。

表 8.7: D-ABDUCTOR と他のシステムとの比較

	TOTAL	BU	TD	被験者数	作業回数
I SuperPaint	0.633	0.722	0.608	1	3
II D-ABDUCTOR (モード 1)	0.337	0.649	0.266	5	20
III D-ABDUCTOR (モード 2)	0.150	0.414	0.0903	5	20
D-ABDUCTOR (モード 3)	0.136	0.403	0.0753	5	20
D-ABDUCTOR (モード 3a)	0.137	0.415	0.0740	5	20
II ISOP	0.392	0.293	0.433	1	1
KJ エディタ	0.627	0.920	0.560	1	1
郡元	—	0.927	—	約 60	21

表 8.6は、表 8.4中の AM と PM の比および FH と SH の比をモード別に示したものである。AM の平均と PM の平均の比は、平均が 0.948、標準偏差が 0.02187 である。同様に FH と SH の比は、平均が 0.943、標準偏差が 0.06299 である。平均がほぼ同じなのに対して標準偏差は大きく異なっている。

つまり 1 日の前半と後半とに分けると習熟の仕方にモード間の差はあまりないが、1 ユニットの前半と後半とに分けるとモード間の差が大きい。よって、1 ユニット内での習熟による影響を軽減できたと言えよう。

8.3.5 他のシステムとの比較

表 8.7は、高機能図エディタである D-ABDUCTOR と他のシステムの比較のために、それぞれのシステムにおける作業効率を示したものである。SuperPaint [Sil91] は、MacDraw [Cla91] に代表される実用的な汎用作図システムの一つとして比較の対象とした。また他の代表的な KJ 法支援システムの作業効率も併せて示した。

表 8.7において、ローマ数字はシステムの大まかな分類を表している。I は図の論理構造を管理していないもの、II は論理構造は管理しているが図の自動描画は行なえないもの、III は論理構造を管理していて自動描画も行なえるものである。D-ABDUCTOR をモード 1 で利用する場合と、モード 2, 3, 3a で利用する場合とをここでは別のシステムであるとして分類した。TOTAL は KJ 法のあるいは KJ 法的な作業全体の作業効率で、BU は作業前半のグループ編成段階だけの作業効率、TD は作業後半の図解化段階だけの作業効率である。被験者数は各システムの作業効率を求めるための実験作業を行なった作業者の数で、作業回数はそのときに行なわれ

た実験作業ののべ回数である。ただし作業回数には基準となる手作業の回数は含まない。

D-ABDUCTORの作業効率(表8.4のクラス3での作業時間を基にした) SuperPaintの作業効率はD-ABDUCTORの被験者の内1名KMが、D-ABDUCTORの場合と同様に単純LCR法で記述された作業を行なって得たもので、図の規模はクラス3で統一した。

表より、D-ABDUCTORの方がどのモードにおいてもSuperPaintより効率が良いことが分かる。たとえば、作業全体の効率(TOTAL)と比較すると、モード1では1.88倍、モード2では4.22倍、D-ABDUCTORの方が効率がよい。

SuperPaintとD-ABDUCTOR(モード1)との大きな違いは図の論理構造を管理しているかどうか、さらにD-ABDUCTOR(モード2)との違いは自動描画ができるかどうかである。表8.8はその違いがKJ法的な作図作業のやり方や戦略にどう影響したかを整理したものである。D-ABDUCTOR(モード1)の特徴は、論理構造を管理していることでCreate-Group(グループ・ノード作成)やCollapse(コラプス)、Expand(エキスパンド)のような図の構造に関わる操作が利用できること、また関係線がノードの移動の伴って自動的に修正されることである。またD-ABDUCTOR(モード2)の特徴はモード1に加えて図の自動描画ができることでありLayout操作を利用できることである。これらの機能あるいは操作を使うことで作業工数の削減が可能になり効率の向上に結び付いたと考えられる。

ISOP、KJエディタ、郡元のデータは文献[YKH93,MHN94]に発表された評価実験の結果に基づいている。ISOPとKJエディタの評価実験は完成図を参照しながらKJ法的手順で行なう図の作成作業によって、郡元の評価実験は実際の様々なテーマでのKJ法の作業によって行なわれたものである。本実験では文字入力(筆記、タイプ)時間を考慮に入れないため、発表されたデータからもそれらは差し引いた。ただし、郡元のデータについては文字入力時間が不明のため含まれたままである。また、手作業と郡元で作成された図の規模が異なっている(手作業の場合リーフ数平均63.8、グループ数平均10.3;郡元の場合リーフ数平均36.4、グループ数平均7.0)、ISOPとKJエディタに関しては実験に用いた図の規模が不明である。

ISOP、KJエディタ、郡元に関しては、それぞれ実験法や図の規模などが異なるため厳密な比較はできないが、表8.7からD-ABDUCTORでの作業効率がかなり良いと推測される。特に図解化段階(TD)で自動レイアウト機能を利用した場合(モード2, 3, 3a)の効率が良く、グループ編成段階(BU)ではD-ABDUCTORよりもISOPの効率の方が良いが、ISOPでのグループ編成作業は他と違って図的な手法ではない。

8.3.6 被験者の感想

表8.9は作業後のインタビューによって集めた被験者のコメントの一部を整理したものである。モード別に肯定的コメントと否定的コメントとに分けてある。モー

表 8.8: SuperPaint と D-ABDUCTOR(モード 1, 2) での作業内容の比較

作業			
	I SuperPaint	II D-ABDUCTOR (1)	III D-ABDUCTOR (2)
BU グループ作成	グループにするノードをマウスでドラッグして集め、その側にテンプレート・ノード(あらかじめ用意しておく)をコピーしてドラッグ	グループにするノードを(およその配置を考慮しつつ)マウスでドラッグして集め、それらをマウスでセレクトして Create-Group を指示	グループにするノードをマウスでセレクトして Create-Group を指示
コラプス	Align Objects (オブジェクトの整列)を指示したあと、Group (オブジェクトのグループ化)を指示	グループ・ノードに Collapse を指示	グループ・ノードに Collapse を指示
TD エキスパン	Ungroup (オブジェクトのグループ解除)を指示したあと、(およその位置を決めるために)ドラッグし、またグループ・ノードのリサイズを行なう	グループ・ノードに Expand を指示したあと、ノードをドラッグし、またグループ・ノードのリサイズを行なう	グループ・ノードに Expand を指示したあと、Layout を指示
関係線作成	Line ツールを選んでマウスで線を描く(ただし、修正が面倒なのでなるべくノードの配置が決定してから行なう)	マウスでドラッグして関係線を作成(線の交差などの考慮に有利なので早い時期に作成する)	マウスでドラッグして関係線を作成
配置決定	マウスでノードをドラッグしながら、グループ・ノードをリサイズし、併せて関係線の修正を行なう	マウスでノードをドラッグしながら、グループ・ノードをリサイズする	Layout を指示、必要なら修正

表 8.9: 被験者のコメント

モード	肯定的コメント	否定的コメント
0	体感時間はあまり長く感じない。作業としては楽しかった。	間違いやすく、修正が不可能／困難。ノードの位置決めが困難。紐によるグループの境界は不便で分かりにくい。
1	モード0と比べてグループ作成、展開が容易。再利用も便利であろう。間違いが少ない。修正が可能／容易で安心感がある。図の構造を計算機が知っているため、普通の作図ツールよりも楽。	展開後の配置を考えてグループをつくる必要がある。1回やったらたくさん。モード2を経験するとつらい。
2	一番気に入った。最も自動レイアウト機能のありがたみを感じた。モード1とは劇的に違う。モード1が終わってモード2に移るとき、安堵感を覚える。細かいことを考えなくてよいので、思考に集中できる。精神的余裕ができたので、ノードをアルファベット順に並べ替えた。	
3	急いでいるとき、ノードが多いときに適している。エキスパンド後のレイアウトが自動化されていることがありがたい。	関係線作成で勝手にレイアウトされるのでメンタル・マップが壊されるため不利だ。同じ作業はまとめてやりたいのに、操作ごとにレイアウトが変わるのがいや。
3a	モード3やモード2よりよい(特に時間的余裕がある時)。画面がたびたび変わるがイライラは緩和され、楽しいと感じる。モード3の欠点が補えている。関係線作成のローカルな方針を崩さずに作業できる。	

ド2と3aに関しては特に否定的コメントはなかった。被験者のコメントは本実験の目的である定量的な評価には直接関係ないが、システムの改良には有益な興味深いものが多かった。

8.4 考察

本稿の主目的は、D-ABDUCTOR という一つの思考支援システムに関して操作レベルにおいてできるだけ客観的な評価を行なうことであり、その結果は前章で述べた。ここではさらに、(1)LCR法は操作レベルの評価法として有効であったか、(2)評価実験に基づく知見から同種の思考支援システム一般に対してどのような示唆が得られるか、という2点に関して考察を加える。

8.4.1 評価法としての LCR 法の有効性

本稿で目的とした操作レベルの評価とは『思考作業 (たとえば KJ 法) に含まれる、カードをさがす、集める、並べる、といった「操作」をどう支援できるかを評価する』ことであった。その意図する評価が正しく行なえるためには、実験に用いる作業に評価の対象がすべて含まれていること、および評価の対象でないものの影響を受けないことが必要である。LCR法で記述された作業は連続して行なえる一連の作業において、KJ法に含まれる上記の操作をすべて含んでいる。さらに単純 LCR法では、KJ法では深く考える必要がある部分を、形式的で単純な作業に置き換えることで思考対象であるテーマへの慣れやテーマごとの難易度の違いによる影響を取り除いている。それに加えて、実験においては作業の繰り返しによる習熟の影響もユニットを適切に設計したことで軽減することができ、結果として操作レベルにおける評価が目的通りに行なえたと考ええる。

8.4.2 同種システム一般に対する示唆

先に述べた評価実験の結果や経験を踏まえて、D-ABDUCTOR および同種の思考支援システム一般に対して、次のような示唆を得た。

- 計算機を利用すると手作業に比べて顕著な作業時間の短縮が見られる、また自動レイアウトなどの機能が増えるとさらに作業時間の短縮が見られた。これらから思考作業の計算機による支援が有効であり、機能を充実させることで作業効率を向上させることができると言える。
- 利用できる機能が多くなるほど、作業者の作業時間のばらつきが減少し、手作業における作業時間との相関も減少した。このことから経験の程度に関係なく計算機による支援が有効であると言える。

- アニメーション機能の効果は時間的には顕著でなかったが「楽しい」という被験者のコメントがあったことは重要である。思考支援では時間では計れないこのような精神的な効果も重視し活用すべきである。

本稿で示した実験は一システムに限定して行なったものであり、その結果だけに基づくこのような一般的主張は信頼性という点では不十分である。しかしながら、今後このような示唆が蓄積されることで同種のシステムを開発する際の指針として有益な情報となると考える。

第 9 章

結論

本章では結論として、本研究の工学的な貢献について整理する。また、研究の今後の展望ならびに課題についても述べる。

9.1 本研究の貢献

本研究の貢献は、強力な視覚表現形式の図の自動描画技術を開発し、それを基に自動描画機能を備えた DP を実現することで、情報処理の上流工程で有効と考えられる図の編集、作成のプロセスを改善した点にある。技術的な観点からは、図の自動描画技術として領域連結複合系の図の配置技術、装飾技術、表示技術など基礎技術を開発し、DP として KJ 法を参考にした図的発想支援システム D-ABDUCTOR を実現した。それぞれの概要は次の通りである。

9.1.1 図の配置技術

領域連結複合系の図の論理情報を、有向グラフの拡張であり、辺が「隣接」と「包含」の 2 種類の型を持つ「複合グラフ」によって定式化した。また、図の美しさ見やすさの基準として、描画規約 C1: 頂点を長方形の領域として表現、C2: 包含辺を領域の包含関係で表現、C3: 隣接辺を直線または折れ線で表現、C4: 階層表現 — 頂点を平行な帯上に配置し、その際すべての辺の向きを揃える、および描画規則 R1: 隣接辺の交差数の最小化、R2: 隣接辺と頂点の交差数の最小化、R3: 長い隣接辺の直線化、R4: 隣接頂点の近接化、R5: 隣接辺の対称配置とその規則間の優先順位 ($R1 > R2 > R3 > R4 > R5$) を定めた。そして、その基準を満足する自動描画アルゴリズムを開発した。そのアルゴリズムは有向グラフの自動描画アルゴリズムとして有名な Sugiyama アルゴリズムを基礎とするもので、複合階層化、正規化、順序決定、座標決定の四つのフェーズからなる。隣接辺の交差数最小化などの描画規則を満足する最適解を求めることは NP 完全であることが知られているが、開発したアルゴリズムでは発見的解法を用いることで高速化を達成している。

ここで開発した複合グラフの自動配置技術により、DPは領域連結複合系の図を処理対象とすることが可能となった。そして本研究において実現した図的発想支援システムD-ABDUCTORでは、この自動配置技術をKJ法支援のための自動描画機能として利用することで、KJ法の作業過程の効率を大幅に向上させることに成功した。この配置技術はKJ法で使われる図に特化しているわけではないので、それ以外にも、呼びだし関係とモジュールのネスト構造を同時に表現するプログラム構造の可視化や、テキストとアンカーの包含関係とアンカーからテキストへの隣接関係を持つハイパーテキストのマップの自動作成などにも応用可能である。さらにはアルゴリズムの高速性によってそれらの図を印刷物のように静的なものとして利用するだけでなく画面上で動的に処理することも可能である。

9.1.2 図の装飾技術

図の「見方」に関するユーザの要求(分類、適合、可視化、強調、詳細化、概略化など)を視覚属性(論理的視覚属性:可視性;質的視覚属性:きめ、模様、色、形;量的視覚属性:サイズ、濃淡)に関して整理し、視覚属性の決定のために使われる情報(構造属性、意味属性、視点属性)およびその決定の仕方をも「重要度」と呼ばれる数値によって表現した。そして、領域連結複合系の図の装飾技術の一つとして、重要度を利用して視覚属性を図の応用情報、論理情報、およびユーザの視点から自動的に決定する方式「図ドレッシング」を開発した。図ドレッシングは各幾何的オブジェクトの重要度を計算する「重要度関数」と、重要度を基に視覚属性を決定する「自動図ドレッシング」から構成される。

図ドレッシングを実装したD-ABDUCTORでは基本関数の重み係数の変更のためにスライド型のユーザ・インタフェースを用意している。ユーザはスライドを操作するだけで重要度関数を変更することができ、新しい重要度関数に基づく図を直ちにみることができる。視覚属性の決定のために使われる「重要度」は図の「見方」の直接的な表現にはなっていないため、ユーザが自分の望む「見方」を重要度あるいは重要度関数で形式的に表現することは直感的とは言えないが、親しみやすいユーザ・インタフェースを提供することで重要度関数の変更に対する試行錯誤を容易にしている。

9.1.3 図の表示技術

図の表示技術に関して、ユーザの立場から表示方式への五つの要求(詳細性、全体性、同時性、表示像の単一性、写像の適性)を整理した。そして、図の重要な性質の一つである一貫性を活かすために、限られた領域内で全体的な情報と詳細な情報をいかに組み合わせるかに表示するかという問題「全体視・詳細視問題」を明確にした。その全体視・詳細視問題を表示方式に対する五つの要求に基づいて解決するために、非線形な写像を用いて、ユーザの視点付近を拡大しそれ以外の部分を縮小しながら

も図の全体を一つの画面内に表示する方式「多視点遠近画法」を開発した。多視点遠近画法の具体例として魚眼表示方式、直交魚眼表示方式、二様表示方式の3種類の写像を与え、それらの性質を、相似性、正距性、正積性、正方位性、直交性、直線性、同相性、および拡大率について整理した。その結果、二様表示方式が図の配置技術として開発した自動描画アルゴリズムによって描かれる複合グラフに適した表示方式であることを示した。

9.1.4 システム統合化

領域連結複合系の図に対する基礎技術を、さらに論理情報や幾何情報の編集機能やそれらを包括するユーザ・インタフェース機能などと組み合わせて、一つの視覚的対話型システム「D-ABDUCTOR」として統合した。その際、応用領域として、発想法として広く知られているKJ法を「図を描き替えながら思考を展開する過程」としてとらえ参考にした。本システムは次のような特徴をもつ。

- (1) 図を用いた思考過程の対話型支援システムである。
- (2) 図の自動描画技術を利用した高機能図インタフェースに最も重点を置いている。
- (3) 直接操作環境とアニメーション環境によりユーザの手間と認知的負担を軽減している。
- (4) 通信機能やマルチメディア機能などにより新しい可能性を目指している。

本研究においては応用領域として思考支援を想定した対話型システムD-ABDUCTORを開発したが、他にも様々な応用分野が考えられる。たとえば、先に配置技術の応用としてハイパーテキストのマップ自動作成をあげたが、これを対話型システムとして応用するとWorld-Wide Webのブラウザとして探索履歴マップや自分独自の認知マップを作成する機能を持つものが考えられる。探索履歴マップの作成に関しては大規模ハイパーテキストにおける迷子問題を解決するためのシステムとしてD-ABDUCTORを利用したものがすでに試作されている[Nak95]。

9.1.5 システムの評価

本研究においては技術の対象が人間であるため、客観的には人間に与えた効果を評価する必要がある。しかしながら、そのような評価はこれまで不十分であったため、方法も手探りの状態である。そこで、実現したシステムの評価としては、アンケートと実験という二つの異なる手法を試みた。

まず10人のユーザに実際の作業に対して自由に使用してもらい、システムおよびシステムが提供する各機能に対するユーザの満足度について調査した。アンケートの結果、筆者らがシステムの応用として想定した思考作業に関しては総合的に5段階の4という評価を得た。そして、試用したユーザへのアンケートの結果、複合グラフのクラス拡張、自動描画スタイルの多様化、図の編集コマンドの充実化、テ

キスト入力・表示機能の改良、図のマルチウィンドウ表示のサポート、などの機能の追加が望まれた。

また、システムの定量的な評価を得るための実験も行なった。KJ法のような思考作業に含まれる「操作」をどう支援できるかという観点から思考支援システムを定量的に評価することを目指して、まず評価実験のための厳密な作業を設計した。その作業は、(1) 時や人によって作業内容が変わらない、(2) 実際の一連の作業全体を基にしている、(3) 作業を中断なく連続的に進められる、(4) 必要な思考の量や質が可変である、といった特徴を持つ。その作業を基に D-ABDUCTOR で使える機能をいろいろと制限した五つのモード (計算機を使わない手作業も含む) において実験を行なった。被験者5名、のべ245回の作業時間の分析によって次のような結果が得られた: (i) 手作業に比べると、D-ABDUCTOR の基本的な編集機能だけを使用しても約1/3の時間で、さらに自動レイアウト機能を使うと、約1/6の時間で作業が行えた。(ii) 使える機能が多いほど作業者ごとの作業時間のばらつきが減少し、手作業における作業時間との相関も減少する傾向にあった。(iii) D-ABDUCTOR と汎用の作図システム SuperPaint を作業効率で比較すると、SuperPaint の0.633に対して D-ABDUCTOR(モード3) は0.136であった (数が小さいほど高効率)。

以上の評価から D-ABDUCTOR および同種の思考支援システムに対して、次のような示唆を得ることができた。

- D-ABDUCTOR は KJ 法支援 (発想支援) という応用分野を想定して設計開発したにも関わらず、他分野にも広く利用された。つまり、DP の応用分野の潜在的な広がりが見られる。
- 計算機を利用すると手作業に比べて顕著な作業時間の短縮が見られる、また自動レイアウトなどの機能が増えるとさらに作業時間の短縮が見られた。これらから思考作業の計算機による支援が有効であり、機能を充実させることで作業効率を向上させることができると言える。
- 利用できる機能が多くなるほど、経験度の異なる作業者の作業時間のばらつきが減少し、手作業における作業時間との相関も減少した。このことから経験の程度に関係なく計算機による支援が有効であると言える。
- 自動レイアウト機能やアニメーション機能に対して「安堵感を覚える」、「思考に集中できる」、「精神的余裕ができた」、「楽しい」という被験者のコメントがあった。図の視覚化の過程において時間的な効率の向上だけでなく、精神的な負担をも軽減しているととれる点が重要である。時間では計れないこのような精神的な効果も思考支援では重視し活用すべきである。

9.2 今後の課題

本研究は、基礎技術の開発からシステムへの統合そして評価というように一貫して進められてきたが、研究として完結したわけではなく、また残された技術的な課題もある。本節では、本研究を基にした、描画技術の発展、システムの発展、評価法の発展という三つの観点から、今後の展望やそこでの研究課題について述べる。

9.2.1 描画技術における課題

描画技術における今後の課題は、描画対象のクラスの拡張や規模の拡大、そのための描画手法の発展、あるいは「図」という概念そのものの拡張などが考えられる。

対象クラスの拡張

開発した複合グラフの自動描画アルゴリズムにより対象となる図を連結系から領域連結複合系へと広げることができたが、それでも領域連結複合系の全体ではなく扱える図はその一部である。今後は描画対象の拡張として、複合グラフに課せられた制約の排除や、有向・無向隣接辺の混在する複合グラフの取り扱いなどを行っていく必要がある。

すでに、複合グラフの自動描画法に続いて、cigraph [Lai93], nested graph [Noi93b, Noi93a], ranked graph [Nor], clustered graph [FEC95a, FEC95b] など強力な視覚表現形式のグラフ描画法に関する研究がいくつか現れてきている。また、有向辺と無向辺が混在したグラフ (混在グラフ) の自動描画アルゴリズムなども開発されている [MS94b, SM95b, SM95a]。関係情報の増加や複雑化に伴いこのような図の自動描画技術は今後ますます重要になると思われる。

対象規模の拡大

関係情報は今後、増加と共にその規模をどんどんと拡大してくるものと思われる。したがって、描画技術における対象規模の拡大は重要な課題である。配置技術としてのグラフの自動描画法の多くは対象の規模に制限を設けているわけではないが、実用上は処理速度の問題と共に表示上の問題が避けられない。そのためには、図ドレッシングや多視点遠近画法のような装飾技術や表示技術を発展させる必要がある。

表示技術における写像の改良や実装上の工夫、あるいは機能のハードウェア化などで高速化を図ると共に、装飾技術と表示技術を視点情報などユーザの要求を表すような情報を共有するように組み合わせることで、より大規模な図も小さい領域でユーザの作業にとって効果的に表示できるようにすることが重要である。また、装飾や表示の効果をより効果的なものにするためには、それらの機能が提供するパラメータをユーザに調整させるのではなく、過去のユーザの振舞いから計算機が学習あるいは自動的に適切な値へと調整する仕組みが望まれる。

描画手法の発展

描画手法に関しては、対象のクラスや規模の拡張あるいは処理速度の向上なども重要であるが、多様な美的基準への対応も重要な課題である。

本研究において開発した複合グラフの自動描画アルゴリズムは描画の際の美的基準としては表4.1に示した描画規約、描画規則、そして規則間の優先関係に従うものである。しかしながら、それらは美的基準としてこれまでに利用されている様々な基準 [Sug88b] の内のごく一部である。さらに深刻な問題は、それらの基準がアルゴリズムに組み込まれているということで、基準の変更はほぼ不可能である。つまり、描画規則を置き換えたり優先順位を変えたりするためにはアルゴリズム自身を変えなければならない。

描画機能をより柔軟なものにするためには美的基準の制御は重要な課題であり、新たな手法が求められる。その一端としてスプリング・アルゴリズムに磁場を導入したマグネティック・スプリング・アルゴリズムが提案されている [MS94b, SM95b, SM95a]。この手法では、磁場の強さをパラメータで連続的に変化させることができるため、有向辺に関する基準を動的に制御可能である。その他にも、シミュレーティッド・アニーリングやジェネティック・アルゴリズムなどがますます利用されると思われる。

「図」の概念の拡張

メディアとしての図の拡張として、今後、領域連結複合系に限らず様々な複合系の図が利用できることが期待されよう。しかしながら、それだけではなく、計算機の利用は「図」の概念をも拡張していくものと思われる。たとえば、動きを伴う「動的な図」、2次元平面に制限されない「多次元の図」など、紙の上では不可能な拡張ができる。そのような図の利用を可能にするためには、まず動的な（あるいは時間概念を含む）図や多次元の図の視覚表現形式や美的基準というものを明確にする必要がある。

9.2.2 システムにおける課題

システムにおける今後の課題は、支援レベルの向上、思考環境として発展、また汎用プラットフォーム化などが考えられる。

支援レベルの向上

秘書レベルにおいて文房具の機能が充実してくると、つづいて会議室や黒板のような機能を充実させる方向が期待される。すなわち、DPをグループウェア的な環境として発展させる方向である。たとえば、異なる図の結合や差分の抽出のような図の演算的な操作体系が確立されれば、別々の人が別々の時に作成した図をマージする、それらの違いを見えるというような機能が実現できる。図の共通部分や差分を可視化できれば、協調作業に携わるメンバーがそれぞれのアイデアの間の共通点や

相違点を客観的かつ明確にでき、図を合意形成のような協調的作業のためのメディアとして利用できるようになると期待される [OK95]。そのためには、グループでの図の参照や編集が可能となるような機能や操作体系の開発が必要となるであろう。

第7章で説明した図的発想支援システム D-ABDUCTOR では開発に際して KJ 法を参考にし、KJ 法の作業を計算機上で可能にするための機能も用意された。それらに加えて KJ 法を行なうためのガイド機能などを整備することで、その支援レベルを枠組パラダイムレベルへと向上させることも可能である。KJ 法以外にも経験的な発想法や創造技法は数多く存在し、その中には KJ 法のように図的な表現を利用するものも少なからず存在する [Tak84, Sug91]。そのような技法における図の利用の仕方を分析することは、図の表現形式を新たに定めたり、図の操作やユーザ・インタフェースを設計する際に役立つと思われる。それにより、DP を枠組パラダイムレベルへと発展させることが可能になる。さらに、そのような技法における人間の作業を計算機で代行させることで、生成レベルへの可能性も拓けるであろう。

環境としての発展

DP を「図を扱うための環境」として捉えたと、環境としての発展という方向が期待される。そのうちの一つの重要な方向はユーザ・インタフェースの拡張である。本研究において構築した DP はパーソナル・コンピュータやワークステーションのスクリーンを表示のためのインタフェースとするものであったが、それ自身を拡張し、たとえば壁一面がスクリーンであるような「思考ルーム」というような大きい環境へと広がっていく可能性もあろう。第6章において説明した多視点遠近画法は元々計算機の画面の小ささあるいは解像度の低さを補う技術として開発した。しかし、思考ルームというような大きい環境を想定した場合には、逆に大きな画面に図を効果的に表示するための技術というようなものの開発が必要になろう。たとえば、バーチャル・リアリティのような技術の導入も必要と思われる。

汎用プラットフォーム化

本研究は、情報処理の上流工程で有効と考えられる図の編集・作成を支援する機能を提供することで情報処理の最上流に位置するネックを取り除こうとするものである。そこで開発した技術やシステムは、創造的アイデア生成、知識獲得、文書作成、ソフトウェア上流プロセスなど、上流において人間に係わる情報処理の様々な分野で期待されかつ役立つと考えられる。

しかしながら、情報処理全体の支援という観点からは上流のある区間だけを支援すれば十分ということはなく、下流へ続く一貫した支援が求められる [Sug93b]。つまり、たとえば研究のためのアイデア生成を支援するのなら最終的な論文作成や論文管理まで支援すべきであり、ソフトウェア開発なら要求獲得からコード作成や保守まで一貫して支援すべきである。そのためには、本研究で開発した DP にそれぞれの応用向けの機能を整備する必要がある。

ただし、第7章で示したアンケートによると、ユーザはD-ABDUCTORをKJ法的な作業に限定して利用することはなく、むしろ図に関連する作業に広く自由に利用していたと言える。つまり、D-ABDUCTORのような図を扱う対話型システムが様々な用途に対して求められていると考えられる。そして、ユーザが求める様々な応用に対してあらゆる機能を用意することは不可能である。

そこで、そのようなシステム開発を効率化するためには、DPを図を扱う一般的な機能を完備した汎用なプラットフォームとして発展させることが有効と思われる[MNS+96, ML96]。汎用グラフ描画システムであるEDGE [PT90]、GraphEd [Him95]、daVinci [FW95]などのようにアプリケーション・プログラムへのインタフェースを整備強化することで、様々な応用に対して高機能なプラットフォームを提供できるようになる。現在、筆者らはD-ABDUCTORの汎用プラットフォーム化をすでに開始している。これに関しては付録Aで簡単に紹介する。

9.2.3 評価における課題

評価における今後の課題は、評価法や実験法の開発および改良と、思考過程の科学的解明へ向けての思考実験が考えられる。

評価法や実験法の開発および改良

本研究で行なった評価実験では、D-ABDUCTORの操作性を定量的に評価することを目指した。そして、そのために、実験で用いた作業も、できるだけ思考(特に深い思考)を含まないように単純LCR法を利用した。今後の課題としては、まず思考レベルの評価があげられる。そこでは、D-ABDUCTORのようなDPが思考の効率にどう影響するかという点を評価しなければならない。この問題に対しては、漢字LCR法や構造的LCR法を利用して実験することを考えている。単純LCR法に比べて作業時間が思考に要する分だけ増えると予想されるが実際はどうか、増えるとしたらその増分はモードつまり利用できる機能に依存するか、といった観点からの実験が思考レベルの評価へつなげると考えている。

また、評価実験の被験者から、「安堵感を覚える」、「思考に集中できる」、「精神的余裕ができた」、「楽しい」といった精神的な影響を示唆するコメントが与えられた。このような精神的な影響は時間的な効率にも関連はあろうが、むしろ思考の「質」に大きく関わってくるものと思われる。実際、自動レイアウト機能が利用できるモード2に対して「精神的余裕ができたので、ノードをアルファベット順に並べ替えた。」というコメントがあった。これは、作業効率の改善という思考にとっては間接的な支援が、精神的に好影響を与えることで生産物の質を向上させる可能性を示唆している。このような精神的な効果およびそれによる知的生産物の質の計量は大変難しい問題であるが重要な課題である。

思考過程の科学的解明

人間の思考過程の科学的な解明を目標とする研究領域では、認知科学や人工知能からのアプローチとして図を活用する「図による推論」への関心が高まってきている [NA94a]。そこでの問題意識は、「図の持つ意味は何か、図のもたらす情報は何か、図を用いた推論とは一体どのような情報処理なのか、思考において図の果たす役割は何か、図による推論は記号的・言語的な推論とどのような相互作用をするのか」[NA94b] といったものである。このような研究が進むことで、経験的に認識されている思考における図の有効性が科学的に明らかになるかも知れない。

そのような研究のためには思考過程の詳細な分析が重要と考えられる。たとえば、KJ 法を対象にした場合、伝統的に紙のカードを用いた手法では、作業履歴の記録はビデオに撮る程度のことしかできない。しかし、計算機 (D-ABDUCTOR) 上で KJ 法を行えば、作業履歴を細かい操作単位で記録することが可能になる。つまり、思考過程の科学的解明を目指す研究を進めるための実験環境としても、本研究で構築したような DP は重要であると考えられる。

謝 辞

本論文は、筆者が富士通株式会社(後に、株式会社富士通研究所へ編入)において1986年4月以来行なって来た研究成果をまとめたものです。

それらの研究成果をまとめるにあたりましては東京大学大学院工学系研究科航空宇宙工学専攻助教授の堀浩一先生に終始御指導をいただきました。同時に早稲田大学理工学部情報学科教授の大須賀節雄先生には様々な貴重な御助言を受け賜いました。また、本論文をまとめるにあたり、東京大学工学部機械情報工学科教授の井上博允先生、東京大学工学部電気工学科教授の田中英彦先生、東京大学工学部計数工学科教授の武市正人先生、東京大学工学部機械情報工学科助教授の寺田実先生には、貴重な御助言と有益なご示唆を頂きました。また、東京理科大学理工学部情報科学科教授の井上謙蔵先生には、様々な御相談にのっていただくと共に励ましの言葉を頂きました。

豪 Newcastle 大学 Peter Eades 教授にはグラフ自動描画全般に関して、また豪 Edith Cowan 大学 Wei Lai 博士にはメンタル・マップの研究に関してご支援頂きました。英 Edinburgh 大学の Robert Inder 博士にはプラットフォーム化に関してご支援頂きました。

株式会社富士通研究所の戸田光彦情報社会科学研究所長(現、HPC 研究センター長)、中川徹主席部長、竹島卓部長には諸般の研究活動において日頃より大変お世話になりました。神田陽治主任研究員、渡部勇研究員、平岩真一研究員、小柴健史研究員、新田清研究員にはシステムの開発およびその応用に関しご支援頂きました。また、同研究所書記の高木郁子さん、大井のり子さん、野中綾子さんには研究支援業務の様々な面においてお世話になりました。

システムの評価のためのアンケートや実験には、上記の方以外にも、筑波大学の中野勝次郎さん、鈴木孝幸さん、富士通株式会社の姫野望さん、上坂博亨さん、大河原弘美さん、株式会社富士通研究所の浜田健生さん、山中英樹さん、下山武司さん、神戸隆行さんに参加して頂きました。

最後に、1986年の富士通株式会社入社以来現在に至るまで、上司でありまた共同研究者でもある株式会社富士通研究所主席研究員の杉山公造博士には研究の進め方から論文発表の仕方まで懇切丁寧に御指導頂き、本論文をまとめるにあたって最終ご支援を頂きました。

以上の方々に心より感謝し、厚くお礼申し上げます。

付 録 A

汎用プラットフォーム化

図は情報の構造や関係を全体的かつ直截的に表現できるため、ユーザ・インタフェースのメディアとして優れていると考えられる。図的ユーザ・インタフェース (DUI: Diagrammatic User Interface) [Lin93] は、人間と計算機との対話においてそのような図を利用するユーザ・インタフェースである。

DUIを支える重要な技術の一つがグラフの自動描画である。ただし、すでに述べた様にグラフの自動描画は一般に難しい問題であり、これがDUI開発のコストを引き上げていると言える。さらに、DUIを備えたアプリケーション・システムはより複雑な機構を必要とする。当然複雑な機構はより高い開発コストを必要とする。そこで、DUIを備えた様々なアプリケーションを素早く効率的に開発できる柔軟な土台、すなわちDUIプラットフォームが求められる。

第9章で述べたように、筆者らはシステムの今後の方向の一つとして、D-ABDUCTORの「DUIプラットフォーム」化を目指し、D-ABDUCTORの拡張機能やカスタマイズ機能を充実させ、それをベースにしたプログラムの試作を通して構想の検討を行なっている [MNS+96, ML96]。

本章では、DUIプラットフォーム化に向けて行なったD-ABDUCTORの拡張について説明し、その拡張されたD-ABDUCTORを基に試作したアプリケーションをいくつか紹介することで、DUIプラットフォームの可能性を示す。

A.1 D-ABDUCTORの拡張機能

第7章で説明した基本機能に加えて、D-ABDUCTORに次のような拡張を施した。

外部プログラムを起動するためのメニューを追加可能。アプリケーション開発者が、外部プログラムを起動するためのメニュー項目を追加できる。新しいメニュー項目の追加は、組み込み機能や他の外部プログラムには影響を与えない。その

ため、アプリケーション開発者は組み込み機能を犠牲にすることなく特定のアプリケーションのための機能を追加可能である。

自動描画のトリガーにより外部プログラムを起動可能。アプリケーション開発者が、組み込みの配置機能と装飾機能を外部プログラムで置き換えることができる。そして、外部プログラムとして供給される配置機能や装飾機能に対しても、自動描画のトリガーが働く。また、アニメーションによる表示機能、直接操作環境など組み込み機能の場合と同様に働く。

言語 Simple による外部プログラムとの双方向通信。D-ABDUCTOR は、ファイルあるいはパイプを通して、外部プログラムと Simple のステートメントを交換する。外部プログラムは言語 Simple を利用することで、画面上の図の状態やそれに対するユーザの操作をモニタしたり、逆に D-ABDUCTOR をコントロールしたりできる。すなわち、言語 Simple はアプリケーション・プログラムとのインタフェース (API: Application Program Interface) として機能する。外部プログラムを記述するための言語は Simple のステートメントを扱えなければならないが、それは単に文字列であるので、言語にとっては深刻な制約ではない。普及している多くの言語が外部プログラムの開発に利用可能である。

A.2 アプリケーションの例

前節で示した拡張により、D-ABDUCTOR は様々な方法でアプリケーション開発が可能になった。本節では筆者らがこれまでに試作したいくつかのアプリケーションを紹介する。

A.2.1 構造アナライザ

構造アナライザは記述形式のアンケート結果のようなテキスト・データの断片群を整理、組織化するために利用される (図 A.1 参照)。様々な自動配置機能 (組み込み機能、スプリング・レイアウト機能 [Ead84]、マグネティック・スプリング・レイアウト機能 [SM95a] など)、装飾機能 (たとえば、最短パスを見つけ色を変更する、不可到達頂点の集合を見つけそれらを不可視にする、など)、あるいはマクロな編集機能などを備え、それらの機能により、テキスト・データ群がもつ関連情報を表現した図を操作することで、ユーザはその情報のもつ様々な特徴を可視化することができる。様々な追加機能が外部プログラムとして提供されていてすべてメニューから起動できるよう設定されている。

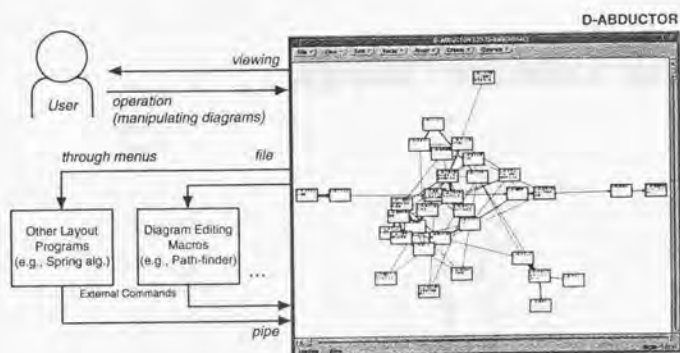


図 A.1: 構造アナライザ

構造アナライザの構成とアンケート結果の分析過程の1画面である。ノードがアンケートの回答を、それらの間のエッジが回答の間の関連を表わしている。回答の間の関連は D-ABDUCTOR とは別のシステム Keyword Associator [Wat94] によってテキスト処理によって自動的に計算されたものである。この例は、二つの興味深い回答の間の最短パスが見つけられたあと、マグネティック・スプリング・アルゴリズムによって自動配置されたものである。見つけられたパスが左から右へと水平に配置され、それ以外の部分は座標とは独立に関連度に応じた距離で配置されている。

A.2.2 ディレクトリ・ブラウザ

ディレクトリ・ブラウザはファイルやディレクトリを管理するために利用される (図 A.2 参照)。メイン・プログラムは、Perl で記述されていて、UNIX のファイル・システムと通信することでディレクトリに関する情報を得る。メイン・プログラムはディレクトリの情報を受けとって、ユーザがこれまでにたどったディレクトリを表わす木を構成する。さらに、そのプログラムは D-ABDUCTOR のログ情報をモニタしていて、ログ情報にノードのセレクト・コマンドを見つけたら、セレクトされたノードが表わすディレクトリについての情報を改めて収集し木を更新すると共に、ディレクトリの内容を図として描き加える。

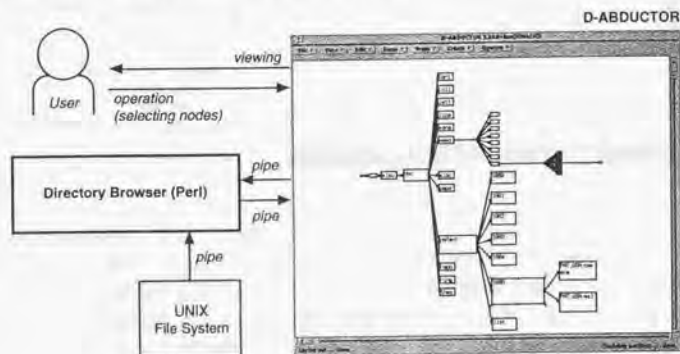


図 A.2: ディレクトリ・ブラウザ

ディレクトリの再帰構造が木によって表示される。ノードはファイルあるいはディレクトリを表わし、その間のエッジは始点に対応するディレクトリが終点に対応するファイルあるいはディレクトリを含むことを表わす。D-ABDUCTOR が提供する図ドレーシング機能によって、カレント・ディレクトリに対応するノードが最も大きく表示され、それ以外のノードもカレント・ディレクトリに (グラフ理論的な距離において) 近い程大きく表示されている。

A.2.3 ネットワーク・モニタ

ネットワーク・モニタは LAN の状態を実時間においてモニタするために利用される (図 A.3 参照)。メイン・プログラムは、Perl で記述されていて、UNIX のコマンド `etherfind` と `runtime` を利用して、イーサネット上のパケットと各ワークステーションのロード・アベレージに関する情報を得る。メイン・プログラムはそれらのコマンドの出力を受け取り、LAN の現在の状態を表わすグラフを構成する。メ

イン・プログラムはさらにCで記述されたスプリング・レイアウトを行なう外部プログラムを起動し、その結果に基づいてノードを移動するための言語 Simple のステートメントをD-ABDUCTORへ送る。

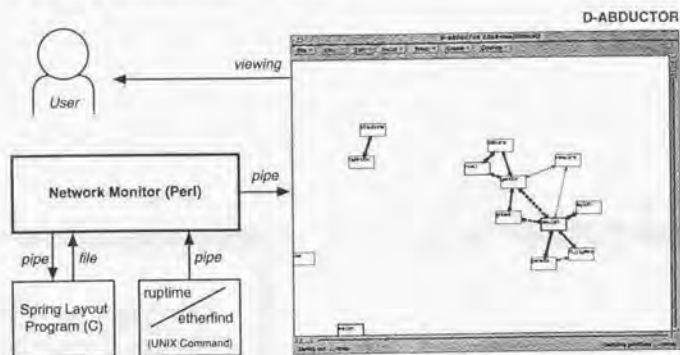


図 A.3: ネットワーク・モニタ

LANの状態が動的に変化する図によって表現される。ノードはワークステーション(WS)を、その間のエッジがWS間のバケットの存在を表わす。図のレイアウトはバケットを多く交換するWS同士が近づくように数秒間隔で刻々と変化する。ノードやエッジの視覚属性はより詳細な情報を表現している。たとえば、より多くのバケットを交換しているWS間のエッジやよりロード・アベレージが高いWSに対応するノードはより太い線で描かれる。

A.2.4 グラフィカル・ハイパーテキスト

グラフィカル・ハイパーテキスト GDS-info はハイパーテキストのドキュメントに対してテキスト・ビューとグラフ・ビューの異なる二つのビューを提供する(図A.4参照)。GDS-info は広く利用されているテキスト・エディタ Emacs の info システムを拡張することで開発された。筆者らはまず、D-ABDUCTOR との通信関数群を emacs-Lisp のパッケージ abd.el として用意した。アプリケーション開発者は、このパッケージを利用することで、低レベルの API の詳細を気にすることなく lisp の関数を記述することができる。すなわち、abd.el は言語 Simple と D-ABDUCTOR との通信に関する詳細をアプリケーション開発者から隠す働きをしている。

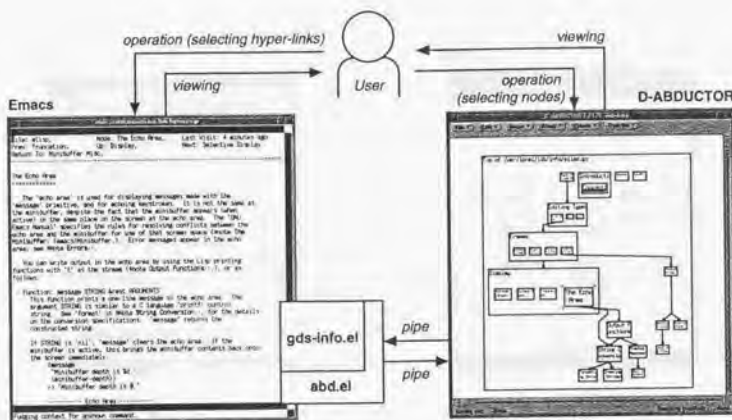


図 A.4: グラフィカル・ハイパーテキスト GDS-info

テキスト・ビュー(左)ではハイパーテキストの1ページが表示され、グラフ・ビュー(右)では過去に辿ったページとページ間のリンクが図示される。GDS-infoの基になっている Emacs の info はメニューとクロス・リファレンスと呼ばれる2種類のハイパーリンクを備えている。グラフ・ビューではメニューが包含関係で表わされ、クロス・リファレンスがエッジで表わされる。両方のビューは相互に連携しているため、どちらのビューでも操作が可能で、ユーザはテキスト・ビューで通常のナビゲーションができるだけでなく、グラフ・ビューでノードをクリックすることで過去におとずれたページへ直接もどることも可能である。

A.2.5 グラフィカル・アウトライン・プロセッサ

グラフィカル・アウトライン・プロセッサ GOL [NIMS96] も GDS-info と同様に、ドキュメントに対してテキスト・ビューとグラフ・ビューの異なる二つのビューを提供する(図 A.5参照)。GOL はテキスト・エディタ Emacs の outline モードを、先に説明した emacs-Lisp のパッケージ abd.el を用いて拡張することで開発された。

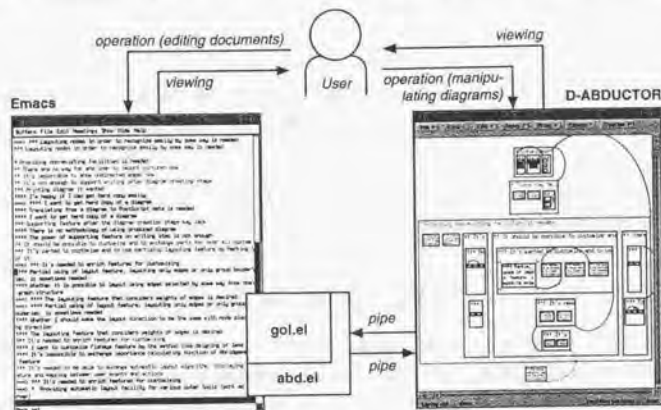


図 A.5: グラフィカル・アウトライン・プロセッサ GOL

テキスト・ビュー(左)では章立ての構造をインデントで表現したテキストが表示され、グラフ・ビュー(右)では章立ての構造と順序および段落間の参照関係を表示した順序付複合グラフ [NIMS96] が表示される。ユーザはこれら二つのビューを画面上で同時に見て、どちらでも文書を編集できる。すなわち、一方のビューでの操作結果は直ちにもう一方のビューへ反映される。

参考文献

- [Ado93] Adobe Systems Incorporated, 1585 Charleston Road, P.O. Box 7900, Mountain View, CA 94039-7900 USA. *User Guide Adobe Illustrator 5.0*, 1993.
- [BBL92] P. Bertolazzi, G. Di Battista, and G. Liotta. Parametric graph drawing. Technical Report 6/67, Dipartimento di Informatica e Sistemistica, University di Roma "La Sapienza", Italy, September 1992.
- [Ber82] Jacques Bertin. 図の記号学. 地図情報センター, 1982.
- [BETT93] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for automatic graph drawing: An annotated bibliography. Technical report, Department of Computer Science, Brown University, 1993. available via anonymous ftp from wilma.cs.brown.edu (128.148.33.6), files /bup/gdbiblio.tex.Z and /pub/gdbiblio.ps.Z.
- [Cla91] Claris Corporation, 5201 Patric Henry Drive, Box 58168, Santa Clara, CA 95052-8168 USA. マックドロー Pro ユーザーズガイド, 1991.
- [Cra93] Craft, Inc. ISOP for Windows 操作マニュアル, 1993.
- [Ead84] Peter Eades. A heuristics for graph drawing. *Congressus Numerantium*, Vol. 42, pp. 146-160, 1984.
- [EFK85] Peter Eades, I. Fogg, and D. Kelly. SPREMB: a system for developing graph algorithms. *Congressus Numerantium*, Vol. 66, pp. 123-140, 1985.
- [ELL93] Peter Eades, Tao Lin, and Xuemin Lin. Two tree drawing conventions. *International Journal of Computational Geometry and Applications*, Vol. 3, No. 2, pp. 133-153, 1993.
- [ELMS91] Peter Eades, Wei Lai, Kazuo Misue, and Kozo Sugiyama. Preserving the mental map of a diagram. In *Proceedings of COMPUGRAPHICS '91*, pp. 34-43, 1991.

- [Eve79] S. Even. *Graph Algorithms*. Computer Science Press, Maryland, 1979.
- [FE91] T. Fruchterman and E. Eisingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, Vol. 21, No. 11, pp. 1129-1164, 1991.
- [FEC95a] Qing-Wen Feng, Peter Eades, and Robert F. Cohen. Clustered graphs and c-planarity. Technical Report 95-04, Department of Computer Science, The University of Newcastle, Australia, 1995.
- [FEC95b] Qing-Wen Feng, Peter Eades, and Robert F. Cohen. Planar drawing of clustered graphs. Technical Report 95-05, Department of Computer Science, The University of Newcastle, Australia, 1995.
- [Fuj86] 富士通株式会社. OASYS 100FII 文書作成ガイド, 1986.
- [Fur86] George W. Furnas. Generalized fisheye views. In *Proceedings of CHI'86, Human Factors in Computing Systems*, pp. 16-23, 1986.
- [FW94] M. Froehlich and M. Werner. The graph visualization system daVinci - a user interface for applications. Technical Report 5/94, Department of Computer Science, Universität Bremen, 1994.
- [FW95] M. Froehlich and M. Werner. Demonstration of the interactive graph visualization system daVinci. In *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 1994, Proceedings*, pp. 266-269, 1995. Lecture Notes in Computer Science 894, Springer Verlag.
- [GNV88] Emden R. Gansner, Stephen C. North, and Kiem-Phong Vo. DAG - a program that draws directed graphs. *Software - Practice and Experience*, Vol. 18, No. 11, pp. 1047-1062, 1988.
- [Har88] David Harel. On visual formalisms. *CACM*, Vol. 31, No. 5, pp. 514-530, 1988.
- [HCMM89] J. G. Hollands, T. T. Carey, M. L. Matthews, and C. A. McCann. Presenting a graphical network: A comparison of performance using fisheye and scrolling views. In *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pp. 313-320, 1989.
- [Him90] Michael Himsolt. GraphEd user manual. Technical Report MIP - intern 01/90, Fakultät für Mathematik und Informatik, Universität Passau, 1990.

- [Him95] Michael Himsolt. GraphEd: A graphical platform for the implementation of graph algorithm. In *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 1994, Proceedings*, pp. 182-193, 1995. Lecture Notes in Computer Science 894, Springer Verlag.
- [Hir93] 廣田 隆一郎. イソップマーケティング戦略情報システム. 第2回「発想支援ツール」シンポジウム講演論文集, 1993.
- [Hor92] 堀浩一. ヒューマンインタフェース, 知識工学講座, 第10巻, 第2章, pp. 19-32. オーム社, 東京, 1992.
- [Hor94] Koichi Hori. A system for aiding creative concept formation. *IEEE Trans. SMC*, Vol. 24, No. 6, pp. 882-894, 1994.
- [IYA86] 出原 英一, 吉田 武夫, 渥美 浩章. 図の体系 — 図的思考とその表現. 日科技連出版社, 1986.
- [Jon89] Oliver Jones. *Introduction to the X Window System*. Prentice-Hall, Inc., 1989.
- [Kam88] Tomihisa Kamada. *On Visualization of Abstract Objects and Relations*. PhD thesis, The University of Tokyo, 1988.
- [Kaw86] 川喜田 二郎. KJ 法. 中央公論社, 1986.
- [KI92] 小池 英樹, 石井 威望. フラクタルの概念に基づく提示情報量制御方式. 情報処理学会論文誌, Vol. 33, No. 2, pp. 101-109, 1992.
- [KKO92] 小山 雅庸, 河合 和久, 大岩 元. カード操作ツール KJ エディタの実現と評価. コンピュータソフトウェア, Vol. 9, No. 5, pp. 38-53, 1992.
- [Koh91] Youji Kohda. Toward the semi-meaningful interface: Beyond the direct manipulation interface. In *Proceedings of International Conference on Multimedia Information Systems*, pp. 85-96, 1991.
- [Koj79] 小島 雅春. パースの Existential Graphs について. 日本科学哲学会編(編), 言語と非言語, pp. 77-91. 早稲田大学出版部, 1979.
- [KSH94] 金原 史和, 佐藤 真一, 濱田 喬. 図形間の幾何的および概念的関係を用いた作図支援. 情報処理学会論文誌, Vol. 35, No. 5, pp. 897-907, 1994.
- [Kun93] 國藤 進, 他. 特集「発想支援システム」, 人工知能学会誌, Vol. 8, No. 5, 1993.

- [KWM+93] 神田 陽治, 渡部 勇, 三末 和男, 平岩 真一, 増井 誠生. グループ発想支援システム: GriPS. 人工知能学会誌, Vol. 8, No. 5, pp. 601-610, 1993.
- [LA93] Y. K. Leung and M. D. Apperley. A taxonomy of distortion-oriented techniques for graphical data presentation. In *Proceedings of HCI International '93*, pp. 104-109, 1993.
- [Lai93] Wei Lai. *Building Interactive Diagram Applications*. PhD thesis, Department of Computer Science, The University of Newcastle, Australia, 1993.
- [Leu89] Y. K. Leung. Human-computer interface techniques for map based diagrams. In *Proceedings of HCI International '89, Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pp. 361-368, 1989.
- [Lin93] Tao Lin. *A General Schema for Diagrammatic User Interfaces*. PhD thesis, Department of Computer Science, The University of Newcastle, Australia, 1993.
- [Mak88] Erkki Mäkinen. Experiments on drawing 2-level hierarchical graphs. Series of Publications A A-1988-1, Department of Computer Science, University of Tampere, January 1988.
- [Mak90] Erkki Mäkinen. How to draw a hypergraph. *International Journal of Computer Mathematics*, Vol. 34, pp. 177-185, 1990.
- [MELS95] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, Vol. 6, No. 2, pp. 183-210, 1995.
- [Mes88] Eli Benjamin Messinger. Automatic layout of large directed graphs. Technical Report 88-07-08, Department of Computer Science, University of Washington, July 1988.
- [MHN94] 宗森 純, 堀切 一郎, 長澤 庸二. 発想支援システム群元の分散協調型 KJ 法実験への適用と評価. 情報処理学会論文誌, Vol. 35, No. 1, pp. 143-153, 1994.
- [Mis89] 三末 和男. 無向枝と有向枝の混在するグラフの描画法について. 情報処理学会第 39 回全国大会講演論文集, No. 3L-9, pp. 81-82, 1989.
- [Mis90a] 三末 和男. 図の概略化と発想支援への応用. 情報処理学会ヒューマンインタフェース研究会資料, No. 90-HI-31-1, 1990.

- [Mis90b] 三末 和男. 図的発想支援システム:D-ABDUCTOR のための図の操作について. 情報処理学会第 41 回全国大会講演論文集, No. 6G-4, 1990.
- [Mis93a] Kazuo Misue. D-ABDUCTOR 2.0 user manual. Research Report IAS-RR-93-9E, FUJITSU LABORATORIES, IAS, 1993.
- [Mis93b] 三末 和男. 図的発想支援システム D-ABDUCTOR のグループウェア化手法. 情報処理学会グループウェア研究会資料, No. 93-GW-2-3, 1993.
- [Mis94] 三末 和男. 図的思考支援を目的とした図ドレッシングについて. リサーチレポート ISIS-RR-94-5J, 富士通情報研, 1994.
- [ML96] Kazuo Misue and Tao Lin. Functional analysis of DUI systems: Toward PizzaSystem development. Research Report ISIS-RR-96-11E, FUJITSU LABORATORIES, ISIS, 1996.
- [MNN90] 松浦 敏雄, 直田 創, 中村 真. 図形の部品化および接続包含関係の保存機能をもつ 作図ツール Key3. 電子情報通信学会論文誌 D-I, Vol. J73-D-I, No. 11, pp. 864-872, 1990.
- [MNS+96] Kazuo Misue, Kiyoshi Nitta, Kozo Sugiyama, Takeshi Koshiba, and Robert Inder. Techniques for DUI platforms: Developing graph drawing applications on D-ABDUCTOR. Research Report ISIS-RR-96-7E, FUJITSU LABORATORIES, ISIS, 1996.
- [MRC91] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of CHI'91*, pp. 173-179, 1991.
- [MS88a] 三末 和男, 杉山 公造. カード・システムを抽象化した複合グラフとその階層的描画法について. 情報処理学会グラフィクスと CAD 研究会資料 32, No. 32-2, 1988.
- [MS88b] 三末 和男, 杉山 公造. 複合階層グラフとその描画法について — 発想支援系の基礎技法の開発 —. 情報処理学会第 36 回全国大会講演論文集, No. 5Z-5, pp. 2131-2132, 1988.
- [MS89a] 三末 和男, 杉山 公造. グラフィック・インタフェースのための全体と詳細を同時に示せる遠近画法. 第 5 回ヒューマン・インタフェース・シンポジウム論文集, No. 2317, pp. 463-468, 1989.
- [MS89b] 三末 和男, 杉山 公造. 図的思考支援を目的とした複合グラフの階層的描画法について. 情報処理学会論文誌, Vol. 30, No. 10, pp. 1324-1334, 1989.

- [MS90] 三末 和男, 杉山 公造. 複合グラフ描画における全体と詳細を同時に見せる遠近画法. 情報処理学会ヒューマンインタフェース研究会資料, No. 90-HI-29, 1990.
- [MS91a] Kazuo Misue and Kozo Sugiyama. Multi-viewpoint perspective display methods: Formalization and application to compound graphs. In *Proceedings of HCI International '91 - Human Aspects in Computing (18B)*, pp. 834-838, 1991.
- [MS91b] 三末 和男, 杉山 公造. 図的思考支援を目的とした図の多視点遠近画法について. 情報処理学会論文誌, Vol. 32, No. 8, pp. 997-1005, 1991.
- [MS93] Kazuo Misue and Kozo Sugiyama. An overview of diagram based idea organizer: D-ABDUCTOR. Research Report IIAS-RR-93-3E, FUJITSU LABORATORIES, IIAS, 1993.
- [MS94a] Kazuo Misue and Kozo Sugiyama. How does D-ABDUCTOR support human thinking processes? In *Proceedings of CG International '94*, 1994.
- [MS94b] 三末 和男, 杉山 公造. マグネティック・スプリング・モデルによるグラフ描画法について. 情報処理学会ヒューマンインタフェース研究会資料, No. 94-HI-55, 1994.
- [MS94c] 三末 和男, 杉山 公造. 思考支援システムの評価法および D-ABDUCTOR の評価実験について. リサーチレポート ISIS-RR-94-22J, 富士通情報研, 1994.
- [MS94d] 三末 和男, 杉山 公造. 図を対話メディアとする発想支援システム D-ABDUCTOR. 1994 年情報学シンポジウム講演論文集, pp. 143-152, 1994.
- [MS94e] 三末 和男, 杉山 公造. 図的発想支援システム D-ABDUCTOR の開発について. 情報処理学会論文誌, Vol. 35, No. 9, pp. 1739-1749, 1994.
- [MS95a] Kazuo Misue and Kozo Sugiyama. Evaluation of thinking support systems from operational points of view. In *Proceedings of HCI International '95 - Symbiosis of Human and Artifact (20A)*, pp. 697-702, 1995.
- [MS95b] 三末 和男, 杉山 公造. 思考支援システムの評価法 および D-ABDUCTOR の評価実験について. 計測自動制御学会 第 17 回システム工学会部研究会「発想支援ツール」資料, pp. 61-68, 1995.

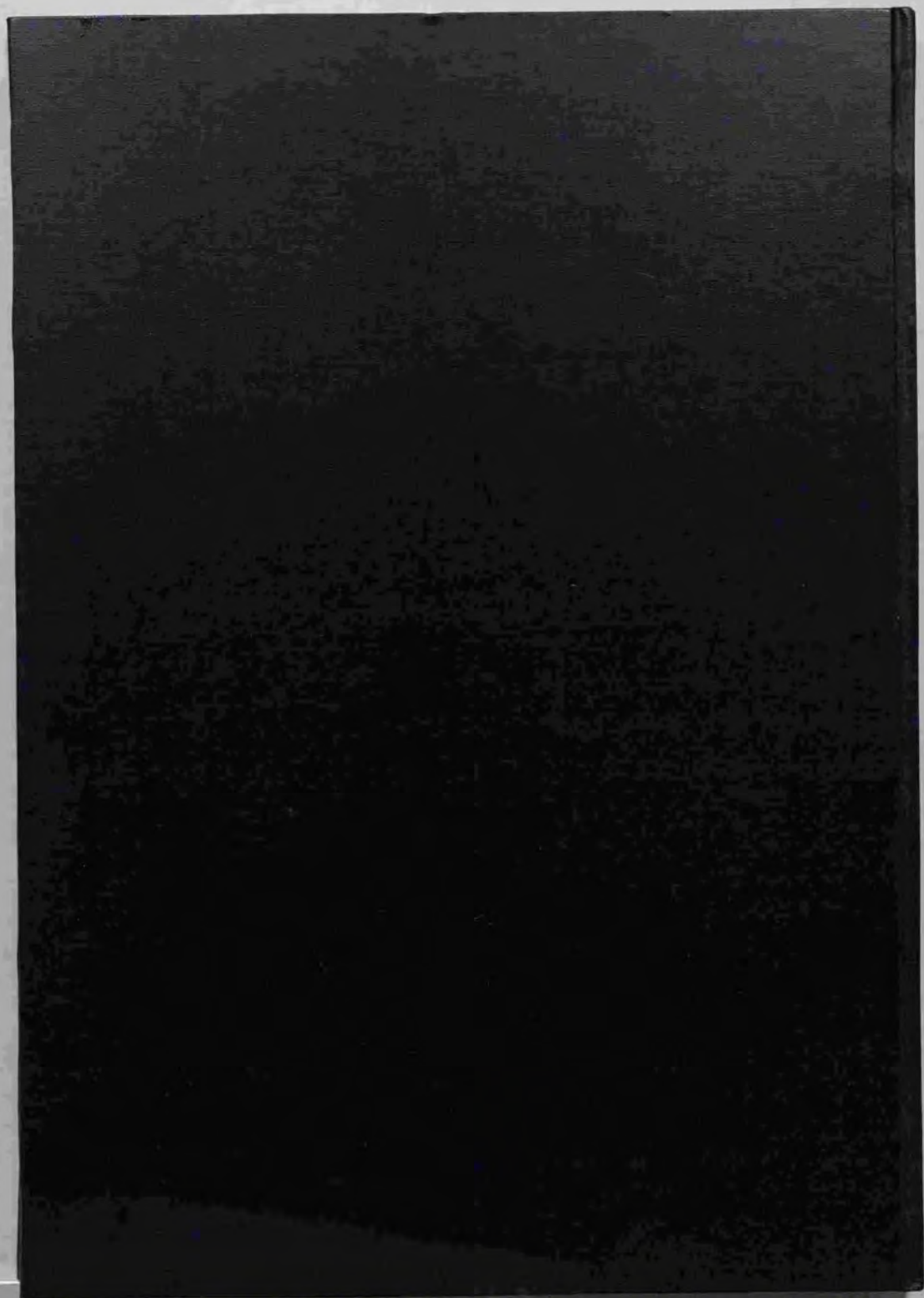
- [MS96] 三末 和男, 杉山 公造. 図的発想支援システム D-ABDUCTOR の操作性の評価. 情報処理学会論文誌, Vol. 37, No. 1, pp. 133-143, 1996.
- [Mye89] Brad A. Myers. User-interface tools: Introduction and survey. *IEEE Software*, Vol. 6, No. 1, 1989.
- [Mye92] Brad A. Myers. Demonstrational interfaces: A step beyond direct manipulation. *IEEE Computer*, Vol. 25, No. 8, pp. 61-73, 1992.
- [NA94a] 西田 豊明, 安西 祐一郎, 他. 特集「図による推論」. 人工知能学会誌, Vol. 9, No. 2, pp. 2-35, 1994.
- [NA94b] 西田 豊明, 安西 祐一郎. 特集「図による推論」にあたって. 人工知能学会誌, Vol. 9, No. 2, p. 2, 1994.
- [Nak95] 中川 格. 大規模ハイパーテキストシステムのための 迷子問題解決支援環境の構築, 1995. 図書館情報大学 図書館情報学部 図書館情報学科 卒業論文.
- [NIMS96] Kiyoshi Nitta, Robert Inder, Kazuo Misue, and Kozo Sugiyama. GOL: Graphical outline processor — simultaneously using a text view and a graph view. In *Asia Pacific Computer Human Interaction (APCHI'96)*, pp. 469-478, June 25-28 1996.
- [Noi93a] Emanuel G. Noik. Exploring large hyperdocuments: Fisheye views of nested networks. In *Hypertext '93: ACM Conf. on Hypertext*, pp. 192-205, Seattle, WA, November 1993.
- [Noi93b] Emanuel G. Noik. Layout-independent fisheye views of nested graphs. In *VL '93: IEEE Symposium on Visual Languages*, pp. 336-341, Bergen, Norway, August 1993.
- [Nor] Stephen C. North. Drawing ranked digraphs with recursive clusters. November 3, 1993 DRAFT.
- [OK95] 女部田 武史, 國藤 進. 複数の KJ 法図解の差異や共通部を可視化するシステムの実現について. 計測自動制御学会 第 18 回システム工学部会研究会「発想支援技術」資料, pp. 21-28, 1995.
- [OKK90] Hajime Ohiwa, Kazuhisa Kawai, and Masanobu Koyama. Idea processor and the KJ method. *Journal of Information Processing*, Vol. 31, No. 1, pp. 44-48, 1990.

- [PT90] Frances Newbery Paulisch and Walter F. Tichy. EDGE: An extendible graph editor. *Software - Practice and Experience*, Vol. 20(S1), pp. S1/63-S1/88, 1990.
- [Raa91] Thomas Van Raalte, editor. *XView Reference Manual*. O'Reilly & Associates, Inc, 1991.
- [RCM93] George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. *CACM*, Vol. 36, No. 4, pp. 57-71, 1993.
- [SA82] Robert Spence and Mark Apperley. Data base navigation: an office environment for the professional. *Behaviour and Information Technology*, Vol. 1, No. 1, pp. 43-54, 1982.
- [SB91] M. Sarkar and Marc H. Brown. Graphical fisheye views of graphs. Technical Report CS-91-61, Department of Computer Science, Brown University, 1991.
- [Sco95] Aaron Scott. A survey of graph drawing systems. Technical Report 95-1, Department of Computer Science, The University of Newcastle, Australia, 1995.
- [Shi93] 篠原 靖志. 知識整理支援システム CONSIST-II — CONSIST の評価と改良について —. *人工知能学会誌*, Vol. 8, No. 5, pp. 593-600, 1993.
- [Shn83] Ben Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, Vol. 16, No. 8, pp. 57-69, 1983.
- [SHO93] 杉本 雅則, 堀 浩一, 大須賀 節雄. 設計問題への発想支援システムの応用と 発想過程のモデル化の試み. *人工知能学会誌*, Vol. 8, No. 5, pp. 575-582, 1993.
- [SHO94] 角 康之, 堀 浩一, 大須賀 節雄. テキストオブジェクトを空間配置することによる 思考支援システム. *人工知能学会誌*, Vol. 9, No. 1, pp. 139-147, 1994.
- [Sil91] Silicon Beach Software, Inc., 9770 Carroll Center Road, Suite J, San Diego, CA 92126-4551 USA. *Aldus SuperPaint (3.0) User Manual*, June 1991.
- [Sil92] Silicon Beach Software, Inc., 9770 Carroll Center Road, Suite J, San Diego, CA 92126-4551 USA. *IntelliDraw (1.0) User Manual*, June 1992.

- [SK94] 鈴木 和彦, 鎌田 富久. グラフ構造理解のための単純無向グラフの自動描画アルゴリズム. 第35回プログラミング・シンポジウム論文集, pp. 71-82, 1994.
- [SM86] 杉山 公造, 三末 和男. ヒューマンインタフェースとしての図形言語 ii — 領域・網図系と手描き様図 —. 第2回ヒューマン・インタフェース・シンポジウム, No. 2412, pp. 479-480, 1986.
- [SM90] Kozo Sugiyama and Kazuo Misue. "Good" graphic interface for "good" idea organizers. In *Proceedings of INTERACT '90, Cambridge, U.K.*, pp. 521-526, 1990.
- [SM91] Kozo Sugiyama and Kazuo Misue. Visualization of structural information: Automatic drawing of compound digraphs. *IEEE Trans. SMC*, Vol. 21, No. 4, pp. 876-892, 1991.
- [SM95a] Kozo Sugiyama and Kazuo Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, Vol. 6, No. 3, pp. 217-231, 1995.
- [SM95b] Kozo Sugiyama and Kazuo Misue. A simple and unified method for drawing graphs: Magnetic-spring algorithm. In *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 1994, Proceedings*, pp. 364-375, 1995. Lecture Notes in Computer Science 894, Springer Verlag.
- [SM96] Kozo Sugiyama and Kazuo Misue. A generic compound graph visualizer / manipulator: D-ABDUCTOR. In *Graph Drawing, Symposium on Graph Drawing, GD '95, Passau, Germany, September 20 - 22, 1995*, pp. 500-503, 1996. Lecture Notes in Computer Science 1027, Springer Verlag.
- [STT81] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. SMC*, Vol. 11, No. 2, pp. 109-125, 1981.
- [Sug85] 杉山 公造. ヒューマン・インタフェースとしての図形言語: 網図系を中心として. 第1回ヒューマン・インタフェース・シンポジウム論文集, pp. 127-132, 1985.
- [Sug87a] Kozo Sugiyama. A cognitive approach for graph drawing. *Cybernetics and Systems*, Vol. 18, pp. 447-488, 1987.

- [Sug87b] 杉山 公造. ヒューマン・インタフェースとしての図形言語 III: 認知的基準によるグラフ描画. 第3回ヒューマン・インタフェース・シンポジウム論文集, pp. 303-308, 1987.
- [Sug88a] 杉山 公造. 図的思考展開支援に関する基礎的研究 — 発想系情報学の構築にむけて — 第I部 図的思考展開過程の分析と基本的枠組み. 国際情報社会科学研究所報告 第24号, 富士通株式会社, 1988.
- [Sug88b] 杉山 公造. 図的思考展開支援に関する基礎的研究 — 発想系情報学の構築にむけて — 第II部 図の修辞術: 認知的基準によるグラフ描画. 国際情報社会科学研究所報告 第25号, 富士通株式会社, 1988.
- [Sug89] 杉山 公造. 図的思考展開支援からみた KJ 法の分析. 第5回ヒューマン・インタフェース・シンポジウム論文集, pp. 325-330, 1989.
- [Sug90] 杉山 公造. 発想支援のためのインタフェース研究 - 発想系情報学に向けて. Research Memorandum IIAS-RM-90-1J, 富士通株式会社, 1990.
- [Sug91] 杉山 公造. 思考支援ツール. 電子情報通信学会誌, Vol. 74, No. 2, pp. 159-165, 1991.
- [Sug93a] 杉山 公造. グラフ自動描画法とその応用. 計測自動制御学会, 1993.
- [Sug93b] 杉山 公造. 収束的思考支援ツールの研究開発動向 — kj 法を参考とした支援を中心にして —. 人工知能学会誌, Vol. 8, No. 5, pp. 32-38, 1993.
- [Sun90a] Sun Microsystems, Inc. OPEN LOOK スタイルガイド, Addison-Wesley, 東京, 1990.
- [Sun90b] Sun Microsystems, Inc. OPEN LOOK ファンクショナルスペック, Addison-Wesley, 東京, 1990.
- [SW93] Aaron Scott and Richard Webber. Grammar oriented visualisation environment (GOVE). Honours thesis, Department of Computer Science, The University of Newcastle, Australia, 1993.
- [Tak84] 高橋誠. 問題解決手法の知識. 日本経済新聞社, 1984.
- [Wat94] 渡部 勇. 発散的思考支援システム「Keyword Associator」第二版. 計測自動制御学会 第15回システム工学会研究会資料, pp. 9-16, 1994.
- [Wir76] Niklaus Wirth. *Algorithms + Data Structures = Programs*. Prentice-Hall, 1976.

- [YKH93] 山口 ふみ, 國藤 進, 林 義樹. 智的問題解決技法イソップの試用評価実験. 第2回「発想支援ツール」シンポジウム講演論文集, 1993.
- [Yos77] 吉川弘之. コンピュータグラフィック論. 日科技連出版社, 1977.
- [You89] Lawrence F. Young. *Decision Support and Idea Processing Systems*. Wm. C. Brown Publishers, Dubuque, Iowa, 1989.
- [Zha93] Rui Zhao. Incremental recognition in gesture-based and syntax-directed diagram editors. In *Proc. of INTERCHI '93*, pp. 95-100, 1993.





Kodak Color Control Patches

Blue Cyan Green Yellow Red Magenta White 3/Color Black

Kodak Gray Scale

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

C Y M

© Kodak, 2007 TM Kodak