

データベース構築促進及び技術開発に関する報告書

知的ネットワークを活用したデータベース構築に関する調査研究

平成4年3月

財団法人 データベース振興センター

委託先 株式会社新世代システムセンター

本報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて作成したものである。

序

データベースは、わが国の情報化の進展上、重要な役割を果たすものと期待されている。今後、データベースの普及により、わが国において健全な高度情報化社会の形成が期待される。さらに海外に対して提供可能なデータベースの整備は、国際的な情報化への貢献および自由な情報流通の確保の観点からも必要である。しかしながら、現在わが国で流通しているデータベースの中でわが国独自のものは1/3にすぎないのが現状であり、わが国データベースサービスひいてはバランスある情報産業の健全な発展を図るためには、わが国独自のデータベースの構築およびデータベース関連技術の研究開発を強力に促進し、データベースの拡充を図る必要がある。

このような要請に応えるため、(財)データベース振興センターでは日本自転車振興会から機械工業振興資金の交付を受けて、データベースの構築および技術開発について民間企業、団体等に対して委託事業を実施している。委託事業の内容は、社会的、経済的、国際的に重要で、また地域および産業の発展の促進に寄与すると考えられているデータベースの構築とデータベース作成の効率化、流通の促進、利用の円滑化・容易化などに関係したソフトウェア技術・ハードウェア技術である。

本事業の推進に当って、当財団に学識経験者の方々と構成されるデータベース構築・技術開発促進委員会（委員長 山梨学院大学教授 蓼沼良一氏）を設置している。

この「知的ハイパーメディアを活用したデータベース構築に関する調査研究」は平成3年度のデータベースの構築促進および技術開発促進事業として、当財団が株式会社新世代システムセンターに対して委託実施した課題の一つである。この成果が、データベースに興味をお持ちの方々や諸分野の皆様方のお役に立てば幸いである。

なお、平成3年度データベースの構築促進および技術開発促進事業で実施した課題は次表のとおりである。

平成4年3月

財団法人 データベース振興センター

平成3年度 データベース構築・技術開発促進委託課題一覧

分野	課題名	委託先
社会	1 形態学的コメントを含む病理データベースのプロトタイプ作成	(株)エス・ピー・オー
	2 交通事故調査データベースのプロトタイプの作成	(財)日本自動車研究所
	3 シルバーエイジの医療と福祉情報の実的な活用を目的としたデータベースの構築	美崎高齢者福祉互助会 美崎生活館
	4 気候情報データベースの構築	(株)エムテーエス雪氷研究所
	5 地下水情報データベースシステムの構築のための調査研究	(株)パスコ
	6 ファジィに関する文献データベースシステムの開発	(財)日本情報処理開発協会
	7 大学におけるデータベース利用教育システムに関する調査研究	日外アソシエーツ(株)
	8 マルチメディア型社会科用データベースの開発	(株)新学社
中小企業振興 地域活性化	9 異分野研究のための知的オリエンテーション・データベースシステムの構築可能性調査	(株)けいはんな
	10 瀬戸内圏公共図書館の郷土資料データベースの構築	(株)中国新聞社
	11 記事データベースアクセス用パイロットシステム構築	(株)河北新報社
	12 商業調整支援データベースの構築	(株)日本統計センター
	13 地域流通最適化に必要なデータベースの構築に関する研究	(社)日本ボランティア・チェーン協会
	14 情報源検索データベースのプロトタイプ作成	セントラル開発(株) 情報図書館RUKIT
海外	15 有価証券報告書のMTデータ変換ソフト開発と英訳辞書作成	コムラインインターナショナル(株)
	16 海外の主要国際・国家規格データベースの構築	日本電子計算(株)
	17 アジア太平洋交流データベースの研究－プロトタイプ作成－	(株)西日本新聞社
	18 先端産業分野における専門用語の電子辞書データベース化の調査研究	科学技術情報研究所(株)
技術	19 書誌データベース用ダイナミックソース・エンジンの構築と自然言語検索システムへの応用	(株)紀伊國屋書店
	20 知的ハイパーメディアを活用したデータベース構築に関する調査研究	(株)新世代システムセンター
	21 CD-ROMによる光学材料データベースの構築	(株)リアライズ社

I. 課題性の研究

II. 知的ハイパーメディアによる
データベースの構築

1章 知的ハイパーメディアの動向

- 1. 1 知的ハイパーメディアの考え方..... 3
- 1. 2 ハイパーメディアの製品化と応用..... 6
- 1. 3 知的ハイパーメディアのニーズ..... 18
- 1. 4 知的ハイパーメディアのもたらす効果..... 19
- 1. 5 新しい情報メディアとしての知的ハイパーメディア..... 20
- 1. 6 今後の動向..... 20

2章 知的ハイパーメディアに関する技術..... 23

- 2. 1 知的ハイパーメディアの要素技術..... 23
- 2. 2 知的ハイパーメディアの基盤技術..... 49
- 2. 3 知的ハイパーメディアのマルチメディア化..... 74

3章 データベース構築技術における知的ハイパーメディア..... 101

- 3. 1 データベース・システム..... 101
- 3. 2 知的ハイパーメディア応用システムの構築..... 114
- 3. 3 知的ハイパーメディアとオブジェクト
指向データベースシステム..... 117
- 3. 4 ハイパーメディアとオブジェクト指向
データベースシステム..... 153
- 3. 5 知的ハイパーメディアによるデータシステム
構築上の問題点..... 153

4章 知的ハイパーメディア技術の研究事例..... 157

- 4. 1 オブジェクト指向データベース用ハイパー
メディア・インターフェース..... 158
- 4. 2 情報検索のための媒体間の操作可能な
コード化されたもの..... 185
- 4. 3 インテリジェント・マルチメディア
情報検索システム..... 207

III. 今後の課題 223

I. 課題性の研究

1. 目 的

データベースにおいて情報を多角的に活用するには、別々に存在する断片的な情報を集めるだけでなく、それらの間に、関連性や類似性を認め、それにもとづいて関連情報を相互に結び付ける必要性が高まっている。つまり、音や映像、文書、図形等の多様な情報を人間的な思考方法や発想方法、認識方法に従って検索や管理ができれば、データベースの活用において画期的な効果をもたらすことになる。このように、関連する情報を相互に結び付けて連想網を構成し、この網を辿ることによって情報の連想的な探索を支援する技術として、最近、ハイパーメディアが脚光を浴びている。さらに、今後は、これにインテリジェントをもたせ、人間との親和性をより向上させた知的ハイパーメディアが、データベース構築の重要な役割を担おうとしている。

こうした視点から、知的ハイパーメディアに関して様々な側面から調査・研究を実施し、データベース構築技術の向上を図るとともにわが国におけるデータベースの発展に寄与することを目的としている。

1. 実施内容

本調査研究では、以下の4項目を主たる内容としている。

1) 知的ハイパーメディアの現状と動向について

知的ハイパーメディアの現状とその応用、知的ハイパーメディアのニーズとそのもたらす効果、新しい情報メディアとしての知的ハイパーメディアと今後の動向等についての考察

2) 知的ハイパーメディアに関する技術についての調査研究

知的ハイパーメディアの要素技術、基盤技術、マルチメディア化技術等について、現在利用できる技術さらに今後必要とされる技術等について調査研究を行う。

3) データベース構築技術としての知的ハイパーメディアに関する調査研究

知的ハイパーメディア応用システムの構築、知的ハイパーメディアとオブジェクト指向DMSとの関連、知的ハイパーメディアによるデータベース構築上の課題等について調査研究を行う。

4) 知的ハイパーメディア技術とデータベースに関する研究事例についての考察

現在、世界の様々な研究機関で行われている代表的な研究開発事例について考察する。

1. 実施体制

新世代システムセンター内に当該事業遂行のための事務局を設置し、このもとで「調査・研究会」を編成して、調査研究を実施した。なお、必要に応じて専門家によるプレゼンテーションあるいはヒヤリング等も併せて実施した。

1) 調査研究会メンバー

滝沢 誠 (東京電機大学理工学部・経営工学科助教授)

山島雄嗣 (財)日本情報処理開発協会・調査部長)

市川 隆 (財)日本情報処理開発協会 AI・ファジー振興センター所長)

道田國雄 (新世代システムセンター企画部長)

近藤伸一 (新世代システムセンター調査部長)

2) 専門家によるプレゼンテーションおよびヒヤリング等

*北海道大学工学部教授 田中 譲

* (株) 日本電子化辞書研究所 横井俊夫

* Dr.Peter Zorkoczy Director,Center for Electronics Education,Open University

Ⅱ．知的ハイパーメディアによるデータベースの構築

1章 知的ハイパーメディアの動向

1.1 知的ハイパーメディアの考え方

1.1.1 情報メディアの統合化

われわれは、産業・社会の活動や家庭生活において様々の情報を大量に利用している。これらの情報は、数字や文字、あるいは絵画やテレビ映像、または音楽などの人工音や風・雨などの自然音であったり、人との会話あるいは演説・講演であったりする。コンピュータの応用の拡がりとともに様々の形態になっている情報をコンピュータで処理しようとする希望も、また拡がってきた。近年、マルチメディア、ハイパーメディアといった言葉や概念が喧伝されるのも、ニーズや需要の高まりがあり、これを実現する技術や製品化が進んできたからということができる。

1965年にアメリカのダートマスで開催されたコンピュータ会議は、コンピュータで何ができるかとして、人工知能（AI）についての議論が行われたことで知られている。人間の知的活動のうち、コンピュータによってどこまで実現できるか。人間との対話ができるようになるのか。といった命題をどのように研究し解決していくかが議論されている。現在の話題であるマルチメディアの利用については、ある意味では既にコンピュータが登場したときから議論されていたのである。

その後のコンピュータ技術の発展には目覚ましいものがある。ここにきて漸く、様々の媒体（メディア）に蓄積された情報の利用に目が向けられるようになってきた。特に、利用者の思考に応じた情報処理、言い換えると情報がどのようなメディアになっているかを意識することなく、また一方では多様化されたメディアを統合的に利用しようという考え方が議論されるようになってきた。

情報メディアを統合化して利用しようという考え方は、人間の思考過程に準じ、文章の作成や会話のなかで思うがままに、様々の情報を利用することになる。従来のコンピュータは、予め扱う情報メディアを限定しなかった。また、画

像や音声のメディアを文字と同一に扱うことが困難であった。あるいは経済的にできなかった。近年マルチメディアに関心が高まってきた背景には、経済的に、パソコンを利用するように手軽に利用するための製品が比較的適切な価格で提供されるようになってきたことがあげられる。

情報技術の進歩が文化の発展を促し、場合によっては急速な変化をもたらしている。マルチメディアは、従来、人間が特別に情報メディアを意識しないで利用していたのが、ようやくコンピュータ技術で取り扱うことが可能になってきた。これが文化、社会、個人、等の広い分野で大きなインパクトを与えることが予想される。情報化のもたらすインパクトは、情報の処理や通信が国境や人間思想の枠を超えてコミュニケーションが発達し世界を変えてきた。マルチメディアの統合的利用は、様々なチャンネルを通じたコミュニケーションが進むとともに、伝えたい情報、理解容易な情報といった観点から一層バラエティのあるメディアによってコミュニケーションが行うことができるようになってきているのである。

1.1.2 マルチメディアの定義

われわれは、様々な情報メディアを扱っている。

記号—文字、数字、記号

画像—図面（写真、絵画）、静止画（記録図）、動画（映画、ビデオ、テレビ画面）

音声—動物の音声、自然音、人工音

マルチメディアは、これらの情報をコンピュータを使用して統合的に扱うこととしている。より具体的には、[Latta91]によれば、マルチメディアとは、コンピュータがツールを提供する。これら2つの要素が合体されるとマルチメディアは空間ベースと時間ベースのメディアにパソコン技術を応用したものと定義されるとしている。このようにLatta博士は、マルチメディアでは、単に複数のメディアを統合的に扱うということから、一歩進めて、コンピュータの役割を強調しているのが特徴である。デジタル化した情報メディアを統合的に扱うためには、マルチメディアコンピュータを欠くことができなく、そのメディア製作の環境となるコンピュータシステムおよびソフトウェアツールが重要な役割を果たすということである。

もう一つの用語であるハイパーメディアは、ハイパーとメディアを統合した用語であ

る。文字どおりとすると、従来の情報メディアを超えるものということになるが、それが何を意味することなのか。

〔Wood head 90〕では、次のとおり定義されている。

Hyperとは

This prefix has dictionary meaning of above,excessive or beyond. Its use in computing,as exemplified below,is perhaps related to analogous jargon or buzzwords coined by the science fiction community, e.g.hyperspace,a speculative extra physical dimension. The hyper prefix now has a strong historical momentum, and seems likely to endure. However,with hindsight other prefixes snch as meta-(among,about or between),or inter-might have been more accurate descriptors of the structural emphasis of hypermedia; the 'hyper-level' is strictly at a metalevel to the semantic or content level.(It may be interesting to note that in the meantime,metatext has become a favourite term of the Structuralist schools of literary criticism,and Intermedia is the proprietary name of an advanced hypermedia system.)

Hypermediaとは

Taken to be a generic approach to constructing non-linear,computer supported materials,as embodied by a number of commercial and academic program shell; the term is also used to describe the materials them-selves. Hypermedia itself is a subest of the more general class of interactive multimedia not all implementations of which support 'hyper' functionality.

また、〔原-91〕では、メディアの新しい利用形態ととらえており、より具体的には、ハイパーメディアは、パイパーテキスト構造をマルチメディア表現で実現するためのアプローチとしている。ハイパーテキスト構造とは、関連情報を直接関係づけ、対話的に検索表示できる構造であり、マルチメディア表現とは、文書ばかりでなく、図形、静止画、動画、音声、音といった複数のメディアが扱える表現のことである。ハイパーメディアとは、このパイパーテキスト構造のマルチメディア表現とを有する情報利用環境である。と定義している。

1.1.3 知的ハイパーメディアの考え方

最近のコンピュータ技術、デジタル化技術の発展によって、情報メディア（文章、画像、音声等）を統合化して利用することが可能になりつつある。その期待は、とくにデータベースや情報検索の領域に高まっている。

Ted Nelson は、1974年にハイパーメディアの概念を提唱した。コンピュータを計算機械や情報検索マシンから、情報メディアの処理機械になり、究極的には、情報メタファとして、仕事、著作、様々の行動（振る舞い）などの主要な機能を果たすようになる。とくに個人（パーソナル）的に使用することができるようになると、生活や仕事に大きな変革をもたらすと述べている。

知的ハイパーメディアは、ハイパーメディアに人工知能技術を応用する考えである。ハイパーメディアに人工知能やニューロネットワーク技術を応用することによって、システムのインテリジェントを持たせようとする考え方が知的ハイパーメディアである。

具体的には、知識ベース機能と推論機能を有するエキスパートシステム、電子化辞書や特定領域（ドメイン）の大規模知識ベース、自然言語理解システム、ニューラルネットワーク等々との結合による新しい応用システムである。

1.2 ハイパーメディアの製品化と応用

1.2.1 ハイパーメディアの製品化

1. マルチメディアパーソナルコンピュータ

ハイパーメディアは、パーソナルコンピュータワークステーションを稼働環境として発展してきている。

マルチメディア用パーソナルコンピュータとしては、米国 Apple Computer 社の Macintosh シリーズ、同じく米国の NeXT Computer 社の NeXT シリーズ、富士通の FM/TOWNS シリーズなどがある。

最近になって汎用パーソナルコンピュータにマルチメディア対応機器を接続できるようになっている。

1) Apple Computer社のMacintoshシリーズ

Macintoshシリーズは、次表に示す機種が販売されている。このマシンは、グラフィックユーザーインターフェース（GUI）を基本仕様として、OSレベルでユーザーに提供されているのが特徴である。GUIを通してアプリケーションの操作を統一的に行うことができるので、初心者のユーザーでも容易に操作方法を習得できるということから、米国では学校や個人などの利用が多く、最近になってビジネス分野にも需要が増している。

OSはMacintoc-OSで、マルチウインドウ表示をサポートし、アイコンおよびマウスにより操作することとしており画像データの取り扱いを容易にしている。このため、デスクトップパブリッシング（DTP）の編集作業やコンピュータグラフィックスなどに利用されている。ハイパーメディア対応ツールとしては、HYPER CARDを同社がサポートしている。

Apple Computer社のMacintoshシリーズ

〔発表時期〕	〔機 種〕	〔形 状〕
1984年 1 月	Macintosh	デスクトップ
85年 3 月	Macintosh512	デスクトップ
86年 1 月	Macintosh Plus	デスクトップ
87年 3 月	Macintosh SE	デスクトップ
87年 3 月	Macintosh II	デスクトップ
88年 9 月	Macintosh IIfx	デスクトップ
89年 1 月	Macintosh SE/30	デスクトップ
89年 3 月	Macintosh IIfx	デスクトップ
89年 9 月	Macintosh Portable	デスクトップ
89年 9 月	Macintosh IIfx	デスクトップ
90年 3 月	Macintosh IIfx	デスクトップ
90年10月	Macintosh Classic	デスクトップ
90年10月	Macintosh LC	デスクトップ
90年 3 月	Macintosh IIfx	デスクトップ
91年10月	Macintosh Classic II	デスクトップ

91年10月	Macintosh PowerBook100	ノートブック
91年10月	Macintosh PowerBook140	ノートブック
91年10月	Macintosh PowerBook170	ノートブック
91年10月	Macintosh Quadra700	デスクトップ
91年10月	Macintosh Quadra900	スタンドアロン

2) NeXT Computer社のNeXTシリーズ

NeXTシリーズは、マルチタスク、マルチユーザーを目的としたワークステーションで、次表の機種が販売されている。

NeXT Computer社のNeXTシリーズ

〔発表時期〕	〔機 種〕	〔形 状〕
1991年 4 月	NeXT station	デスクトップ
91年 4 月	NeXT station Color	デスクトップ
91年 4 月	NeXT cube	デスクトップ
91年 4 月	NeXT dimention	デスクトップ

NeXTコンピュータは、Macintoshコンピュータの設計者でもあるスティーブジョブスが、ソフトウェアの開発者や大学等の高等教育用への市場をねらいに設計したマシンである。CPUは、モトローラMC68030、MC68040を採用。

OSは、UNIX43BSDと互換性をもつMachを採用して、マルチタスク、マルチユーザーを実現している。

3) 富士通(株)のFM/Townsシリーズ

富士通は、パソコンによるマルチメディア化を行うことにより家庭や学校等の市場確保を目的にして、FM/Townsシリーズを開発した。FM/Townsシリーズの機種は、次表に示すとおりである。

富士通(株)のFM/Townsシリーズ

〔発表時期〕	〔機 種〕	〔形 状〕
1989年 2 月	FM Towns 1 F	デスクトップ
89年 2 月	FM Towns 2 F	デスクトップ

89年11月 FM Towns 1 H ディスクトップ

89年11月 FM Towns 2 H ディスクトップ

FM Townsのモデル1 Fおよび2 Fはフロッピーディスクを、また1 Hおよび2 Hはハードディスクを搭載している。開発方針は、GUIを基本にして初心者が容易に操作できること、マルチメディア機能を装備して豊富な表現力を提供すること、CPUに32ビットを採用し、CDドライブを標準にして、長期的に継続できる基本アーキテクチャを採用することとしている。

OSは、FM/Towns-OSとマイクロソフト社のMS-DOS (TM 版) の2種をサポートしている。FM/Towns-OSは、MS-DOS (TM) にDOS拡張機能である386DOS-EXTENDED (TM) を組み合わせている。このOSでは、グラフィックユーザーインターフェース (GUI) でアプリケーションを起動するTowns MENUを用意している。

マルチメディア指向のプログラミング環境のためにTowns GEARをサポートしている。

2. マルチメディア対応パソコンOS

パーソナルコンピュータのオペレーティングシステム (以下パソコンOS) にマルチメディア対応を標準機能とした製品が登場している。アップル社のQuick-TIME、米国マイクロソフト社のMULTIMEDIA-WINDOWS、富士通のTowns-OSである。

これまでのパソコンOSは、汎用を目的として、マルチメディアごとに個別に対応していたが、標準機能とすることによって、マルチメディアのアプリケーションの開発をより効率的に容易に行うことができるようにしている。

3. マルチメディア対応周辺機器

周辺機器としては、パソコンの標準機器のほかに接続音声関係 (マイクロフォン、ヘッドフォン等)、AV関係 (CD-ROM、ビデオディスク、VTR、テレビジョン) などである。

4. マルチメディア対応ソフトウェア

ソフトウェアは、マルチメディア対応パソコンOSのほかに様々なマルチメディアアプリケーションソフトウェアが製品化されている。

1.2.2 ハイパーメディアの応用

ハイパーメディアの応用領域は、情報メタファとして、人間の様々の知的活動領域を考えることができる。現在、応用システムとして研究開発されているものの事例は、つぎのとおりである。

- ・ DB……電子美術館、歯科患者データベース、電子図書館、マルチメディア日本史辞典、国文学マルチメディアデータベース
- ・ 計画……地図情報システム
- ・ TV 会議……マルチメディア分散在席会議システム
- ・ 教育……聴覚障害児のためのマルチメディア CAI
- ・ 電子出版……電子美術館
- ・ 案内……商品案内、ドライブコース案内、マンダラの見方（美術観賞案内）

1. ハイパーメディアの応用事例〔1〕

〔出典または参考資料：電総研ニュース No.496-1991〕

システム名：電子美術館——ART MUSEUM

組織名：通商産業省工業技術院 電子技術総合研究所

システム概要：

1) 次世代のマルチメディアデータベースとマルチメディア対話概念とその応用のパイロットシステムとして研究

2) 研究の目的

(1)画像理解などのマルチメディア処理を組み込んだマルチメディアデータベースモデルの確立

(2)画像や言葉を媒介とするマルチメディア対話方式の開発

(3)マルチメディアデータベースの構築とマルチメディア情報システムへの応用

3) 応用システム——ART MUSEUM——の機能

(1)利用者が描いたラフな手書きのスケッチをキーとしてオリジナルの絵画や類似の構図の絵画を検索する。

〔例示画検索機能（「百聞不如一見」型検索）〕

(2)各利用者の感性モデルにもとづいて、利用者が表す主観的な印象語に相応しい色づかいの絵画を推定する。

〔感性検索機能（「聞一以知十」型検索）〕

2. ハイパーメディアの応用事例〔2〕

〔出典または参考資料：歯科医療情報システム研究大会論文集——1991〕

システム名：患者情報のマルチメディアデータベース化と情報検索

組織名：大阪大学歯学部

システム概要：

1) 患者のあらゆる情報、すなわち種々の所見データ、画像データ、分析データなどの統合を図り、マルチメディアデータベースを構築し、診療支援、診断支援機能の充実を図る。

2) システム設計の基本理念

(1)データの入力を簡単にし、発生源入力方式を基本とすること。

(2)データの転送、照会が臨床における適切な時間内で行えること。

(3)従来の文字情報に加えて、画像データ、分析データなどを統合したデータベース、すなわちマルチメディアデータベースを構築し、これの有効利用が可能であること。

(4)データベース化された情報の検索が迅速に行えること。

3) 患者情報

(1)基本情報……病院カルテ番号、矯正科ファイル番号、患者氏名、生年月日、電話番号、住所、初診日、初診医、受付日、受付医、診療種別、不正咬合分類、紹介者、備考、担当医名。

(2)電子カルテ内データ……診療記録、問診情報、身長データ、歯列表及び咬合状態などに関する所見、診断支援のための各種分析結果、画像データ。

(3)画像データ……顔面、口腔内写真、各種X線写真（セファロ、パノラマX線写真、手根骨X線写真、咬合X線写真、顎関節X線写真等）、セファロの透写図。

注：セファロは患者の口腔模型や側面位頭部X線規格写真をいう。

4) システム構成

ホストは32ビット汎用コンピュータ ACos3300モデル 6、端末PC-9801UX、外部記憶は光ディスク装置、磁気ディスク装置、磁気テープ装置。

画像関係は、CCSカメラ Fotovix II、イメージスキャナー VS-1、画像処理装置 VP-150 wo25台の端末に接続。

5) 画像データの処理

画像データの処理は、CCDカメラで読み取り、そのデータは、A/DカードでRGB各8ビットのデジタルデータに変換し、圧縮処理のうえ VP-150のメモリーに書き込まれる。

圧縮処理は、非可逆的圧縮方式で1/4～1/6に圧縮。

6) データベース管理システムは、RIQS (Relational Information Query System)、データ検索はエンドユーザー言語 TQF (Table Oriented Query Facility)。

3. ハイパーメディアの応用事例〔3〕

〔出典または参考資料：NEC技報 V44 No.6-1991〕

システム名：マルチメディア分散在席会議システム——MERMAID

組織名：日本電気(株)

システム概要：

1) マルチメディア分散在席会議システム MERMAID (Multimedia Environment for Remote Multiple-Attendee Interactive Decision-Making) は、地理的に離れた複数地点にいる人達がそれぞれの関についたままワークステーション (WS) を利用して、会議をはじめとする様々なグループ協同作業を行うことができるシステム ISDN や LAN、専用線を結合したネットワークを利用して、文学、図形、イメージ、手書きから構成されるマルチメディア文書に加え、動画と音声をもリアルタイムに同報し、参加者全員で共有・処理することができる。

2) MERMAID は実用的にも有効性が実証され、すでに遠隔7地点、20数台のWSを接続し、簡単な打合せから技術討議、ソフトウェアの協同開発、ホストコンピュータへのリモートアクセス機能を利用した分散シミュレーションなどの研究開発支援などに2年近くの間実用に供している。

3) グループ通信アーキテクチャ (GCA)

MERMAIDでは、将来のB-ISDNを含む多様な通信ネットワークや分散コンピューティングなどのコンピュータ環境に対しても拡張可能にするための設計概念GCAを提案。GCAの要求条件は、つぎのとおり。

- (1)マルチメディア情報の効率的な同報通信
- (2)グループ調整機能
- (3)情報処理機能との連携
- (4)多様なネットワーク (ISDN、衛星、IVDLANを含むLAN等：IVDは音声データ統合型)

4) システムの特長

- (1)GCAにもとづく広域多者会議の支援
- (2)音声と動画の交換、活用
- (3)マルチメディア文書のリアルタイム編集、保存、検索、再利用、配布
- (4)共有画面への書き込み権に関する移動・調整機能制御
- (5)マルチウインドウ、メニューマウス操作を基本とするビジュアルで簡易な利用者インタフェース
- (6)共有画面と個人画面の同時表示とその間の情報の受け渡し

5) システム構成

- (1)ワークステーションはEWS4800 (CISC、RISC共)
- (2)通信ネットワークは、ISDN、IVDLANを含む各種LAN、衛星回線
- (3)高速回線も利用可能。
- (4)マルチメディア入出力としては、キーボード、マウスのほか、多者間音声。
- (5)通信インターフェース、イメージスキャナ、手書用タブレット、ビデオカメラ、動画像コーデックス
- (6)ソフトウェアとしては、OSはUNIX、マルチウインドシステムはXWindow、ユーザーインタフェース構築支援ツールは鼎。低位通信プロトコルはTCP/IP、高位はマルチメディア文書通信はISO/ODA

4. ハイパーメディアの応用事例〔4〕

〔出典または参考資料：画像電子学会年次大会予稿——1991〕

システム名：画像データベース——電子図書館構築のために——

組織名：テレマティーク国際研究所

システム概要：

1) 電子図書館のための画像データベース技術についてまとめている。

2) 入力技術

(1)ビデオ、マイクロフィルム、スキャナーからの入力考えられる。マイクロフィルムは図書館の利用者が多いが、フィルム分解能が200dpiで精度が低く、雑音が多いのが問題。

(2)画像精度としては、漢字の場合最低400dpi以上必要。

(3)画像情報量圧縮技術では、とくに文字と画像が混在する図書の場合が問題文字画像を認識し、コード情報化が考えられる。

3) 画像データベースは、市販のDBMSを利用。

A	B	画像	C
---	---	----	---

キーワードはBの部分におく。

画像データを管理するデータは、別のディスクに持つことによって検索時間の短縮を図る。光ディスク等膨大なデータファイルの場合は画像データ自身に持たすことも可能。

5. ハイパーメディアの応用事例〔5〕

〔出典または参考資料：電子情報通信学会技術研究報告——1991〕

システム名：マルチメディアを用いた日本史ハイパー辞典

組織名：上智大学理工学部

システム概要：

1) 歴史辞典にハイパーテキスト概念を導入。ノードは歴史上の事件、文化、人名、歴史用語、さらにその中に含まれる事件名、文化名、人名、歴史用語を参照点とし、相互に関連するそれぞれのノードへとつながっている。

ノードは文字のほか、静止画、音声出力も可能。

2) 検索ノードの出力画面では、属性をもった歴史用語、人名等には視覚的に判断でき、属性を知ることができるように、文字色の違いで表示することとした。

3) 検索方法としては、年表検索、人名検索、歴史用語検索、家系図検索などの属性検索が出来る。このほか、利用者が求める語句を直接入力して検索する方法ノードからノードへと履歴をたどりながらサーチする検索する方法も可能。

4) 補助機能として、歴史的地図と現在の地図とを比較するようにしている。

5) 使用コンピュータはPC9801

6. ハイパーメディアの応用事例〔6〕

〔出典または参考資料：人口知能学会研究会資料——1991〕

システム名：都市計画支援システムのハイパーメディア化

組織名：(株)東芝 総合研究所

システム概要：

1) 都市計画支援システムとして、従来の地図情報システムを越えたハイパーメディア化による地図情報の体系化を図ることとした。

2) マルチメディアデータとして、画像データ、図形データ、属性データ（文字記号、数値データ）および音情報。

3) ベースとなる地図情報

(1) 1/2500の都市計画基本図を用い、住宅地図（1/1500）、住居表示台帳図（1/500）、地下埋設管理台帳図（1/1000）など縮尺の異なる地図や航空写真（カラー）などと組み合わせて画像データとして扱う。

(2) 図形レイヤ情報

地図情報を図形化して計測や解析に利用する。図形化には点要素化、線要素化、面要素化の3種があり、図形要素の内容をまとめレイヤごとに（例えば、建屋図形レイヤ、土地利用図形レイヤ）定義する。

(3) 属性としてのマルチメディア情報

地図上の各図形要素にその属性がある数値や文字情報を関係づける。そして、静止画（例：建物）、動画（例：日影の時間変化）、図形データ（例：建物の設計図面）、音情報（例：騒音の録音データ）などをリンクして、マルチメディア情報化する。

d. 解析・シミュレーション情報、コンピュータグラフィックス画像。

4) このシステムは、東京都新宿区を対象に実データにより試作。

属性データは、建屋（構造、階級、用途）、土地利用は種別（道路管理種別）、用途地域、リレーショナル型データベース化した。画像データは、住居表示台帳図、道路台帳図、地下埋設管台帳図、地質地盤資料、住宅地図、用途容積地域地区図、航空写真、その他、合計IGバイト

5) データ格納媒体は、図形及び属性データは磁気ディスク、画像データは光ディスクを用いている。

7. ハイパーメディアの応用事例〔7〕

〔出典または参考資料：人文科学とコンピュータ——1991〕

システム名： マルチメディア作品——オルセー美術館——

組織名： パイオニアLDC(株)

システム概要：

1) フランスODA社が制作したレーザーディスク（LD）とパイオニアLDC社が制作したスタッフウェアで構成。

2) LDには、60分の美術映画（動画）、絵画・工芸品・写真などオルセー美術館所蔵作品2,100点を11,000枚の静止画ファイルで収録。

3) スタッフウェアは、収録作品を多角的に検索するインデックス機能を担う。マッキントッシュのハイパーカードで作成。

4) このマルチメディアソフトは、LDとパーソナルコンピュータを組み合わせたインタラクティブマルチメディアシステムによって利用できる。

5) パーソナルコンピュータ（Macintosh）LDプレーヤー（LD-V530またはLD-V800）ソフトウェアは漢字TALK60X、日本語ハイパーカード1,2X、LD用ドライバーソフト、LDソフト。

8. ハイパーメディアの応用事例〔8〕

〔出典または参考資料：情報の科学と技術 41巻1号—1991〕

システム名： 国文学におけるマルチメディアデータベース

組織名： 国文学研究資料館

システム概要：

1) 国文学研究資料館は、国文学に関する文献資料を調査研究し、収集、整理、保存し、これを広く研究者一般の利用に供して、国文学研究の進展に寄与することを目的として、設立された大学共同利用機関。

2) 同館では、国文学研究支援システムの開発および、国文学マルチメディアデータベースの構築を計画している。

3) 国文学研究支援システムは、国文学研究のため情報システムデータベースの多角的利用、さらに全国的な情報流通を図ることを目的としている。

4) 国文学データベース (注*:計画中または開発中)

〔形態〕 〔データベース〕

0 次情報 原文献資料データベース

1 次情報 国文学作品本文データベース

(1)語牽引データベース

(2)本文データベース (本体)

(3)書誌データベース*

(4)ユーティリティデータベース*

(5)注釈データベース*

2 次情報 文献資料目録データベース

(1)古典籍総合目録データベース

(2)所蔵原本目録データベース

(3)マイクロ資料目録データベース

(4)和古書目録データベース

研究情報データベース

(1)逐次刊行物目録データベース

(2)論文目録データベース

(3)文字セットデータベース

(4)用語データベース*

(5)著者、書名典拠データベース*

1.3 知的ハイパーメディアのニーズ

ハイパーメディアの知的化、インテリジェント化のニーズとしては、次の知的機能が考えられる。

1.3.1 メディアオブジェクトの知的化

ハイパーメディアにおけるメディアオブジェクトを全てダイナミックオブジェクトと定義し、内容の提示のみならず、その編集機能、動作機能、対話機能、プログラム機能などを具備するとともに、これらの機能におけるAI手法の応用を図る。さらにメディアオブジェクトの部品化にもとづくメディアオブジェクトの直接合成機能も含まれる。

1.3.2 リンクの知的化

問題に応じた情報について特定の条件を満たす探索経路の推論、ルール記述によるリンクの定義とその推論による経路探索、状況に応じた結合先の変更、結合の強度やあいまい度の導入と推論、利用頻度による結合の強度の適応などの拡張機能が含まれる。

1.3.3 文章、画像、音声等作成の知的化

文章、画像、音声などのパターン認識による入力と幸造解析および組織化、文章等作成の知的支援、自動抄録の作成、アウトライン作成、アイディア作成など

1.3.4 文章、画像、音声等探索の知的化

情報の探索支援、推論による探索、探索学習、類似性にもとづく推論、連想推論など。

1.3.5 文章、画像、音声等利用の知的化

自然言語理解、機械翻訳、チューニング機能、利用者スキルやメディアの種類等の利用状況に応じた環境の設定（変更機能を含む）など。

また、知識情報システムとの結合では、エキスパートシステムとの統合が考えられる。これには、ハイパーメディアを既存のエキスパートシステムのフロントエンドとして

用いる方法とハイパーメディアにルールやファクトを記述するメディアオブジェクトを導入するとともに、推論機能をもたせる方法がある。実現技術としては、ハイパーメディアと論理プログラムの結合、ハイパーメディアとプロダクションルールシステムとの結合、ハイパーメディアと演繹DBまたはオブジェクト指向DBとの統合などの拡張が考えられる。

1.4 知的ハイパーメディアのもたらす効果

知的ハイパーメディアは、知的活動の強力なツールとして位置付けられる。この効果は計り知れないものがある。

ここでは、1つは個人生活、社会、ビジネス、学術などの諸分野における豊富な情報の提供と利用とによって、人類の幸福を指向した情報化社会の到来が予想できることである。さらには、新しい文明や文化の創設に発展していくことになる考える。

知的活動では、資料や情報を探し内容を分析し解釈をするといった「調べる」ということがある。ハイパーメディアでは、資料や情報の項目を「ノード」としておき、「ノード」と「ノード」は「リンク」で接続できるようにする。この機能をコンピュータの画面を見ながら利用することによって必要とする適切な情報を捜し出すことができる。これも利用者個人の経験や知識を予め入力しておくことによって、纏めようとする内容に応じた情報を連想しながら探すことができる。勿論、途中で判らない用語があると辞書を引くこともできる。どのように資料を作るかは思うがままに、文章化し図表化しビデオに撮り音声で説明することも可能となるのである。2. 2の応用で例示したように教育分野でも美術館や博物館などの社会活動、医療分野でもハイパーメディアの利用領域は広がっていく。そして、正にA. トフラーが予測した「第3の波」のように、ハイパーメディアの恩恵を受けて人々が真の情報化を享受することになるであろう。

もう1つは、既存産業のブラッシュアップによる活性化、付加価値化、新技術、新製品をベースとしたベンチャビジネスの登場が期待され、雇用機会の増大、産業、経済の発展につながっていくと予想される。

ハイパーメディアは、コンピュータとソフトウェアによって既存のメディアに付加価値を図るものである。印刷、出版、新聞、テレビ放送といったメディア産業は、新しい技術を利用したメディアを世間に送り出すことができる。メディア産業以外の他の産業においても新製品開発や新サービスにハイパーメディアを活用するであろう。とくに、期待されるのは、ハイパーメディアの応用技術やアイデアを持ったベンチャービジネスの登場活躍の場面が増加することである。個人の知的活動が拡大することが予想されるなかで、これを支援し小回りの効いたベンチャービジネスが求められるであろう。

1.5 新しい情報メディアとしての知的ハイパーメディア

知的ハイパーメディアは、人間の知的活動における情報メタファとして、また新しい情報メディアとして考える。今後、ハイパーメディアのツールが様々な商品として登場し利用されるとともに、ハイパーメディアに求める機能もますます高度になっていく。そのときに、人工知能（AI）技術を活用して、ツールの機能を増強したり、アプリケーションの機能を強化することになろう。創作支援など下記の各段階でハイパーメディアのツールと結合させたエキスパートシステムを開発することや、著作物にまつわる知識ベースを利用することなどである。

- ・ものごとを考える段階……………計画立案、著作物の枠組み、構成要素、内容展開、でき上りイメージ
- ・著作物の制作段階……………作文、作曲、計画、およびその出力（考えた枠組みにもとづき）
- ・行動段階……………基本計画にもとづく行動計画、行動指示、行動評価

これらの段階の情報メタファとして、情報の収集、加工、利用において、一貫して管理することができる。

1.6 今後の動向

現在のところマルチメディア、ハイパーメディアは、緒についたところで、今後の動

向に注目する必要がある。

1.6.1 マルチメディアおよびハイパーメディア対応の製品化が進む。

とくに、マルチメディアを標準機能にもパソコンおよびパソコンOS、マルチメディア対応AV機器。

1.6.2 マルチメディア関連の標準化

昨年（1991年）11月にISO（国際標準化機構）は、MPGE（Moving Picture Coding Experts Group）の作成した、動画（カラーテレビジョン信号）とオーディオ信号についての圧縮符号化方式とデータフォーマットの国際標準案を発表した。原案は、テレビジョン信号とオーディオ信号を1.5Mビット/秒に圧縮する方式とMPEGIデータフォーマットから構成されている。

この標準案に合わせて、動画を圧縮/伸長する専用チップが米国のIntegrated Infomation Technology社から「Vision Processor」が、また、ソフトウェアが米国のXing Technology社から「VT-Monition Scalable MPEG Software」が販売された。

1.6.3 マルチメディア、ハイパーメディアにおける知的所有権問題

人間がハイパーメディアを使用して様々の知的活動を行う時、現行の法制度で問題になるのが知的所有権問題であるが、とくに著作権法の取り扱いを考えておく必要がある。

- ・コンピュータに蓄積されたマルチメディアDBをネットワークを介して入手し、それを創作構想に応じて自由に加工し新しく作品を作り上げる。情報メディアの入手、その加工程度（同一性保持との問題）、新作品の権利など。

- ・著作権法で認めている個人使用の適用除外の問題。レンタルビデオのコピーが問題を提起しているように、ハイパーメディアの時代では個人の使用が想像できない範囲で拡がることが予想される。利用を阻害しない形での対応が必要となろう。

- ・社会性のあるマルチメディアDBについては、公共財として、図書館や情報センター等の公共機関を介して利用容易にする方式、利用料の無料あるいは低価格でサービスする方策が求められる。

引用参考文献

- *〔高橋——91〕：高橋三雄 監修「わかりやすいコンピュータ用語辞典」ナツメ社
- *〔Woodhead——90〕：Nigel Woodhead「Hyper text&Hyper media——Theory and Applications」SIGMA PRESS
- *〔Berk——91〕：Emily Berk and Joseph Devlin「Hypertext／Hypermedia お Handbook」McGraw-Hill Publishing Co.,Inc.
- *〔津久井——89〕：津久井 凌「知識メディアの潮流（Turing Machine 1989.8）」(株)スペック
- *〔奥地——89〕：奥地耕司「ハイパーメディアとESとの統合（Turing Machine 1989.8）」(株)スペック
- *〔寺野——89〕：寺野隆雄 他「マルチメディア知識システムへの試み（Turing Machine 1989.8）」(株)スペック
- *〔富士通——90〕：富士通「特集FM Towns（富士通ジャーナル1990.7）」
- *〔日経——92〕：日経データプロ・EDP 1992.1
- *〔Sculley——87〕：J.Sculley&J.Byrne「ODYSSEY（日本語訳 スカリー会津泉訳 早川書房）」
- *〔JIPDEC——91〕：(財)日本情報処理開発協会「知的ハイパーテキストに関する調査研究報告書 1991.4)」
- *〔Latta——91〕：John N.Latta「Databases for Multimedia and Virtual Reality for the 21st Century」
- *〔原——91〕：原良憲「ハイパーメディア入門（bit vol.22, No.8）」

2章 知的ハイパーメディアに関する技術

2.1 知的ハイパーメディアの要素技術

2.1.1 序

近年、情報化社会の進展に伴い、ワードプロセッサやパーソナルコンピュータ、ワークステーション等が広く普及してきている。このため、生成される文書情報も急速に増加してきており、近い将来膨大な量に達するものと予想されている。知的ハイパーメディアにおいては、このような膨大な情報の中から、いかに的確にかつ迅速に所望の情報を探し出すかが、個人の知的な創造活動を支援するためにも、また企業の戦略を決定する上でも重要な課題となっている。こうした状況下で、これまでのインデックス方式に基づく情報検索システムでは、登録文書数が増大するとインデキシング作業が膨大になったり、キーワードの選択に専門的な知識が要求されるため一般のユーザでは操作が困難になってきている。また、文書データベースが大規模化してくると十分な絞り込みができなくなるということや、技術文献データベースでの技術用語の目まぐるしい変遷に起因する検索精度の低下なども大きな問題となっている。

こうした問題に 대응するものとして、インデックス情報を用いない自由な言葉による検索を目的としたフルテキストサーチ（自由語全文検索）技術の研究が行われた。

このインデックスを不要とするフルテキストサーチを実現するためには、テキストデータを蓄積ファイルから高速に読み出し、この中から所望の言葉を高速に探し出す技術が開発された。また、実用規模のデータベースで現実的な検索時間を実現するためには、更に検索速度を等価的に高める検索加速技術が開発されている。

情報の戦略的価値とともに得られる情報量が増大する一方、情報の種類も多様化しているため、マルチメディア情報のファイリングシステムが求められている。特に、情報価値が高い学術論文や技術雑誌などは、テキストのみならずカラー写真や図表などが重要な情報を担っており、これら異種媒体の混在した文書、即ち、マルチメディア文書を効率的に蓄積し、かつ検索しやすいデータ構造に変換するマルチメディア構造化技術が求められている。この技術を用いれば、文書情報を最小限の人間の介在で検索できる形で光ディスクなどの大容量記憶システムに入力することが可能となる。

カラー写真などを含むマルチメディア文書画像は、テキスト情報を再現できるよう高解像度を要求する。そのため、画像データは膨大なデータ量となり、従ってテキストや写真の画像としての特質に着目して、それぞれ最適なデータ表現とデータ圧縮をすることにより、記憶するために必要な情報量を大幅に圧縮する方法が考えられる。このためには、文書画像の種類に応じて文字、写真、図表など個々の領域を構成要素として自動的に分離するところのマルチメディア分離抽出技術が不可欠である。更に、マルチメディア文書画像の高度な検索のためには、文書画像の入力に際して検索情報の自動抽出が必要である。そのため、文書画像から表題ページを識別して、そこから表題や著者名などの書誌的事項を自動抽出して文字認識することや、本文テキストの文字認識を行ってインデキシングをすること、テキストの章や節の単位を同定すること、図表や数式の所在を抽出すること、あるいは、これら抽出した要素間の相互関係を抽出することなどを行う文書構造抽出機能が求められている。

2.1.2 マルチメディア内容検索のシステム概念

1. システムの概要

マルチメディア内容検索システムは、高解像度のフルカラーイメージスキャナ、文書画像を解析し構造化するプロセッサ、大容量光ディスク装置、カラープリンタ、高性能ワークステーション等からなる。カラー文書は、白黒の文書より、情報の強調、識別、正確さ等の点で優れており、カラー情報が取り扱えることは、将来の電子化文書ファイリングシステムでは不可欠である。

このシステムのユーザは、イメージスキャナを用いて紙の文書の表面情報を入力し、のちほど所望の情報を容易に検索することができるよう、それを光ディスクに格納する。システムは、書誌事項だけでなく原文書情報を格納しているので、ユーザは必要な情報を直接に検索することができる。例えば、断片的な記述をユーザが入力することにより、所望の写真領域を検索し、それを高精細ディスプレイを介して見ることができる。更に、例えば、その写真を説明しているテキスト領域をブラウザのコマンドにより検索し、該当するテキストを流し読むことができる。

2. マルチメディア内容検索の課題

このようなマルチメディア内容検索システムを実現するためには、次のような課題を解決する必要がある。まず、第一の課題として、高効率の画像圧縮方法が必須である。なぜなら、フルカラーイメージスキャナで採取した画像のデータ量は膨大であり、また、2値画像であるテキストに対して高解像度を保証し、かつ、カラー写真に対して高階調度を維持することができるような効率的な唯一の圧縮符号化方式が必要である。次に、第二の課題としては、柔軟な内容検索を支えるために、書誌事項のみならずインデックス情報を自動的に抽出しなければならない。なぜならば、この柔軟な内容検索では内容をアクセスするためのポイント、即ち、インデックスを多数登録しておく必要があるためである。更に、第三の課題としては、知的な文書ブラウジングを実現するために、システムは文書の論理構造、即ち、本文テキスト、説明文、写真等の文書の構成要素の相互関係を抽出することができなければならない。

3. 文書画像の構造化

カラー写真、カラー文字などを含むマルチメディア文書を、カラースキャナを用いて画像データとしてコンピュータに入力する。カラースキャナで、赤、緑、青色の三原色成分データを採取する。次いで、文書の書式に関する知識を基に、文書をその構造要素に分離、抽出する。ここでは、文書から書誌事項を抽出するとともに、レイアウト構造、論理構造を抽出し、構成要素であるカラー写真、図形等を分離する。そして、これら構造化した文書データをマルチメディアファイルに格納する。このマルチメディアファイルを用いて、文書の内容に基づく検索を行うことができる。

この文書画像の構造化方式は、文書の書式知識や色情報の知識ベースを用い、カラー文書画像データ入力、カラー情報分離、論理構造抽出、文字認識、などの基本処理が実行されることになる。これらの処理の内、カラー情報分離では、三原色成分の色彩情報を用いた領域分離を行うとともに、領域の色識別を行う。また、論理構造抽出処理では、表題ページにある表題領域、著者名領域、要旨領域、本文領域、並びに参考文献領域を抽出する。これらの領域の一般的な位置関係は文書書式として、通常定ま

っているものであり、この概略の文書書式を構造化のための制約条件として用いることができる。

2.1.3 マルチメディア技術

1. CD-ROM

1) CD-ROM

CD-ROMはデジタルオーディオデータを記録する目的で開発されたCD (Compact Disc) に一般のデジタルデータを記録したものである。物理フォーマットはCDのものをそのまま使っており、約540MBの容量がある。

CD-ROM規格では論理ファイルフォーマットは定められていない。しかし、ハイシエラフォーマットなどの標準化の動きがあり、ISO9660標準ができています。また、CD-ROM規格では、テキストや画像などのコンピュータデータのほかに、CD-DAトラックにCDと同じフォーマットの音声を置くことができます。

2) CD-I

CD-I規格は、一枚のCD-ROMに画像・音声・動画・プログラムを記録するシステムの規格で、100%の互換性を確保するためにCD-Iプレーヤのハードウェアも規定している。主な仕様は以下のようである。

(1)CPUはMC-68000系

(2)OSはOS/9を基本としたCDRTOS

(3)画面解像度は、360×240、720×240、720×480

(4)オーディオレベルは、CD-DA、Level-A、Level-B、Level-C

オーディオについてはCD-DAの他に、次のような圧縮した音声データをサポートしている。

CD-DA 16bit PCM	44.1kHz stereo	75分
Level-A 8 bit ADPCM	37.8kHz stereo / mono	150分 / 300分
Level-B 4 bit ADPCM	37.8kHz stereo / mono	300分 / 600分
Level-C 4 bit ADPCM	18.9kHz stereo / mono	600分 / 1200分

この音声データはADPCMインターリーブドオーディオと呼ばれ、テキストや画像などのデータと交互に置かれるため、音声を出しながら画像データを取り出して表示することができる。一つの画像データに対して複数の音声を用意しておくこともでき、たとえば多国語を切り換えて使うようなことができる。

3) CD-ROM XA

CD-ROM XA規格では、CD-I規格のようにハードウェアを規定することはしない。オーディオについては、CD-I規格のLevel-B、Level-Cのインターリーブドオーディオをサポートしている。CD-ROM XA規格ではハードウェアもOSも規定していないので、UNIXでもMS-DOSでもこのフォーマットを使うことができる。CD-I規格ではディスプレイのハードウェアも規定しているが、CD-ROM XA規格では標準を640×480ドット256色としている。論理フォーマットは、ISO 9660標準に準拠している。

4) LD-ROM

CD-ROMがデジタルオーディオを記録するために開発されたCD（コンパクトディスク）をもとにしたのに対して、LD-ROMはアナログビデオ信号を記録するためのLD（レーザーディスク）をもとにしている。データの論理フォーマットはCD-ROM XA規格と同じで、ADPCM音声も記録できる。

2. インタラクティブ・ビデオ

インタラクティブ・ビデオは、ユーザの応答または指示によってビデオの再生場面が

変わるようなシステムである。基本的には、ビデオプレーヤとそれを操作するコンピュータから成る。コンピュータはビデオが再生している場面を把握できるので、それに合わせてコンピュータ画面を変え、ユーザの入力を受け付ける。逆に、自分が持っているストーリーに合わせて再生場面を変えこともできる。CAI、プレゼンテーションなどに用いられる。

1) ビデオ

インタラクティブ・ビデオのためのコンピュータ・プログラムはビデオとは別のメディアで用意されることが多い。たとえば、ビデオ用のレーザーディスクとプログラム用のフロッピーディスクの組で、ひとつのインタラクティブ・ビデオ・ソフトウェアとする。こうすれば、コンピュータもレーザーディスクプレーヤも既存のものに近いもので済む。しかし、異なるメディアを組にして扱うのは面倒であるとか、プログラムが大きくなってくるとフロッピーディスクの数が増える、と言う欠点がある。このため、ビデオとプログラムを一つのメディアに記憶するシステムもある。

インタラクティブ・ビデオで使われるビデオプレーヤは、コンピュータからの操作で目的のシーンを素早くアクセスできることが要求される。現在使われているビデオメディアは大きく分けて次の3つがある。

(1)ビデオカセットテープ（ベータ、VHS、8mmビデオ、Uマチックなど）

家庭用のビデオデッキを利用すればかなり安価であり、ビデオソースを撮影し編集するのも手軽であるが、テープであるためアクセスタイムが非常に長いのが大きな欠点である。コンピュータからの操作という面では、最近の製品ではフレームナンバーやタイムコードなどでの正確な制御が可能になってきている。

(2)ビデオディスク（レーザーディスク、CD-V、追記型ビデオディスクなど）

アクセスタイムは十分に短い製品もある。コンピュータからの操作という面でもフレーム単位で制御できる。画質はとても良い。レーザーディスクやCD-Vなどは大量生産には向くが、少量ではディスクをスタンプするのにお金と時間がかかるのが欠点である。追記型ビデオディスクはまだ高価で一般的ではないが、一枚ずつ書き込みができるので少量でも制作できる。

(3) デジタルディスク (DVI、CD-I、CD-ROM、XA など)

ビデオ信号をデジタル化して、磁気ディスクやCD-ROMなどのランダムアクセスメモリに記録する。ビデオ信号をデジタル化するとデータの量が膨大になるので圧縮する。デジタルデータであるためコンピュータとの相性はよいが、圧縮しているために画質にまだ少々問題がある。

2) コンピュータ

インタラクティブ・ビデオでは、ビデオ画面とコンピュータ画面とは同一のディスプレイに重ね合わされて表示されるシステムが多い。これをスーパーインポーズ機能と呼ぶ。この機能があれば、ユーザは一つのディスプレイを見ていればよいし、ビデオ画像にコンピュータが絵を付け加えて、ソフトをよりインタラクティブにすることができるなどの利点がある。しかし必ずそうでなければならないというわけではなく、ビデオとコンピュータが別にディスプレイに表示されるシステムもある。

スーパーインポーズするディスプレイには大きく分けて次の二通りがある。

(1) ビデオ用ディスプレイに表示する。

ビデオがディスプレイ全体に表示される。コンピュータの表示はその上に乗るが、解像度はビデオに合わせることになるので余り上げることはできない。パーソナルコンピュータ以下では一般にこの方式が取られる。

(2) コンピュータ用ディスプレイのウィンドウの中にビデオを表示する。

ビデオはディスプレイ中の一つのウィンドウの中に表示される。解像度や画面構成が自由になる。ワークステーション以上ではこの方式が取られることが多い。

現在のインタラクティブ・ビデオでは、要求される機能にもよるが、コンピュータに大きな処理能力は要求されていないシステムが多い。極端な例では、ビデオプレーヤ内蔵のマイクロプロセッサとコマンドだけでもインタラクティブ・ビデオのようなことが実現できる。しかし、ハイパーテキストやネットワークなどのために大きな処理能力を使うようなシステムが現われている。

3) インタラクティブ・ビデオ・ソフトウェア

インタラクティブ・ビデオのソフトウェアを作るには、オーサリングツール、テキストエディタ、ペイントツールなどを使う。

インタラクティブ・ビデオのソフトウェアを制作する場合、コンピュータプログラムを作ることの難しさもさることながら、ビデオの撮影と編集に大きな手間がかかる。コンピュータ・プログラムについては優れたオーサリング・ソフトウェアがあれば専門家が不要になる可能性があるが、ビデオ撮影は専門家とそうでない者で出来上がったものの質に大きな差がある。

3. 異種メディア間の関連付け

ハイパーテキストでは、ある文書のある部分からそれに関連する他の文書を簡単に呼び出すためにリンクを作っておく。このリンク付けの対象となるメディアとしては次のものが考えられる。

- 1) テキスト
- 2) 静止画像
- 3) 音声
- 4) 動画像
- 5) プログラム

プログラムもメディアの一種だと言える。あるプログラムは何か計算してその結果を表示するし、あるプログラムはユーザの指示でどこかのデータベースを検索して、その結果を表示する。アクセスしたときにユーザに対して、表示することを含めて何かをすることに変わりはない。こう考えると、メディアとしてはプログラムの方がむしろ一般であると言えるかも知れない。テキストや静止画像や音声や動画像もそのデータ（ビット列）だけではなんの意味もなく、それを表示、再生するプログラムと共にあって始めてメディアなのである。

一般にテキストや静止画像などの静的なメディアからのリンクは簡単だが、音声や動

画像などの動的なメディアからのリンクは難しい。動的なメディアはデータがすぐ巨大になりやすく、静的なメディアと同じに扱うのは問題がある。

異種メディア間の関連付けと言うことでは、もう一つ問題がある。動的なメディアと静的なメディア、あるいは動的なメディア同士を表示する場合の時間的な同期の問題である。これはマルチメディア・システム構築上の問題であるが、音声を出しながらそれにタイミングを合わせて静止画あるいは動画を表示するなど、動的なメディアをきちんと扱う場合には対応しておく必要がある。またCD-ROMで音声データを読み出している間は画像データなどの他のデータを読み出せないとすると、ソフトウェアの表現にかなりの制限になる。実際には、CD-ROM XAなどではADPCMインターリーブオーディオという方法で音声データと他のデータを同時に読みだすことができるようになっている。

2.1.4 構築支援機能

1. オーサリング・ツール

オーサリング (Authoring) とは著作物を作成することを言う。これには、文書を書くこと、絵を描くこと、ハイパーメディアのノードとリンクを作ること、また、その前段階として思考を整理すること、などがある。

1) 著作物の種類から捉えた支援ツール

対象とする著作物の種類でオーサリングツールは異なる。著作物の種類の分け方の視点として以下のものがある。

- (1)内容 (技術資料、教科書、思考整理メモ、百科事典、ガイドブック、小説、など)
- (2)記録メディア (紙、コンパクトディスク、ビデオテープ、ネットワーク上ハードディスク、など)
- (3)個人利用／共同利用
- (4)著者と読者の区別
- (5)文書型／プレゼンテーション型

2) オーサリング過程から捉えた支援ツール

オーサリングをまず便宜的に著者のオーサリングと、読者・視聴者のオーサリングに分ける。著者のオーサリングを関連資料を収集するところから、著作物を作成するまでの過程で捉えれば以下の過程が抽出できる。

- (1)情報収集・整理
- (2)思考・概念整理
- (3)情報提示順序の指定
- (4)データ入力・編集
- (5)リンク付け
- (6)追加プログラミング
- (7)記録メディアへのデータ変換
- (8)記録（印刷、コンパクトディスクのプレス、オンライン・データベース登録、など）

3) オーサリング形態から捉えた支援ツール

オーサリングの形態で考えると、

- (1)個人オーサリング
- (2)共同オーサリング

がある。これもオーサリング過程とのマトリックスで考えられる。共同オーサリングは、例えば概念整理過程における共同作業支援（討議支援）、また思考の文字化の過程における共同作業支援（共同執筆）などがある。

ハイパーメディアのオーサリングツールとしてKMSはネットワークに結合している他の場所にある二つのワークステーションの情報間にリンクを張る機能を提供する。また、討議や執筆過程の相互チェックに利用できる、ハイパーメディアを使った電子メールや掲示板の機能を提供している。

2.1.5 人工知能技術

将来型のハイパーメディアとしての知的ハイパーメディアの実現には、人工知能技術

が要素技術として多くの面で重要な役割を果たす。知識表現・推論技術、自然言語理解技術、音声や文書などのパターン認識技術、知識の獲得および学習技術などを有効に用いることができる。また、技術としてのみではなく、人工知能研究の認知心理学的な側面が、使いやすく解かりやすいマンマシンインタフェースの実現にとって重要である。

1. 知識表現と知的検索

1) 知識の表現技術

知識の表現は知的なシステムの実現にとって基本的な要素である。知識の表現モデルとしては、大きく分けて以下の4つがある。また、あるシステムでは複数のモデルを併用してハイブリッドシステムを構成している。

(1)意味ネットワークモデル

(2)フレームモデル

(3)プロダクションモデル

(4)論理モデル

以下、それぞれについて簡単に解説する。

(1) 意味ネットワークモデル

意味ネットワークの歴史は古く、その基本的な構成はノードとリンクとからなり割合単純である。その単純さと解かりやすさから、主に自然言語処理の中で多く用いられてきた。ノードは物理的な物や抽象的な概念を代表する。この概念ノードは上下関係を表すIS-Aリンクで相互に連結することにより、全体として分類階層木を構成する。この上下関係（IS-A関係）は包摂関係（Subsumption Relation）とも呼ばれる。

また、各ノードには属性を定義することができる。属性の値は別の概念ノードでもよいし文字列でもよく、リンクを介して属性を付けたノードに関係付けられる。例えば、「鳥」が概念ノードとして登録されているとき、「鳥は翼を持つ」という事実は、「鳥」

と「翼」の二つの概念ノードを「持つ」(HAS-PART-OF) という属性リンクで結ぶことにより表現できる。一般に属性リンクは関係リンクとも呼ばれ、HAS-PART-OF という属性 (関係) は特に部分全体関係 (Part-Whole Relation) と呼ばれる。

ノードを縦方向に結ぶIS-Aリンクはいわば縦糸であり、属性あるいは関係を定義するリンクは横糸である。このような構成でネットワーク (網目) を構成するのが意味ネットワークである。

意味ネットワークの一つの機能的特長は、IS-Aリンクを介した属性の継承である。

例えば、「鳥」の下位概念として登録されるすべての概念ノードには、「それは翼を持つ」という属性が継承される。この機構を属性継承と呼ぶ。ある概念ノードをとってみると、複数の上位概念を持たせたいことがある。従って、複数の上位概念とIS-Aリンクで結ぶことを可能にするシステムもある。この複数の上位概念からの属性継承を多重継承という。

このような属性継承機構により上位概念による質問への応答が可能となる。例えば、「翼を持つ動物は何か」という質問に「それは鳥である」と答えることができる。これは一種の推論であり、分類型推論 (Taxonomical Reasoning) と呼ばれる。

意味ネットワークにおける推論は基本的に部分ネットワーク (グラフ) のパターンマッチングである。組み込みの推論機能は、包摂関係と部分全体関係を用いたものである。他の意味の表現はユーザが定義するそれ以外の関係リンクに委ねられている。すなわち、関係リンクに与えられた名前とそれに関連付けられているプログラムにより規定される。従って、各々のシステムを越えた普遍的な意味処理機構はパターンマッチング以上にはない。

意味ネットワークモデルを用いた主な知識表現言語としてはKL-ONEが良く知られており、自然言語理解システムなどで用いられている [Brachman85a]。意味ネットワークによる知識表現とそれによる推論は、単純さと解かりやすさがある半面、機能的には限界がある。そのため、ほかのモデルとの融合も試みられている。例えば、Brachmanは論理モデルを導入したハイブリッド型の知識表現推論システム、KRYPTONを開発した [Brachman85b; Brachman87]。KRYPTONは、ものや概念の集合を静的に記述する知識を格納するTBOXと、それら概念を用いて記述する事実を格納するABOXとを持つ。前者は意味ネットワーク型の知識表現を用い、主に概念の定義を与える。

後者には後に述べる論理モデルによる知識表現を用いて、概念間の関係を蓄積する。後者における推論機構としては1階述語論理を用いた定理証明システムを利用している。これにより推論機能の強化を図っている。

(2) フレームモデル

フレームモデルは、視覚認知モデルのために考案されたフレーム理論 [Minsky75] に基づいた知識の表現形式である。フレームとは、ものを理解するときのいわゆる枠組みである。[もの] は物理的な物でもよいし、抽象的な概念でもよい。フレームにはそのものの名称がフレーム名として付いており、フレームの中にそのものを規定する属性を記述することができる。属性を記述する記述子をスロットといい、スロットはスロット名とフィラーとよばれる値とから構成される。スロット名は属性名を表し、フィラーは属性値を表す。例えば、先の「鳥」は以下のようなフレームで表現することができる。

```
(鳥      (IS-A      生き物)
          (HAS-PART-OF 翼)
          (HAS-PART-OF 足))
```

スロット IS-A は、意味ネットワークの場合と同じように概念の上下関係を表しており、フレームモデルの場合にも、フレームの分類階層木が構成される。属性はこの階層木に添って下位に継承される。

スロットは同フレームの属性を規定するものであるが、ほかのフレームとの関係を記述しているとも考えられる。上記の例で、属性値であるところの「翼」や「足」は別途定義されるフレームを指していてもよい。すなわち、スロット HAS-PART-OF は鳥フレームと翼フレームとの間の関係を規定している。従って、翼フレームにスロット IS-PART-OF を用意して、同フレームを鳥フレームと以下のようにつなぐことができる。

```
(翼      (IS-PART-OF 鳥))
```

すなわち、この場合、スロット HAS-PART-OF とスロット IS-PART-OF は一つの関係リンクをそれぞれ反対方向に辿ったものと考えることができる。従って、フレームの

集合は意味ネットワークと同様にネットワークを構成する。

このように基本的な考え方としては、フレーム型知識表現は意味ネットワークの見方を変えたものと見ることができ、これら二つの表現モデルは関係が深い。しかし、実際のフレーム型知識表現言語では、各フレームには属性の記述以外にも、手続き的なプログラムを付加することができるなど、知識表現機能が強化されている。さらに、思想的にはオブジェクト指向プログラミングとも共通した考え方を保有している。代表的なフレーム型知識表現言語としてはKRLが知られている [Bobrow77]。KRLは、航空機座席予約システムを自動化する自然言語理解システムGUSを台材として研究された。その中で、先に述べた手続き付加機能やデフォルト値機能など、フレームモデルシステムの実現において有効な各種のアイデアが開拓された。

(3) プロダクションモデル

意味ネットワークやフレームは以上のように、分類体系や、ものの静的な記述に適している。一方、知識には、「このようなときはこのようになる」というような、ものの因果関係があり、推論のための重要な要素となっている。

このような種類の知識は、規則の形で記述される。「このようなとき」は条件であり、“IF…”と表し、「このようになる」は結論であり、“THEN…”と表される。従って、このような規則を、IF-THENルールあるいはプロダクションルールと呼ぶ。

専門家システム（エキスパートシステム）は、このようなルールを数多く記述、蓄積し、問題解決を行う推論システムである。これらのルールを左（条件部）から右（結論部）へ順方向に辿る推論を前向き推論、その逆に結論から条件へ遡る推論を後向き推論と呼ぶ。これまで実用に供せられてきた多くのAIシステムはプロダクションモデルを用いた専門家システムであり、病気の診断コンサルテーションシステムにはじまり、投資、資金運用などのコンサルテーションシステム、大規模システムの故障診断システムなどが開発されている。

(4) 論理モデル

論理モデルは述語論理の体系により知識を表現するものである。述語論理は、あるもののそれ自身や、それらの間に存在する関係について記述しようとするものである。

(補:命題論理は言明(ステートメント)が真であるか偽であるかにのみ主眼がある論理であり、言明あるいは記述の内部に立ち入らない。) 述語論理で、例えば、論理式:

$$\forall X. \text{BIRD}(X) \rightarrow \text{HAS-WING}(X)$$

は、「すべてのXに対して、もしBIRD(X)が真なら、HAS-WING(X)は真である」ことを表す。すなわち、同論理式は「Xが鳥であるなら、それは翼を持つ」ということを表現している。ここでBIRD(X)やHAS-WING(X)は述語といい、「真」または「偽」のいずれかの値を取る。記号 $\forall$ は限定子の一つ、全称限定子であり「すべての~に対し」という意味を持つ。限定子にはもう一つ、存在限定子 $\exists$ があり、「~が存在する」という意味を表す。限定子の変数(Xなど)にのみかかり、述語にはかからない論理体系を1階述語論理という。論理型プログラミング言語Prologは、この1階述語論理に基づいて構築された言語である。

論理モデルを用いる利点は、知識を明快に宣言型で記述できることと、推論機構が強力であることである。例えば、Prolog言語を用いれば、推論を用いて問題解決を行う部分のプログラムを自分で書く必要はない。しかしながら、実用規模の知識を相手にした場合は、論理モデルでは計算量が大きくなるという問題があり、これまでのところその利用は研究レベルにとどまっている。

知識の表現と推論の技術は実用に供しようとするはまだ完全なものにはなっていない。これまでに開発されているシステムでも、発見的手法の部分が大きな部分を占めていることが報告されている。これは意味を扱う推論機構が強化されている論理モデルでもそうであり、本質的な課題がまだ残されていることを意味する。

実際問題に適用しようとする、まずどのように意味の原始(プリミティブ)を選ぶか、どのように知識を体系化すべきか、あるいは実世界をどのように捉えたらよいか、ということが問題となる。また、どのように例外を扱ったらよいか、あるいはシステム内に記述されていない知識はどのように解釈すべきか、といった問題がある。

実世界の問題では、いわゆる常識 (Common Sense) が極めて重要な役割を演じている。すなわち、システムを真に知的にするためには、我々が意識することなく頭の中で推論に用いている「常識的な知識」が不可欠である。従って、人間が保有している常識をすべてシステムに取り込むことが当然の帰結として求められるが、それには巨大な規模の知識ベースを構築する必要がある。そのため、その代わりに、デフォルトという考え方を導入し規定されていない場合の知識の扱いを可能にすることも考えられている。また、実際問題では必ず例外があり、これをどう扱うかも問題である。このような問題に対して、例外記述と多重継承とを許容する場合の統一的な推論機構や、デフォルト論理をサポートした非単調推論機構などが研究されているが、まだ解決されていない問題も多い。

しかしながら、システムがすべてを自動的に推論して問題解決するのではなく、人間と協調的に知識処理を行うシステム、あるいは人間の良きパートナーとしてのツールとして考えるのであれば、かなり応用範囲は広がると考えられている。従って、これからはAI (Artificial Intelligence) ではなく、IA (Intelligence Amplification) が重要であるといわれている [Davis87]。特に、知的ハイパーテキストでは、システムに完全な問題解決を期待するのではなく、システムの支援あるいは協調により人間の知的能力を増幅することが目的、狙いであり、部分的にでも応用できる技術要素は多い。

2) 知的検索技術

コンピュータによる意味の処理が考えられたときから、その目的の一つはコンピュータの中に情報を蓄え、そこから欲しい情報を意味的に取り出すことであった。その後も自然言語理解の分野で質問応答システム (Question&Answering System) を題材とした研究が多くなされてきた。これらは知的な情報検索システムであるということが出来るが、応用システムの開発を目指すというよりは、科学的な見地からのものが多く、研究のためのピークルであった。また、質問応答システムという名称が示すように、「人間の代わりをする知的なシステム」を目指したものであった。

これに対して、近年、データベースが広く実用になるにつれて、その利用方法が素人には難しいことが実感されるようになってるとともに、それを解決すること自体も易しく

はないということが分かり始めた。それも単に、操作方法が難しいというだけではなく、情報検索において、欲しいものを如何にコンピュータに伝えるかという本質的な問題の存在が改めて分かってきた。このような理由から、従来よりも工学的なアプローチによる知的な情報検索システムの研究が増えてきた。

さらに、情報検索は、検索文をシステムに与えて関連するデータを取り出すという単なる一方通行的なやり取りではなく、真のニーズに応えるためには、検索結果を次の検索文に反映させながら、欲しい情報に順次絞り込んでいくプロセスが重要であることも認識されてきた。さらに、あちこちを歩き回る感覚で拾い読みする機能、すなわちブラウジング機能が重要であることも認識されてきた。

これらの新しいアプローチは、マルチウインドウを用いた知識表現言語システムを応用することで具現化されつつある。これまでRABBIT、ARGON、UNIFILEといったシステムがこのようなアプローチで試作されている。

これらのシステムは、意味ネットワークモデルあるいはフレームモデルに基づいた知識表現方式を用いて検索対象の情報アイテム（データやオブジェクト）を記憶するとともに、それら情報アイテムが何であるかを記述する一般的な知識体系をも知識ベースの中に保有している。この点は従来の情報アイテムのみを記憶するデータベースとは大きく異なっている点である。また、これらのシステムは検索文を随意に編集する機能や、知識ベースの中身を色々な見方で表示する機能、あるいは視点を随意に移動する機能などを備えており、直接的操作方法を用いてモードレスで対話することが可能である。

検索機能の実現にはパターンマッチングを用いた分類型推論機能を利用している。これにより、上位概念語（一般名称）を用いた部分的記述から具体的な概念（すなわち、欲しい情報）を検索することを実現している。

例えば、

「ベル研究所は研究所である。」

「ベル研究所はニュージャージー州にある。」

「ニュージャージー州は米国の一部である。」

「UNIXはオペレーティングシステムである。」

という知識が登録されていたとすると、抽象概念「米国にある研究所が開発したオペ

レーティングシステム」は、具体概念「ベル研究所が開発したUNIX」を包摂することが推論される。すなわち、「米国にある研究所が開発したオペレーティングシステムは何か」という検索文から「ベル研究所が開発したUNIX」を検索することが可能である。

これらのシステムはビジュアルインタフェースにも従来の情報検索システムにもない特徴をもっている。例えば、端末の画面を複数のウィンドウに分割し、それぞれを検索文の編集、知識ベースのブラウジング、ブラウジング中の世界の位置表示などに割り当てたとする。ユーザは自然言語風の検索文をマウスにより指示しながら、順次自分の欲しい情報の記述文に仕立て上げて行くことができる。検索文中のある名詞（例えば「会社」）を指示すると、システムは自動的にメインウィンドウ中に「会社」を含む上位概念である「組織」の部分世界を概念階層木の形で表示する。その表示中の、例えば「コンピュータメカ」をマウスでクリックすると、検索文中の「会社」は「コンピュータメカ」に自動的に書き替わる。あるいは、この計算機メカを限定するために、メニューから可能な「関係」、例えば、「～を開発している」という関係を選んで、検索文を「メインフレームを開発しているコンピュータメカ」というように編集することができる。

このような過程を通じて、ユーザはおぼろげな記憶からでも検索文をシステムからの刺激を受けながら対話的に作成していくことができる。さらに、検索結果がもし多すぎるような場合には、検索文の修飾を強めたり、概念をより絞り込んだ下位概念に変更することが容易に行える。このシステムの基本思想は、一方向的な対話ではなく、このようにシステムからも積極的に関連する情報を選択して表示して、システムのユーザとが協調的に目的を達成する所にある。

これらの試作システムでは、数千のオーダーの概念あるいはデータ項目が登録され、研究者の身の回りの多種多様な個人用情報検索システムとして試用されている。その結果、このようなアプローチが有効なものであることが報告されている。これらのアプローチは、知的検索システムは知的ハイパーメディアの有望な一つの応用分野であることを示していると考えられる。

2. 自然言語処理

自然言語処理は、談話理解、テキスト理解、機械翻訳、自然言語マンマシンインタフェース、情報検索、あるいは日本語の場合には仮名漢字変換などに応用できる技術を含み、主に言語を理解するための方法として研究されてきている。その意味で「自然言語理解」ということも多い。

自然言語理解は、形態素解析を含む講文解析と意味解析というプロセスで構成される。歴史的には、講文解析するための手法がまず研究され、ATN (Augmented Transition Network) 文法などが考案された。ATNでは、有限状態遷移図に英語独特の組み込み的な表現を処理する機能を付加して、文脈自由文法の解析能力を強化している。この文法を用いた講文解析により英語をデータベースの検索文(形式言語)に変換する質問応答システムLUNARが開発された。このシステムは米国NASAが月から持ち帰った石の素性分析結果のデータベースに自然言語で質問することを実現した。また、ほぼ同じ時期に積み木の世界を操作する対話システムSHRDLUが開発されている。これらのシステムは自然言語理解のためには言語知識のほかに、対象とする世界の知識が不可欠であることを実証した。

これに対してSchankは文や句の意味を表現するために、概念依存理論 (Conceptual Dependency Theory) を展開した。この理論の背景には、どんな文にも意味が同一なら言語に依存しない意味表現方法が唯一に存在するはずである、という考え方がある。この考え方は、中間言語を介する機械翻訳方式の基本にもなっている。概念依存理論では、意味の原始となる原始動作 (Primitive Acts) を11個抽出している。そしてこれらの原始動作と「もの」の間には、動作主、対象、受け手、手段、などの依存関係を定義している。さらに、これらを基本としてすべての動詞を分類体系のなかに位置付けることが考えられる。文の理解とは、入力された文を講文解析するとともにこれらの概念依存関係に射像させることをいう。この概念依存関係の構造は、計算言語学における格構造にもよく似ている。自然言語理解における意味の抽出の有望な方法はこの流れを組んだ方式であると考えられている。

このような自然言語理解方式をとるためには、従来のような言語的な辞書のほかに、概念の体系を表すネットワークと各概念を定義する概念辞書を構築することが必要で

ある。

一方、自然言語理解の難しさの一つはあいまい性の解消方法である。すなわち、複数の解釈のうちどれが発話者（筆記者）の意図に沿うものであるかを同定することである。これを解決するためには、文脈や常識が重要であるとして、常識推論や類推推論などの高度な推論方式の研究や、常識を含んだ巨大な知識ベースを構築することなどが進んでいる。あいまい性が自然言語理解の重大な課題である一方、それを逆手にとって如何にうまく利用するかという観点も同時に重要であると思われる。

3. パターン認識

知的ハイパーテキストに応用できるパターン認識技術は主に音声認識・音声合成技術と文書処理・文書認識技術である。これらの技術は、マンマシンインタフェースの改善やデータベースの自動構築手法として応用することができる。また、パターン認識技術の内、特徴抽出技術やパターンマッチング技術をうまく用いると、マルチメディア情報の検索の高度化を実現することができる。

1) 音声認識と音声合成

音声情報処理の直接的な応用は、音声認識と音声合成を用いたマンマシンインタフェースとしてである。もう一つの応用は、音声の符号化技術であり、自然音や音楽を含めた音声データの圧縮符号化に不可欠なものである。

音声認識は最も長い間、コンピュータとの対話手段として求められてきており、期待も大きく、一部は製品にもなっている。また、これまで多くの音声認識システムが試作、開発されてきている。しかしながら、誰もが自在に用いるにはまだ技術的課題が残されている。その意味で、それぞれの認識システムは何らかの制限がついたものとなっている。制限のパラメータには、話者は特定か不特定か、発話内容は単語か文章か、発話は離散か連続か、および語彙数はどれぐらいか、という4つのパラメータがある。従って、音声認識システムとしては、これらのすべての組み合わせがありうる。以下に主なシステム形態を説明する。

(1) 離散単語認識システム

単語毎に区切って発話するシステムで、単語としては数詞や命令単語があり、学習を行なった特定話者に対して実用レベルの精度（97%程度以上）で認識ができる。語彙数は通常100から200語程度である。しかしながら、各話者はすべての単語を学習のため数回以上、事前に発生する必要がある。ここでいう学習とは、サンプルパターンから認識で用いる標準パターンを作成することである。

認識方式としては、音声波形を周波数分析することにより特徴量を発話音声単位で抽出し、学習してある標準パターンと特徴空間でマッチングを行うことにより認識するものである。マッチングにおける技術的問題である発話速度の非線形な変動については、動的計画法（Dynamic Programming）を用いて吸収する方法（DPマッチング法）が有効な方法として知られている。この方法では発話音声の中がどのような音韻から構成されているかは見ておらず、発話されたパターンを全体としてマッチングしているのみであり、そこに限界がある。

(2) 連続単語音声認識システム

複数の単語を連続して発声しても良いように拡張されたシステムである。音声の標準パターンは単語単位に用意されていて、マッチングを行う際に単語の境界を最適に求める方式を用いる。ここにも動的計画法が有効に用いられる。例えば、2段DPマッチング法が開発されている。応用としては、電話番号のように連続した数字音声の認識がある。連続してもよい単語の数は認識精度の点から4個から5個位である。このシステムでも話者は学習を事前に行った特定話者である。また、認識できる語彙の数はあまり多くはできない。

(3) 大語彙単語認識システム

1000から5000、あるいは1万という語彙を対象にするシステムである。これだけの大語彙となるとすべての単語を予め発話して学習することは事実上不可能であるので、

上記の方式とは異なり、音韻単位の認識を経た単語認識の手法をとる。すなわち、標準パターンファイルには、音韻認識のための標準パターンと、各単語の発音情報（音韻列など）が記憶されている。システムの方式にもよるが、100種程度の音韻を学習しておくことが必要である。

この学習にはいくつかの方法がある。音韻認識を経る音声認識での問題に調音結合とよぶ現象があるが、これを解決するために、音韻学習を単語や文章を連続発声した音声データの中から学習する方法がある。また、すべての音韻をいちいち学習するのではなく、サンプル音声からその話者の特徴のみを取り出して適応させる方法もある。この方法は話者適応という。

大語彙単語認識システムにおける認識精度は、現在まだ実用レベルには達していない。

（４）連続音声理解システム

連続的に発声した文書を対象としたシステムであり、意味の理解まで狙ったものである。現在も活発に研究されている技術である。これまでに、HMM (Hidden Markov Model) 手法やベクトル量子化など、興味ある有効な手法が開発されている。また、自然言語理解技術との関係も深い。

現在、人工知能システムに良く用いられている黑板モデル (Blackboard Model) は、この連続音声理解の研究の中から生まれたものである。

最近、ニューラルネットワークの音声認識への適用も盛んに研究されている。従来の認識方式を部分的に越える認識精度を達成したという報告もあるが、実用にはさらに時間が必要であると思われる。

音声認識はマンマシンインタフェースにとって極めて重要、かつ有効な方法であることがいわれているが、パターン認識の中では最も難しい技術の一つであり、現在のところワークステーションなどの対話手段として日常的に使われる状態にはなっていない。音声は人間と人間との対話手段として最も自然で、かつ自在に操ることができるものであることは疑いの余地はない。しかしながら、コンピュータシステムとの対話手段としては、音声認識技術の向上のみならず、その在り方や利用の仕方を見直す必要性があると思われる。

一方、音声合成技術としては、テキストからの規則合成も可能になりつつあり、知的ハイパーテキストへの応用も考えられる。従来は、人間が喋った音声进行分析して符号化したうえで、コンピュータに記憶させ、必要に応じて出力するいわゆる分析合成型の音声合成が用いられていた。これは、システム形態としては情報圧縮が行なえることと、ランダムアクセスが可能である点以外は、テープレコーダと変わりはない。これに対してテキストからの規則合成では、文字情報を入力すると、それを解析して発声情報を求め、さらにアクセントやイントネーションを自動的に付与したうえで音声出力を行う。これを実現するためには、入力テキストの形態素解析を行い単語に分割し文構造を同定する技術、文全体に対してイントネーション（時間関数としての音声ピッチ周波数）を決定する技術、アクセントを含めた発声情報付きの単語辞書の構築、などが必要である。この技術はさらに「概念」からの音声合成に向けて研究が進められている。

これらの技術の一部は、すでに1ボードのテキスト音声自動変換装置の形で商品化されている。テキストからの音声合成は、画面に出しきれない情報の補足説明や、処理状況の報告、あるいはヘルプ情報の読み上げなど、応用範囲は広いと考えられる。

2) 文書処理と文書認識

パターン認識の中で文字認識技術は最も長い実用化の歴史をもっている。商品としては、光学的文字読み取り装置（OCR:Optical Character Reader）が20年近い歴史をもっている。これらは主に数字を中心とした伝票類を高速に読み取る目的で利用されてきている。

これに対し、最近では本や論文などの文書を読み取る装置も商品化されてきており、文字認識技術の応用範囲はさらに拡大されつつある。OCRが読み取り対象としている文字は、数字、アルファベット、片仮名、記号が基本であり、活字文字と手書き文字とが読み取れる。活字は通常数十種類のフォントが読める。活字の認識精度は、文字品質にもよるが、99.995%程度である。手書き文字については、数字の認識率は99.95%程度であるが、アルファベットや片仮名になると精度はそれより若干下がる。

最近、伝票類に漢字が含まれることも多く、特に申込書や契約関係の手書きの書類の読み取りが求められており、これらを読み取る手書漢字OCRも実用化されている。これらの手書漢字OCRは、丁寧な楷書体で筆記されたものを対象にしており、通常、2500から3000字種の手書き漢字を読み取ることが可能である。住所や人名などの単語照合処理を用いることにより、99%以上の認識精度も達成されている。このように、これまでの文字認識技術は情報処理システムへの伝票類のデータ入力手段として応用されてきた。しかしながら、近年光ディスクの文書ファイリングへの応用が始まることにより、文字認識のみならず、文書画像処理への期待も増している。

光ディスク1枚には8万枚から12万枚の文書イメージを登録することができ、これを用いて電子的なファイリングシステムを構築することができる。多数の光ディスク駆動装置を接続したり、光ディスクライブラリ装置を用いることにより、100万件（1000万頁以上）のオーダの文書データベースの構築も可能である。文書イメージは、200dpi（dot per inch）ないしは400dpiの走査密度で入力、蓄積、印刷がなされる。イメージデータはMMR符号化方式（Modified Modified-READ方式）により、文字が主体の通常の文書で10%から15%に圧縮することができる。したがって、A4サイズの文書イメージは30Kバイトから60Kバイトになる。現在のところ、これらのシステムでは、モノクロ2値のイメージ処理が基本である。

光ディスク・ファイリングシステムでは、遠隔地からでもコンピュータネットワークを通じて検索を行い、本文を含めて即座に結果を入手することができる。この点で効果と期待が大きい。従来の文献データベースは書誌的事項（表題、著者名、出典など）を大量に集めたものであり、文献本体は別途取り寄せるなり複写する必要があった。最近では、マイクロフィルムの代わりに、このような文書イメージをCD-ROMに格納したコンパクトな文書データベースも構築されている。例えば、日本の特許公報は、これまでのマイクロフィルムによるサービスをやめて、CD-ROMの形でサービスが行われている。

このようなシステムにおける課題は、文書を登録する際に、検索が可能になるように文書画像からインデックスとなるべき検索情報を抽出して、同時に登録することが必要なことである。従って、この処理を自動化するために文字認識技術を応用することが考えられている。例えば、文書画像を解析して文書構造を抽出した上で書誌的事項

を自動抽出したり、写真や図表などを分離抽出して、内容からの検索を実現しようとする研究がある [Fujisawa88]。また、文書画像のテキスト領域を解析して、各文字パターンを数ビット程度の情報に符号化することにより、その符号をサーチして内容検索を行うアプローチも研究されている。文書の構造を解析するアプローチとしては、文書レイアウトの一般的ルールを知識ベース化して、その知識を援用することによりトップダウン的に文書の論理構造まで抽出する方法も試みられている。

これらの技術は、特にデータベース型の知的ハイパーテキスト・システムの構築に有効であると考えられる。

4. 学習

学習は知的なシステムに不可欠な機能である一方、技術とするまでにはその原理は十分に解明されていない。

学習には、概念の学習、手続きの学習、論理に基づく学習、統計的な学習、あるいはニューラルネットワークにおける学習がある [安西89]。生のデータを多量に集めて概念的な整理をしたり、抽象化した概念を自動的に抽出したりすることが概念の学習である。手続きの学習では、例えば対話的に実行した例からプログラムを抽出することや、プロダクションシステムにおけるルールの学習などが研究されている。

統計的な学習やニューラルネットワークの学習は、主にパターン認識において重要である。このパターン認識における学習の問題は、先の節で述べたようにパターン認識システムの実現に不可欠なものであり、全自動とまではいかないものの100万のオーダーのサンプルを対象にした学習を行った例もある。特に、実用になっているOCRの分野では、適切な学習を行うことによって始めて先の述べた99.99%のオーダーの認識精度が達成されている。

知的ハイパーテキストにおける学習としては、個人への適応化機能や、大量の概念の整理機能が期待される。前者はマンマシンインタフェースの高度化の要求である。ユーザがシステムを利用するにつれ、そのユーザの興味の範囲や癖を学習し、短い対話（プロトコル）でも複数のことができるようにすることが可能となろう。概念の整理に対する要求も大きい。元来、ハイパーメディアシステムでは、大量の情報をユーザ

に如何に分かりやすく提示するかが重要な命題であり、逐次ランダムに入ってくる情報を概念的に分類する機能があれば極めて有効である。

近未来的には、ハイパーテキストのリンクをユーザが辿る毎にその重要性や、前後の関係などを学習し、複雑なネットワークのための適応型ナビゲーション機能を知的ハイパーテキストに付与することが考えられよう。

5. 概念と辞書

知的ハイパーメディア・システムを分かりやすいシステムにするには、自然言語理解機能は欠かせないものとなる。また、情報を格納する知的な入れ物として用いるのであれば、データベースにおけるスキーマに対応する意味データモデルが重要である。これにも自然言語の理解が重要な役割を演じる。すなわち、自然言語は最も強力な知識表現言語であるとも言える。

このためには、概念を整理し、それを表現する単語の集合を体系化し、概念間相互の関係を記述することが求められる。現在、EDR（日本電子化辞書研究所）において、約40万の項目からなる概念辞書が開発中である [EDR88]。この辞書の第一の応用先は機械翻訳であるが、巨大知識ベースの構築も目標に掲げられている。EDRで開発中の電子化辞書には、単語辞書、概念辞書、共起辞書、対訳辞書が含まれる。概念辞書は概念体系と概念記述とからなり、全体で概念のネットワークを構成する。このネットワークの構成は先に述べた意味ネットワークの構成をとる。

知的ハイパーテキストにおいては、大規模な知識ベースとしての形態もあり、概念辞書はその基盤技術として不可欠である。しかし、同時に小形なシステムとすることも求められており、必要最小限の概念項目を抽出し、コンパクトな概念辞書を構成することも工学的には重要である。そのような観点からは、網羅的に概念を収集整理するのではなく、基本となる動詞の体系や、それら動詞の意味を規定するに必要な基本名詞の集合を明らかにしていくことも肝要である。

2.2 知的ハイパーメディアの基盤技術

2.2.1 オブジェクト指向プラットフォーム

ここでは、知的ハイパーメディア構築に際してそのマルチメディア情報処理とユーザーインタフェースの設計に密接に関係すると考えられるオブジェクト指向パラダイムを提供する基本技術について考察する。

オブジェクト指向という言葉は、1970年代後半から80年代初頭にかけてに Xerox 社が相次いで発表したデスクトップ・メタファをユーザインタフェースとするオフィス・システム「Star」とオブジェクト指向型プログラミング言語「Smalltalk-80」により一躍脚光を浴びた。

オブジェクト指向は、一般的にデータ抽象型プログラミング言語の発展形態として位置づけられているが、ソフトウェアの拡張可能性と再利用性の改善といった生産性向上の観点から「パッケージ化、部品化」、処理方式から「並列計算モデル」、データベースから特にマルチメディア・データベースの支援としての「モデル化表現」「情報隠蔽」、人工知能からのモデル化表現や「知識のモジュール化、フレーム化」などの基礎研究やプログラミング理論に加え、マルチメディア・ユーザインタフェース、ハイパーメディアなど様々な応用分野から同時発生的に提起されてきた。そのためか、多くの議論が今日まで行われているがいまだにオブジェクト指向の考え方自体に体系的な基礎が確立されているとは言えない。

ここでは、オブジェクト指向の基本概念を元祖ともいえ、発表以来、常に議論の中心となっている Smalltalk-80 から求めてみると、

「オブジェクト指向は、ソフトウェアの体系を人間同志が会話する形を抽象モデル化して表現しているもので、ソフトウェアシステムの中心となるオブジェクト（人間）とメッセージ（言葉／会話）、という単位から構成する」

である。実世界におけるモノは記憶（位置や大きさなど）を持ち、いくつかの作用に反応する。これらをひとかたまりにした記述の単位をオブジェクトとする。オブジェクト間でやりとりされる情報は、解釈者としてのオブジェクトとそれにたいするメッセージとの対である。メッセージは記号としての物理的变化を担うものである。この考え方はソフトウェア開発者（又は、使用者）の思考パターンに沿ってソフトウェア

を開発する（又は使用）上で何をしたいかという目的（WHAT）を考えるだけでソフトウェア手順をどう組むか（又はどう使用操作するか）（HOW）を意識しなくても利点がある。つまりメッセージをどのオブジェクトへ送るかということを選択することがプログラミングであり、操作となる。

ソフトウェア工学的に言えば、

「オブジェクトとは、手続と記憶領域の融合体で外部から隠蔽したもの」

と定義することができる。つまり従来風に言えば、内部変数と内部手続きを持ってカプセル化されたレコードあるいはストラクチャと呼ばれるデータ構造のようなものである。オブジェクトは内部状態を持ち、ある特定の要求（メッセージと呼ぶ）に応答して、内部状態を変化させるように動作するものである。

オブジェクト指向のソフトウェア開発は、今日、一般的に行われているライブラリーから関数を呼ぶ構造化方式ではなく、再利用コンポーネント（オブジェクト）を組み立てる方式である。ちなみにオブジェクト指向言語のひとつである Objective-C ではこのような再利用コンポーネントを Software IC と呼んで、全く新しい IC を作るのではなく、既存の再利用可能な IC に少しの変更を加えたものを組み合わせることで新しい回路を実現することと表現している。

また、思想を表現する技法としての観点からは、オブジェクト指向は、「モノ（オブジェクト）のあり方に着眼し、現実の世界をモデル化するパラダイム」であり、「現実の世界はモノの集まりで構成されており、モノ同志が互いにメッセージを通信しあってモノごとが進む。制御中心でなく、こうしたモノの振る舞いを中心にしてコンピュータ上の処理体系を考えるもの」ともいえる。

オブジェクト指向プラットフォームとは、「オブジェクト指向型の思考を表現し実装するためのツール」といえる。

1. オブジェクト指向ユーザインタフェース

最近、システムや製品の成否を決めてしまうほどの主要な要因のひとつとして、ユーザインタフェース設計の重要性が高まってきた。しかしながら、その対象となる範囲が広く、また、ユーザとシステムの置かれている環境や文化的側面の影響も多く受け

系統的な理論や体系はまだ確立していない。

しかしながら、システムのユーザが増加し使い勝手の面から、抽象から具体へ、単一メディアからマルチメディアへと取り扱う情報量が増加し複雑になるに従ってその希求度は急速に高まっている。また、異なった開発者やメーカーが作った異なったアプリケーションでも、一貫性のあるユーザインタフェースで操作できる要求も急速に増している。

1) Look and Feelとオブジェクト指向

新しいユーザインタフェースの概念として「ユーザが、抽象物でなく具体的な物を相手にする感覚を重視する」今日流の「Look and Feel（見た目と使い心地）」の概念が生まれたのは1970年代、Xerox社のパルアルト研究所（PARC）においてである。

“Look”がマルチウィンドウやアイコンの多用であるとすれば、“Feel”は対象領域をあたかも直接操作する感覚である。

Look and Feelはオブジェクト指向の考え方から出現した発想であり、より自然なユーザインタフェースとしては「抽象性より具体性が優れている」という観点から、「情報という抽象を具体物としていかに錯覚させることができるか」ということである。逆にいえば、オブジェクト指向パラダイムが「Look and Feel」の実現を可能にしたということである。

つまり、従来の手続き型のパラダイムは、関数という作用や述語論理という関係といった、どちらも目に見えず、手に触れることはできない抽象を処理対象の中心にしてきた。一方、オブジェクト指向はその原点に、オブジェクト（もの）という、目に見え、手に触れるものを対象にしているためLook and Feelの実現には最適な優れた「計算モデル」といえる。まさに抽象から具体への変換である。もちろん、モデル化やメタファはあくまで「たとえ（比喩）」であって自然でも完全でもないことは明らかである。

2) アイコン 2 次元ユーザインタフェース

典型的なオブジェクト指向ユーザインタフェースのひとつとして現在、最も流行しているのはアイコン・ユーザインタフェースである。

アイコンはスクリーン上の 2 次元の像とか絵であり、机や鉛筆といったものをメタファ（たとえ／比喻）として表現するものとして使われる。

但し、アイコンの語源が、「神の力を持つ宗教的な彫像とか絵（聖母マリアの像とか絵といった宗教のご利益のあるもの）」であるように、前述の Star システムでは「アイコンは、振る舞いとそのものに本来備わっている属性をもつオブジェクト」と定義している。

従って、スクリーン上の単なるシンボルやラベルはアイコンとはいえない。

アイコンは正にメディアオブジェクトしかもダイナミックメディアのひとつといえる。尚、アイコン・ユーザインタフェースとオブジェクト指向ユーザインタフェースとはしばしば混乱してもちいられているが同義ではない。

しかしながら、その関係は、両者がユーザとシステム側または開発者の間で同一の情報の隠蔽の原理原則を使用していることである。つまり、オブジェクト指向言語では「オブジェクトの中にデータとそのデータ上のオペレーション（メソッド）でカプセル化」しており、アイコン・ユーザインタフェースでは、「データをアプリケーション内で管理されているデータに適用できるオペレーションでカプセル化」している。また、アイコンの複製や再利用はオブジェクト指向言語の継承の概念、アイコンの管理や整理（文書メタファと保管庫）にはクラス／インスタンスの概念なども各々、類似対応している。

ユーザはこのユーザインタフェースにより、

「コンピュータシステムの線形の空間の世界では意味をもたない距離、所有権、対話といった実世界の概念や量や空間という物理の概念を疑似的ではあるが直接操作感がえられる」

「コンピュータの時間と処理は逐次的であるが、実世界のように時間が全オブジェクトにたいして独立に、しかも同時に進んでいるような疑似感覚がえられる」

などを実現可能にしている。

2. オープンプラットフォーム

多種多様なアプリケーションやシステム、ハードウェア、ソフトウェアが続々と登場してきている。そこには、当然、システムやハードウェアに依存しない共通基盤が求められてくる。つまりユーザの視点からは一つのユーザインタフェースの上に必要な種々のアプリケーションを統合して、しかも統一のとれた一貫性のある操作性で実現する要求がでてきた。つまり、異なるアプリケーションに対して共通のインタフェース構築の要求である。

1) ユーザインタフェース・アーキテクチャ

オブジェクト指向が普及するにつれて、ユーザインタフェースを機能やアプリケーションから分離することが意識されてきている。即ち、ユーザインタフェースに係わるウィンドウなどのソフトウェアモジュールが各種のアプリケーションソフトウェアから切り出されて独立し始めた。使い勝手は当然機能と絡み合い、密接な関係をもつため、機能から使いやすさの工夫をきりだすのは一般には決して容易ではないが、この分離は急速に進行している。そこに一石を投じてたのが、Smalltalk-80のMVC (Model-View Controller) パラダイムである。

MVCはSmalltalk-80の中で導入されているマルチウィンドウを使ったユーザインタフェースのモデル化技法であり、ビットマップディスプレイ上のウィンドウを通じた、操作と操作されるデータの関係性をオブジェクト群によって表現したものである。目的とするところはユーザインタフェース、特にウィンドウシステムを「入力によって操作され表示されるデータを管理するオブジェクト即ちアプリケーション対象・モデル」、「表示されるウィンドウ即ち表示・ビュー」、「入力されるデータまたは入力行為即ちユーザからの入力・コントローラ」に分離し、各々をオブジェクトとした集合体としてユーザに必要とされるウィンドウの機能を構築し、アプリケーション領域のモデルとビュー各々の再利用性と結合を容易にすることを提唱している。

仮想端末レベル (グラフィックス・プリミティブ：一般にはOSレベル) とユーザの間に、アプリケーション層に相当するものとしてモデル、プレゼンテーション層に相当

するものとしてビューとコントローラが位置づけられる。

モデル (Model) : アプリケーションを構築するために使われるオブジェクト

ビュー (View) : モデルを出力表示するオブジェクト

2) コントローラ (Controller) : ユーザからモデル及びビューへの入力制御オブジェクト

モデルは、ユーザに提供される方法に関係なく実現される。完全に受動的なものであり、ユーザインタフェースに関する情報は保持していない。ビューとコントローラからなるユーザインタフェースがユーザやシステムの都合により分離されたり置換されたりする可能性が大きいからである。モデルは常にビューから参照されるがその逆はない。各モデルは複数のビューとコントローラの組合せをもっている。

ビューは特定のモデルに対するユーザインタフェースをコントローラとともに実現する。一定のモデルは複数のユーザをもっているため、一般にはユーザの固有の要求を満たすため、様々なビューがつくられる。例えば、あるユーザはグラフィックディスプレイであったり、あるユーザは文字ディスプレイであったりする。デスクトップメタファ、つまりビューが机の上で重なり合った紙のように振る舞うユーザインタフェースはここで実現される。ビューはモデルをもとにデータの表示をし、モデルへのポインタをもつ。各ビューは必ず一つだけのモデルと一つのコントローラが対応する。コントローラは、ユーザからの入力（キーボードやマウス）を制御し、モデルとビューのオブジェクトの状態を指示する。従ってコントローラはモデルとビューへのポインタをもつ。多くの場合一つのビューに一つのコントローラが対応する。

ここで、ユーザの入力制御を別の独立したコントローラに持たせるの必要性については議論があるところである。つまり各ビューが両方のチャンネルを管理させる。

しかし、入力装置と出力装置の組合せの多様化など今後の発展動向を考えた場合に、オブジェクトの流通性、再利用性の観点からは今から分離しておくことも妥当と思われる。

3) オープンプラットフォーム

知的ハイパーメディアのアプリケーションの開発からみたプラットフォームはユーザインタフェースである。その実現上、重要な要素は、前述のようにアプリケーションがユーザインタフェースレベルでの互換性をもって提供されることと同時に、多くのユーザに共通のインタフェースとして使用できることである。つまりプレゼンテーション層にあたるビューとコントローラの設計ツールと設計方式の土台となるマルチウィンドウシステムがユーザインタフェースのプラットフォームと考えられる。

ユーザインタフェースの構築は、基本的にはプレゼンテーション層に属し、ウィンドウのオープンやクローズ、サイズ表示、移動、切り替えなどを管理するウインドインゲシステム、フォントやグラフィックの表示方法を定義するもので、スクリーンやウィンドウ、ウィンドウ内のテキストの書体やサイズをしてを表示するためのイメージングモデル、プログラミング言語のためのファンクションコール群であるアプリケーションプログラム・インタフェースの3つの要素から構成される。

これらの各層の標準化のための作業がOpen System FoundationグループのOSF/MotifやUII (Unix International Institute) グループのOpenLookなどを中心に行われているがプロトコルや機能自体が十分には検討されておらずいまだ決め手のない状況である。現状、実質的に標準となりつつあるのは、ウインドインゲシステムのX WindowとイメージングモデルのDisplay PostScriptである。

2.2.2 オブジェクト管理

1. オブジェクト管理

オブジェクト管理は、オブジェクト指向の考え方と密接に関係している。つまり、オブジェクトはあくまで「記憶+手続」の情報のみで構成され、自分自身の識別子を持っていないということである。名前がなくても犬は犬であり人は人であるようにオブジェクトも名前がなくてもオブジェクトである。何故ならば、カプセル化の基本によりオブジェクトの内部情報自体は外部から隠蔽されて保護され、名前自体も外部からは隠蔽されているからである。

したがって、オブジェクトの存在自体が他のオブジェクトとを識別する要素となる。

つまり、同種のアイコンは名前で区別されているのではなく、各々の表示された位置や大きさなどを表示されることによって識別される。これは従来のシステムがもつ各データの本質は識別子であるといったメタファが解消され、実世界がより適切に写像されるためマルチメディア情報の取扱いが有効なことである。一方、プログラム開発上はオブジェクト指向以外の言語では全く問題にならないオブジェクトを識別し管理するといったことが必要になってくる。

Smalltalk-80を始めとして多くのオブジェクト指向システムが提供しているオブジェクト管理のためのメカニズムはウィンドウを中心としたものになっている。つまり、システム内の各オブジェクトはユーザと対話できるようにウィンドウ表示機能をもつViewというオブジェクトと関連を持つオブジェクトとして保持される。ユーザはViewを特定し、その選択されたViewのもつ各オブジェクトとの関連をたどることで目的のオブジェクトを特定するというものである。

また、C言語とのオブジェクト指向を組み込んだObjective-Cなどの混成型言語では、オブジェクト識別子を新たなデータ型として定義して、オブジェクト指向の機構をバイパスしてオブジェクト固有情報を直接アクセスすることができる。これはオブジェクトを特定するという意味でハードウェア効率などの面からみると混成型言語の最大の利点でもある。しかしカプセル化という原理に例外を許すという意味で最大の欠点にもなっている。

2. オブジェクトベース

オブジェクト指向によって作成されたオブジェクトやアイコンはどのように保管し再利用すればよいのであろうか。多くのオブジェクト指向システムは基本的に次の二つの概念を提供している。

1) クラス (Class) とインスタンス (Instance)

実世界においては、何の意味も持たずに集まったものを烏合の衆というが、一般に同じような性質や機能、目的をもったものの集まりや目的をもった集まりをクラスと呼んで分類していく。オブジェクト指向においても、同じ内部記憶構造と同じ手続をもつオブジェクト群は、同じクラスに属すると考えられる。一つのクラスを指定したと

き、そのクラスに属するオブジェクトをそのクラスのインスタンスと呼んでいる。

2) 継承 (Inheritance)

一般により上位の概念がもつ性質や機能は、より下位の概念によって受け継がれる。親子関係などの遺伝の法則などはその一例といえる。オブジェクト指向においても、下位のクラスに属するオブジェクトは、上位のクラスを特徴づける性質や機能を受け継ぐ機構を有する。これを継承 (Inheritance) と呼ぶが、オブジェクト指向の世界では実世界と異なり、下位のクラスのオブジェクトを定義するとき、その上位のクラスの名前を指定することにより、その上位のクラスの性質や機能を自動的に継承することにすれば、後はその上位のクラスのオブジェクトはもたないが下位のクラスのオブジェクトに必要な機能や性質を付加的に記述するだけでよい。継承は対象世界の概念的上下関係をオブジェクトのクラスとしてモデル化しそれを簡潔に記述するのに有効であると共にオブジェクト再利用にとって重要である。

こうしたクラスと継承の概念によりオブジェクトやアイコンを整理したり生成したりすることがオブジェクト管理の基本になっている。

クラスと継承はプログラムの再利用に大きく寄与するが、プログラムが依存している基礎をなすソフトウェアが変わったときにアプリケーションが動作しなくなる可能性を皆無にすることはできないことに留意する必要がある。

また、オブジェクト指向は自律的要素オブジェクトからなる世界であり、オブジェクトは互いに情報を交換して全体的な作業を行うということから、当然、情報ネットワークを介して個人ベースから共有ベースへとオブジェクトを共有して使用したり参照したりするといった分散と協調のシステムに適応することが考えられる。

その時には、オブジェクトベースの履歴、同期、保護といった要素の研究開発が必要である。特に、継承の取扱いはオブジェクト指向の概念とのものと係わり合いのある基本的な問題となってくる。

2.2.3 プログラミング言語

1. コンカレント・オブジェクト指向言語

並行処理の基本的な考え方は、ひとつの仕事を複数に分離してそれらを同時に実行すればより高速に仕事ができるということであるが、実際上の並行処理の実現にとって重要なことは、分割した仕事を結合することにある。つまり分割した実行された仕事は各々相互作用をもったり統合したりすることが必要である。

オブジェクト指向の計算モデルの最大の魅力のひとつは、逐次的な従来のコンピュータのアーキテクチャモデルから抜け出すことである。オブジェクトがメッセージによって通信しあうという、小さな独立したオートマトンがそれぞれ並行して動作を続け、必要なときに必要な所へ必要なだけ情報を提供するという計算機科学の夢である並行処理計算モデルの導入に適しているということである。

しかしながら、Smalltalk-80を始めとする現在のオブジェクト指向言語は、逐次コンピュータ上で実現されているためオブジェクトという単位の他に、プロセスという処理単位が存在するということからくる並行モデルの実現上の難点があった。第一は、並行に実行されているプロセス同士の交信において、メッセージ交換でなく遠隔手続呼び出しの時に生じる本来対等であるべきオブジェクト間での従属関係の処理、第二は、並行に実行されているプロセス同士の交信における共有オブジェクトの処理、つまりオブジェクト間で共有されているオブジェクトにメッセージが送られてきた時の処理順序の問題、第三は、オブジェクト間、及びオブジェクト内に複数のアクティビティを許す時に生じる同期と排他制御の問題などである。

これらの問題解決に沿ってオブジェクト指向プログラミング言語に並行処理記述能力を持たせる試みが行われている。大きくはオブジェクトを並行処理単位としてメッセージ送信により計算を抽象化しようとする Actor や Concurrent Smalltalk や ABCL / 1 などの方式とオブジェクトは保護されたデータ領域としてこれに並行実体としてのプロセスを付加する Ada の流れがある。

2. ユーザインタフェース構築ツール

ウィンドウベースのユーザインタフェースを構築するための技術分野を最近、グラフ

イカル・ユーザインタフェース (GUI) とかユーザインタフェース・ビルダー (User Interface Builder) と呼ぶようになってきた。GUIは、先にのべたようにXerox社のSmalltalk-80や、Starを原点としてApple社のMacintoshで開花した。

そして最近NeXT社が開発したNextStepに装備されたInterfaceBuilderで大流行のきざしがある。現状は各種のものが乱立している状態といえる。

GUIは、基本的にはプレゼンテーション層に属し、主に次の3つの要素から構成される。

*ウィンドウイング・システム

*イメージングモデル

*アプリケーションプログラム・インタフェース (API)

ウィンドウイング・システムは、ウィンドウのオープンやクローズ、サイズ表示、移動、切り替えなどを行うもので、スクリーン上に表示されるウィンドウ、メニューなどを構成するためのプログラミングツールとコマンドの集合である。代表的なものとしてはX WindowやMS Windowがある。尚、このレベルはOSに依存する部分が多い。

イメージングモデルはフォントやグラフィックの表示方法を定義するもので、スクリーンやウィンドウ、ウィンドウ内のテキストの書体やサイズを指定して表示するための一般にはページ記述言語といわれるプログラミングツールとコマンドの集合である。最もよく知られているものとしてはAdobe社のDisplay PostScriptやそのベースとなったXerox社のInterpressなどがある。MacintoshのQuick Drawなども有名である。PostScriptやInterpressは、元来は多様なレーザープリンタへの出力を意識したイメージングモデルであり、事実その分野への応用が既に行われている。

アプリケーションプログラムインタフェース (API) はプログラミング言語のためのファンクションコール群である。システム開発者はこれを用いて、どのウィンドウ、メニューアイコンを表示するかを指定する。OpenLookのX ViewやHP社のX—Widgetsなどがある。ここがユーザインタフェース構築の鍵を握っている。

しかしながら、現状の多くは上記のような要素に分解できず、独自の構成を持つものが多い。例えばHewlett—Packard社のNewWaveやNeXT社のNextStepのようにユーザインタフェースだけでなくアプリケーション開発のためのツールまで含めて提供し

ているものもある。

標準化の動きも Open System Foundation グループの OSF / Motif や UII (Unix International Institute) グループの Openlook などにより活発に行われているが未だ決めていない状況である。

尚、NeXT 社の NextStep のアプリケーションとして作成された Interface Builder はアイコンを使って自分の欲しいメタファを容易に作れるツールであり、注目を大いに浴びている。このようなものが今後続々と登場してくると思われるが、一番の基本であるユーザインタフェースの互換性の問題はなにひとつ解決されていない。

2.2.4 分散システム

ハイパーメディアは、人間の連想構造に基づいて情報を組織化し、情報間の関連性を含めた形で情報を伝達ないし記録し、それによって多角的に情報を活用できるようにするシステムである。ハイパーメディアをツールとして使用して CSCW (Computer-Supported Cooperative Work) のためのグループウェアを実現しようとした場合、大勢の人々がアイデアや意見を出し合い、それらの情報を互いに関連付けて組織化し、互いの意思疎通を図り共同して作業を進めていくことが必要である。このような CSCW のための環境をコンピュータネットワーク上に実現するためには、ネットワーク上に分散した大量の情報を、統一的に管理し、利用を助けてくれるような分散システムが必要となる。

分散システムに必要とされる機能としては、次のものが考えられる。

1) オブジェクト空間ないし名前空間の管理

ネットワーク上に分散したオブジェクトの識別子ないし名前を統一的に管理する。オブジェクトが存在している場所を意識せずに、オブジェクト名を指定したり検索したり出来ることが望ましい。

2) オブジェクトのアクセス機能

ネットワークをまたがってオブジェクトをアクセスする機能を提供する。オブジェクトが存在している場所によらず、同じアクセス方法でアクセスできる必要がある。

3) 多重実行制御とオブジェクトの同時アクセス制御

複数の利用者からの要求を、一つのシステムで同時に処理するために、複数のプロセ

スを多重に実行するような制御が必要である。

また、同じオブジェクトに対して、同時にアクセス要求が発生することがあり、このような時に、オブジェクトの内容やオブジェクト間の関係に矛盾が生じないように制御する機能も必要である。

4) 分散並列処理

一つのアプリケーションの処理で、異なるコンピュータ上に分散したオブジェクトを相互に同期をとりながら処理するような場合、通信コストの低減と負荷分散のために、それぞれのオブジェクトが存在する場所で分散して並列に処理を実行し、必要に応じて同期を取るような制御を行う。

5) オブジェクト管理

ネットワーク上に分散したオブジェクトを統一的に管理し、オブジェクトの利用を助けるための機能を提供する。検索機能、索引機能、質問言語による問い合わせ機能、バージョン管理機能、履歴管理機能、セキュリティ管理機能などがある。オブジェクトが存在している場所を意識せずに、これらの機能が使えることが望ましい。

ハイパーメディアのオブジェクトを管理するような分散システムの場合には、さらに次のような機能も必要とされる。

6) オブジェクトの性質と振舞いの管理

ハイパーメディアのオブジェクトは、ダイナミックメディアとしての性格を備えていると考えられる。即ち、リンクを辿ってナビゲーションする機能や、メディアを編集する機能などが、そのオブジェクト自身、ないしは、ボタンオブジェクトのようなそのオブジェクトの構成要素のオブジェクトに備わっていると考えられる。

このようなオブジェクトそのものが持つ性質や振る舞いなどを、オブジェクトの型やクラスとして管理するような機能が必要である。

7) オブジェクト間の関係の管理

ハイパーメディアでは、オブジェクト間にリンクが結ばれる。これは、人間の連想構造に基づいたナビゲーションのためのリンク関係である。また、ハイパーテキストの文書は、ボタンやテキストや図形などの部品を組み合わせたり、他の文書の一部を複製したりして、作成されることが多い。文書の構成要素もオブジェクトとして管理し

ているようなシステムでは、作られた文書とその構成要素との間に部品関係が存在する。また、型やクラスをオブジェクトとして管理するようなシステムでは、性質や振る舞いの継承のための関係が出来る。これらの関係をオブジェクトに付随した形で保持して、それに基づいて関係するオブジェクトを見つけて、必要な情報をアクセスする機能が必要である。分散システムでは、ネットワークをまたがってオブジェクト間の関係を結ぶ必要がある。

ハイパーテキストのダイナミックメディアとしての側面に着目すると、物とその振る舞いをまとめてモジュール化するという、オブジェクト指向の考え方を採り入れ、それを分散システム化することが考えられる。一方、オブジェクトをデータベースのように統一的に管理し、その利用を支援する機能も重要である。以下では、これら二つの側面から見た分散システムについて説明する。

1. 分散オブジェクト指向システム

オブジェクト指向は、データとそれを操作する基本的な操作・手続き群をひとまとめにし、それをモジュールとみなそうとする考え方である。ダイナミックメディアとしてのメディアオブジェクトは、メディアを操作するための機能がメディアそのものに付随しているという点で、オブジェクト指向のオブジェクトそのものである。

また、オブジェクト指向は、次に示すように、ネットワークのノードプロセッサにオブジェクトを分散させたシステムを表現するモデルとしても、優れている。

*オブジェクト指向のオブジェクトは、データと操作が一体になったものであり、それ自身が一つの機能単位として独立している。各オブジェクトに関する計算はそれぞれのオブジェクトのもとで、分散して行われる。このため、オブジェクトの分散の仕方に依存することなく計算を行える。また、分散処理が自然に実現されており、並列処理の導入も容易である。

*オブジェクト間の相互作用は、オブジェクトから別のオブジェクトへメッセージ送信することによって実現される。このため、ネットワークでの通信との整合性が良い。オブジェクトの性質の継承メカニズムとクラス階層の考え方は、メディアオブジェク

トを部品として組み合わせて複合オブジェクトとしての文書を作るようなメディアの部品化に適している。また、データベースとしての機能を備えることも可能である。これらの理由から、ハイパーテキストの基盤としての分散システムを実現するためのモデルとして、オブジェクト指向が最も重要視されている。

1) 分散並列処理とオブジェクトアクセス機能

オブジェクト指向システムには、オブジェクトを保存・管理するためのオブジェクト管理システムとしての機能と、オブジェクトを定義したり計算を実行したりするためのオブジェクト指向言語としての機能とがある。分散オブジェクト指向システムは、これらの二つの機能を兼ね備えている必要がある。この実現の仕方には、両方の機能を一体にして実現した分散並列オブジェクト指向システムと、オブジェクトを管理するサーバと言語を実行するプロセスとを分離して server—client モデルで実現したシステムの、二つの実現形態がある。

(1) 分散並列オブジェクト指向システム

Actor や Smalltalk のような、オブジェクトにメッセージを送ることで計算を抽象化しようというパラダイムに基づいたシステムでは、オブジェクトの管理と言語という二つの機能が一体になっている。ここでは、オブジェクトのアクセスは、内容の読み書きという意味ではなく、メッセージを送ってメソッドを起動して、オブジェクトのもとで計算を実行するという意味になる。このようなシステムから発展した分散オブジェクト指向システムでは、分散したオブジェクトのもとで計算を行う処理の分散化が行われることになり、オブジェクトを並列処理の単位にして分散並列処理を実現することになる。このような分散オブジェクト指向システムとしては、Actor、ABCL／1、Orient84／K などがある。

オブジェクト単位の並列処理は、メッセージのやり取りによって同期が取られる。同期の取り方に応じて、メッセージ通信の方法を次の 4 通りに分類する。

①同期メッセージ送信：送信側はメッセージを送ると、受信側がそれを受理するまで、

次の動作の実行を待つ。

②非同期メッセージ送信：送信側はメッセージを送ると、受信側の受理を待たずに、次の動作を行う。

③同期メッセージ交信：送信側はメッセージを送ると、受信側からの返答メッセージが来るまで、次の動作の実行を待つ。

④非同期メッセージ交信：送信側はメッセージを送ると、受信側からの返答メッセージを待たずに、次の動作を行う。返答メッセージは必要になった時点で受け取る。

オブジェクト単位での並列処理の他に、オブジェクト内でのメソッドの並列処理を許しているシステムもある。この時のメソッドの実行順序を制御する方式としては、特殊なメッセージの送信方法を使った時そのメッセージを優先して実行するような方式や、メソッドに優先度を付けておき、それに基づいてメソッドの実行順序を制御する方式や、メソッドないしオブジェクトの状態によってメソッドの実行順序を制御する方式など、様々な方式が考えられている。

(2) Server-Clientモデルの分散オブジェクト指向システム

AdaやC++のような、コンパイラ言語にオブジェクト指向の考え方を導入した用な言語では、オブジェクト管理システムとしての側面が欠如している。このような言語では、オブジェクトの存在がプログラムの実行中だけに限られているため、永続的 (persistent) なオブジェクトを扱うための機能を言語に追加することが必要となる。このような言語から発展した分散オブジェクト指向システムでは、オブジェクトとは別にプロセスという計算の実行単位を導入し、オブジェクトを管理するためのオブジェクトデータベースのサーバが、そのプロセスと独立して存在するような server-client モデルで実現されることが多い。オブジェクトが存在している場所で計算を実行するのではなく、プロセスが作られた場所でサーバ上のオブジェクトをアクセスして計算を実行することになる。オブジェクトのアクセスは、文字どおりオブジェクトの内容の読み書きとなる。

分散並列処理を実現するためには、明示的にプロセスを生成する機能を言語に導入することになる。このような server-client モデルのシステムとしては、ODEというオ

ブジェクト指向データベースシステム環境で実行されるオブジェクト指向言語OPLなどがある。なお、Kamui環境は、オブジェクトを並列処理の単位とするような分散並列オブジェクト指向システムであるが、言語とは独立した形で作られており、server-clientモデルの実現も考慮している。

2) オブジェクト空間ないし名前空間の管理

ネットワーク上に分散したオブジェクトの識別子ないし名前を統一的に管理する必要がある。ハイパーテキストでは、オブジェクトから他のオブジェクトにリンクが張られる。他のコンピュータ上のオブジェクトにリンクを張るためには、ネットワーク全体の中でオブジェクトを一意に指定できるような、統一的に管理された識別子ないし名前が必要である。

2. 分散オブジェクト管理

ネットワーク上に分散したオブジェクトを統一的に管理し、オブジェクトデータベースとしてオブジェクトの利用を支援するための機能について説明する。このような機能としては次のものが考えられる。

1) 検索機能 (search)

オブジェクトの内容や性質によって該当するオブジェクトを検索する機能である。ネットワーク上に分散した大量のオブジェクトの中から、目的とするオブジェクトを見つけだすには、検索を並列に実行して処理の高速化を図る必要がある。また、条件に合致するものが、大量に見つかってしまう場合があり、利用者の側が検索範囲を絞り込んでおくなどの、利用上の工夫が必要となる。

2) 索引機能 (indexing)

内容や性質に基づいて索引を作り、それを使って必要なオブジェクトを見つける機能である。索引の付け方を統一しておくことが、索引を活用する上で重要である。広域ネットワーク環境では、環境が動的に変化するため、索引とオブジェクトとの関係を正しく維持するための技術を研究する必要がある。

3) 質問言語による問い合わせ

同種のオブジェクトを集めたものは、一種のデータベースと考えることができる。このような集合体に対して、データベースの質問言語のような機能を使って集合演算や問い合わせを行う機能が、情報の有効活用のために必要とされている。

4) バージョン管理・履歴管理

複数の人間が共同作業を行って、同じオブジェクトを互いに修正しあうような場合には、版数（バージョン）を管理したり、修正履歴を管理したりする機能が必要となる。オブジェクトが他のオブジェクトの内容を引用している時に、原本を明確にしておくような管理をする場合にも、これらの機能が必要となる。

5) セキュリティ管理

悪意による情報破壊や情報漏洩を防止するために、アクセスする権利を特定の利用者限定したり、情報を暗号化したりする機能。広域ネットワーク環境には、多種多様の利用者が参加しているため、アクセス権の割り当て方に、一定のルールを導入する必要がある。

6) 個人および組織の管理

広域ネットワークには、多種多様の個人および組織が参加している。個人が属している組織や共同体などに応じて、その人の情報空間を限定したり、アクセス権や版權を管理するためには、ネットワークに参加している個人や組織をモデル化して、個人や組織に付随するルールや権利とともに、管理することが必要になってくると思われる。

2.2.5 ダイナミック

1. ダイナミック・メディアの機能

一般的に、情報の伝達は何らかのメディアを介してなされ、メディアは伝達内容に何らかの意図や意味を付加する。メディアは情報の伝達媒体であり、伝達の形態の多様化を求めて、現在までに、種々の機能を持った多様なメディアが発達した。それらの機能の内のいくつかを列挙すると以下ようになる。

1) 可視性

2) 記録可能性

3) 保存可能性

- 4) 可搬性
- 5) 複製可能性
- 6) 速報性
- 7) 同報性 その他

コンピュータがメディアとして使われるようになると、さらにいくつかの機能が上述のリストに加えられた。主なものを列挙すると以下のようになる。

- 8) 編集機能
- 9) 検索機能
- 10) 動作機能
- 11) 対話機能
- 12) プログラム機能
- 13) 連想機能
- 14) 統合管理機能
- 15) 異種メディアの統合
- 16) メディア合成機能、その他

紙に印刷された文書はもはや編集することができないが、コンピュータ上の文書は自在に編集できる。情報に索引を付け、条件検索機能によって膨大な情報の中から所望の情報を抜き出すことも容易になった。何らの動作も定義されていない静的な文書だけでなく、動作することができる動的な文書も扱われるようになった。利用者との対話機能を持ったメディアも定義できるようになった。メディアの動作や対話の仕方を自在に定義したり変更できるように、プログラム機能が提供されているハイパーカードのようなメディアも登場した。関連する情報を結合リンクで結び連想機能を提供する意味ネットワークやハイパーテキストのようなメディアも登場した。データベース化の思想に見られるように、関連する種々の情報を互いの整合性を保ちながら一括管理する統合管理システムの考えかたも浸透してきた。さらには、テキストのみならず、図面、画像、音声、映像などの異種メディアの各々に対して、これらをコンピュータ上で処理する技術が進歩してきたことに対応して、複数の異種メディアの混在を許し

つつ、個々の情報の関連を保ちながら種々の情報を統合管理するマルチメディアシステムも開発が盛んになってきている。さらに近年は、利用者が目的にあわせてメディアの形態や機能を自在に合成して構築できるように、メディアのキットを提供するシステムの開発が望まれ始めている。

メディアはその状態が時間とともに動的に変化するか否かで

「スタティック・メディア」と「ダイナミック・メディア」

に分類できる。状態が動的に変化するメディアを「ダイナミック・メディア」という。

「ダイナミック・メディア」は、さらに、

「非対話型ダイナミック・メディア」と「対話型ダイナミック・メディア」

に分類できる。非対話型ダイナミック・メディアは、ビデオ映像や動画をそのまま表示するダイナミック・メディアである。対話型ダイナミック・メディアはユーザからの指示に応じて反応する。対話型ダイナミック・メディアはさらに、

「パッシブ・メディア」と「アクティブ・メディア」

に分類することができる。パッシブ・メディアは他からの刺激を得て始めて動作するのに対し、アクティブ・メディアは自律的に動作する。アクティブ・メディア一種のエージェントである。

任意のスタティック・メディアはダイナミック・メディアとしてモデリングすることができる。さらに、任意の非対話型ダイナミック・メディアは対話型ダイナミック・メディアとしてモデリングすることができる。同様に、任意のパッシブ・メディアはアクティブ・メディアとしてモデリングすることができる。システム構築の観点からは異なる分類のメディアを何種類も区別して実現することが煩わしく、統合管理の観点からも望ましくない。その結果、アクティブ・メディアを取り扱わないシステムでは、すべてのメディアがパッシブな対話型ダイナミック・メディアとしてモデリングされ、アクティブ・メディアを取り扱うシステムでは、すべてのメディアがアクティブ・メディアとしてモデリングされる。前者がシングル・プロセスのシステムとして構成できるのに対し、後者はマルチ・プロセスのシステム構成になる。

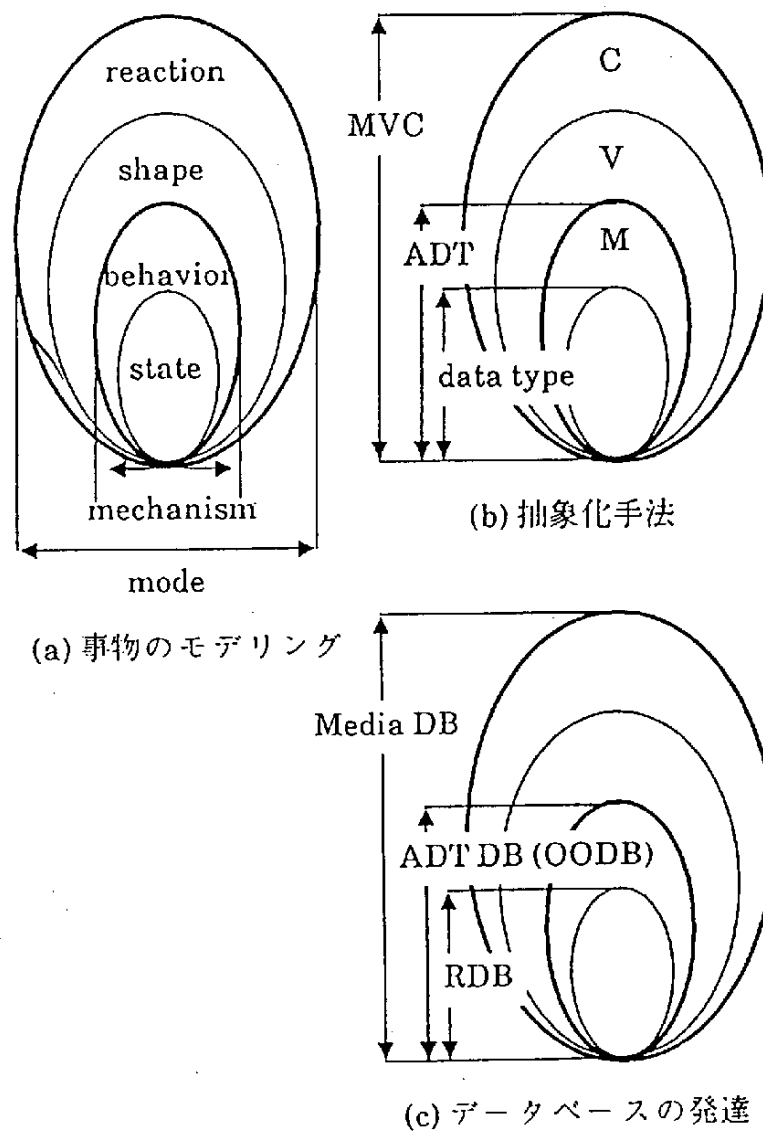


図2-1 事物のモデリングとデータベース

ダイナミック・メディアはユーザがその動作をプログラムすることができるか否かで、「ノンプログラマブル・メディア」と「プログラマブル・メディア」に分類することもできる。

2. ダイナミック・メディアのモデリング

ダイナミック・メディアはオブジェクト指向の考えに準拠してモデリングされることが多い。一般に事物（オブジェクト）のモデリングには、図2-1に示すように事物の（1）機構（mechanism）と（2）様態（mode）の2つを定義することが必要であり、機構を定義するには、（1.1）状態（state）と（1.2）動作（behavior）の2点を、様態を定義するには、（2.1）形態（shape）と（2.2）反応（reaction）の2点を定義する必要がある。反応というのは、直接操作に対する事物の反応を意味する。事物の形態の表示をも含めて事物をモデリングすることは、種々の事物からなる系全体の機構を記述したプログラムにグラフィカルな入出力部を加えてヒューマンインタフェースを設けることとは異なる。後者の場合は前者の場合のように各事物が個別に形態も含めてモデル化されている訳ではない。

プログラム言語においては、図2-1（b）に示すようにデータの抽象化手法がモデリング手法に相当する。抽象化手法の歴史を振り返ると、データ型の導入で事物の状態の記述が可能となり、抽象データ型の導入で事物を状態と動作の対で記述することが可能となった。これはさらに、オブジェクト指向プログラミングへと発展した。これらは、事物の機構のみを記述したものである。事物の機構と様態を共に記述することは、オブジェクト指向プログラミング言語の1つであるSmalltalk-80で導入されたMVCプログラミングスタイルによって初めて可能となった。MVCは、対象をModelとViewとControllerに分けて記述するプログラミング手法である。Modelが対象の状態と動作を定義し、Viewが画面上における対象の見え方、つまり形態を定義する。ユーザは画面上でマウスを用いて、このViewに対して種々の操作を行うことができる。Controllerはこれらのマウス操作に対する反応を定義する。

データベースの分野においても、図2-1 (C) に示すように、最初は事物の状態のみが蓄積の対象であったが、状態と動作を持ったものとして事物を蓄積することが種々の応用分野で必要になり、抽象データ型を蓄積単位とした抽象データ型データベース (ADT データベース) が提案された。さらには、オブジェクト指向の考えを入れて、状態と動作をもったオブジェクトとして事物を記述し、これらのオブジェクトを蓄積単位としたオブジェクト指向データベースも提案されている。しかし、これらはいずれも事物の機構のみを記述したものであり、データベースを知的活動のツールシステムにも応用するには、機構のみでなく様態も含めて事物を記述し、これを蓄積単位とすることが望まれる。そこではMVCのようなモデリング手法が必要になる。メディアの記述には機構のみでなく様態も含めた記述を要し、MVCないしはそれに準ずるモデリング手法の採用が不可欠である。

ダイナミック・メディアのプログラミングには汎用言語を用いる場合と専用言語を用いる場合とがある。後者の場合、専用言語としてイベント駆動型言語が用いられることが多い。ダイナミック・メディアのプログラミングに用いるために用意されたイベント駆動型言語をスプリクト言語という。HyperCardのHyperTalkはこの例である。HyperCardでは、HyperTalkはイベント駆動の記述と、簡単な演算処理の記述にのみ用いられ、一般の複雑な処理の記述はC言語を用いて行われるが、HyperTalk プログラムはこの処理プログラムを外部コマンドを用いて呼ぶことができる。

ダイナミック・メディアは必ずしもマルチプロセスを必要としないが、ダイナミック・メディアをコンカレント・プロセスとして実現すると、各メディアを自律的なアクティブメディアとして実現することも可能になる。このような自律的メディアはエージェントと呼んで差し支えない。こうしたコンカレント・プロセスによるオブジェクト・プロセスを容易にするために、コンカレント・オブジェクト指向プログラミング言語が種々提案され始めている。このような言語がメディア・システムの構築に活用されるには、MVCのV (View) を自在に定義しプログラムできる機能の充実が必要である。

3. メディア・ツールキット

1) ツールキット

新しい種類のダイナミック・メディアを定義するのにスクリプト言語を用意することにより、メディアが容易にプログラム可能になった。一般に、メディアは種々のコンポーネントを含む。ドキュメントであれば、テキスト領域、画像領域、表領域、チャート領域等を含む。ドキュメントが動作機能を持つダイナミック・メディアであれば、これらの各領域もまたダイナミックである。さらに、ビデオ画像領域や動画領域なども含みうるように考えなければならない。メディアに指示を送るための各種指令ボタンも必要になる。

メディアに対してプログラム可能性を保証するだけでは、同じ機能の領域であっても、新しいメディアの一領域としてその領域が用いられる度ごとに、その領域の機能をメディア定義プログラムの一部としてプログラムする必要がある。しかし、いったんこれらの標準的機能を持った領域がメディアを構成する部品として定義されると、これらを組み合わせることにより、利用者は簡単に種々の機能領域を持った新しいメディアを合成することができる。このような部品をツールキットという。

ツールキットは、各部品をユーザインタフェース構築のための部品と考える場合にはユーザインタフェース・ツールキット（UIツールキット）と呼ばれる。各部品もまたメディアであると考えられる場合にはメディア・ツールキットと呼ばれる。この分類は設計者の意図を問題としており、必ずしも明確に分類できるわけではない。

ユーザインタフェース・ツールキットはここ10年間に誕生した数々のウィンドウ・システムにおいて、アプリケーションにグラフィカル・インタフェースを容易に追加できるようにし、アプリケーションの移植を容易にすることを目指して開発が盛んになってきた。

UIツールキットは次の3つの部分からなる。

(1) グラフィカル・ユーザインタフェースの既製部品の集合 (スクロール・バー、メニュー、ボタン、ダイヤルなど)。

(2) アプリケーション開発者がこれらの部品をアプリケーションに固有のコードで組み立てるためのフレームワーク。

(3) ウィンドウ・アプリケーションが必要とする各種通信サービスの集合。

ツールキット・フレームワークの主たる機能の一つは、マウス・クリックなどの入力イベントが画面上のどこで生じたかを分析し、それがスクロール・バー上であればスクロール・バー手続きにこのイベントを送り、プッシュ・ボタン上であればこのボタンに割り付けられた手続きにこのイベントを送る、イベント分岐ループ (event dispatch loop) の機能である。UI ツールキットを用いることにより、アプリケーション・プログラマは画面の仕組みや入力イベントに応じた処理、タイミングなどに対する考慮から解放される。ツールキットのなかには、さらに「組合せ」と「継承」の機能をフレームワークに含めて提供するものもある。組合せ機能は部品の結合を直接的に定義することを可能にする。継承機能を利用することにより、既存の部品をカスタマズして新しい部品を定義することができる。

2) これまでに開発されたツールキットの代表例

(1) Smalltalk : Xerox

(2) Tajo : Xerox

(3) Sun View : Xerox

(4) Xt : MIT

(5) Toolbox, Mac App, Hyper Card : Appie

(6) Andrew Toolkit : CMU

(7) OPENLOOK ツールキット

① X View

② Xt⁺

③ NDE (News Development Environment)

(8) Motif ツールキット :

(9) Next Step ツールキット : NeXT 社

(10) Intelligent Pad : 北海道大学

2.3 知的ハイパーメディアのマルチメディア化

2.3.1 序

コンピュータによる情報処理の歴史を振り返ってみると、まず数値処理に始まり次にテキスト処理、さらには画像、音声といったコード化できないデータまで扱えるようになった。ある情報をコンピュータを通して伝えようとするとき、今までの主流はその情報を数値あるいはテキストという形に変えて表現（伝達）していた。しかし、実世界に存在する情報の意味を知るためには数値やテキストの形を経由しないで（人間の思考や作業を介さないで）現実のものになるべく近い状態で直接伝えるに越したことはない。つまり、うぐいすの鳴き声を伝えるには「ホーホケキョ」という文字を見るよりは「 ～ ～ ！（うぐいすの鳴き声）」と声そのものを聞いた方がよいのである。また人間という好奇心旺盛、かつ、わがままな動物は、鳴き声を聞いただけでは飽き足らず、うぐいすの姿を見たいとか、飛び方（動いているところ）を見たいとか、同じような姿かっこうの他の鳥や同じような飛び方をする他の鳥を見たいといった、うぐいす一つから様々な情報提供の要求を出すものである。こういった要求（考え方そのものかも知れない）を満たすためにハイパーメディア概念が用いられるのであり、その要求される情報を人間に理解し易い状態で提供するためにコンピュータでマルチメディアを扱うことが必要となるのである。

現在、個々のメディアをコンピュータで扱うことは可能となっている。それをハイパーメディアの概念を導入することで有機的に結びつけ、ハイパーな世界を築くことが、「知的ハイパーメディアのマルチメディア化」におけるテーマであろう。

ハイパーメディアのマルチメディアには、次の2通りが考えられる。

1) テキストベースのハイパーメディアシステムのドキュメントをマルチメディア化する

2) 現存するメディアの操作に対してハイパーメディアの概念を導入する

「マルチメディア化する」を、ここでは、

- 1) multiple（マルチ）なメディアをサポートすること。
- 2) それらのメディアの統合が可能なこと。つまり、異なるメディアを任意に構造化できること。

3) 対話であること。つまり、情報を選択する順序とその詳細レベルをユーザーが選択できること。

と捉えることとする。

知的ハイパーメディアをマルチメディア化する場合に、その応用分野として設計・計画、通信・伝達、教育、編集、仮想体験、開発環境に分野わけし、それぞれに対して個人、オフィス、公共とその適用範囲を簡単に分類したのが表2-1である。

表2-1 マルチメディアの応用

	個人	オフィス	公共
設計・計画		図面管理 地図システム CAD	
通信・伝達		マルチメディアLAN	プロフィールサーバー
教育	対話型ビデオ マルチメディアCAI プレゼンテーション	言語教育・LL 電子マニュアル 商品紹介システム	電子百科辞典 案内板 医療診断用プレゼンテーション
編集	対話型ビデオ デジタルビデオ編集	DTP モンタージュ	
仮想体験	シミュレーション実験室 景観シミュレーション ビデオゲーム アーケードゲーム	人工現実感生成 フライトシミュレータ	
開発環境		CASE	マルチメディア出版支援ツール

2.3.2 マルチメディア

1. メディアの定義

マルチメディアとしてとらえられる「メディア」を分類したのが表2-2である。

表2-2 メディアのカテゴリー

カテゴリー	例
1 人間による情報の第一次表現	言語、非言語
2 データの物理的表現形態	テキスト、絵、図、表、音、動画
3 データの通信・蓄積媒体	紙、CD、VTR、光ファイバー
4 情報の伝達手段	
5 社会的な情報の伝達システム	

ここでは、「メディア」は上のカテゴリー2を指す。しかし、コミュニティ化を図ることによりカテゴリー4で既存のメディアを越える情報伝達メディアに進化して行くことが考えられる。

2. メディアの種類

次にカテゴリー2に属するメディアをマルチメディアの観点で分類してみる。

物理的形態としては、聴覚に訴えるものとして「音声」を、視覚に訴えるものとして「図面」「画像」「映像」をあげる。ここで、「図面」とはすべてコード化できる静的な図または絵であり、「画像」とはビットマップによる静止画であり、「映像」とは「図面」および「画像」に時間軸を設けた動的なものである。コンピュータによるハイパーテキスト処理を考慮すると、コード化できるかどうか大きなファクターとなる。各メディアの分類に対して各々のメディアの例をコード化が可能かどうかに分けたのが表2-3である。

表2-3 メディアの分類

	コード化が可能	コード化が困難
図面	・表 仕様表、時刻表、名簿、帳簿 ・グラフ ・図面 地図、設計図、回路図	
画像	・合成 CG	・録画 文書 絵画（人物、風景、静物） 写真（風景、報道、医療）
映像	・合成 CGアニメーション	・録画 ビデオ
音声	・合成 合成音、合成音成	・録音 音楽、音、声

3. メディアの特性

メディアを見たときの人の感じ方でその特性を分けると、表2-4のようになる。

表2-4 メディアの特性

特性	該当するメディア
実感性： 実物を思い浮かべる。実感する。 その場面に自分が参加しているような意識をもたせることができる。	写真、ビデオ、録音など 画像、映像、音声でコード化が困難なもの
象徴性： 物事の関係、数量的な比較などが視覚的に象徴されている。 物事がある側面からとらえたときの特性を直感的に理解させるのに適している。 その特性を操作するためのインタフェースとして有効である。	図、表、グラフ、メーターなどの人工的なメディア（図面）
情感性： 楽しさ、悲しさなどの気持ちを感じる。 感情がつけ加わることで、実感性も高まる。	音楽、声など聴覚的なメディアのうちコード化が困難なもの
説明性： 言葉や数式から指向を経て意味を理解する。 物事を、論理的・説明的に正確に理解させたいときに欠かせない。	テキスト、図、表など

2.3.3 メディア別技術

表2-2のメディアにテキストを加え「画像」と「映像」からコード化できるものとして「コンピュータグラフィクス（CG）」を分離したものを対象メディアとし、各メディアに対する処理をメディア別技術としてまとめたのが表2-5である。

表2-5 メディア別技術

	編集・操作	認識・整理	データ表現	検索
テキスト		文書要約技術 機械翻訳	テキストデータの圧縮	テキスト検索技術 キーワード抽出と自動リンク付け
図面	ペイントツール 図形、イラストの自動生成	図形の識別	多分木による図形データ表現	類似図形検索
画像	画像処理 静止画の編集ツール	文書画像の文字認識 OCR パターン認識 画像認識 画像の領域分割	文書構造の表現 高効率圧縮符号化	高速文書ページめくり 文字列検索 静止画検索

映像	動画の編集ツール 映像効果	動きの認識 ビデオのカット 切り	データ圧縮 DVI技術 フラクタル圧縮	動画検索
CG	CGによる動画制作(アニメーション) 光源、陰の操作 視点の操作	モデル化 情報の補完 ・隠れた部分 ・動画の場面間		
音声	音声合成 音質変換 ・男声/女声 ・子供/大人の声 音声データ編集 ツール	音声認識 ・indexing ・要素分離	音声データの圧縮	音声検索 (類似 波形検出)

2.3.4 システム化技術

マルチメディア化した知的ハイパーテキスト・システムを構築するためにはメディアごとの技術とは別にそれらのメディアを統合してシステムとしてまとめるための技術、あるいはシステムを処理・操作する技術を必要とする。これをシステム化技術として各技術要素ごとに以下に述べる。

1. プラットフォーム

ハイパーメディアに限らずマルチメディアを扱うときにそのプラットフォームの必要性は必ず話題に上ることである。プラットフォームの必要性は、Macintoshがマルチメディアを意識したシステムをプラットフォームとして実現し、現在マルチメディアの実用化に向けて一歩先を行っているのを見れば明らかであろう。ここでは、知的ハイパーメディアのマルチメディア化において必要と思われるプラットフォーム技術をシステムと言語に分けて挙げる。

システムに関しては、各メディアの特性を意識しないで使えるようにするためにマル

チメディア・マネジメントシステムなるものを、また各メディアを扱うアプリケーションを統合するためにオブジェクト指向のオープンアーキテクチャを実現する必要がある。

言語に関しては、シンセティック・プログラミング、オブジェクト指向プログラミングが主流である。さらに、各メディアの特性を意識しないための理想的な言語として、マルチメディアの共通表現言語なるものの実現が望まれる。

2. データ管理

マルチメディア・データを一元的に管理するために最終的に必要とされるのがマルチメディア・データベースである。マルチメディア・データベースは、まず各メディアのデータベース（画像データベース、ビデオデータベース、音声データベースなど）を構築しそれをマルチメディア・データモデルに基づいて統合化することで実現する。オブジェクト指向データベースはマルチメディア処理に向いているといわれるが、「オブジェクト指向データベース＝マルチメディアデータベース」ではなく、オブジェクト指向データベースをマルチメディアというアプリケーションに応用したものがマルチメディア・データベースということである。

3. リンク技術

ハイパーテキストを「テキストからテキストへのリンク」と定義するならば、そのマルチメディア化におけるリンク技術は「メディアからメディアへのリンク」を実現するための方法論と言うことになる。

リンク技術はイクスプリシット・リンクとインプリシット・リンクに分けて考えることができる。

＊イクスプリシット・リンク:選択肢、参照、索引など

＊インプリシット・リンク:検索（パターン認識）、推論、追跡など

コンピュータで実用的に扱えるメディア（テキスト、音声、図面、画像、映像）の中で、テキストのようにコード化が可能なものと困難なもので行えるリンクの種類や規模が違ってくるため、2者は区別して考えた方がよい（表2-3参照）。

コード化できるメディアに対しては、テキストに適用可能なリンク技術を適用し易い。コード化が困難なメディアに対しては、各々になんらかの形でコード情報を付加することでExplicit Linkを、音声認識や画像認識によりそのデータに含まれる意味情報を抜き出すことでImplicit Linkを実現する必要がある。

一方、ハイパーテキスト・システムがマルチメディアのデータに対してリンクを張ることによりリンク技術にも以下のような拡張が必要となってくる。

- * リンク操作の空間的解釈
- * Navigation Spaceのマルチメディア化
- * ノード、リンクのグラフィカル表現
- * 異種メディア間のリンク——例えば音声から何かへのリンク
- * メディアの相互関連——同期、差分
- * リンク構造を更新したときの波及箇所の検索

4. インタフェース

マルチメディアを扱う際にその対話性は重要な要素である。マルチメディアを操作・理解するためのインタフェース手法には、AI技術が重要なポイントとなる。さらにはデータの入出力操作に対してマルチメディアを駆使することも考慮しなければならない。

1) 技術要素

- * 知的対話技術（ユーザメンタルモデルなど）
- * パターン認識応用対話インタフェース（音声対話技術）
- * ビジュアルインタフェース技術
- * ビジュアル言語
- * 臨場感の評価技術（メディアの選択技術）

2) 入力手段

- * データグローブ
- * 触感
- * 身振り、言語
- * 操作の標準化

3) 出力手段

- * ハイビジョン
- * ホログラム
- * テキスト、静止画、動画の合成表示
- * 音声ガイド
- * 音響効果

5. 編集・出版

マルチメディア化したハイパーテキスト・システムの編集を考えると、マルチメディア・ドキュメントに対するオーサリングツールが必要である。

オーサリングツールはその目的や対象によって以下のふたつに分けて考えた方がよい。

- * 高品質なものを編集することを目的とする。対象はマルチメディア出版。
- * 簡単なもの手軽に編集できることを目的とする。対象はパーソナルユース。

技術要素としては次のようなものが考えられる。

- * 各種エディタの結合
- * メディア交換
- * メディア間のデータ交換
- * 視覚以外のデータの可視化
- * マルチメディアを格納する技術（例えばCD-ROM）
- * マルチメディア出版支援ツール

6. ネットワーク

マルチメディアのハイパーテキスト・システムの発展形として、既存のメディアを越える情報伝達メディアに進化して行くべきである。その際、ネットワーク技術は必須のものと考えられる。

ネットワークの応用

- * メディアの流通に利用
- * 分散化システムの構築
- * ISDN等マルチメディア処理に適した通信技術との融合

技術要素

* ネットワーク透過なマルチメディア技術（プロトコル）

2.3.5 システム例

マルチメディアに対応したハイパーメディア・システム及びパッケージを表2-6及び表2-7に示す。

表2-6 マルチメディアシステム

名称	開発者	目的・特徴
MEMEX	V. Bush	個人用マルチメディア技術文書ライブラリ
Intermedia	Brown大学	教育用、異なる種類のアプリケーションを統合化できる環境を提供することを目指す
HyperCard	Apple社	個人用情報管理
Palenque	Bank Street大学	児童教育用、カメラで画像をとって自分だけのアルバムをつくれる
知識メディアステーション	三菱電機	知識整理（Hypermedia+Interface）
Video Book	日本電気	英語学習ソフト、シナリオ言語あり、PC上でDVIが可能
LEAD	ビー・ユー・ジー	英語教育用、映像とセリフの同期がとれる
電脳机	YHP社	技術情報・図面管理
ビデオカット切り	北海道大学	Video Indexing、画面の切り替えごとに取り込む
Transmedia	北海道大学	文書画像管理・文字列検索（英文）

表2-7 マルチメディアパッケージ

名称	開発者	目的・特徴
マンダラの見方	大日本印刷、岩波書店	CD-ROM上に音声と画像を格納して曼陀羅の見方を説明する
MUSIC DATA CITY	SONY社	CD-ROMを用いた音楽最新情報
ボイジャー		HyperCardでCD-ROMを使う

2.3.6 知的ハイパーメディアと音声

1. 応用例

音声に関する各分野で例えば次のような応用システムが考えられ、実現されているものもある。

1) 語学教育

画面に表示されているストーリー（文章）中のあるセンテンスを選択すると、その訳文が表示され同時にインストラクタの音声が出力される。学習者は、その音声に従って発生し、さらに学習者の発生とインストラクタの発音とを比較して評価する。

2) 音楽情報

アーティスト名、タイトルなどによる検索に該当したCDの曲名一覧が表示され聞きたい曲を選択するとそのイントロが聞けるといったシステムがSONYのNewsやAppleのMac上のアプリケーションなどで出されている。

3) 電子百科辞典

広辞苑のデータをCD-ROMに納めて音声データを持つものに関しては。

それを選択することで鳴き声、音楽などを聞くことができる。

2. 入力・編集

表2-3における分類でコード化が可能な合成と、コード化が困難な録音とでは音声データの入力並びに編集の方法はおのずと違ってくる。

合成音声の場合は、元となるデータはテキストのコードとなるため音声データ特有の

入力方法なるものは存在せず、むしろそのコードデータから音声を合成する知的アルゴリズムの方に重きが置かれる。音楽、効果音等の合成音の場合は、楽譜による音譜コードの入力や波形操作による効果音の入力等があり、それぞれ入力する音声データの種類の合わせた入力・編集ツールが必要である。

録音データはマイクロフォン等で外界から入力するか出版されている音声メディア（CD-ROM等）をそのまま使うかすることになる。録音データのための編集ツールはどのようなものになるか、果してそのようなものが必要なのか、は分からない。

3. 加工

合成データはコード化されているため加工はしやすい。例えば、音声合成において、音質変換により、男声と女声の変換や、子供と大人の声の変換などができる。

コード化が困難なメディアの場合は加工するのが難しくなる。加工のためにはまず連続したデータを要素分離しなければならない。人間の操作によって機械的に要素に分けていくのも一つの方法だが、要素への分離さらにはそのindexingをどう自動化するかが課題となる。

音楽の場合は曲と曲の間のPauseを区切りと見なすといった程度でよく、音譜単位で要素化したければ合成による音（FM音源による演奏など）を用いればよい。人の声を要素分離するには一般に音声認識の手法を用いるが、声をすべてコード化するのではなく録音された声のデータをそのまま使用するのであれば要素の区切りとなる部分を識別するための音声認識となろう。

4. 検索

コード化されている合成音、合成音声についてはテキストサーチと同様の方法で検索は可能である。録音データに関しては生のデータで直接検索することを考えると、検索したい録音データと類似したデータを持つ録音データを捜さねばならずそのためには音声の類似波形検出機能が必要である。生データで直接検索する必要がある場合は、要素分離したものをindexingすることでそのindexを基に各要素へアクセスする形の検索を行えばよい。

5. 格納

音声データを格納、保存するのに、音声データの圧縮・伸張技術が必要となる。格納媒体としてはCD-ROMが一般的であり、XAというCD-R-OM仕様の圧縮モードでは映像やテキストに加え連続16時間（最大）の音声の再生が可能である。

2.3.7 知的ハイパーテキストと図面

ここで、「図面」とは全てコード化できる静的な図または絵であり、表・グラフ・地図・設計図などが挙げられる（表2-3参照）。

1. 応用例

1) 地図システム

まず広域の地図があってその中のある地域を指定することで（よりローカルな地図とリンクされている）徐々に的を絞っていき最後に詳細な情報の記されているローカルな地図に行き着く。さらに地図上の各建物などを指示することでその名称、説明といった詳細情報を見ることができる。

2) 電子本、電子マニュアル、電子百科辞典

テキストによる本文から図・グラフ・表へリンクが張っており本文から該当する図・グラフ・表を参照できる。つまり、テキストだけのハイパーテキストシステムのノードに図・グラフ・表を対応させたもの。

3) 図面管理

設計図などの図面を図面単位で管理し関連する図面間にリンクを張ることでハイパーテキストシステムを実現したもの。

2. 入力・編集

コンピュータによる描画ツールとしてペイントツールなるものがあり、これによって各種のXX表、XX簿、XX図、XXグラフを描くことができる。

さらに、各図面ごとに特化した専用ツールも存在しうる。こういったツールにより生成されるものは「座標どどこを始点としどどこを終点とする矩形」というふうな図形情報の集合をコード化したものである。

一方、図形を描くための元データから図形を自動生成することにより図面を作成する方法もある。例えばスプレッドシートのデータ群を元にグラフを自動生成したり、設計データから設計図を自動生成したりすることである。両者の違いは、図形情報そのものをデータとして持つのか、図面によって表現したい意味内容の方をデータとして持つのかの違いである。後者の方は、意味内容を図形情報に変換するためのモデル化が必要となる。

3. 加工

図面データは基本的に全てコード化されているため加工はしやすい。図形情報そのものを持っている場合は、コンピュータグラフィックスの図形処理により図形の変形などを行う。自動生成する場合は、元データを変更するとそれに伴って図面が更新される。例えば元データの変更に応じてグラフが更新される。

4. 検索

図面を図形の集合としてとらえるならばその検索には以下のものが考えられる。

1) 全く一致する図面を検索する

各図面は全てコード化されてデータを持っているため、そのコードが全て一致するかどうかを見れば、全く一致する図面を検索することはできる。

これは、グラフ・表・規格の決まっている設計図等比較的自由度の少ない図面に有効である。

2) 類似図形を検索する

地図・イラストなど自由度の高い図面に対しては、類似図形を検索することで条件に一致する図面を得るほうが現実的である。例として、意匠・商標などの大量の図形から、スケッチを入力して類似の図形が検索できるようになりつつある（電子技術総合研究所のシステム）。

3) ある特徴を持った図面を検索する

例えば、「50%を越えるパイを持つ円グラフを探す」とか、「交差点の数が1キロ平米当たり5個以上あるエリアマップを探す」といった条件に合う図面を検索する場合、図形識別をさらに知的なものにする必要がある。

2.3.8 知的ハイパーメディアと画像

ここで「画像」とは、文書画像・絵画・写真などのビットマップイメージの静止画である。

1. 応用例

1) 文書画像

*北海道大学のTransmediaシステム

画像テキスト文書をスキャナで読み取ってビットマップイメージをそのままデータとして持ち、キャラクタ単位で要素分解した情報も同時に持つ。

文書中の任意のワードの検索を行う際にワード単位でパターンマッチングを行い類似した文字列を候補として順次挙げていくもの。

*日立中央研究所の文書画像理解システム

文書イメージをスキャナで読み取り、そこから自動的に各種メディア（写真、図、表、適すと）を分離抽出し、自動Indexingを行ったり、テキストの論理構造を抽出することが可能になりつつある。これによりマルチメディア文書の内容検索が可能になる[Yashiro89]。

2) 電子百科辞典

広辞苑のデータをCD-ROMにおさめて画像データを持つものに関してはそれを選択することで対応する画像を見ることができる辞典。

3) 商品紹介システム

店頭で商品の画像データベースを検索するシステム。CD-ROMでパッケージ化したも

のとして“マンダラの見方”(大日本印刷と岩波書店の共同製作)がある。

4) 案内板システム

デパートや会社のビル案内を、画面上の见たいフロアや部屋を指示することで絵や写真の画像により表示するもの。

5) ゲーム

アドベンチャーゲームやロールプレイングゲームにおいて指示した内容(“右のドアを開く”とか“戦う”等)に応じて画面(画像)が移り変わっていくもので、今最も一般に広まっている知的(?)ハイパーテキストシステムと言えよう。

2. 入力・編集

入力デバイスとしては、カメラ・スキャナ(イメージリーダー)等がある。コンピュータグラフィクス(CG)によって入力・編集する方法もある。

3. 加工

画像データの加工としては一般に、取り込み時に発生するノイズ(ゴミ)の除去、イメージデータをベクトルデータに変換する(コード化)といったnormalizeが挙げられる。

CGによる画像では、光源や陰の操作ならびに視点の操作をすることも可能である。そのためには画像中に描かれている物体の3次元情報を持つ必要があり、隠れた部分の補完も必要となる。つまり、描く対象にたいして物体の位置・形状・大きさに関するモデル化を行うことにより、どこに光源があってもどこに視点があっても描く画像が一意に決まる様にしておくのである。

4. 検索

文書画像にたいして、文字認識やOCRを行うことで、文字列検索が可能となる。

静止画にたいして、画像認識の技術を用いて類似画像を検索することも可能となりつつある。

画像内の意味情報にしたがった領域分割とindexingをシステムが自動的に行うことができる。画像内の必要な部分を検索することができる。

5. 格納

画像を効率良く格納、保存するためにビットマップイメージ（モノクロ・カラー）の画像圧縮・伸張技術が必要となる。

2.3.9 知的ハイパーテキストと映像

ここで「映像」とは、「図面」や「画像」に時間軸が加わった動的なものであり、録画ビデオやCGアニメーションによる動画像のことをいう。

1. 応用例

1) 対話型ビデオ

元々シーケンシャルに収められているビデオ情報をランダムにアクセスすることによって、観る側がインタラクティブに操作できるようにするものでスタンフォード大学やMITメディアラボ等で研究されている。

2) 景観シミュレーション

景観情報をビデオに収めそれをコンピュータディスプレイ上であたかも実際に観て回っているように視点や距離などを変えてみるができるもの。代表的なシステムとして、The Bank Street College (New York) のPalenque Projectがある。Palenqueでは、ビデオ録画を観ながらカメラで撮影し（静止画をとる）それを自分のアルバムに登録するといったこともできる。

3) フライトシミュレータ

コンピュータグラフィックスを用いた仮想体験システムのひとつ。飛行機（戦闘機）の飛行に関するモデル化を行い、操縦の仕方や外界の条件に応じて飛行状態をアニメーションで表現する。

2. 入力・編集

入力デバイスとしては、録画のためのビデオカメラ、カメラ、スキャナ等がある。コンピュータグラフィクス (CG) によって入力・編集する方法もある。

映像の作り方には以下の3通りが考えられる。

1) ビデオ録画をそのまま動画情報として使用する。

2) 図面・画像といった静的な情報を組み合わせて、あるいは動画の場面を一部だけ入力して映像を作成する。

3) あるモデルに基づいてCGアニメーションにより映像を作成する。

2) は基になる画像または映像の間を補完する技術が必要となり、そうした加工・補完の技術で効率良く映像を作れるような編集ツールが望まれる。

3) は表現したいものを正確にモデル化することが重要である。このモデル化は4D (3D+時間) のものであり、これができるいればどのような表現の仕方 (見方) でもできるすなわちハイパーな処理が可能となる。

3. 加工

例えば、アニメーションの1カットごとの画像を作成し、各画像間の補完によりなめらかな動きの動画 (アニメーション) を作ることができる。

連続した (時間軸を持つ) 動画を要素分けによってindexingすることも重要な加工技術である。ひとつの手段として、北海道大学のビデオカット切りの手法がある。これは、ビデオ映像の中で場面が切り換わるタイミングで自動的に区切りを付けるもので、これを実行すると一連のビデオ映像の中に出てくる場面の一覧を表示したり、その中の選択した場面からビデオをスタートさせたり、といったことができるようになる。

4. 検索

検索のためには映像をなんらかの形でindexingしなければならない。オーソドックス

な方法として、入力または編集の時点においてマニュアル操作により（オーサリングツールなどを用いて）indexingする方法がある。この操作をシステムが自動的に行うものとしてビデオのカット切りによるindexingがある。

5. 格納

映像データを格納するうえでデータ圧縮・フラクタル圧縮などの圧縮・伸張技術が必要となる。DVI技術はCD-ROMにおけるデータ圧縮技術の標準を目指したもので注目に値する技術である。さらに、CD-ROMXAやCD-Iといった標準規格があり、いずれにしろ映像という大量のデータを扱うためには不可欠の圧縮技術である。

2.3.10 異種メディアの統合技術

よく、マルチメディアあるいはハイパーメディアの技術はまだデモの段階にとどめていられるが、その原因のひとつにマルチメディアが真の意味で統合化されていないことが挙げられよう。では“真の意味の統合化”とはどういうことだろうか。本節では“統合化”に必要なと思われる技術、1) マルチメディアデータベース、2) 異種メディア間のリンク、3) メディア交換、について述べる。

1. マルチメディア・データベース

横浜国立大学の有澤博教授は「現状でも、映像や音声を扱うことができるデータベースシステムはある。しかし、それらは本来のマルチメディア・データベースではない。真のマルチメディア・データベースでは、データのフォーマットがアプリケーションに依存することなく、あらゆるメディアデータが統合され、それらを構造的に、また抽象的に扱えなければならない。」と言う。こういったマルチメディア・データベースを構築するためのハードウェア・ソフトウェアは整備されつつある。今のところオブジェクト指向データベースがマルチメディア・データベースに向いているという一致した見解はあるが実用システムはまだ存在せず、目下ホットな研究課題と言えよう。現在のマルチメディア用のソフトでは、アプリケーションがデータの操作と格納を制御しており、アプリケーション間のデータ交換は統合化ユーザインタフェースソフトが行っているため以下のような欠点があると、同氏は指摘する。

* アプリケーションが共通なオブジェクトとして認識されていないため、メディアが統合化されていない。

* オブジェクトの操作が統一されていない。

* オブジェクトが構造化できない。つまり、オブジェクト間にリンクが張れない。

* 他のアプリケーションで作られたオブジェクトを検索できない。

* オブジェクトの格納法がアプリケーション固有なため、再利用できない。

従って、マルチメディア・データベースに必要な技術要素は次のようになる。

* マルチメディアのデータ表現を統一するためのモデル論の確立。

* データ操作の統一（ユーザインタフェースの向上のため）。

* 内部表現形式の統一（データ互換性のため）。

* 大規模記憶メディアの扱い（特に動画）。

2. 異種メディア間のリンク

異種メディア間でリンクを張るうえで、技術的な課題となるのが動きのあるメディア（音声・動画・アニメーション）に対するリンクをどう張るかということである。それぞれのメディアに対するリンクは各認識技術によりそのデータに含まれる意味情報を抜き出すことで行う必要があるが、ここで問題となるのは、どのメディアとどのメディアをどういうふうにリンクさせる（関連づける）のか、そしてその相互関連はどういう意味を持つのかということであろう。

メディアの相互関連のしかたとして、時間軸を持つデータ（映像・音声）と他メディアとの同期をとる問題がある。以下にメディア間で同期をとる例を示す。

1) ビデオとテキスト

ビデオ（映像と音声）とその内容に関するテキストデータがあって、ビデオの流れに同期してテキストデータが追従していく。BUG社のLEADシステムでは英語教材のビデオで人が話すとそれに対応した字幕（せりふ）が表示される。

2) 映像と音声

語学の発音の教育で、口の開け方や舌の動かし方を映像で流し、その映像に同期して

音（声）が提示される。

3. メディア変換

MIT メディア・ラボのNegloponte 博士は「マルチメディアを名乗るなら、五感の間で情報を自由に変換できるようにしなければならない。」と言う。つまり、どんなに多種類のメディアを（同時に）扱うことができてそれらのメディアが単に混在しているだけではだめで、それらのメディアが互いに情報を変換しあってこそ人間にとって意味を持つということだ。例えば車のナビゲーションシステムで、地図をCD-ROMに入れ自動車の位置検出システムと組み合わせることは、画像と数値データというメディアの混在ではある。しかし両手をハンドルにとられ、目で窓の外を見なければならぬ運転者にはなんら貢献しない。ところが、これに音声認識とAIの手法を付加し運転者の声にたいして音声で答えることで、はじめて、人間にとって意味を持つわけである。ここでは、地図と位置検出という視覚に訴える情報が音声という聴覚に訴える情報に変換されている。

上記例でも分かるように、メディア変換には各メディア情報に関するパターン認識（文字認識、画像認識、音声認識）及びそれに付随するAIの技術が重要な要素となる。

参考文献

- * [Akscyn 88] R. M. Akscyn, D. L. McCracken, E. A. Yoder, "KMS:A Distributed Hypermedia System for Managing Knowledge in Organizations " *Communications of ACM*, Vol.31, No.7, pp.820-835, 1988.
- * [Catlin 89] T. Catlin, P. Bush, N. Yankelovich, "InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration," *Proceedings of the HYPERTEXT'89*, pp.365-378, 1989.
- * [Conklin 87] J. Conklin, "Hypertext: An Introduction and Survey," *COMPUTER*,
- * [Hypertext 88] "Hypertext on Hypertext (IRIS Version)," ACM Press, 1988.
- * [Yankelov 88] N. Yankelovich, B. J. Haan, N. K. Meyrowitz, S. M. Drucker, "Intermedia: The Concept and the Construction of a Seamless Information Environment," *COMPUTER*, Vol.21, No.1, PP.81-96, 1988.

- * [ISO 88] "Volume and File Structure of CD-ROM for Information Interchange," ISO 9660, 1988.
- * [共同計画 89] 共同計画出版事業部、世界CD-ROM総覧、紀伊國屋書店、1989.
- * [伊藤 89] 伊藤、本位田「プロトタイプینگ支援ツール」情報処理学会誌、Vol.30, No.4, pp.387-395, 1989.
- * [上田 88] 上田、森、久保、「組版機能を備えたHypermediaシステム (1)」第36回情報処理学会全国大会予稿集、pp.2301-2302, 1988.
- * [小川 89] 小川、佐原、金子、「音声、動画を含むハイパーメディア作成支援システム」情報学シンポジウム、pp.43-52, 1989.
- * [Akscyn 88] R. M. Akscyn, D. L. McCracken, E. A. Yoder, "KMS:A Distributed Hypermedia System for Managing Knowledge in Organizations," *Communications of ACM*, Vol.31, No.7, pp.820-835, 1988.
- * [Conklin 87] J. Conklin, "Hypertext: An Introduction and Survey," *COMPUTER*, Vol.20, No.9, pp.17-41, 1987.
- * [Halasz88] F. G.Halasz, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedeia Systems," *Communications of ACM*, Vol.31, No.7, pp.836-855, 1988.
- * [Marchion88] G. Marchionini, B. Shneiderman, "Finding Facts vs. Browsing Knowledge in Hypertext Systems," *COMPUTER*, Vol.21, No.1, PP.70-80, 1988.
- * [Streitz 89] N. A. Streitz, J. Hannemann, M. Thuring, "From Ideas and Augments to Hyperdocuments: Travelling through Activity Spaces," *Proceedings of the HYPERTEXT'89*, pp.343-364,1989.
- * [Yankelov 88] N. ankelovich, B. J. Haan, N. K. Meyrowitz, S. M. Drucker, "Intermedia: The Concept and the Construction of a Seamless Information Environment," *COMPUTER*, Vol.21, No.1, PP.81-96, 1988.
- * [Salton 83] G.Salton and M.J.McGill, "Introduction to Modern Information Retrieval," McGraw-Hill, New York, 1983.
- * [Fujisawa 86] H.Fujisawa, A.Hatakeyama, and J.Higashino, "A Personal Universal Filing System Based on the Concept-Relation Model," *Proc. First Int. Conf. Expert Database Systems*, pp.31-44, 1986.
- * [加藤 89] 加藤、藤澤、大山、川口、畠山、兼岡、秋沢, "全文検索用テキスト

- サーチマシンの開発”，電子情報通信学会研究会資料 DE89-38, pp.17-24, 1989.
- * [Higashino 86] J. Higashino, H. Fujisawa, Y. Nakano and M. Ejiri, "A Knowledge-based Segmentation Method for Document Understanding," *Proc. 8th Int. Conf. Pattern Recognition*, pp.745-748, 1986.
 - * [Shima 88] Y. Shima, T. Murakami, J. Higashino, Y. Nakano and H. Fujisawa, "A Segmentation Method of Color Document Images for Multimedia Contents Retrieval," *Proc. RIAO 88, User-oriented Content-based Text and Image Handling (AFIPS)*, Cambridge, vol.2, pp.1001-1008, March 1988.
 - * [Yashiro 89] H. Yashiro, T. Murakami, Y. Shima, Y. Nakano and H. Fujisawa, "A New Method of Document Structure Extraction Using Generic Layout Knowledge," *Proc. Int. Workshop on Industrial Applications of Machine Intelligence and Vision (MIV-89)*, pp.282-287, April 1989.
 - * [田中 88-1] 田中「オブジェクト指向データベース・システムーその背景と概念」
b i t, Vol.20, No.6, pp.83-90, 1988.
 - * [田中 88-2] 田中, 吉川「オブジェクト指向データベース・システムー研究開発の動向」
b i t, Vol.20, No.7, pp.51-61, 1988.
 - * [日経89]「オブジェクト指向データベース」日経コンピュータ、5月22日号、
pp.108-121, 1989.
 - * [Andrews 87] T. Andrews, C. Harris, "Combining Language and Database Advances in an Object-Oriented Development Environment," *Proceedings of OOPSLA*, pp.430-440, 1987.
 - * [Bancilho 88] F. Bancilhon, "Object-Oriented Database Systems," *Proceedings of Principles of Database*, pp.152-162, 1988.
Vol.20, No.9, pp.17-41, 1987.
 - * [Banerjee 87] J. Banerjee, H. Chou, J. F. Garza, W. Kim, D. Woelk, N. Ballou, "Data Model Issues for Object-Oriented Applications," *ACM Transactions on OIS*, Vol.5, No.1, pp.3-26, 1987.
 - * [Kim 88] W. Kim, N. Ballou, H. Chou, F. Garza, D. Woelk, "Integration an Object-Oriented Programming System with a Database System," *Proceedings of OOPSLA*, pp.142-152, 1988.

- * [Kim 89] W. Kim, E. Bertine, F. Garza, "Composite Object Revised," *Proceedings of ACM SIGMOD*, pp.337-347, 1989.
- * [Smith 87] K. Smith, S. B. Zdonik, "Intermedia: A Case Study of the Difference Between Relational and Object-Oriented Database Systems," *Proceedings of OOPSLA*, pp.452-465, 1987.
- * [Lecluse 88] C. Lecluse, P. Richard, F. Velez, "O2, an Object-Oriented Data Model," *Proceedings of SIGMOD*, pp.424-433, 1988.
- * [Penney 87] D. J. Penney, J. Stein, "Class Modification in the GemStone Object-Oriented DBMS," *Proceedings of OOPSLA*, pp.111-117, 1987.
- * [(財)日本情報処理開発協会：知的ハイパーテキストに関する調査研究]
- * [安西 89] 安西祐一郎：認識と学習、岩波講座ソフトウェア科学16、岩波書店、1989.
- * [大須賀 87] 大須賀、佐伯共編：知識の獲得と学習、知識工学講座 3、オーム社、1987.
- * [木内 88] 木内伊都子、畠山敦、藤澤浩道：「知的ファイリングシステムのビジュアルインタフェース」、情報処理学会、文書処理とヒューマンインタフェース研究会資料、21-3、1988.
- * [斎藤 81] 斎藤収三、中田和男共著：音声情報処理の基礎、オーム社、1981.
- * [信学会誌 87] 電子情報通信学会誌、音声情報処理特集、Vol. 70, No. 4, 1987, pp331-438.
- * [情報処理 89] 情報処理学会誌、大特集「自然言語理解」、Vol. 30, No. 10, 1989, pp.1141-1267.
- * [中田 78] 中田和男：パターン認識とその応用、コロナ社、1978
- * [長尾83] 長尾真：画像認識論、コロナ社、1983
- * [Barr 81] A. Barr and E. A. Feigenbaum (Ed.), "The Handbook of Artificial Intelligence," Vol. 1-3, William Kaufmann Inc., 1981-1982.
- * [Bobrow 77] D. G. Bobrow and T. Winograd, "An Overview of KRL, a Knowledge Representation Language," *Cognitive Science*, 1, 1977, pp.3-46.
- * [Brachman 85a] R. J. Brachman and J. G. Schmolze, "An Overview of the KL-ONE

- Knowledge Representation System," *Cognitive Science*, 9, 1985, pp.171-216.
- * [Brachman 85b] R. J. Brachman, V. P. Gilbert, and H. J. Levesque, "An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of KRYPTON," *Proc. IJCAI-85*, Los Angeles, CA, 1985, pp.532-539.
 - * [Brachman 87] R. J. Brachman and H. J. Levesque, "Tales from the Far Side of KRYPTON," *Proceedings from the First International Conference*, Benjamin/Cummings Publishing Company, Inc., 1987, pp.3-43.
 - * [Davis 87] R. Davis, "Knowledge-Based Systems: The View in 1986," in W. E. Grimson and R. S. Patil (Ed.), "AI in the 1980s and Beyond," Cambridge, Mass., MIT Press, 1987, pp.13-41.
 - * [EDR 88] EDR TECHNICAL REPORT, 「EDR電子化辞書」、日本電子化辞書研究所、TR-016, 1988
 - * [Kato 88] T. Kato, H. Shimogami, T. Mizutori, and K. Fujimura, "TRADEMARK: Multimedia Database with Abstracted Representation on Knowledge Base," *Proc. 2nd Int. Symposium Interoperable Information Systems, ISIIS'88*, Tokyo, 1988, pp.245-252.
 - * [Fujisawa 86] H. Fujisawa, A. Hatakeyama, and J. Higashino, "A Personal Universal Filing System Based on the Concept-Relation Model," *Proceedings of the First Int. Conf. Expert Database Systems*, Charleston, 1986, pp.31-44.
 - * [Fujisawa 88] H. Fujisawa, H. Yashiro, J. Higashino, Y. Shima, Y. Nakano, and T. Murakami, "Document Analysis and Decomposition Method for Multimedia Contents Retrieval," *Proc. 2nd Int. Symposium Interoperable Information Systems, ISIIS'88*, Tokyo, 1988, pp.231-238.
 - * [Marchionini 88] G. Marchionini and B. Shneiderman, "Finding Facts vs. Browsing Knowledge in Hypertext Systems," *Computer*, Vol. 21, No. 1, Jan. 1988, pp.70-80.
 - * [Minsky 75] M. Minsky, "A Framework for Representing Knowledge," in P. Winston (Ed.), "The Psychology of Computer Vision," New York, McGraw-Hill, 1975, pp.211-277.
 - * [Patel 84] P. F. Patel-Schneider, R. J. Brachman, and H. J. Levesque, "ARGON: Knowledge Representation meets Information Retrieval," *Proceedings of the First Conference on Artificial Intelligence Applications*, 1984, pp.280-286.

- * [Quillian 68] M. R. Quillian, "Semantic Memory," in M. Minsky (Ed.), "Semantic Information Processing," Cambridge, Mass., MIT Press, 1968, pp.227-270.
- * [Raphael 68] B. Raphael, "SIR: A Computer Program for Semantic Information Retrieval," in M. Minsky (Ed.), "Semantic Information Processing," Cambridge, Mass., MIT Press, 1968, pp.33-145
- * [Schank 73] R. C. Schank, "Identificaiton of Conceptualization Underlying Natural Language," In Schank and Colby (Ed.), "Computer Models of Thought and Language," San Fransisco, Freeman, 1973, pp.187-247.
- * [Schank 77] R. C. Schank and R. P. Abelson, "Scripts, Plans, Goals, and Understanding," Hillsdale, N.J., Lawrence Erlbaum, 1977.
- * [Tanaka 88] Y. Tanaka and H. Torii, "Transmedia Machine and Its Keyword Search over Image Texts," *Proceedings of Conference on User-Oriented Content-Based Text and Image Handling, RIAO-88*, Cambridge, Mass., 1988, pp.248-258.
- * [Tou 82] F. N. Tou, M. D. Williams, A. Henderson, T. Malone, "RABBIT: An Intelligent Database Assistant," *Proceedings AAAI-82, Pittsburgh, Pennsylvania, American Association for Artifcail Intelligence*, 1982, pp.314-318.
- * [Winograd 72] T. Winograd, "Understanding Natural Language," New York, Academic Press, 1972.
- * [Woods 70] W. A. Woods, "Transition Network Grammars for Natural Language Analysis," *Communications ACM*, 13, 1970, pp.591-606.
- * [Woods 73] W. A. Woods, "Progress in Natural Language Understanding: An Application to Lunar Geology," *AFIPS Conference Proceedings 42, 1973 National Computer Conference*, AFIPS Press, pp.441-450.
- * [Yashiro 89] H. Yashiro, T. Murakami, Y. Shima, Y. Nakano, and H. Fujisawa, "A New Method of Document Structure Extraction using Generic Layout Knowledge," *Proc. Int. Workshop Industrial Applications of Machine Intelligence and Vision, MIV-89*, Tokyo, 1989, pp.282-287.
- * [Krasner 88] G. Krasner, A Cookbook for using the Model-View-Controller User Interface Paradigm in Smalltalk-80, 1988.
- * [Krasner 85] G. Krasner, Smalltalk-80 Bits of History, Words of Advice, Addison-Wesley ,

1985.

- * [Johnson 89] Jeff Johnson, The Xerox Star: A Retrospective, *Computer*, Sept. 1989, pp11-29, 1989.
- * [Cox 89] B. J. Cox著、前川 守 監訳、オブジェクト指向のプログラミング: (株) トッパン、1988.
- * [Goldberg 87] Adele Goldberg、相磯 秀夫 監訳、Smalltalk 言語詳解: オーム社、1987.
- * [Adobe Systems 89] Adobe Systems、アスキー出版 監修、(株)アスキー、PostScriptチュートリアル&クックブック、1989.
- * [Ishikawa 88] 石川 裕、オブジェクト指向並行プログラミング言語、情報処理、Vol.29, No 4, pp325-333, 1989.
- * [Campbell 88] Brad Campbell & Joseph M. Goodman, *Communications of the ACM*, Vol.31 No.7, pp856-861, 1988.
- * [増永 87] 増永良文、マルチメディア・データベース総論、情報処理、Vol.28 No.6, pp671-683, 1987.
- * [Yashiro 89] Hiroshi Yashiro, Tatsuya Murakami, Yoshihiro Shima, Yasuaki Nakano and Hiromichi Fujisawa, *International Workshop on industrial Applications of Machine Intelligence and Vision(MIV-89)*, Tokyo, April 1989.

3章 データベース構築技術における知的ハイパーメディア

3.1 データベース・システム

現在の情報システムは、データベースシステム（DBS）を基本として構成されている。データベースシステムとは、利用者にあるデータモデルを提供するシステムである。

3.1.1 データベースシステムとは

データベースシステムは、ある組織体（または企業体）が組織活動を行うための必要とする情報を、統合化して管理するシステムである。組織体の活動は、複数の応用から構成される。例えば、学校を例として考えてみる。学校の業務として、学生の成績管理、カリキュラム管理、授業料管理、教室管理といったものがある。成績管理では、各学生とその成績のデータが必要であり、カリキュラムでは、授業科目、先生、教室、授業時間といったデータが必要となる。これに対して、データベースシステムでは、学校という視点にたって、学校全体の業務が必要とするデータの構造を一つに定めて、これらのデータを学校として管理していくものである。即ち、組織体内の全応用が必要とするデータを、統合化したシステムである。統合化（integration）とは、複数の応用がそれぞれの形式で必要としているデータを、組織体の目的に合うような一つの形式にまとめることである。このことは、各応用単位に管理されていたデータを、組織体としてデータを集中管理することである。

このように、統合化の概念に基づいて、データベースシステムが登場してから、既に、20～30年になってきている。60年代から70年代にかけては、情報システムは、大型の汎用コンピュータを中心として、利用者は、端末を通じてデータベースシステムを利用していた。1980年代から、高性能なワークステーションが情報システムの主要な要素となり、かつLAN、OSIといった標準化された通信により相互結合されてきている。こうした情報システムでは、データベースシステムは大型コンピュータ内だけでなく、各利用者は、自分のワークステーションにも構成される。例えば、大型コンピュータ内のデータベースシステム内のデータをワークステーションにダウンロードして利用する例がある。このように統合化の概念のもとに、一つのコンピュータ内に集中化されてきたデータベースシステムが80年代から分散化されだしてきている。各応用単位

に管理されてきたデータベースシステムが、情報システム全体としての緩やかな管理に従っていく形態であり、これを連邦化（federation）という。

3.1.2 アーキテクチャ

次に、データベースシステムの論理的な構成について考える。データベースシステムの論理構成を考えるには、次の二つの観点がある。

1. 利用者からみたデータ構造と操作演算の階層構造。
2. データベースシステムがどのような要素から構成されるかというデータベースシステムのアーキテクチャ。

まず、1.の点について考える。

1.三階層構造〔ANSI／SPARC〕

データベースシステムは、複雑なシステムであり、ディスク装置、磁気テープといった二次記憶装置内のデータから、利用者が実際の応用において用いる帳票、伝票、設計図といったデータまで様々なデータの表現とこれに対する操作がある。利用者には、応用が必要とするデータ構造とこれに対する操作だけを提供することが望ましい。従って、応用が必要とするデータ構造とこれに対する操作だけを提供することが望ましい。従って、二次記憶装置内のデータから応用が必要とするデータを抽象化する必要がある。このためには、データベースシステムでは、利用者に対して、以下の独立性を提供する必要がある。

- 1) データ独立性（data independence）
- 2) 論理独立性（logical independence）

データ独立性とは、応用がデータの物理的な格納構造、アクセス方法と独立であることである。言い換えると、データの物理的な記憶構造とアクセス方法が変化しても、応用には影響がないことである。利用者に、データの論理的な側面だけを示し、データがどのように実際の計算機内で実現されているかを見せないことである。

次の論理独立性とは、応用の必要とするデータと組織体全体が関心を持つデータとを分離することである。データベースシステムは、組織体全体のデータを統合化したシ

システムである。組織体の各応用は、組織体全体が関心を持つデータの一部だけを利用している。従って、組織体全体のデータ構造とアクセス方法が変化しても応用に影響を与えないことが望ましい。これを論理独立性という。

以上の二つの独立性は、抽象化の段階を示す三つの階層構造を設けることにより解決できる [図3-1]。

- (1) 外部層 (external layer)
- (2) 概念層 (conceptual layer)
- (3) 内部層 (internal layer)

内部層は、最下位の階層であり、データの物理的な格納構造とアクセス方法に関係している階層である。概念層は、内部層の上位の階層であり、組織体全体にとって意味のある論理的なデータ構造と操作方法が上位層に提供される。この階層の利用者は、データの物理的な特性を意識せずに、論理的なデータ構造とこれに対する論理的な操作を行える。即ち、データ独立性が提供される。外部層は、概念層の上位にあり、各応用毎に、応用が必要とするデータ構造と操作方法が提供される。ここでは、各応用は、自分の必要とするデータだけを操作でき、論理独立性が提供される。三層の階層構造は、米国規格協会 ANSI により、データベースシステムの標準の階層構造となっている。この階層構造は、データベースシステムを理解するときの基本となるものであり、重要である。

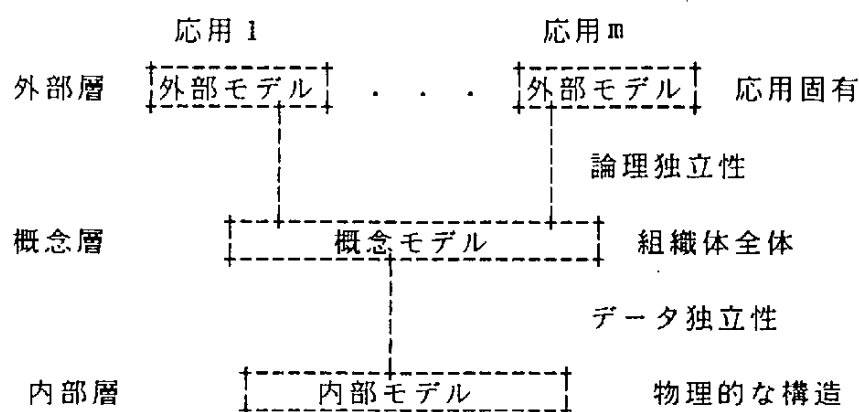


図3-1 三階層構造

以上の三階層構造を考えることにより、次のような利点がある。

* データ独立性と論理独立性を独立に達成できる。

* 応用の多様化と物理構成の変化に対する適応が容易となる。応用の多様化に対しては、概念層と応用層の変換を用意すればよい。計算機の世代交代などによる物理構成の変化に対しては、物理層と概念層の間の変換だけを変化させればよい。

2. データベースシステムの構成

データベースシステムがどのような要素から構成されるかについて考える。

データベースシステムとは、以下の構成要素から構成されるシステムである。

〔図3-2〕

- 1) データベース (database)
- 2) 利用者 (user)
- 3) 応用プログラム (application program)
- 4) データベース管理システム (DBMS : database management system)
- 5) データ辞書／ディレクトリ (DD / D : data dictionary / directory)

1) データベース

データベースは、データの集合である。データは、コンピュータのディスク装置等の二次記憶装置内に記憶される。

2) 利用者

利用者は、データベース内のデータを利用するものである。利用者には、データベースシステムに対する以下の三種の役目 (role) がある。

- (1) 一般利用者 (casual user 又は end user)
- (2) 応用プログラマ (application programmer)
- (3) データベース管理者 (DBA : database administrator)

応用プログラマは、コンピュータの専門家であり、各応用に対する応用プログラムを作成する。一般利用者は、コンピュータの非専門家であり、秘書、事務員、設計者等である。一般利用者は、応用プログラムを用いて、自分の業務を行う。データベース管理者は、データベースシステム全体の管理を行う。データベース管理者は、データベースシステム全体の最高権利者である。

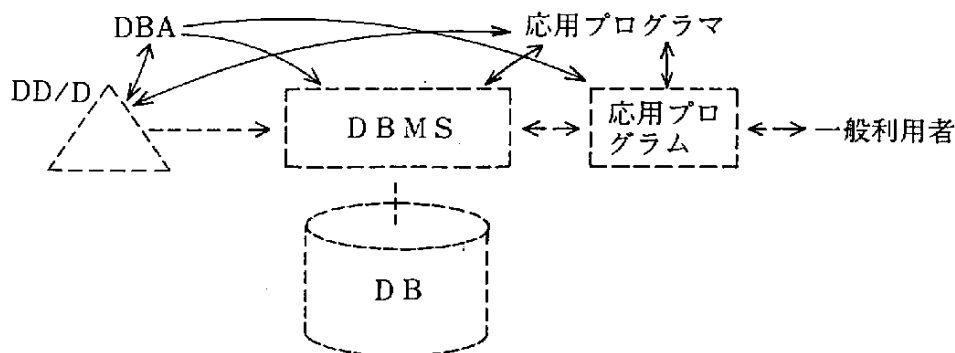


図3-2 データベースシステムの構成

2. データ構造

まず、データモデルのデータ構造について考える。データ構造 (data structure) とは、データがどのような要素から構成され、各要素が互いにどのような関係があるかを示している。

1) スキーマとスキーマ例

データ構造には、スキーマ (schema) と例 (instance) という二つの側面がある。スキーマとは、データ構造のなかで時間不変な論理的なものである。これに対して、スキーマの例とは、スキーマに実際の値を与えたもので、時間可変なデータ構造である。例えば、ファイルの名前、項目構成はスキーマであり、レコードの集合がスキーマの例である。データ構造を、スキーマとその例にわけるとはデータベースの設計時に重要である。例は又、実現値 (occurrence)、状態 (state) ともいわれる。

データベースシステムのデータ構造は、スキーマとその例であるデータベースから構成される。

データベースシステムでのデータ構造としては、網型データ構造と関係型データ構造がある。

データベースシステムの構造は、一般に、少量のスキーマと、大量のスキーマ例から構成されている。

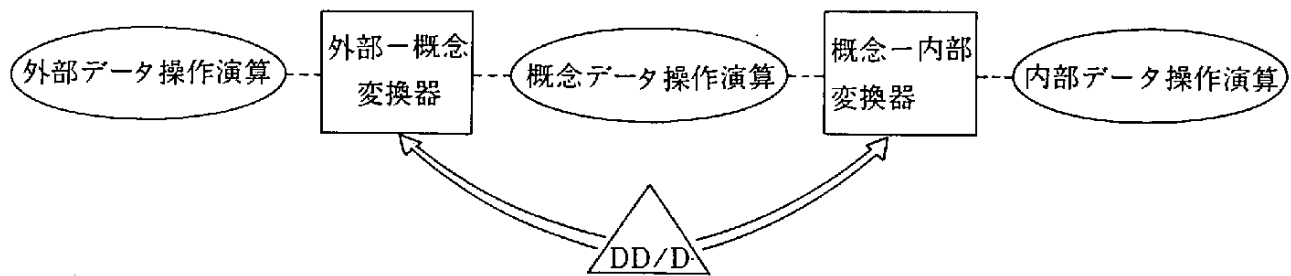


図3-3 層間写造

ここで、データベースのインテグリティ制約とは、データベース内のデータが保つべき制約である。以上の目標を達成するために、データベース管理システムは、以下の制御機構を有している。

- 1) 同時実行制御 (concurrency control)
- 2) 信頼性制御 (reliability control)
- 3) 質問最適化 (query optimization)
- 4) 安全性制御 (security control)

データベースシステムの目的の一つは、複数の利用者により、データベース内のデータを共有することである。同時実行制御とは、複数の利用者のもとでデータベースのインテグリティ制約を保ち、データベースシステムのスループットを向上させるための制御である。信頼性制御とは、データベースシステムがこうむる種々の障害（例、電源断、ディスクのクラッシュ）に対して、データベースのインテグリティ制約を保つための制御である。安全性制御とは、不正な利用者により、データベースと応用プログラムを不正に利用されることを防止するための制御である。質問最適化制御とは、利用者からのデータ操作演算を、応答時間を最小とするように物理的なデータ操作演算に変換するための制御である。

また、DBMSは、利用者にインタフェースを提供する。

5) データディクショナリ／ディレクトリ

データディクショナリ／ディレクトリ (DD/D: data dictionary/directory) は、データベースシステム全体の管理を行うためのデータベースシステムである。データベースシステムがどのようなデータ、利用者、プログラムを持ち、それらの間にどのような関係があるかについての情報をデータとして持っている。DD/Dを用いて、データベース管理者は、データベースシステムの管理を行うと同時に、利用者間の通信のために用いられる。

3.1.3 データモデル

利用者から見たデータベースシステムとは何かについて考える。利用者から見たデータベースシステムのモデルをデータモデル (data model) という。ここでは、データモデルとは何かについて考える。

1. データモデルとは

人間が物事をどのように認識するかを考える。例えば、自動車を考えてみる。自動車は、車体、タイヤ、ハンドル、エンジンといった種々の部品から構成される。自動車とは、こうした部品から構成され、個々の部品が互いにどのように関係づけられているかが分かることがまず必要になる。しかし、これだけでは自動車を理解したことにはならない。自動車であるためには、自動車としての動作を行える必要がある。例えば、ブレーキを踏めば、自動車は停止し、アクセルを踏めば自動車は加速する。また、ハンドルを左に回せば、自動車も左にまがる。更に、自動車が動作できるためにはガソリンが入っていなければならないというように、自動車が自動車として動作できるためには、ある一定の条件を満足しなければならない。自動車を理解するためには自動車がどのような部品から構成され、それぞれがどのように結合されているかという構造 (structure) と、自動車がどのように振る舞うかという動作、更に自動車が満足せねばならない条件 (これをインテグリティ制約という) の二点が理解される必要がある。データベースシステムも同じように、データベースシステムが保有しているデータ構造、これに対して何を行えるかというデータ操作演算、及びデータ構造が満足すべきインテグリティ制約が理解されるならば、利用者はデータベースシステムが何かを理解できる。データベースシステムを以上の三点に抽象化したものをデータモデルという。即ち、利用者から見たデータベースシステムのモデルをデータモデルである。データモデル (data model) M とは、利用者から見て、データベースシステムが、どのようなデータ構造 D を持ち、どのようなデータ操作 O を行え、データ構造が保つべきインテグリティ制約 I が何かを示すモデルである [図 3-5]。即ち、 $M = (D, O, I)$ 。データモデルの概念は、ソフトウェア工学における抽象データ型 (abstract data type 又は ADT)、Smalltalk 等のオブジェクトの概念と類似した概念である。

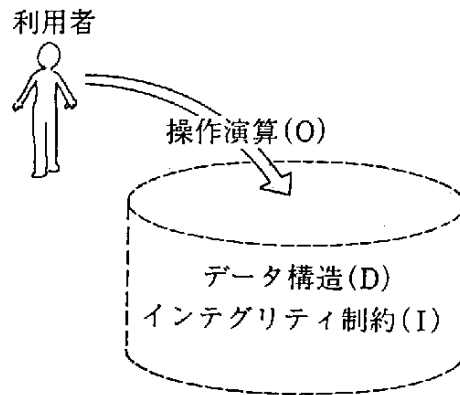


図3-5 データモデル

2. データ構造

まず、データモデルのデータ構造について考える。データ構造 (data structure) とは、データがどのような要素から構成され、各要素が互いにどのような関係があるかを示している。

1) スキーマとスキーマ例

データ構造には、スキーマ (schema) と例 (instance) という二つの側面がある。スキーマとは、データ構造のなかで時間不変な論理的なものである。これに対して、スキーマの例とは、スキーマに実際の値を与えたもので、時間可変なデータ構造である。例えば、ファイルの名前、項目構成はスキーマであり、レコードの集合がスキーマの例である。データ構造を、スキーマとその例にわけるとはデータベースの設計時に重要である。例は又、実現値 (occurrence)、状態 (state) ともいわれる。

データベースシステムのデータ構造は、スキーマとその例であるデータベースから構成される。

データベースシステムでのデータ構造としては、網型データ構造と関係型データ構造がある。

データベースシステムの構造は、一般に、少量のスキーマと、大量のスキーマ例から構成されている。

3. インテグリティ制約

インテグリティ制約 (integrity constraint) とは、スキーマ例 (データベース) が保つべき制約である。例えば、成人のデータからなるデータベースに対して、 $\text{age} > 20$ は、インテグリティ制約である。

4. 操作演算

1) 操作演算

データ構造に対する操作演算としては、スキーマ例 (データベース) に対するデータ操作演算とスキーマに対するデータ定義演算とがある。データ操作 (data manipulation) 演算とは、スキーマ例 (データベース) に対する操作演算である。データ定義 (data definition) 演算とは、スキーマに対する操作演算である。

データ操作演算を表現するための言語をデータ言語 (data language) とする。データ言語には、以下に述べるデータ操作言語とデータ定義言語がある。

2) データ操作演算

データ操作演算とは、データ構造のスキーマ例 (データベース) に対する操作演算であり、以下の種類がある。

(1) 検索演算 (retrieval operation)

(2) 更新演算 (update operation)

検索演算とは、データベースから、利用者の必要とするデータ構造を導出する演算である。導出されたデータ構造のスキーマを目標スキーマ (target schema) という。目標スキーマの例を、スナップショット (snapshot) ということもある。目標スキーマの例をデータベースから与える条件を検索条件という。

更新演算とは、スキーマ例 (データベース) を変化させる演算である。ただし、更新されたデータベースは、インテグリティ制約を保たねばならない。

データ操作演算を表現する言語を、データ操作言語 (DML: data manipulation language) とする。

3) データ定義演算

スキーマの定義、変更、消去を行う演算である。データ定義演算を表現する言語をデータ定義言語 (DDL: data definition or description language) とする。インテグリティ制約もデータ定義演算により定義される。データ言語とは、データ操作言語とデータ定義言語をあわせたものである。

3.1.4 関係型モデル

既存のデータモデルとしては、関係型モデル、網型モデルがあり、商用のデータベースシステムで用いられてきている。

1. 関係型データ構造

関係型モデルのデータ構造は、関係または表 (テーブル) という簡単なデータ構造である。

[例] 例として、従業員 (Emp) と部 (Dept) を示す関係について考える。

Emp	eno	ename	sal	dno	Dept	dno	dname
	1	A	150	1		1	DDD
	2	B	200	1		2	EEE
	3	C	300	3		3	FFF
	4	A	100	2		4	GGG
	5	D	280	3			
	6	E	450	1			
	7	F	130	2			
	8	D	310	2			
	9	G	200	1			
	10	H	280	1			

2. データ操作演算

関係に対しては、国際標準言語である SQL により操作を行える。以下に、例で示す。

[例]

(1) DDD 部の部員を求めよ。

```
select ename from Emp Dept
```

```
where dname = DDD and Dept dno = Emp dno
```

Dept dnoとEmp dnoは、各々関係DeptとEmpの属性であることを示している。

dnameは、関係Deptのなかにだけあるので、Dept dnameと書かなくてもよい。

(2)DDD部員のなかで、給料が200以上の者とその給料を求めよ。

```
select ename sal from Emp Dept
where dname = DDD and Dept dno = Emp dno and
      sal >= 200
```

または、以下のようにも書ける。

```
select ename sal from Emp
where sal >= 200 and
      dno in (select dno from Dept
              where dname = DDD)
```

(3)DDD部員のなかで、給料が200以上の者とその給料を、給料の昇順に求めよ。

```
select ename sal from Emp
where sal >= 200 and
      dno in (select dno from Dept where dname = DDD)
order by sal
```

3.1.5 網型モデル

関係型モデルの他の代表的なデータモデルとして、網型（network型）のデータモデルがある。網型（network）モデルは、関係型モデルと異なり、実際の応用のなかから発展してきたデータモデルであり、データ構造の論理的な側面と物理的な側面とが混在している。即ち、網型モデルは物理的にデータ独立でない。これまでに、定型的な事務処理、オンラインシステム等の大規模で高性能が要求される応用分野で広く利用されてきている。国際標準化が、CODASYL委員会においてなされてきたが、現在ISOにより網型モデルの言語としてNDLが標準化されてきている。

1. 網型データ構造

網型モデルのデータ構造は、レコード型と親子型の二種の要素から構成される。レコード型は、レコード（または、レコード実現地）の集合であり、従来のファイルに対応している。親子型は、二つのレコード型間に存在し、レコード型間の一対多の関係を示す。レ網型のデータ構造は、コードを節点、親子型を有向辺とした有向グラフとして表される。

〔例〕 以下に、関係型モデルの例と同じく、従業員と部の関係を示した網型のデータ構造を示す。

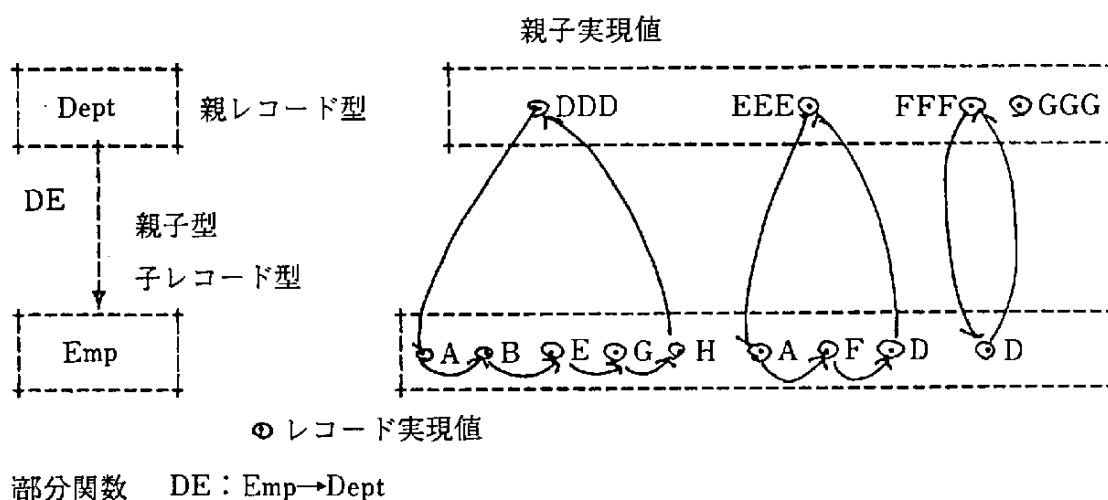


図3-6 論理的網型データ構造の例

2. 網型データ操作演算

網型モデルのデータ操作は、データ構造内を巡行するものである。以下に、例で示す。

『DDD部員のなかで、給料が200以上の者とその給料を求める』

ためのCOBOL DMLの例を以下に示す。

```
move DDD to dname in Dept.
find any Dept.
if not found go to End.
find first Emp within DE.
```



```

        go to L.
N.find next Emp within DE.
L .if end go to End.
    get    Emp.
    if sal in Emp < 200
        go to N.
    move ename in Emp to ename in Out.
    move sal in Emp to sal in Out.
    write Out.
        go to N.

```

3.1.6 意味的なデータモデル

次に、オブジェクト指向のデータモデルについて考える。関係型モデルと網型モデルといった従来のデータモデルでは、CAD等の新しい応用に対処できなくなっている。このために、以下のような新しいデータモデルが提案されてきている。

- 1) E-Rモデル [CHEN76]
- 2) 意味的モデル (semantic data model) [TSUR84, KING85]
- 3) 関数型モデル [SHIP81]
- 4) 論理型モデル [KUPE85]
- 5) オブジェクト指向モデル [MAIER86]

1. E-Rモデル

E-R (entity-relationship) モデルは、以下の二種の要素から構成されている。

- 1) 実体型 (entity type)
- 2) 関連型 (relationship type)

実体は、実世界内の基本要素を示す。各実体は、属性の値の組から構成され、属性は、実体の持つ性質を示している。同一の性質、即ち、同一の属性構成を持つ実体の集合を実体型とする。

関連は、実体間の関係を示す。実体間の関係により与えられる性質を示す属性を持つことができる。同一の性質を持つ関連の集合を関連型という。

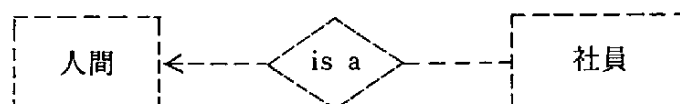
(1) 1:nの関係



(2) n:mの関係



(3) 汎化(generalization)の関係



2. 関数型モデル

関数型モデルのデータ構造は、実体型と実体型間の関数関係から構成される。

3.2 知的ハイパーメディア応用システムの構築

本章では、ハイパーメディア・システムを用いた応用システムについて考える。

3.2.1 ハイパーメディア・システム

現在のテキストでは、テキストが読まれる順番に文書、図、表等が構成されている。例えば、はじめに、第一ページを読み、次に、第二ページを読み、第三ページを…を読むといった具合に、テキストが構成されている。即ち、従来のテキストは、逐次的(sequential)である。

これに対して、ハイパーテキストでは、読まれる要素は、逐次的ではない。テキストの要素は、互いにリンクされ、このリンクに従って、テキストが読まれる。例えば、図3-7で、まず、テキストの要素Aが読まれる。次に、読むものとして、BとCがある。利用者は、このなかでどちらかを選択する。例えば、Bを選択する。さらに、Bから次のテキスト要素として、D、E、Fの三つのオプションがある。このように、利用者は、テキストを順番に読むのではなく、興味の具合により、種々のパス（テキスト要素の系列）により、テキストを読んでいける。このようなテキストの要素をたどっていく操作を、読むというよりは、巡行（navigation）またはブラウズ（browsing）という。

ハイパーテキストの節点（テキストの要素）は、有向リンクにより互いに結合される。二つの節点AとDが有向リンクにより結合されている（A → D）とき、節点Aをアンカ（anchor）節点といい、節点Dを目的（destination）節点という。節点をリンクの方向にたどっていく操作が、巡行である。例えば、図3-7で、A → B → D → Eと巡行できる。これに対して、巡行してきた節点を逆向きにたどることをバックトラック（backtrack）という。このように、一般に、ハイパーテキストは、節点の集合と、節点間の有向リンクからなる有向グラフの構造を持っている。

節点間のリンクは、システムが与えているものの他に、利用者により、動的に節点を自分の観点から結びつけるものとして用いることもできる。この機能を用いることにより、自分の観点にたって情報を構造化し、整理することができる。例として、KJ法でのKJカードの整理にハイパーテキストを利用することがある。

ハイパーテキストは、一般にテキストから構成される。これに対して、ハイパーメディアは、テキストだけでなく音声、画像、イメージといった種々のメディアを構成要素としたシステムである。マルチメディア情報を管理するシステムとしてハイパーテキスト・システムが、ハイパーメディア・システムである。以降、ハイパーテキスト・システムを含めて、ハイパーメディア・システムとして検討していく。

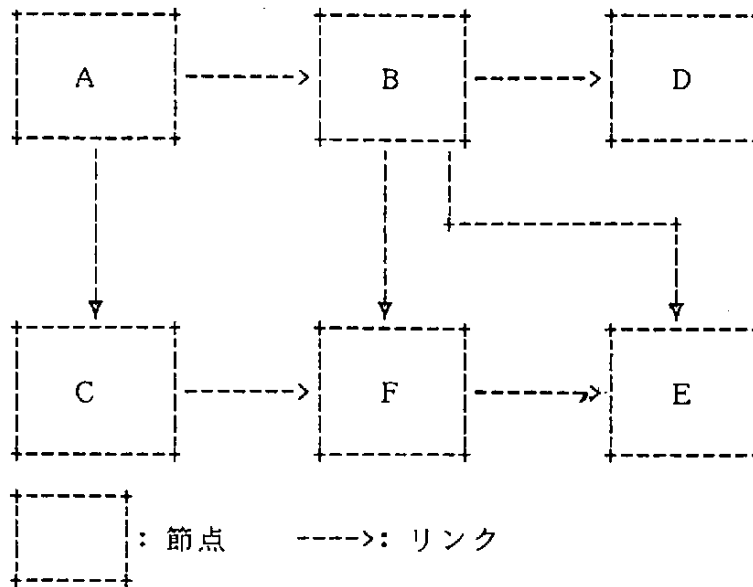


図3-7 ハイパーテキストの例

3.2.2 応用システム

従来のデータベースシステムでは、例えば、関係型のデータ構造に対して、SQLを用いて非手続き的な検索と更新を行うことができる。しかし、データベースシステムでは、データベース内のデータを操作できても、利用者が必要とするデータを柔軟に構成したり、加工することには適していない。このためのインタフェースとして、ハイパーメディアが考えられている。

Ben Schneiderman [SCHN89] は、ハイパーテキスト・システムを適用することが適した分野かどうかを決定するための規則として以下を示している。

〔適用規則〕

- 1) 大量の情報が、多くの断片 (fragment) から構成されている。
- 2) これらの断片が、相互に関連している。
- 3) 利用者は、これらの断片の中の一部のみを用いる。

これらの規則を満足する応用に対しては、ハイパーテキスト (ハイパーメディア) を適用できる。ハイパーメディアの適用例としては、これまでに以下のようなものがある。

- (1) オンラインのドキュメント：ハイパーテキスト・システムの自然な応用である。

利用者は、ドキュメント（例えば、マニュアル）全体を読まずに、必要とする部分、項目と、これらに関連した部分についてのみ読むことができる。

(2) 利用者支援：利用者は、例えば、コンピュータを利用するときに、マニュアルに書かれている以上の情報が必要である。このために、マニュアル、入門用のチュートリアル、オンラインのヘルプ機能等を、ハイパーテキストにより統合できる。

(3) ソフトウェア工学：ソフトウェアのライフサイクル全体に対する支援をハイパーテキストにより行う。

(4) オペレーティング・システム

(5) 事務応用：修理等のマニュアル、辞書、参考文献、監査、カタログ等。

(6) 知的な応用：KJ法等のアイディアの整理、ブレインストーミングの支援がある。

(7) 教育的な応用：外国語の学習、博物館等。

(8) レジャー：旅行者の支援、図書館、対話的なフィクション。

3.3 知的ハイパーメディアとオブジェクト指向データベースシステム

次に、ハイパーメディア・システムとオブジェクト指向データベースシステムの関連について考える。

3.3.1 オブジェクト指向 (object-oriented) システムの背景

まず、オブジェクト指向システムの背景について考える。オブジェクト指向システムが考えられてきた背景には、以下の点がある。

〔背景〕

1. ソフトウェア危機 ソフトウェア要員の不足
2. ソフトウェア生産性と信頼性の向上

こうしたソフトウェアの問題点を解決するものとして、オブジェクト指向の概念が考えられてきた。

ソフトウェア・システムには、以下の二つの捉え方がある。

〔ソフトウェア・システムの捉え方〕

- 1) データ／手続き指向 (data／procedure-oriented) ソフトウェア
- 2) オブジェクト指向 (object-oriented) ソフトウェア

1) のデータ／手続き指向は、従来のソフトウェア・システムの考え方である。ここでは、ソフトウェア・システムは、データと手続き（プログラム）の二つから構成される。

〔データ／手続き指向ソフトウェア〕

- (1) データ＝ソフトウェアにより操作される情報
- (2) 手続き＝ソフトウェアの単位

これに対して、2) のオブジェクト指向のソフトウェア・システムは、データと手続きを一体化したオブジェクトを基本要素として構成される。

〔オブジェクト指向ソフトウェア〕

- (1) オブジェクト＝情報とこの情報を操作する手続きのパッケージ
- (2) メッセージ＝オブジェクトの操作の指示

オブジェクト指向システムにおいてオブジェクトの概念が基本となるが、この「オブジェクト (object)」という概念は、1960年代後半から1970年代初頭にかけて、ソフトウェア工学、人工知能等の種々の分野で独立に考案されてきたものである。

- ① フレーム・モデル (frame model) [Minsky M.]
- ② osにおけるケーパビリティ (capability) [Dennis J.]
- ③ 並列計算モデル (ACTOR) (Hewitt C.)
- ④ Smalltalk-80 [Kay A.]
- ⑤ 離散系のシュミレーション (Simula) [Nygaard.K.]

⑥ 抽象データ型 (ADT : abstract data type)

現在のソフトウェア・システムの問題点を解決するために、オブジェクト指向システムは、以下の点を目的としている。

〔目的〕

(1) データと手続きの一体化。

データとこれに対する操作を行う手続きを一体化 (カプセル化) する。

抽象データ型 (ADT : abstract data type)

(2) モジュール化。

(1)としてデータ構造とこれに対する操作演算を一体化し、一まとまりの実体として定義する。これがどのように実現されているかを知る必要がない。

オブジェクトの概念は、主に、ソフトウェア工学で議論されてきた抽象データ型の概念に基づいている。

〔抽象データ型 (ADT:abstract data type)〕

抽象データ型とは、データ構造とこれに対する操作演算を一体化させたものである。利用者は、提供される操作演算を用いてのみ、データ構造を操作できる。これ以外の方法により、直接データ構造を操作できない。利用者には、抽象データ構造をどのように実現したかが隠蔽されている。

〔特徴〕

(1) 与えられた演算を用いてのみデータ構造を操作できる。

(2) 情報隠蔽 (information hiding)。どのように実現されているかとは独立にデータ構造を利用できる。

抽象データ型に基づいた言語としては、Ada Alghard Clu Euclid等がある。

例として、抽象データ型カウンタを以下に示す。

TYPE counter

DEFINITION

integer range 0..59;

OPERATIONS

up (counters);

down (counters);

zero (counters);

INHERITS

END counters;

TYPE counters

IMPLEMENTATION

PROCEDURE up (VAR a : counters);

begin

if a = 59 then a := 0 else a := a + 1;

end;up

PROCEDURE down (VAR a : counters);

begin

if a = 0 then a := 59 else a := a - 1;

end;down

PROCEDURE zero (VAR a : counters);

begin a := 0;end;zero

END counters;

以下は、抽象データ型スタックの例である。これは、pop push等のスタックの演算の意味を、公理論的に与えている。

specification stack

s is a stack

x is a character

operators

newstack : stack

push : stack character → stack

pop : stack → stack

top : stack → character

size : stack → integer

axioms

pop (newstack) = newstack

pop (push (s x)) = s

top (push (s x)) = x

size (newstack) = 0

size (push (s x)) = size (s) + 1

restrictions

top (newstack) raises an underflow error

size (push (s x)) > n raises an overflow error

利用者からみて、スタックがどのように操作できるかは見れても、どのように実現されているかは見れない。

3.3.2 オブジェクト指向システムの基本概念

オブジェクト指向システムの基本的な概念について考える。オブジェクト指向システムは、以下の基本的な概念に基づいている。

1. オブジェクト (object)
2. メッセージ・パッシング (message passing)
3. 階層構造 (汎化、特殊化)-generalization
4. 継承 (inheritance)

1. オブジェクト

オブジェクト指向システムの基本となるオブジェクトの概念について、まず考える。
オブジェクトは、以下が一体化した概念である。

- 1) 内部状態（データ構造）
- 2) これに対する操作演算（メソッド（method））

メソッドとは、オブジェクトが提供する操作演算である。利用者は、メソッドを用いてのみオブジェクトを操作できる。この意味で、オブジェクトは、メソッドを抽象演算とした抽象データ型である。例として、オブジェクト『銀行』は、メソッドとして、入金、出金、照会といったものを提供している。
利用者は、銀行オブジェクトに対して、メソッド入金を用いて、ある口座に入金を行える。

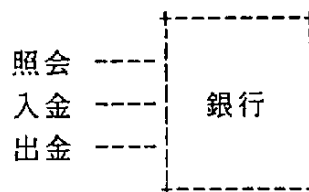


図3-8 オブジェクトの例

2. メッセージ・パッシング（message passing）

オブジェクト指向システムの次の基本概念としては、メッセージ・パッシングがある。これは、システムの起動の方法に関している。従来の、ソフトウェア・システムでは、プログラムは、関数呼び出しにより他のプログラムを起動している。すなわち、プログラムの制御が関数呼び出しの命令に達したときに、プログラムが呼び出され、実行される。これにたいして、オブジェクト指向システムでは、オブジェクトは、メッセージ・パッシングと呼ばれる機構により、活性化される。
オブジェクト間では、メッセージという通信単位の送受信により情報の交換が行われる。オブジェクトの動作を以下に示す。

〔オブジェクトの動作〕

- 1) オブジェクトは、メッセージを受信したとき、活性化される。

- 2) メッセージは、オブジェクトの持つメソッドが含まれる。
- 3) オブジェクトは、内部状態を持ち、状態により受け付けられるメッセージを受信し、メッセージ（セクタ）が含むメソッドを実行し、内部状態を変化させる。

例えば、オブジェクト stack は、セクタとして push を含んだメッセージを受信すると、メソッド push が実行される。これにより、オブジェクト stack の内部状態、すなわち、スタックにあるデータがプッシュ・ダウンされる。

< push > —————> stack

セクタ (selector) セクタの示すメソッド push を実行する。

〔メッセージ・パッシング〕 オブジェクト指向言語は、データを手続きに渡すかわりに、オブジェクト（データ）に演算（手続き）の実行を依頼する。

x: sqrt

x: sqrt□

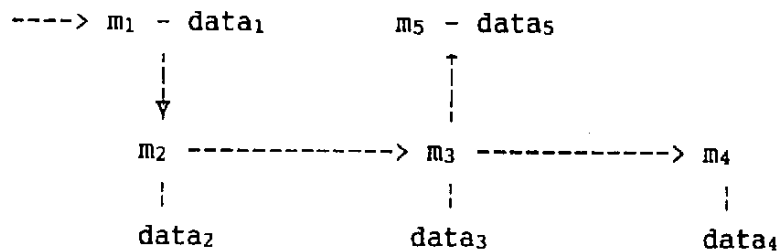


図3-9 オブジェクト指向

これに対して、既存のプログラム言語では、データ／手続き指向である。

既存のプログラム言語は、能動的（アクティブ）な手続きが、これに渡された受動的なデータに作用する。例えば、平方根を計算する関数 sqrt (x) は、

データ x に対して、計算が行われる。

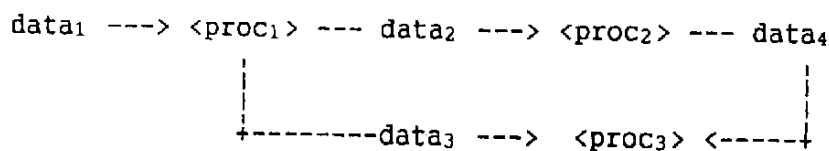


図3-10 データ/手続き指向

こうしたメッセージ・パッシングを実現する方法として、動的結合 (dynamic binding) がある。すなわち、オブジェクトがメッセージを受信したとき、セレクタをこれが見せるメソッドに結合するものである。従来のプログラム言語システムでは、コンパイル／リンク時に、全ての呼び出される関数は結合されている静的結合方式が用いられている。動的結合では、プログラムの実行時に、呼び出される関数が見つけられ、結合される。

オブジェクトの実現方法としては、図**に示すように、メッセージ辞書をオブジェクト内に持たせる方法がよく用いられる。メッセージを受信した時に、メッセージ辞書が探索され、メッセージのセレクタの示すメソッドを探しその実行コードを実行するものである。

オブジェクト

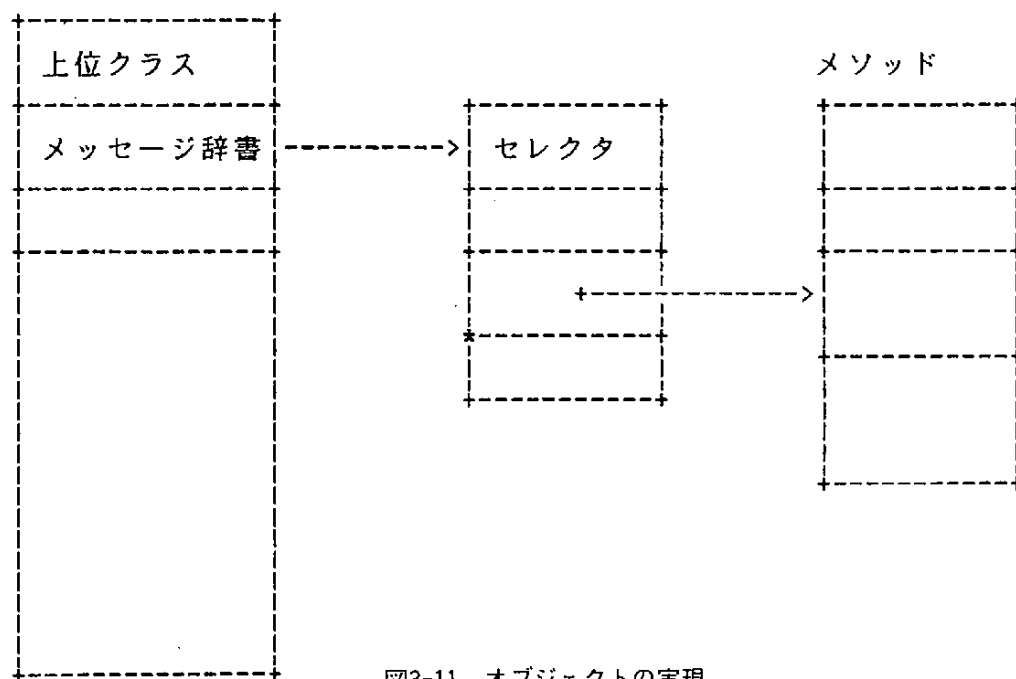


図3-11 オブジェクトの実現

3. 階層化

複雑なシステムを構築したり、理解するためには、オブジェクトを階層化することが必要である。階層化の方法としてまず、抽象化がある。オブジェクトは、抽象化 (abstraction) の概念により階層化される。抽象化による階層化の例としては、通信

における OSI の七階層のモデルがある。

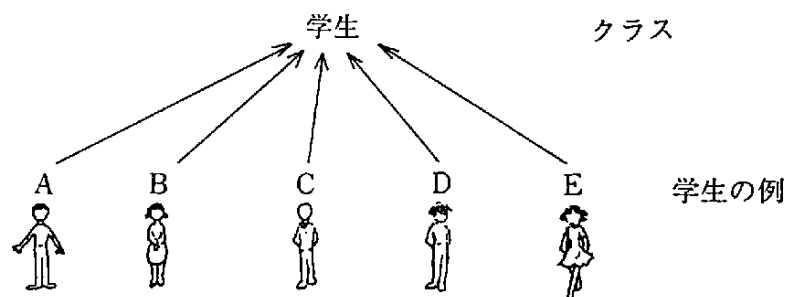
1) クラスと例

〔クラス (class)〕 同一の性質を持つオブジェクトの集合をクラスとする。

〔例 (インスタンス (instance))〕 クラスに属するオブジェクトをクラスの (具体) 例とする。

〔instance-of〕 クラスとそのクラスに属するオブジェクトの関係を instance-of という。

〔例〕



2) 分類 (classification)

同一の性質を持つオブジェクトの集合がクラスである。

〔分類化 (classification)〕 同一の性質を持つ (同一の型の) オブジェクトに対して、クラスを与えることである。オブジェクトの共通の性質を取り出し、個々のオブジェクト固有の性質を取り去ることである。

〔具体化 (instantiation)〕 各クラスに属するオブジェクトを与えることである。

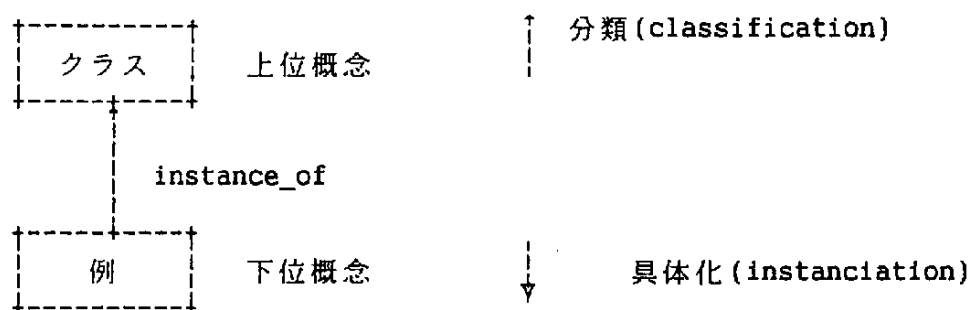


図3-12 instance_of

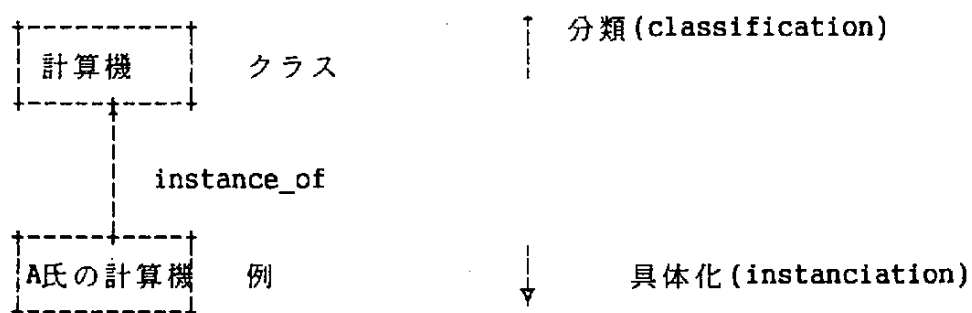


図3-13 instance of

3) 汎化 (generalization) クラス間の関係として、汎化 (または is a) の関係がある。

〔汎化〕 一つ以上のクラスから、共通の性質を抽象化して、より一般的なクラスを定義することが汎化である。

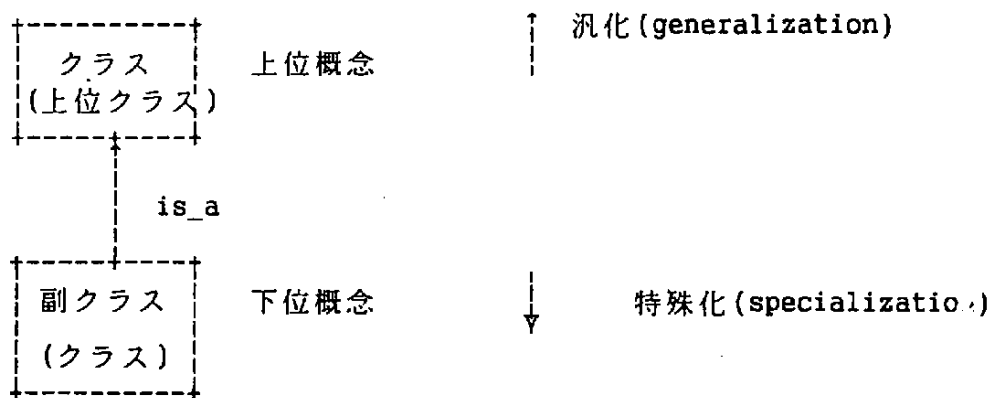


図3-14 階層化

抽象化された上位の概念と、より具体化された下位の概念の関係である。
 下位のクラス (class) に対して、上位を上位クラス (super-class) という。
 又、上位クラスに対して、下位のクラスを副クラス (subclass) ともいう。

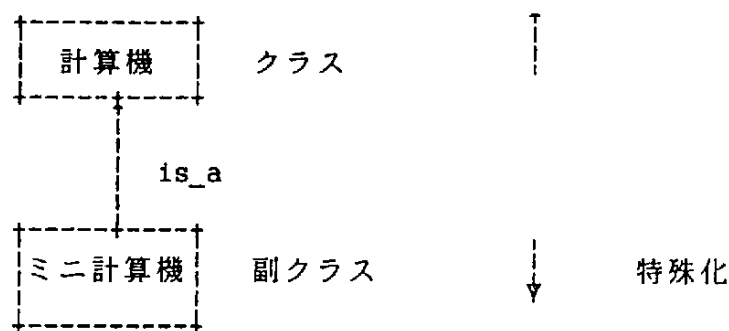


図3-15 階層化

4) 集積化 (aggregation)

集積化は、part-ofといわれる関係である。あるクラスが、他のクラスから構成されていることを示している。

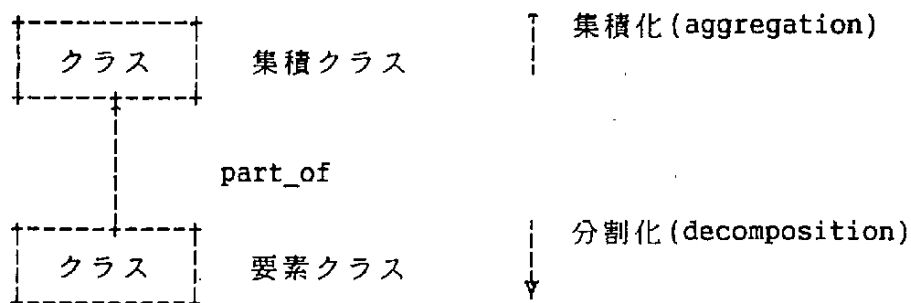


図3-16 集積化

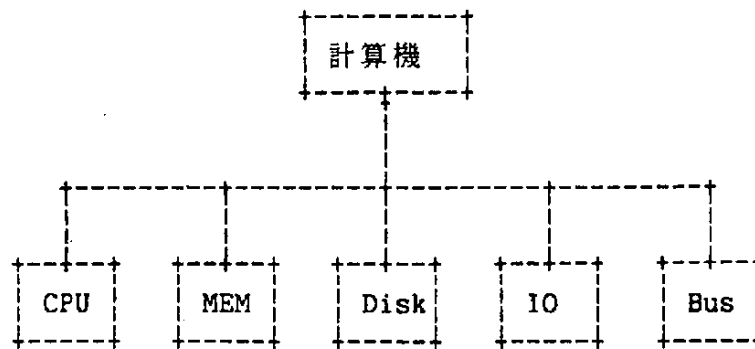


図3-17 集積関係

4. 継承

〔継承 (inheritance)〕一般化では、上位のクラスの持つ性質が下位のクラスに継承される。クラスの性質とは、以下がある。

- 1) データ構造。
- 2) データ構造に対する操作演算。

図の例で、ミニ計算機は計算機である（ミニ計算機 is a 計算機）であることから、ミニ計算機は計算機の性質を継承している。

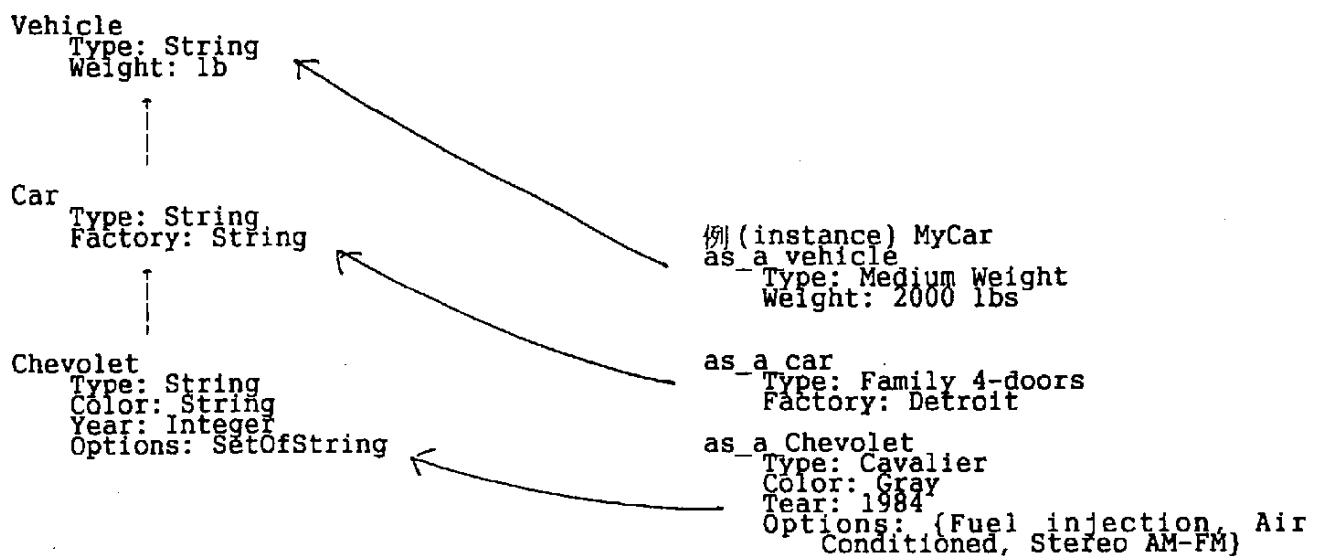


図3-18 継承階層

〔多重継承 (multiple inheritance)〕 あるクラスが複数の上位のクラス $C_1 \dots C_n$ を持ち、各上位クラス C_i から性質 P_i を継承する場合である。

多重継承では、継承される性質が競合する (conflict) 場合が問題となる。

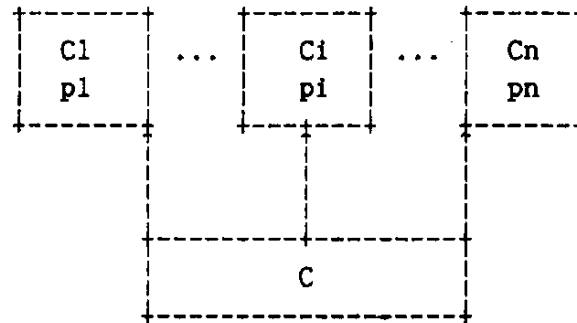
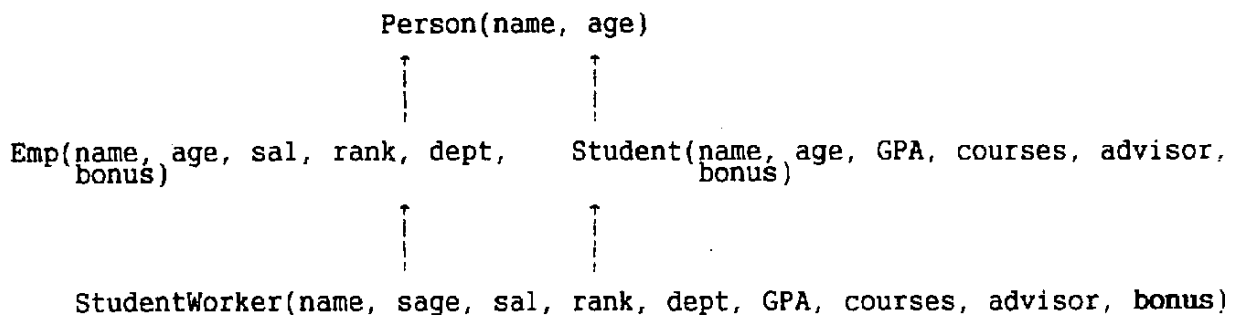


図3-19 多重継承

〔競合の解消方法〕

上位のクラス $C_1 \dots C_n$ に優先度を与えて、優先度の高いクラスの性質が継承されるものとする。



クラスC_jから継承する性質をP_jとする。このとき、P₁.....P_nが矛盾する場合は問題となる。この矛盾を例外を示すために利用することも行われている (Loops Flavor)。

〔汎化関係 (継承関係) の束 (lattice)〕

CCを全クラスと例の集合とし、CCとする。ここで、CSにより、クラスCがクラスSの副クラスである (即ち、CはSの性質を継承する) ことを示すとする。このとき、半順序集合 (CC) は、一般的に (lattice) となる。

(CC) が木構造をしているときは、多重継承は生じない。何故ならば、各クラスCは複数の上位クラスを持たないからである。これに対して、(CC) が束である場合には、各クラスは一般に一つ以上の上位クラスを持てるので、多重継承の問題が生じる。

5. オブジェクト指向計算の利点

〔利点〕

- 1) 情報隠蔽>実現方法との独立性
- 2) データ抽象化>信頼性の向上、実現方法との独立性データ構造と手続きの一体化
- 3) 動的結合>新しいクラスの追加の容易性
- 4) 継承>モジュールの再利用

継承される性質 (データ構造と手続き) を下位オブジェクトが利用できる。

〔欠点〕

- 1) 実行速度メッセージ・パッシング
- 2) 実現の複雑性ガーベッジ・コレクション動的結合
- 3) 大規模システム化が困難。データベース

3.3.3 オブジェクト指向プログラミング・システムの例

オブジェクト指向プログラミング・システムのいくつかについて述べる。

1. Smalltalk-80

1) 概念

Smalltalk-80は、Xerox社のPARCのソフトウェア概念グループが15年の期間をかけて開発したシステムである。「すべての人間の創造的な活動を計算機により支援すること」が目的である。Smalltalk-80は、プログラミング言語とプログラミング環境を統合したシステムである。Starワークステーション、InterLisp-D知識エンジン（Xerox）、Lisa、Macintosh（Apple社）に影響を与えた。OA、CAD、電子出版等に影響を与え、高解像度ディスプレイ、アイコンを用いた利用者インタフェース等、現在のワークステーションの基礎を生み出している。Smalltalkの開発概念としては、以下の3点がある。

[Smalltalk-80開発概念]

- (1) パーソナル計算 個人専用のワークステーション
- (2) 対話 視覚に基づいたグラフィックインタフェース
- (3) オブジェクト指向プログラミング

2) プログラミング言語

データ構造としては、オブジェクト型だけで、複雑なデータ構造の定義は不要である。

Smalltalk-80の構文の要素としては、以下がある。

- 1) リテラル
- 2) 変数
- 3) メッセージ式
- 4) ブロック

3) メッセージ式

(1) メッセージ式の型

単項メッセージ式

キーワード・メッセージ式

(2) 単項メッセージ 引数なしのメッセージ式

宛先のオブジェクト名

メッセージ名

[例] (1) 例を考える。通常のプログラム言語では、 $\sin(\theta)$ と書くところを次のように書く。

$\theta \quad \sin \quad \sin \longrightarrow \theta$

(2) $\text{size}(\text{nameString})$ も同様に、以下のように書く。

$\text{nameString} \quad \text{size} \quad \text{size} \longrightarrow \text{nameString}$

(3) キーワード・メッセージ 引き数を持ったメッセージ式

宛先オブジェクト名

キーワード (セクタ: 引き数) のリスト

[例]

$\text{index} \quad \text{max} : \text{limit}$

$\text{household} \ 1 \ \text{Finances} \quad \text{spendAmount} : 560.00 \text{forReason} : \text{Rent}$

$\text{spendAmount} : \text{forReason}$: メッセージを householdFinances オブジェクトに送信する。

(4) 二項メッセージ

[例]

$3 + 4$

$\text{total} - 1$

$\text{total} \leq \text{max}$

$\text{sum} \quad 3 + 4$

x theta sin

4) ブロック

(1) ブロック [...] (ブロッククラスの例)

(2) ブロックにメッセージが到着するまで、実行が行われない。

(3) 例

```
index <= limit
```

```
ifTrue : [totaltotal + (list at : index)]
```

(4) ブロックは、ヒープ領域から確保され、ガーベージ・コレクタによって破棄される。ブロックを生成したスコープの終了によって、スタックを用いた通常のプログラミング言語の begin...end といったブロック構造のように破棄されない。

5) プログラミング環境

(1) プログラミング言語、サポート・ツール (エディタ、リンカ、デバッカ) オペレーティング・システムが統一され、同一の仮想空間に共存している。

(2) 高解像度のグラフィック・ディスプレイ、ブラウザ

2. C++

ベル研究所で、C 言語に以下のオブジェクト指向機能を付加してを発展させた言語である。

1) C++ 言語の目的

(1) C 言語固有のポータビリティの維持

(2) C++ と C の互換性

(3) 型チェックの強化 (C 言語では、型のチェックが弱い)。

(4) データ隠蔽

2) クラス

クラスは、C の struct を一般化した class 文により定義される。

[例]

```

class fcontrol
    int type date; /* type dateは固有
    public : /* file nchar dateを公開する
                (アクセス可能)

    FILE    * file;

    int nchar;

    char date [128];

```

3) フレンド機能

(1) 固有メンバの参照が許される。

(2) 例

```

class    date
    int    day,month,year;

public :

    friend void setDate (date *,int,int,int);

    friend void nextDate (date *);

    friend void nextToday ( );

    friend void printDate (date *);

```

dateの実現方法を隠蔽し、4つのフレンド関数によってのみdateをアクセスさせる。

dateの利用方法の例を以下に示す。

```

date    myBday,today;

setDate (&myBday,9,12,1950);

setDate (&today,21,7,1989);

printDate (&myBday); nextDate (&today);

```

4) メンバ関数

(1) クラスの例 (インスタンス) に対する演算を定義する。

(2) 例

クラスの定義

```
class    date
    int    day,month,year;
public:
    void  set (int,int,int);
    void  nex ( );
    void  print ( );
```

メンバ関数の参照

```
date  myBday;
...
myBday.print ( );
```

メンバ関数の定義

```
void  date.next ( )
    day = day + 1;
```

.....

5) 導出型クラス

既存のクラスから副クラスを導出できる。クラスから副クラスへの継承機能を持つ。

[例]

```
class    shape
    vector  centrt; int  color;shape  * next;
public:
    void  move (vector);
    vector  where ( );
```

```
virtual void rotate (int);
virtual void draw ( );
```

副クラス virtual 関数は、副クラス毎にオーバーライドされねばならない。

```
class circle : public shape
{
    float radius;
public :
    void rotate (int) ...
    void draw ( );
    .....
};
```

6) 演算子のオーバーライド

(1) 演算子を新しく定義できる。

(2) 例

演算子の定義

```
class vector
{
    int x,y;
public:
    vector (int xc,int yc) x = xc;y = yc /* constructor */
    vector (int xc) x = xc;y = 0 /* constructor */
    vector (vector p) x = p.x;y = p.y /* constructor */
    friend vector operator + (vector,vector);
    friend vector operator + (vector,int);
    ...
};

friend void nextDate (date *);
friend void nextToday ( );
friend void printDate (date *);
```


演算子の利用

```
vector east = (1); north = (0,1);
```

```
vector northeast = north + east;
```

7) メモリ管理

(1) 新オブジェクトは、動的にヒープ領域にアロケートされ、番地により参照される。

(2) オブジェクトの初期化

1) 関数Fに渡されるオブジェクトと、ローカル変数として宣言されているオブジェクトとは、Fが呼び出されるときに初期化される。

2) 関数から返されるオブジェクトは、関数から返るときに初期化される。

3) 関数Fの呼び出しスタック内のオブジェクト (引数、ローカル変数) は、Fから戻ったときに消去される。

クラス開発者は、これらの場合どうするか指定できる。

消滅演算子 ~ <class 名> 自動ガーベージ・コレクションにかわるもの

```
class string
{
    int length;
    char * bytes;
    string (char *); /* string constructor */
    ~string ( ) delete bytes /* string destructor */
public:
    int length ( ) return length
    char * text ( ) return bytes
};

string.string (char * s)
```

```
length = strlen ( s );
bytes = new char [length + 1 ];
strcpy (bytes,s);
```

3.3.4 Objective-C

Objective-Cは、Productivity Products International社が販売している混合型プログラミング言語である。Objective-Cは、既存のC言語を拡張して、オブジェクト指向プログラミングの動的結合を組み込んでいる。

1.データ型

1) オブジェクト型の追加

- | | |
|-----------|----------------|
| 2) オブジェクト | データ・ブロック |
| オブジェクト識別子 | データ・ブロックへのポインタ |
| メッセージの送信 | データへの関数の適用 |

2.メッセージ式

1) 単項メッセージ式

```
aSet = [Set new];
```

c 言語の変数Setが示すオブジェクトに単項メッセージnewを送信する。

この応答は、変数aSetに格納される。

```
print (““The size of this set is % d ¥ n”, [aSet size]);
if ([container isEmpty]) ...
```

2) キーワード・メッセージ式

```
[<receiver> <selector:argument>...]
```

[例]

メッセージ式	セレクタ	引数
--------	------	----

[an Object doThis]	doThis	無し
--------------------	--------	----

[an Object doThis : arg]	doThis :	arg
--------------------------	----------	-----

[an Object do : arg1 with : arg2]	do : with :	arg1, arg2
-----------------------------------	-------------	------------

メッセージ式の型

```
id aSet = [Set new];
```

```
int n = [aSet size];
```

3.3.5 関係型モデルでのオブジェクト指向

以前に概説した関係型モデルは、集合論に基づいて構成されている。しかし、このような単純なモデルでは、CAD、OA、知識ベースといった非標準な応用を処理できない。このために、集合指向の関係型モデルに、オブジェクト指向の概念を取り組むことが試みられている。

1. 現在の関係型モデルでのオブジェクト概念

1) 原子的オブジェクトの提供

〔定義〕 原子的オブジェクト (atomic object) とは、上位の階層により原子的に扱われるデータ単位である。

例としては、実数型、ドキュメント型、イメージ型がある。即ち、利用者は、原子的オブジェクトの内部構造とは独立に利用できる。

既存の関係型データベース管理システム (RDBMS) は、整数型、文字列型といった組み込み型の限定されたデータ型だけを提供しているだけである。これに対する一つの解は、より強力な型を提供することである。例えば、配列、マトリクス、リスト、ビ

ットマップ、イメージといったものがある。型には、単にデータ構造だけでなく、これに対する操作演算が与えられねばならない。

これは、抽象データ型 (ADT) の概念である。

2) 複合オブジェクト

〔定義〕複合オブジェクト (complex object) は、その属性が原子的でなくともよく、階層的な構造を持つオブジェクトである。

オブジェクトに含まれる非原子的なオブジェクトを、副オブジェクト (subobject) 又は要素オブジェクト (component object) という。複合オブジェクトの例としては、論理型プログラミングでの複合項 (complex term) 又はファンクタ (functor) がある。また、OA では、キャビネット等の OA オブジェクトがある。CAD でも、CAD 設計オブジェクトがある。

〔例〕OA オブジェクトキャビネットの関係型モデルによる実現

Cabinet (cab #, properties,...)

Drawer (drw #, cab #, properties,...)

Folder (fdr #, drw #, properties,...)

Document (doc #, fdr #, description,date,author,...)

複合オブジェクト (complex object) 概念を提供するデータモデルについて述べる。

〔定義〕オブジェクトは、以下のように再帰的に定義される。

(1) 整数、実数、論理値、文字列を原子的オブジェクトとし、これらはオブジェクトである。

(2) $O_1, \dots, O_n$ をオブジェクトとし、 $a_1, \dots, a_n$ を属性とする。このとき、 $\langle a_1 : O_1, \dots, a_n : O_n \rangle$ を組オブジェクト (tuple object) とし、これはオブジェクトである。

(3) $O_1, \dots, O_n$ をオブジェクトとしたとき、 $O_1, \dots, O_n$ を集合オブジェクトとし、これはオブジェクトである。

[例]

原子的オブジェクト	10
集合	1, 2, 3, 4
組	$\langle \text{name : Smith, age : 42} \rangle$
関係	$\langle \text{name : Smith, age : 42} \rangle, \quad \langle \text{name : Tom, age : 30} \rangle$
階層組	$\langle \text{name : } \langle \text{first : John, last : Doe} \rangle, \text{ age : 40} \rangle$
集合値を持つ組	$\langle \text{name : Smith, children : Jim, Tom, age : 42} \rangle$
入れ子関係 (nested relation)	$\langle \text{wine \# : 210, on hand : } \langle \text{store : 10, qty : 250} \rangle$ $\quad \langle \text{store : 20, qty : 500} \rangle \rangle$

[定義] 入れ子関係 (nested relation) とは、その属性の値が複合オブジェクトである関係である。

[例] 入れ子関係の例を以下に示す。

Wines = $\langle \text{wine : } \langle \text{w \#},$
 $\text{on hand : } \langle \text{store, qty} \rangle,$
 $\text{vineyard},$
 $\text{vintage},$
 $\text{price : } \langle \text{date, amt} \rangle \quad \rangle,$
 $\text{sale : } \langle \text{date : } \langle \text{month, year} \rangle,$
 $\text{city},$
 $\text{customer : } \langle \text{name},$
 $\quad \text{wine : } \langle \text{w \#}, \text{ qty} \rangle \quad \rangle \quad \rangle \quad \rangle$

3) 複合オブジェクトに対する台数演算

以下の入れ子関係 WINE に対する演算を考える。

WINE

w#	on_hand		vinyard	vintage	price	
	store	qty			year	amt
210	20	510	Chenas	1982	83	5.00
	35	600			84	6.00
					85	6.00
320	10	500	Chablis	1981	85	15.00
	20	450				

(1) 制限

まず、制限演算について考える。

c-restrict (<relation>, <condition>)

・組値を取る属性 例、name.first

*集合値を取る属性 例、price * amt

[例]

c-restrict (WINE, vineyard = "Chenas" price * amt < 6.00)

w#	on_hand		vinyard	vintage	price	
	store	qty			year	amt
210	20	510	Chenas	1982	83	5.0
	35	600				

[例]

Person : < name, children : first name >

c-restrict (Person, "Ann" in Children)

(2) 射影

入れ子関係の射影演算について考える。

c-restrict (relation,c-list)

ここで、

c-list : : = attribute attribute : < c-list >

attribute : < c-list >

< ... > = 組 ... = 集合

[例]

c-project (WINE,on hand,vineyard,price : < amt >)

on_hand		vineyard	price
store	qty		amt
20	510	Chenas	5.00
35	600		6.00
10	500	Chablis	15.00
20	450		

(3) 非ネスト (unnest)

複合関係を第一正規形に変換する演算である。

unnest (c-project (WINE,on hand,vineyard)

store	qty	vineyard
20	510	Chenas
35	600	Chenas

(4) ネスト (nest)

T = unnest (c-project (WINE,w #, on hand))

nest (T,store.WINE : < w #, qty >)

store	WINE	
	w#	qty
10	320	500
20	210	510
	320	450
35	210	600

(5) 結合

c-join (relation1,relation2,condition)

A B ここで

c-join (WINE,unnest (sale), w # = w #)

4) 複合オブジェクトの実現

複合オブジェクトの実現方法について考える。実現方法としては、複合オブジェクトを直接に二次記憶に格納する方法と、従来からの正規形の関係として格納する方法との二つがある。

(1) 直接格納モデル 複合オブジェクトを直接物理的に格納する方法である。

(2) 正規関係による実現 概念層のモデルとして複合オブジェクトの関係が存在し、これを正規形の関係として実現する方法である。

5) オブジェクトの識別

オブジェクト識別子=他の全てのオブジェクトからオブジェクトを識別するオブジェ

クトの性質

関係型モデルでのオブジェクトは、キーにより行われる。これにたいして、複合オブジェクト・モデルでのオブジェクトの識別は以下のようになされる。

(1) オブジェクトの種類

原子的オブジェクト

組オブジェクト

集合オブジェクト

(2) オブジェクト $O = \langle \text{識別子}, \text{値} \rangle$

ここで、

識別子 = O の一意な識別子

値 = O は以下のものである。

1) O が原子的ならば、システム提供の定義域内の要素、

2) $\langle a_1 : I_1, \dots, a_n : I_n \rangle$ ここで、

a_j = 属性

I_j = 識別子

3) $I_1, \dots, I_n$ ここで、 I_j は識別子

例えば、オブジェクトは、以下のSQL内で示されるようにして識別される。

```
select name.WINE.vinyard from SALE
where qty > 100 and WINE.vintage = 1982
```

〔オブジェクトの同一性と同値性〕

二つのオブジェクト O_1 と O_2 の識別子が同一ならば、 O_1 と O_2 は同一である。

O_1 と O_2 の値が同一であるとき、 O_1 と O_2 は同値である。

〔深同一性 (deep equality)〕

同一の値を持つ二つの原子的オブジェクトは、深同一である。 O_1 と O_2 が集合であ

り、集合の要素が一対一対応し、対応する要素が深同値であるならば、O1とO2は深同一である。O1とO2が組であり、同一の属性の値が同一であるならば、O1とO2は深同一である。

3.3.6 オブジェクト指向データベースシステム

1. 概念

1) データモデルの発展

データモデルの発展を以下の図により示す。

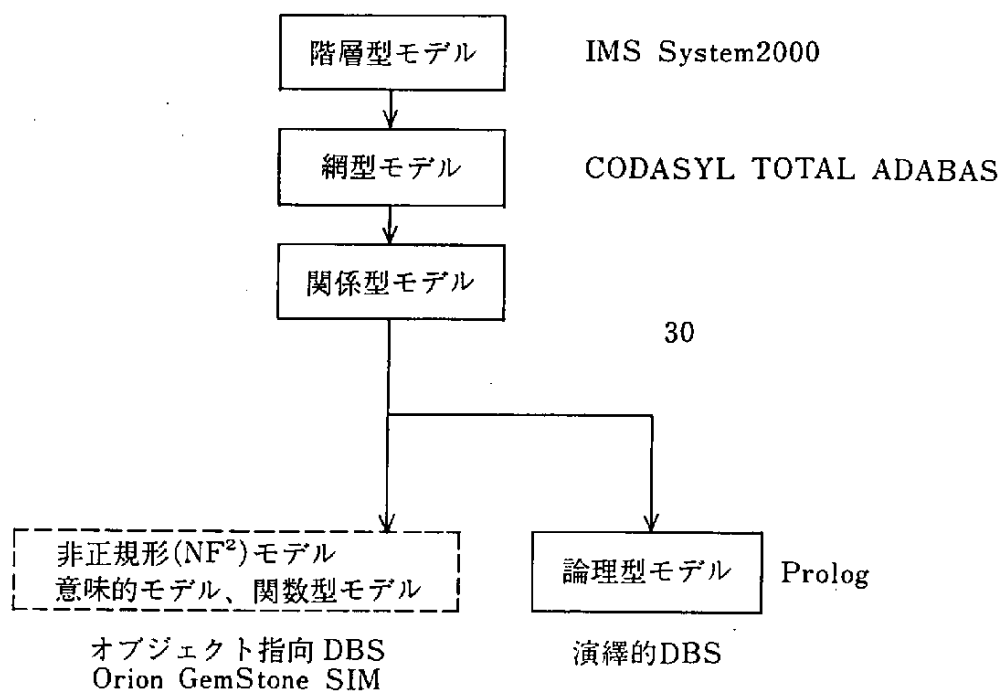


図3-20 データモデルの発展

2) オブジェクト指向データベースシステム

データベースシステムの最小機能として、以下の点をあげることができる。

- (1) 高水準質問言語と質問最適化例えば、SQL 言語と、これの最適化である。
- (2) 原子的トランザクション（同時実行制御と復旧制御）複数利用者からのトランザクションに対して、データベースのインテグリティ制約を保つとともに、システムのスループットの向上がデータベース管理システム（DB-MS）により行われている。
- (3) 複合オブジェクトの格納、高速アクセスのための索引とアクセス方法、索引、

ハッシング、ポインタ、クラスタリング等の物理構成技法がこれである。

オブジェクト指向システムは、以前のべたように、オブジェクト、メッセージ・パッシング、階層構造、継承といった概念に基づいたシステムである。

オブジェクト指向データベースシステム (OODBS) とは、以下のように定義できる。

オブジェクト指向データベースシステム＝

オブジェクト指向システム＋ (a) ＋ (b) ＋ (c)

2. ORION

ORIONは、米国のMCCにより開発されたCAD／CAM用のオブジェクト指向データベース管理システムである。

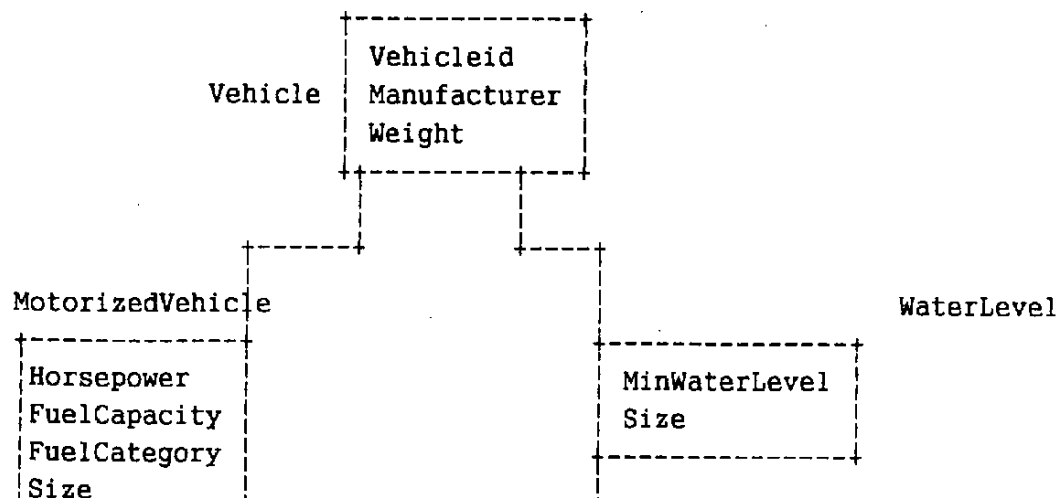
1) 基本概念

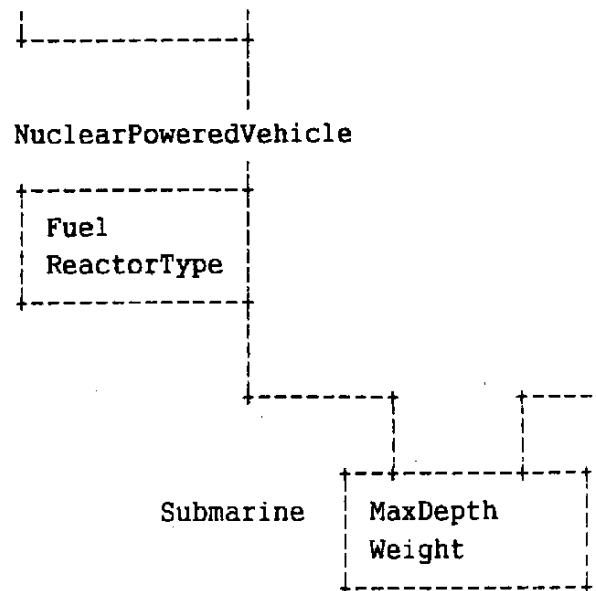
(1) クラス束 (class lattice)

多重継承 (multiple inheritance) での競合

- * クラスCと超クラスSの間での競合
- * クラスCの超クラスS1, ..., Sn間での競合

解決方法 クラス間に優先度を設ける。



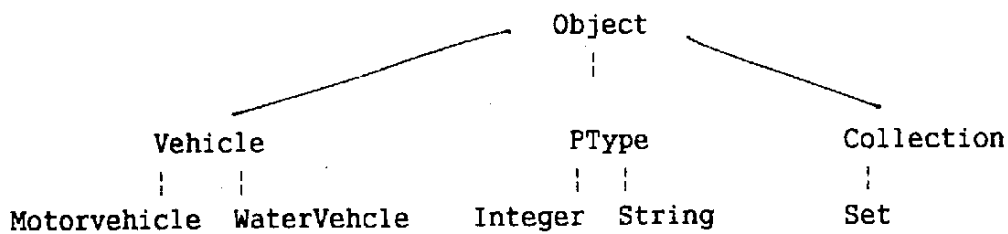
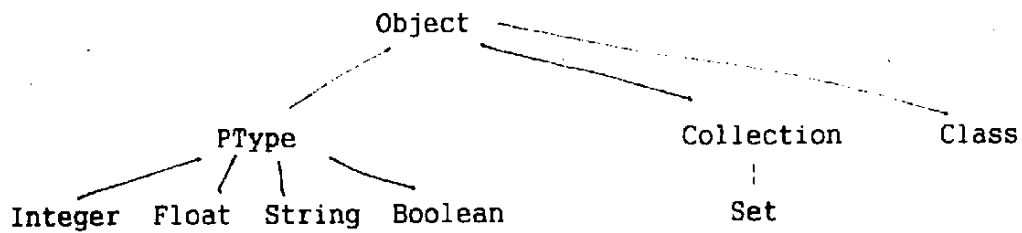


(2) 基本クラス

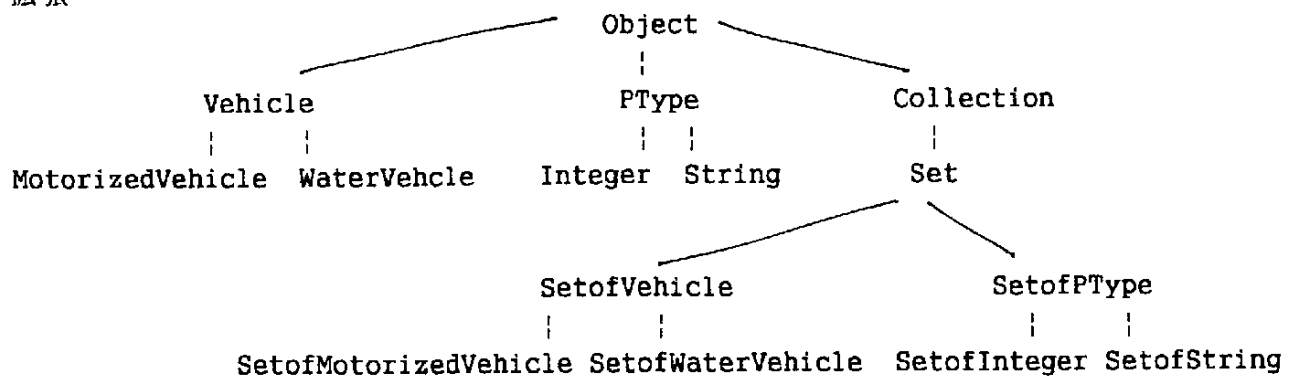
PType = 基本的なデータ型

Collection = 他のオブジェクトの集合であるオブジェクトの集合

Set = 集合



拡張



2) 複合オブジェクト (composite object)

複合オブジェクトは、以下のように定義される。

< Composite Object > :: = < Composite Object Root > (< Linked Dependent > *)

< Linked dependent > :: = < Instance Variable > < Dependent Object >

< Dependent Object > :: = < Leaf Object >

< Dependent Object Root > (< Linked dependent > *)

< Dependent Object > *

3) 版 (version)

ORION は、CAD用であることから、版の管理機能を持っている。版を示すオブジェクト間に、リンク version of を用いている。

4) GemStone

Gemstone は、Smalltalk とデータベースシステムとを組み合わせたオブジェクト指向データベースシステムである。利用者に対しては、OPAL という Smalltalk に基づいた言語がオブジェクト指向データベース言語として提供されている。

(1) Smalltalk を選択した理由

Smalltalk を採用した理由としては、以下がある。

- * ドキュメントが豊富である。
- * 仮想マシン・モデル
- * スキーマ／例の分離

Smalltalkに基づいていることから、Gemstoneは、以下の概念を提供している。

- ① オブジェクトの概念
- ② 複合オブジェクトのモデル化と継承階層の提供
- ③ オブジェクトの動作 メッセージ送受信
- ④ クラス
- ⑤ オブジェクトの型付け

(2)GemStone モデル

① オブジェクト

オブジェクト=例変数 (instance variable) (=フィールド) より構成される。

[例]

Employee (

Empname (firstName / String.lastName / String) / PersonName,

ssNo / SmallInteger,

Address (stNumber / SmallInteger.street / String.city / String) / Home Address,

Salary / SmallInteger)

② メッセージ

メッセージの基本形式は、< receiver > < message >である。

* 単項メッセージ emp firstName

* 二項メッセージ 8 * 3

(empl salary) < (emp 2 salary)

* キーワード・メッセージ

anArray at : 3 put : Ross

③ メソッド

increaseBudgetBy : aPercentage upto : aLimit

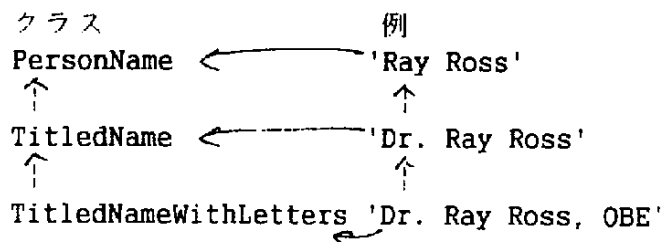
budget := budget + (budget * (aPercentage / 100)).

(budget > aLimit) ifTrue : [budget := aLimit].

budget

④ クラス

クラス階層とその例（インスタンス）を以下に示す。

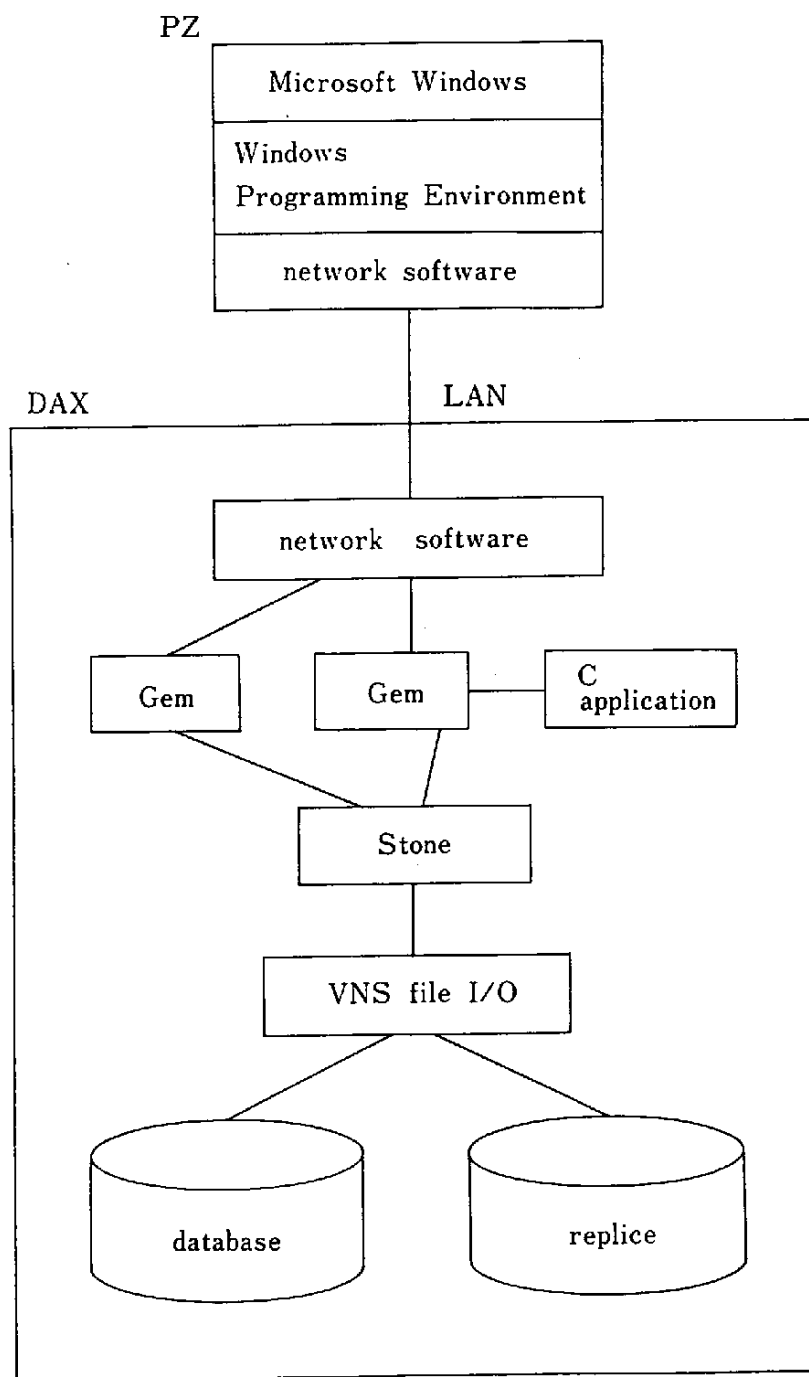


クラス	例変数	メッセージ
PersonName	first	first first: last last: fullname
TitledName	(above)+ title	(above)+ titledName
TitledNameWithLetters	(above)+ letters	(above)+ titledName(new method)

3. Smalltalk からデータベース管理システム (DBNS)

- 1) 多利用者かつディスクに基づいたシステム
- 2) インテグリティ制約
- 3) 巨大オブジェクト空間
- 4) 物理記憶管理
- 5) 他のシステムからのアクセス

4. アーキテクチャ



Gemstore アーキテクチャ

3.4 ハイパーメディアとオブジェクト指向データベースシステム

オブジェクト指向データベースシステムに対して、ハイパーメディアは、利用者インタフェースとして位置づけることができる。ハイパーメディアの各節点を、一つのオブジェクトと考え、節点間のリンクをオブジェクト間の関連を示することにより、ハイパーメディアをオブジェクト指向モデルによりモデル化できる。各節点に対しての操作、例えば、コピー、編集といった操作を、オブジェクトのメソッドと考えることができる。

3.5 知的ハイパーメディアによるデータベースシステム構築上の問題点

ハイパーメディアを用いて、データベースシステムを構築する場合の問題点について考える。ハイパーメディアを用いることの利点は、利用者に必要とする情報をリンクに従って巡行的にアクセスできることと、リンクを設定することにより情報を整理していけることである。また、種々のメディアを各節点内で統一的に扱えることである。このように、ハイパーメディアは、利用者のインタフェースとして優れたものである。ハイパーメディアを用いて、データベースシステムを実現する方法は、データベースシステムとしてオブジェクト指向のデータベースシステム (OODBS) を用いることである。問題点としては、以下がある。

3.5.1 複数の利用者によるハイパーメディアの共有問題

特に、共同執筆や、会議での黒板のように、複数の利用者が、一つの節点（オブジェクト）を共有して操作する場合の同期制御の問題がある。従来のデータベースシステムのように、オブジェクトをロックすることにより、ある利用者がオブジェクトを操作している場合には、他の利用者に操作させない方法を用いると、同時に操作できる利用者の数が減少してしまう。オブジェクトは、従来のデータベースシステムのロックの単位に対して、大きなものであるからである。一つの方法は、各節点オブジェクトの提供する演算（メソッド）間で、メソッドの意味に基づいて、同時に実行できる

かどうかを示す競合関係を定義するものである。例えば、共同執筆の場合に、テキストを書くことと、赤ペンをいれる演算とは、同時に行っても競合しない。しかし、多種大量のオブジェクトからなるシステムに対して、各オブジェクト毎に演算間の競合関係を意味的に定義することは容易ではない。

さらに、各利用者毎に、異なった視野を持つ場合がある。視野毎に、オブジェクトの演算の競合関係が異なる場合がある。こうした矛盾をどのように解決していくかが今後の課題となる。

3.5.2 大規模化

大量の節点オブジェクトから構成されるシステムにおいて、まず性能面の問題がある。複雑な大量のデータ構造を高速にアクセスする機構が必要となる。特に、画像、音声、イメージといったマルチメディア情報を、高速にアクセスする機構が必要となる。大量の節点オブジェクト間のリンクの保守も大きな問題である。

3.5.3 マルチメディア

マルチメディアを、一つのオブジェクトとして考える時に、これらにどのような演算（メソッド）を提供していくか、またいけるかが問題となる。特に、イメージ等のマルチメディアにたいして、曖昧な検索、例えば、『赤っぱいチェックの模様』といった検索に対して、どのように行っていくかである。各メディアに対して、比較、更新といった演算を厳密に定義し、これを実現できることが必要であるが、現在、音声認識、パターン認識、イメージ処理等の研究課題であり、今後の研究が必要である。

参考文献

[BANE87] Banerjee,J..et al..“Data Model Issues for Object-Oriented Applications,” ACM TODS.1987.

[NIEL90] Nielsen,J..“Hyper Text&Hyper Media,”Academic Press,1990.

[SCHN87] Schneiderman,B..“User Interface Design for Hyperties Electronic Encyclopedia,”Proc.of the ACM Hypertext'87Cont..1987,pp.189-194.

[SCHN89] Schneiderman,B.."Reflections on Authoring,Editing,and Managing Hypertext,"The Society of Text,MIT Press,1989, pp.115-131.

4章 知的ハイパーメディア技術の研究事例

序

今日、先進国における研究開発機関では、知的ハイパーメディア技術に関して様々な研究開発が行われている。以下に紹介するのは、今後の知的ハイパーメディア技術に多大のインパクトを与えるものと考えられる代表的研究開発事例である。なお、これら事例は1991年春、スペインのバルセロナで開催されたRIA091会議で発表された論文をもとにしている。

4.1 オブジェクト指向データベース用ハイパーメディア・インタフェース

4.1.1. 序

ここ数年、データベース・アプリケーションに新しい分野が登場してきた。オフィス・オートメーション、CAD/CAM、技術文書などがそれである。これらの分野で扱うデータはその大半が膨大な量であり、また複合化され、マルチメディアを対象としたものとなっている。さらに（経営者、企業意思の決定者など）専門家でないユーザの集団がこれらの膨大なデータベースを扱うようになってきている。

今日、主要なデータベース・システムはリレーショナル・モデルをベースとしている。リレーショナル・モデルにより、SQLによって代表される強力な記述照会言語の開発が可能となった。またQBE、CUPID、GQLのようなユーザ・フレンドリなグラフィック・インタフェースも開発されている。ただし、これらの言語やインタフェースの場合、これらについての理論的な知識が必要であり、また言語の特殊な構文、あるいはコマンドの構造を習得しなければならない。専門家でないユーザがより速く、またかんたんに情報を入手したい場合にこれは障害となる。さらにリレーショナル・モデルは、複合データおよびマルチメディア・データの格納と操作には適していないということが広く認められている。

ここでは、このようなデータの管理を可能とし、また専門家でないユーザを対象としたインタフェースを提示することにある。ハイパーテキスト・システムでは、タイプにより指定されたリンクによって互いに結合されたマルチ・メディア・ノードのネットワークにおいてデータの表現と管理が行われるが、以後、このシステムで使用される手法によって説明を行う。ハイパーテキスト・システムが提供する最新のインタフェースにより、ユーザは、線形ではなく連想による方法でデータを調べることができる（ユーザは連想方式で処理を進める）。この方法による処理は、専門家でないユーザの集団にとっては特に興味深いものである。

まずはじめに、リレーショナル・インタフェース HQL (Hypertext Query Language : ハイパーテキスト照会言語) が考案され、SUN のワークステーションで実現された。HQL は、リレーショナル・データベースの検討にあたって、ハイパーテキスト・システムの思想をいくつか取り入れている。そのあとで、このインタフェースは複合データの管理にまで機能が拡張された。この CHQL (Complex HQL : 複合 HQL) という新しいインタフェースは拡張リレーショナル・モデル NF^2 の思想 (Non First Normal Form) をベースとしている。

これら二つのインタフェースは、専門家でないユーザ向けのインタフェースの分野におけるハイパーテキストの手法に対する関心が高まっていることを示している。ここでは CHQL により、複合オブジェクトにおける照会の表現を容易にするシステムを提示する。ただし、強固な構造を持つオブジェクトに関わる照会のなかにはまだ表現が困難なものもあるため、CHQL には限界がある。さらに、選択されたデータ・モデルによってもデータの共用とマルチメディア・データの管理は容易とはならず、またデータの意味を十分に表現することは不可能である。

これらの弱点の解決策を、ここではオブジェクト指向の手法に求めた。この手法は、オブジェクトの静的な面と動的な面の結合による新しい処理を示唆している。データベースの分野においては、この手法の利点はよく知られており、いくつかのシステムがすでに市場に出ている (O2 [DEUX90]、GBASE [GRAP88]、IRIS [FISH87] など)。しかし、これらのシステムによって提供された照会言語は複雑で、しかも文法は厳格であり (RELOOP [CLUE89]、OSQL [BEEC88])、またエンド・ユーザ向けのインタフェースは、使用するモデルに関して少なからぬ知識を必要とするためすぐに限界を露呈した。

本研究の目的は、オブジェクト指向の手法とわれわれのハイパーテキストの手法の利点を一体化したオブジェクト指向インタフェースを提示することにある。このインタフェースはオブジェクトのベース・スキームを、互いに関連があるノード (クラスと、クラス間の束縛に対応するリンクを表現するノード) のネットワークとして説明して

いる。このインタフェースによってユーザは、既存の大半のインタフェースよりも自然でやさしい方法で、複合オブジェクトとマルチメディア・オブジェクトの管理ができるようになる。

以下では、オブジェクト・インタフェース OHQL (Object Hypertext Query Language : オブジェクト・ハイパーテキスト照会言語) を紹介している。

4.1.2. リレーショナル・インタフェース HQL

リレーショナル・インタフェース HQL は、専門家でないユーザを対象としている。HQL は、ハイパーテキスト・システムの思想をいくつか取り入れ、まとめあげることにより、ユーザにとってより自然な方法でデータベースを記述している。ハイパーテキスト・システムは、データベース・スキームの構造を正確に知らなくても、また入手したいデータについて明確な考えを持っていなくても照会内容を表現することができる。

HQL を使用すれば、ユーザは以下の方法によってリレーショナル・データベースを処理できる。

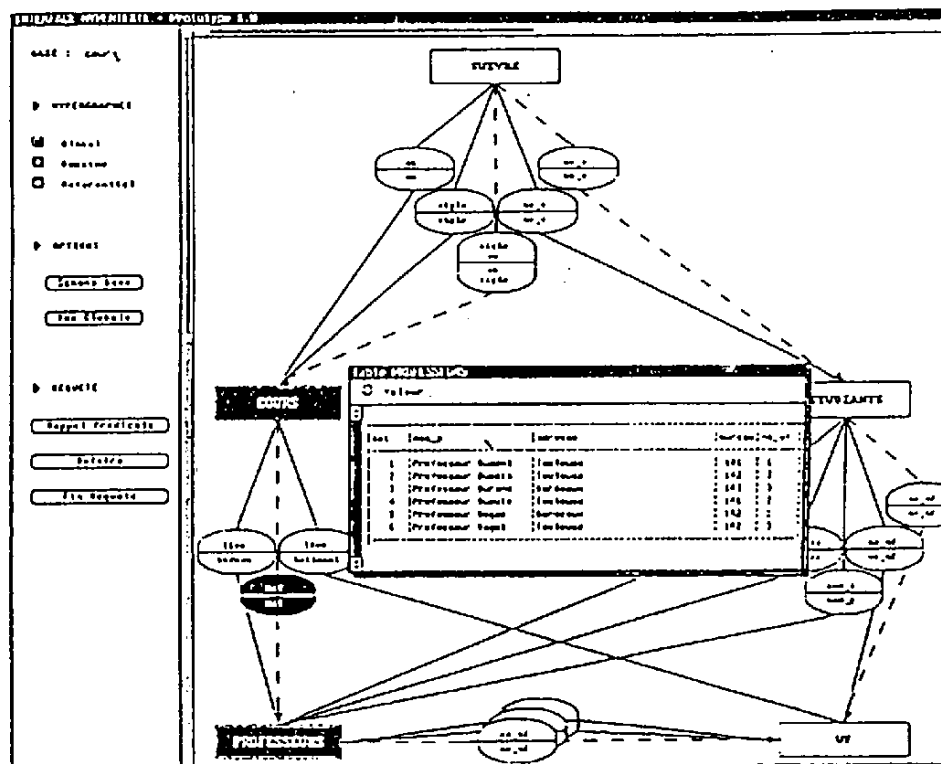
- * ハイパーグラフによるデータ・スキームの視覚化。ハイパーグラフではノードが関係を表し、リンクが (ドメインの、および関係のある) 完全な束縛を表している。

- * 連想方式によって探索しながら少しずつ照会内容のフォーマットを整えていく (リレーショナル・モデルの一貫性は保証される)。

- * リレーショナル代数演算を可能にし、また比較的学びやすいグラフィック言語 [PUJO 90] の使用。この処理には、ノード、およびハイパーグラフのリンクに対するリレーショナル演算の適用が含まれる。

(ノードとしての) 関係の表現と、(リンクとしての) これらの関係の間の束縛によっ

てユーザは、結合の可能性をすぐに見て知ることができる。



データベースのスキームを表現するHQLプロトタイプのハードコピー。結合は行われている（選択されたリンク）。ノード（関係）を構成するものはメイン・ウィンドウに表示される。プロトタイプは操作可能であり、これはSUNのワークステーション上でC言語によって、またUNIXの下で動作するRTIのDBMS INGRES上で開発された。

これは新しい手法であり、以下の重要な二点が根幹を成している。

- * 独特で簡潔な形式による結合の指定（図によってリンクを選択する）。
- * 連想方式によるデータ参照の可能性。

HQLは、エンド・ユーザ向けインタフェースの分野において重要な補助ツールを提供している。

4.1.3. NF²インタフェースCHQL

NF²モデルは強固な構造を持った複合オブジェクトのモデリングに関する機能を提供する。オブジェクトは、標準化されていないNF²の関係によって記述される。NF²の関係は、その属性にアトミック値または複素数値を含む組のセットである。複素数値がNF²の関係となる場合もある。構造化照会言語HDBL [PIST87] はむずかしすぎてエンド・ユーザには使えないということはよく知られている。

インタフェースCHQLはHQLインタフェースの基準をNF²モデルにまで拡張したものである。CHQLはグラフィカル・モードによって照会内容を明確な形にする。専門家でないユーザでも自分の照会内容を少しずつ、そして簡潔な形式で指定できる。ユーザは、HDBLのようにテキストで構成され、きちんと文法に従わなければならない言語を知らなくても、自分の照会内容を明確に表現することができる。

HQLを使用すれば、ユーザは以下の方法によってNF²データベースを処理できる。

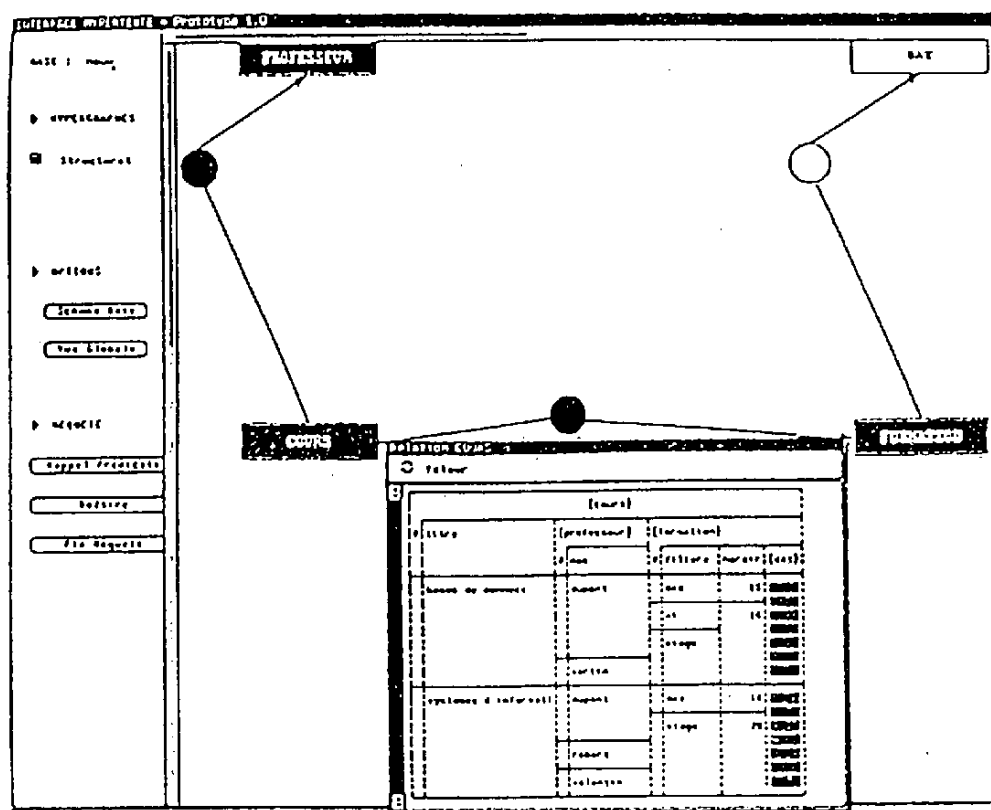
- *ハイパーグラフとしてのNF²の関係を視覚化する。ルートはNF²の関係であり、これ以外のノードはネストされた関係である。リンクは、NF²の関係と、その関係の属性との間の構造的な束縛に対応している。

- *選択したリンクによりユーザの照会内容のフォーマットを整える。二つの関係の間のリンクの選択には、これらの関係を持つ結合演算が含まれる。ノードが目に見えるようになった場合、関係は、アトミック属性と、選択されたリンクに対応する関係属性を持つNF²の表によって表される。

- *NF²の表で選択、更新、削減という操作を可能にするグラフィック言語を使用する。

CHQLは、エンド・ユーザによる複合オブジェクトのハンドリングにおいて重要なステップを成している。グラフィック・ツールを使用すれば、ネストされた関係の操作はさほどむずかしいものではなくなる。すなわちHDBLのような言語は使用せず

む。ただし、このインタフェースではその限界もいくつか明らかになっている。複雑な照会においては条件の表現がむずかしいときもある。さらに、モデルはデータの意味を十分に表現できず、またマルチメディア・データのハンドリング [WOEL86] に関しては限界がある。



NF²データベースのスキームを表現するCHQLプロトタイプのハードコピー。リンクは二つ選択されている。ノード（NF²の関係）を構成するものはメイン・ウィンドウに表示される。プロトタイプは操作可能であり、これはSUNのワークステーション上でC言語によって、また拡張リレーショナル・システム上で開発されたものである。

4.1.4. オブジェクト指向インタフェースOHQL

オブジェクト指向の手法を使用すれば、より自然な方法でオブジェクトによるデータの表現が可能となる。オブジェクト指向の手法は、一般化と継承の概念によってよりすぐれたモデル化の機能を提供する。さらにオブジェクトの静的な面と動的な面の両者を扱う。また複合オブジェクトおよびマルチメディア・オブジェクトの格納、操作、共用を容易にする。この手法によりCHQLインタフェースの問題は解決される。

1. オブジェクト指向モデル

ここで使用するモデルはオブジェクト指向の手法の中核となる思想を取り入れている。
オブジェクト指向モデルの記述は以下のように行われる。

1) オブジェクト、オブジェクトの同一性、属性、メソッド、メッセージ、カプセル化

オブジェクト指向の手法では、概念上のエンティティはすべてオブジェクトとしてモデル化される。オブジェクトは、整数のようなアトミック・エンティティ、すなわちアトミック・オブジェクト、または自動車のような複合エンティティ、すなわち複合オブジェクトを表現することができる。もっと正確に言うと、オブジェクトは、ユニークな参照によって識別される構造化コンポーネント（属性）の集合である。オブジェクトに対するアクセスと処理は、メソッドと呼ばれる処理手順のセットによって行われる。

一般に、オブジェクトには状態と作用がある。

* オブジェクトの状態とは（識別子、値）のペアである。

識別子はユニークである。すなわちオブジェクトの同一性である。これによりその値に関係なくオブジェクトを識別することができる。

オブジェクトの構造は属性の集合によって記述される。オブジェクトの値はその属性の値である。

アトミック・オブジェクトには値があるだけで属性はないが、これに対して複合オブジェクトは属性を持っており、属性によって他のオブジェクト（アトミック・オブジェクトまたは複合オブジェクト）の参照を行う。

* オブジェクトの作用は、オブジェクトの定義の一部であるメソッドによって定義される。メソッドによって、オブジェクトの状態を操作し、これを返すことができる。

オブジェクトはメッセージによって自分以外のオブジェクトと通信を行うことができる。オブジェクトは対応するメソッドの実行により、メッセージに対して応答を行う。メッセージの構文は以下の通りである。

(レシーバ セレクタ 引き数)

レシーバはメッセージを受信するオブジェクトであり、セレクタは実行しなければならないメソッドの名前、そして引き数は発生しうるパラメータである。

カプセル化の概念により、オブジェクトの属性とメソッドを同時にモデル化することが可能となる。オブジェクトの構造はユーザには見えず、ユーザはオブジェクトの外側から見えるメソッドによってこれ进行操作できるだけである。これらのメソッドは公開されており、オブジェクトのインタフェースを構成している。メソッドは自分のサイン、すなわち自分の名前、引き数の型、処理結果の型によって記述される。オブジェクトのインタフェースに記述のないメソッドはユーザには見えず、これは非公開となっている。

2) クラスと継承

同一セットの属性、メソッドによって記述されるオブジェクトは一つのクラスに属している。クラスは抽象データ型を表している。クラスには名前があり、またインテンションとエクステンションを所有している。

*クラスのインテンションは、そのオブジェクトの一般的で変わることのない特性を記述している。公開されたメソッドのサインはクラスのインタフェースを構成している。

*クラスのエクステンションは、クラスのインスタンスのセットである。インスタンスは特定の時点でのオブジェクトの状態である。

クラスはアトミック・クラス（属性を持たない）、または複合クラスとなることができる。いうなればクラスはオブジェクトの製造工場であり、クラスを使用して新しいインスタンス（すなわち新しいオブジェクト）を生成することができる。新しいインスタンスは、インスタンス化・メソッドと呼ばれる `Create __object ()` メソッドを実行するとクラス上に作成される。

クラスのエクステンションは、インスタンス化の手順にしたがいインテンションと結合される。

継承は意味モデルの一般化の概念から生まれたものである。一般化は、特殊化されたオブジェクトと総称オブジェクトの関係“is __ a”を表現している。これによって、（スーパークラスと呼ばれる）より一般的なクラスにおいて、スーパークラスのサブクラスと呼ばれるいくつかのクラスの一般的な特性（属性およびメソッド）を共有することが可能となる。

継承によって、クラスからサブクラスへの特性の伝達が可能となる。サブクラスのインテンションは自ら所有しており、また継承される特性のセットによって記述される。この定義はスーパークラスの洗練化または豊饒化によって行われる。したがって、サブクラスのインスタンスはどれもスーパークラスのインスタンスでもある。サブクラスのエクステンションはスーパークラスのエクステンションのなかに含まれる。この継承は包含による継承である。

このモデルでは多種多様な継承を提供している。またクラスはスーパークラスをいくつでも直接所有することができる。継承される特性の名前が重複しても、特性の名前を変えれば解決できる。

異なるクラス間での継承の関係はクラス・ラティスと呼ばれるグラフによって表現される。このグラフは `OBJECT` というルートを所有している。これは全オブジェクトの作用を共有するクラスである。

3) ドメインと完全な束縛

クラスの各属性は、属性のドメインと呼ばれるクラスで定義される。属性の値はそのドメインのインスタンス（ドメインがアトミック・クラスの場合）であるか、またはドメインのインスタンスへの参照（ドメインが複合クラスの場合）である。このモデルでは、INTEGER、REAL、STRING、PICTURE、TEXTなどの定義済のアトミック・クラスをいくつか提供している。また構造化クラスSET、LIST、および全クラスを記述するクラスCLASSを提供し、クラス・メソッドの操作も含んでいる。

ドメインはアトミック・ドメイン、または複合・ドメインのいずれかである。複合ドメインの意味はそのクラスの意味である。アトミック・ドメインの意味はアトミック・クラスの特異化により表現される。作成されたサブクラスは、リレーショナル・モデルにおけるセマンティック・ドメインの概念に等しい。

両者のクラスが同じであるか、一方が他方のサブクラスであるならば、これら二つのドメインの共存は可能である。

モデルに継承される意味に関する完全な束縛のセットはオブジェクト・ベースの一貫性をサポートしている。本論文では継承の束縛とドメインの束縛は分けて考えることとする。ドメインの束縛は以下のいずれかである。

*属性がアトミック・ドメインで定義される場合、単一ドメインの束縛である（これはリレーショナル・モデルにおけるドメインの束縛に等しい）。

*属性が複合ドメインで定義される場合、複合ドメインの束縛である。これは識別のための束縛である。

2. 実 例

本例では、講座、学生、学位、および大学の研究室のメンバである研究者（哲学博士、

学生、教師) から構成される大学のデータベースについて説明する。クラスを設計した人間の考えを目に見えるかたちにするると以下のようなになる。

Class : LABO
Superclass : STRING

Class : DEGREE
Superclass : OBJECT
Attributes : title : string;
 credits : set (course);
 exams : list (string);
 official : teacher;
Methods : nb_course (credits) : integer;

Class : PERSON
Superclass : OBJECT
Attributes : name : string;
 first_name : string;
 ss# : integer;
 photo : picture;
Methods : age (ss#) : integer;

Class : STUDENT
Superclass : PERSON
Attributes : n°_card : integer;
 prepare_for : degree;
 cv : text;

Class : COURSE
Superclass : OBJECT
Attributes : title : string;
 nb_hours : integer;
 teachers : set (teacher);
 room : integer;
Methods : timetable (room) : list (course)

Class : TEACHER
Superclass : PERSON
Attributes : grade : integer;
 laboratory : labo;
 room : string;
Methods : wage (grade) : real;

Class : PhD_STUDENT
Superclass : STUDENT
Attributes : laboratory : labo;
 reader : teacher;
Methods : research_team (laboratory) : set (PhDstudent)

クラスLABOはアトミック・ドメインの意味を示している。これはクラスSTRINGのサブクラスである。これ以外のクラスはスーパークラスOBJECTのサブクラスである。これらのクラスは属性を持ち、この属性のドメインは単一ドメイン (たとえばtitle : string)、または複合ドメイン (たとえばofficial:teacher,credits:set (course)) となる。

3. インタフェースの機能

(1) ハイパーテキスト・インタフェース

OHQLは、HQLやCHQLで紹介したハイパーテキストの手法の利点を取り入れている。すなわち、

*オブジェクト・ベース・スキームは互いに関連するノード (クラス) のネットワークとして使用される。

*ユーザは連想方式によってオブジェクトを参照し、また照会内容は少しずつ指定される。

*オブジェクトの構造を知らなくてもオブジェクトにアクセスできる。さらに、ハイパーテキストの手法はオブジェクトの手法とも、暗黙のうちにオブジェクトの周辺を動き回る機能ともごく自然に調和する。なぜならばオブジェクトは別のオブジェクトを参照する機能を備えているからである。ユーザはこの参照、すなわちオブジェクトの識別子を通じて別のオブジェクトを手に入れることができる。

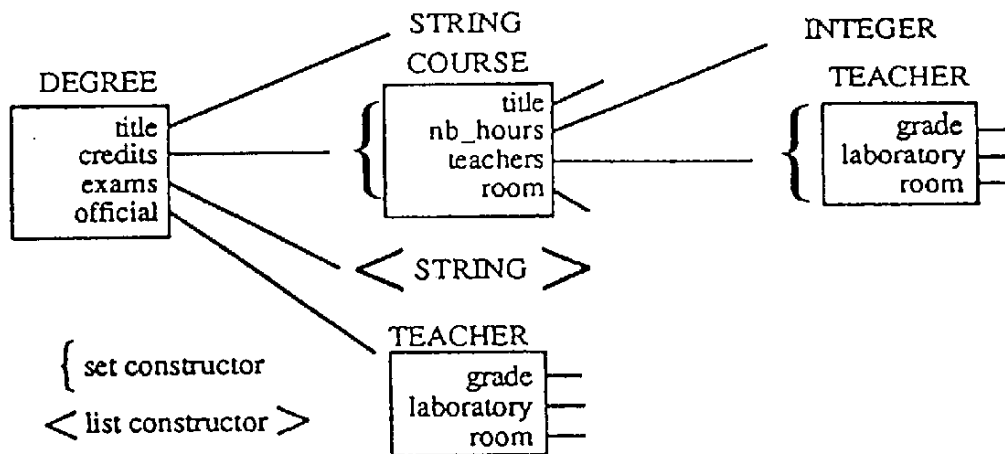
このインタフェースは専門家でないユーザを対象としている。ユーザはSQLのような言語を知っておく必要はない。このような言語は、ある場所から選択する構文をベースとしており、この構文はかなりの精密さを必要とする。ユーザは自分の照会内容をすべて指定しなくてもオブジェクトを入手し、これを調べることができる。ベース・スキームを調べれば、連想方式によって探索し、照会内容を少しずつ具体的に表現していくことができる。

(2) 複合オブジェクト・インタフェース

ユーザは、複合オブジェクトの管理を可能とするオブジェクト・クラスを扱う。これらのオブジェクトは基本的なコンストラクタの組、セット、リストとともに表現される。複合オブジェクトのコンポーネントは、これらコンポーネントのドメインである別のクラスに属している。このインタフェースにより、これらのコンポーネントをわかりやすく自然な方法で見ることができる。したがって、ユーザは複合オブジェクトを調べ、また処理することができる。

たとえば、学位はクラス COURSE および TEACHER にコンポーネントを持つ複合オブジェクトである。クラス COURSE のコンポーネント・オブジェクト自体は、クラス TEACHER が所有する他のオブジェクトから構成される。

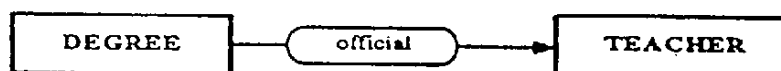
オブジェクトの定義によって、オブジェクトのクラスと、オブジェクトの属性が定義されるクラス（ドメイン）の間にリンクが生成される。たとえば、クラス DEGREE の場合、このような種類の関係を図で示すと以下のようなになる。



1) ハイパーグラフ

オブジェクト指向データベースのスキームはハイパーグラフで表示される。ノードはクラスを表し、リンクは（継承またはドメインの型の）束縛と対応している。

束縛は、特定の条件が現れた場合にシステムが起動するメソッドによって実現される。リンクはメソッドを表し、その引き数はリンクされたクラスである。リンクは、いくつかのクラスが存在するメソッドを図で示している。リンクはどれもボタンと対応している。ボタンによって、束縛と関連する属性の名前を見つけることができる。たとえば、以下のリンクはクラス DEGREE の属性 official と、そのドメイン TEACHER の間のドメインの束縛と対応している。

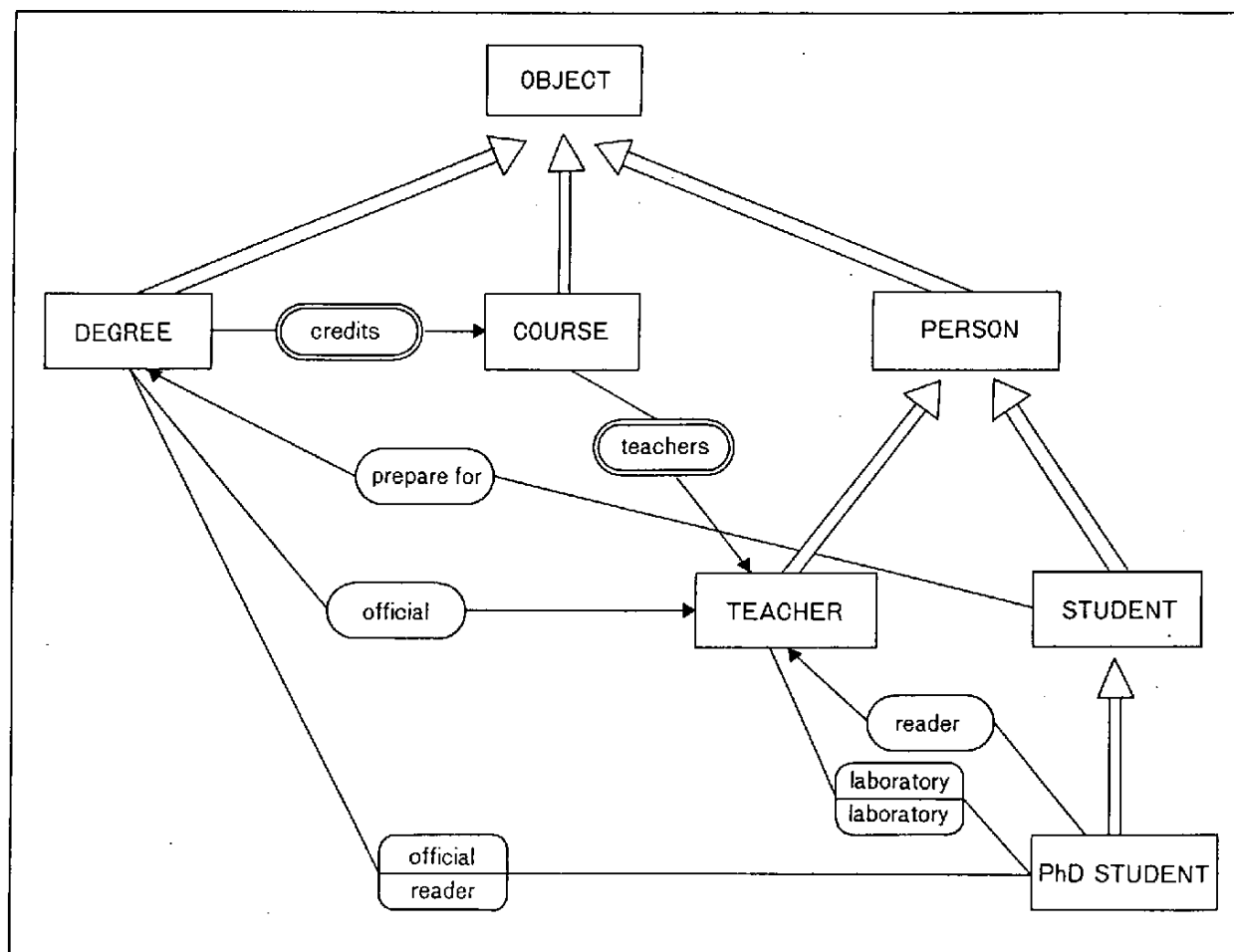


メソッドを実行するには、マウスによってボタンを選択する。そうするとリンクが反

転表示される。

official

グラフはクラス OBJECT と、ユーザの作業が可能なクラスを表示している。これ以外のクラスで定義済またはアトミックであるもの、および定義されていないもの（たとえばクラス LABO）はユーザが見やすいように表示されない。



ユーザに提示されるハイパーグラフ。

リンクの種類

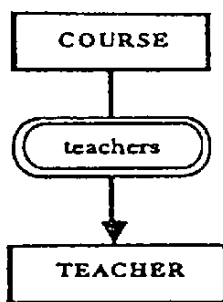
(1) 継承リンク

このリンクによって、スーパークラスとそのサブクラスの間で特性の伝達が可能となる。継承リンクが選択されると、属性とメソッドがスーパークラスからサブクラスへ（たとえば、STUDENT から PhD_STUDENT へ）転送される。そうすればユーザは、

スーパークラスのインタフェースによって完全なものとなったサブクラス・インタフェースを見ることができる。

(2) ドメイン・リンク

このリンクは、単一ドメイン、および複合ドメインの束縛と対応している。ここでは合成ドメインのリンクと、他と共存可能なドメインのリンクは分けて考えることとする。



合成ドメインのリンクはベースのオブジェクト間の合成関係を表している。

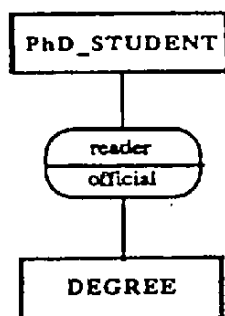
リンク “prepare_for” は、クラス STUDENT の属性がドメイン DEGREE で定義されることを示している。学生のオブジェクトはコンポーネントを持っており、これは学位のオブジェクトである。

リンク “teachers” は、COURSE の属性が、コンストラクタ SET または LIST を持つ構造化ドメインで定義されることを示している。講座のオブジェクトはコンポーネントを持っており、これは教師のオブジェクトの集合である。

リンク “prepare_for” は一つの値しか持たないが、リンク “teachers” は複数の値を持つ。

これらのリンクは複合オブジェクトの合成と、オブジェクトの周辺を動き回る可能性を表示している。リンクの方向は合成の方向を示している。リンクが選択されると、コンポーネントはアクセス可能となる。このコンポーネントは [KIM89] に説明があるクラス間の暗黙の結合と対応している。

他と共存可能なドメインのリンクは、クラス間で明示的な結合がありうることを示し



ている。ユーザは結合の述語を指定しなければならない。

リンク “official reader” は、これらの属性が、他と共存可能なドメインで定義されることを表している。

たとえば、このリンクが選択されると、クラス PhD_STUDENT および DEGREE は結合の述語を照合する phd_students と学位を含むことになる。

注記 定義済のアトミック・ドメイン (INTEGER、STRING など) と対応する単一ドメインの束縛はリンクでは表現できない。ユーザが見ることができるのは、セマンティック・ドメイン間での結合の可能性だけである。したがってユーザの照会内容の一貫性は保証される。

2) オブジェクトとクラスのハンドリング

(1) オブジェクト

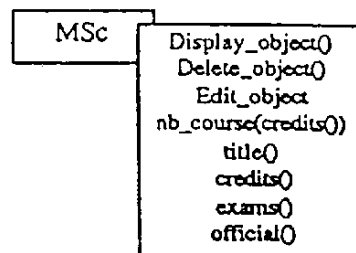
オブジェクトは識別子を持っているが、これはオブジェクトの作成時に与えられた名前か、あるいはシステムによって生成された識別子のいずれかである。システムはオブジェクトをアイコン (矩形の形をした、オブジェクトの識別子) によって表す。オブジェクトのインタフェースは、オブジェクトの公開されたメソッドから成るメニューである。メニューの操作はマウスとそのボタンによって行う。

カプセル化の概念では、オブジェクトの構造をユーザから見えないようにすることが必要である。クラスの属性毎にシステムは、属性と対応し、同じ名前を持つメソッドを自動的に生成する。オブジェクトのために、このメソッドは対応する属性の値を返す。メソッド title ()、credits ()、exams ()、official () は自動的に生成され、オブジェクト degree のインタフェースの一部となる。これは重要なことである。なぜなら

ば、これらのメソッドは、クラスのハンドリングを行うメイン・メソッドで引き数として使用されるからである。

これらのメソッド以外の、公開されたメソッドはスキーム・ベースで定義される。オブジェクトの内容を表示する `Display__object ()`、`Delete__object ()`、`Edit__object` のような標準メソッドもここで定義される。

たとえば、オブジェクトの学位を “MSc” にしてみよう。
このアイコンとインタフェースは以下のようにになる。



システムは、合成ドメインのリンクの方法によりオブジェクトの間を動き回ることを許している。リンクが選択されないと、オブジェクトのコンポーネントにはアクセスできず、オブジェクトの識別子しか見ることができない。この場合、オブジェクトのアイコンは灰色となり、オブジェクトのインタフェースにはアクセスできない。

例をあげれば以下のようにになる。



(2) クラス

クラスの識別子はその名前である。ハイパーグラフのノードは、アイコンによりクラスを表している。システムでは、単一クラスまたは複数クラスの照会内容の表現を可能とする定義済メソッドのセットを提供している。これらのメソッドは、クラスの作成時にクラスとリンクされる。これらはインタフェースの一部である。これらによってオブジェクトのセット・ハンドリングが可能となる（ユーザにとって、クラスはオブジェクトのセットである）。メソッドには以下のものがある。

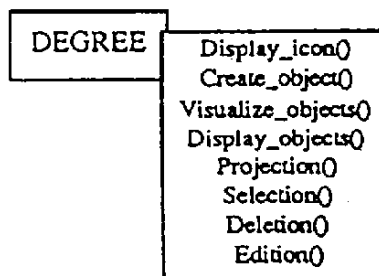
*オブジェクトのセットを表現する `Visualize__objects ()` と `Display__objects ()`、

- * 表示するオブジェクトのコンポーネントを計算する Projection ()、
- * オブジェクトのセットを選択する Selection ()、
- * オブジェクトのセットを更新する Edition () と Deletion ()。

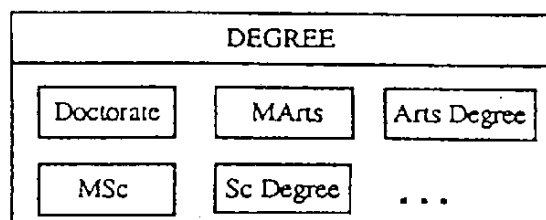
引き数としての数個のクラスによって演算、特に結合、連合、差異のような2進演算を実行するメソッドはクラスCLASSで記述される。このインタフェースの目的として、結合を図で示すことによりこれを容易なものとするのが一つあげられる。したがって、クラスCLASSはユーザからは見えない。これは、ユーザが結合演算以外の設定演算を行うことを望めばハイパーグラフによって目に見えるものとなる。

メソッドCreate__object () により新しいオブジェクトが作成され、Display__icon () によってアイコンとしてのクラスが表示される。

たとえば、DEGREEをクラスにしてみよう。DEGREEのアイコンとインタフェースは以下ようになる。



メソッドVisualize__objects () によりノードの内容にアクセスできる。これはアイコンによってオブジェクトを表示する。



メソッドDisplay__objects () により、全オブジェクトを表示しながらノードの内容にアクセスできるようになる。

DEGREE					
オブジェクト	表 題	単 位	試 験	職 員	
Sc Degree	科学の学位	CD1 CD2 CD3 ...	1. 01/06 2. 03/09 ...	Smith	
MSc	科学博士	CM1 CM2 ...	1. 04/06 2. 03/09 ...	Lewis	
Doctorate	名誉博士	...			

3) 照 会

照会はメソッドのサインのシーケンスである。照会はユーザの手によって発生し、終了する。照会の発生はメソッドの起動、すなわちメッセージの設定と対応している。ハイパーグラフでは、照会は反転表示されたノードとリンクのセットである。

照会が行われた結果、システムによって新しいクラスが自動的に生成される（これはクラス作成のメッセージを通じてクラスCLASSに送られる）。

(1) 単一クラスの照会

① プロジェクション

プロジェクションにより、クラスのあらゆるオブジェクトに対して1個または複数のメソッドが割り当てられる。これらのメソッドは、クラスの属性と関連するメソッドのなかから選択される。したがってプロジェクションは、オブジェクト毎に、起動されたメソッドと対応する値を保持しているクラスを削減する。

クラスのインタフェースでメソッドProjection () を選択することにより、引き数ウィンドウが突然現れ、ユーザはこのなかから割り当てるメソッドを選択する。

たとえば、クラスDEGREEの場合、ウィンドウは以下ようになる。

学位の表題と職員を取得するには、メソッドtitle () およびofficial () を選択する。

Projection Arguments
title()
credits()
exams()
official()

選択すると、メソッドtitle () およびofficial () が割り当てられたオブジェクトdegreeを集めた新しいクラスが作成される。

ハイパーグラフのノードDEGREEはこのテンポラリ・クラスを表している。メソッドDisplay_objects () を実行すると結果は以下のようなになる。

DEGREE		
オブジェクト	表 題	職 員
Sc Degree	科学の学位	Smith
MSc	科学博士	Lewis
Doctorate	...	

メソッドProjection () により作成したアプリケーションは、削除された属性に対して束縛が行われていた場合、ハイパーグラフでのリンクを動的に削除してしまう可能性がある。

②選 択

選択により、属性に関連するメソッド上の、あるいはクラスのスキームで定義されるメソッド上の述語を照合するクラスのオブジェクトを保持することが可能となる。選択によって、述語を満たすオブジェクトへのクラスが限定される。

たとえば、「職員がルイスとなっている学位は何か？」という照会をしてみよう。

述語ウィンドウは以下のようになる。

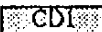
Selection Predicate		
title()	>	AND OR
credits()	>=	
exams()	<	
official()	<=	
nb_course(credits())	=	
official() =		Lewis

これはオブジェクトの比較である。システムは二つのオブジェクトが同じものであるか照合を行う。ハイパーグラフのノードDEGREEは照合結果のクラスを表している。

述語を述べるには、提示されたメソッドのなかから左側のオペランドを選択し、提示された演算子のなかから演算子を選択する。右側のオペランドはアトミック値、オブジェクト名、図によって選択したオブジェクト・アイコン、リスト、オブジェクトのセット、メソッドのいずれかとなる。セットまたはリストの場合、ユーザは、自分の値（アトミック・オブジェクトの場合）、名前、アイコンを指定するオブジェクトのシーケンスを示すことになる。

たとえば、「“CD1”、“CD 2”を教える学位は何か？」という照会をしてみよう。

述語は次のようになる。

credits() >= ( )

削除と編集

この機能によって、同じ方法で表現された述語に従い、クラスのオブジェクトの削除または更新が可能となる。

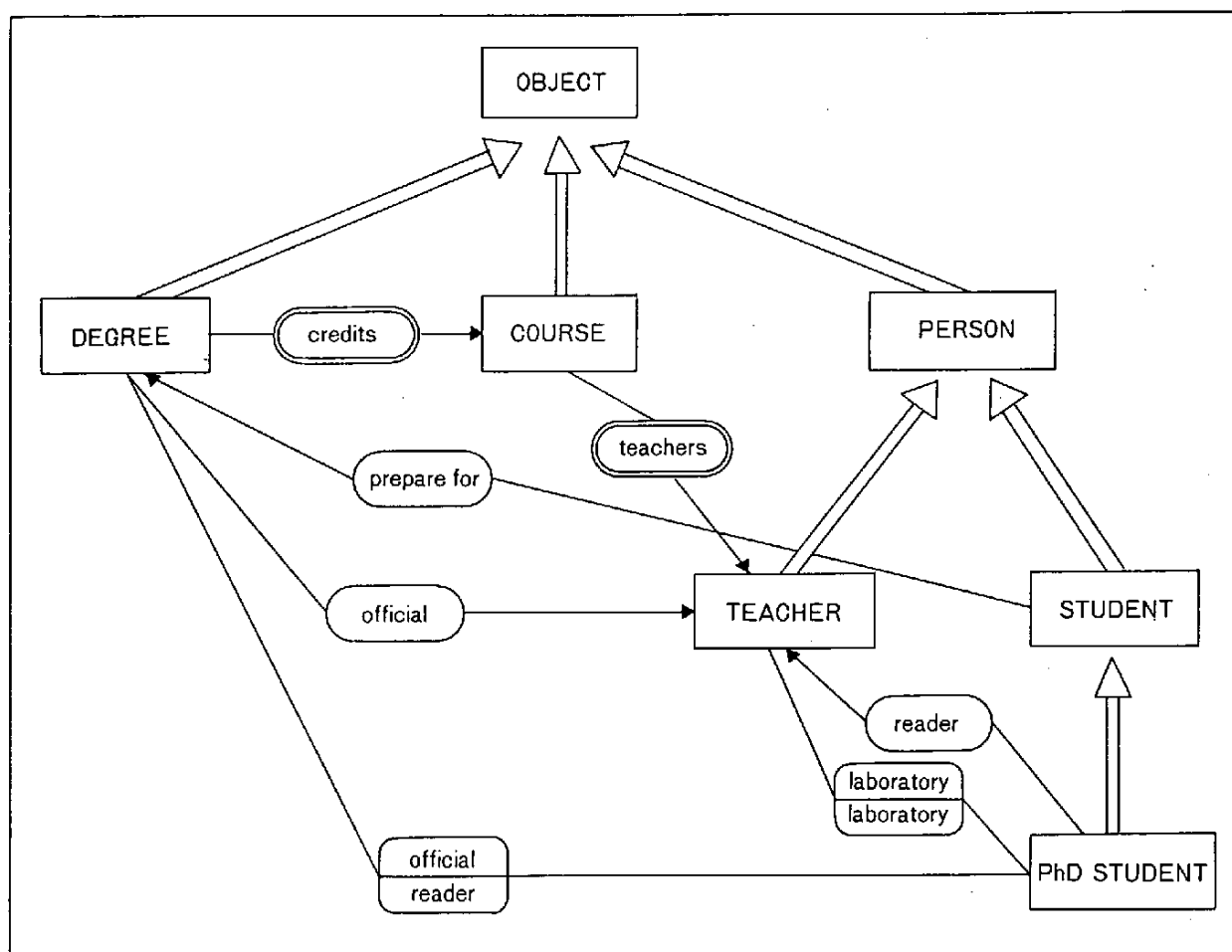
(2) 多重クラスの照会

①合成ドメインのリンクの選択

*このリンクを選択すれば、オブジェクトのコンポーネントを取得できる。これによ

り、複合オブジェクト間を自由に動き回ることが可能となる。

学位の複合オブジェクトについて考えてみよう。以下の例では、ユーザはリンク “credits” および “official” を選択している。



システムは、クラス DEGREE、COURSE、TEACHER 間の結合を暗黙のうちにやっている。学位のオブジェクトはどれも対応する教師のオブジェクト、および対応する講座のオブジェクトと結びついている。

照会に対して返ってくるものは、反転表示されたサブグラフによって表現される複合オブジェクトのセットである。DEGREE はサブグラフのルートであり、これ以外のクラスはコンポーネントのクラスである。ノード COURSE には講座が格納され、この講座はノード DEGREE のグラフにおける単位である。ノード TEACHER には教師が格納され、これは学位の責任者（職員）である。

メソッド Display_object () によりクラス DEGREE を調べると、結果は以下のようになる。

DEGREE					
オブジェクト	表 題	単 位	試 験	職 員	
Sc Degree	科学の学位	CD1 CD2 CD3 ...	1. 01/06 2. 03/09 ...	Smith	
MSc	科学博士	CM1 CM2 ...	1. 04/06 2. 03/09 ...	Lewis	
Doctorate	名誉博士	...			

Display_object()
 Delete_object()
 Edit_object()
 wage(grade())
 grade()
 laboratory()
 room()

講座と教師のオブジェクトのアイコンはもう灰色ではなくなり、オブジェクトはアクセスが可能である。たとえばユーザは、オブジェクトが “Maitrise”（オブジェクトが “Lewis”）である職員をインタフェース・メソッドによって調べることができる。

メソッド Display_object () は以下の処理結果を返す。

TEACHER	grade	laboratory	room
Lewis	245	IRIT	410

クラス TEACHER とクラス PERSON の間で継承リンクを選択すれば、この両者の間で

特性の伝達が実際に行われる。教師のオブジェクトには、対応する人物のオブジェクトの属性とメソッドを含むインタフェースがある。オブジェクト“Lewis”にアクセスすると、処理結果は以下のようになる。

教 師	名 前	洗礼名	ss#	写 真	等 級	研究所	研究室
Lewis	Lewis	John	1540431172274	photo Lewis	245	IRIT	410

ユーザは、“photo Lewis”のボタンを活性化することにより、調査の続行と“Lewis”の写真へのアクセスが可能となる。すると写真が画面に表示される。

同じような方法でユーザは、学位のオブジェクトのなかにある講座のオブジェクトを手に入れることができる。



*複合オブジェクトのセットにおいて選択を行うには、サブグラフのクラスにおいて選択照会を行う。COURSEまたはTEACHERにおける選択は、コンポーネント・オブジェクトにおける条件の設定と同じである。

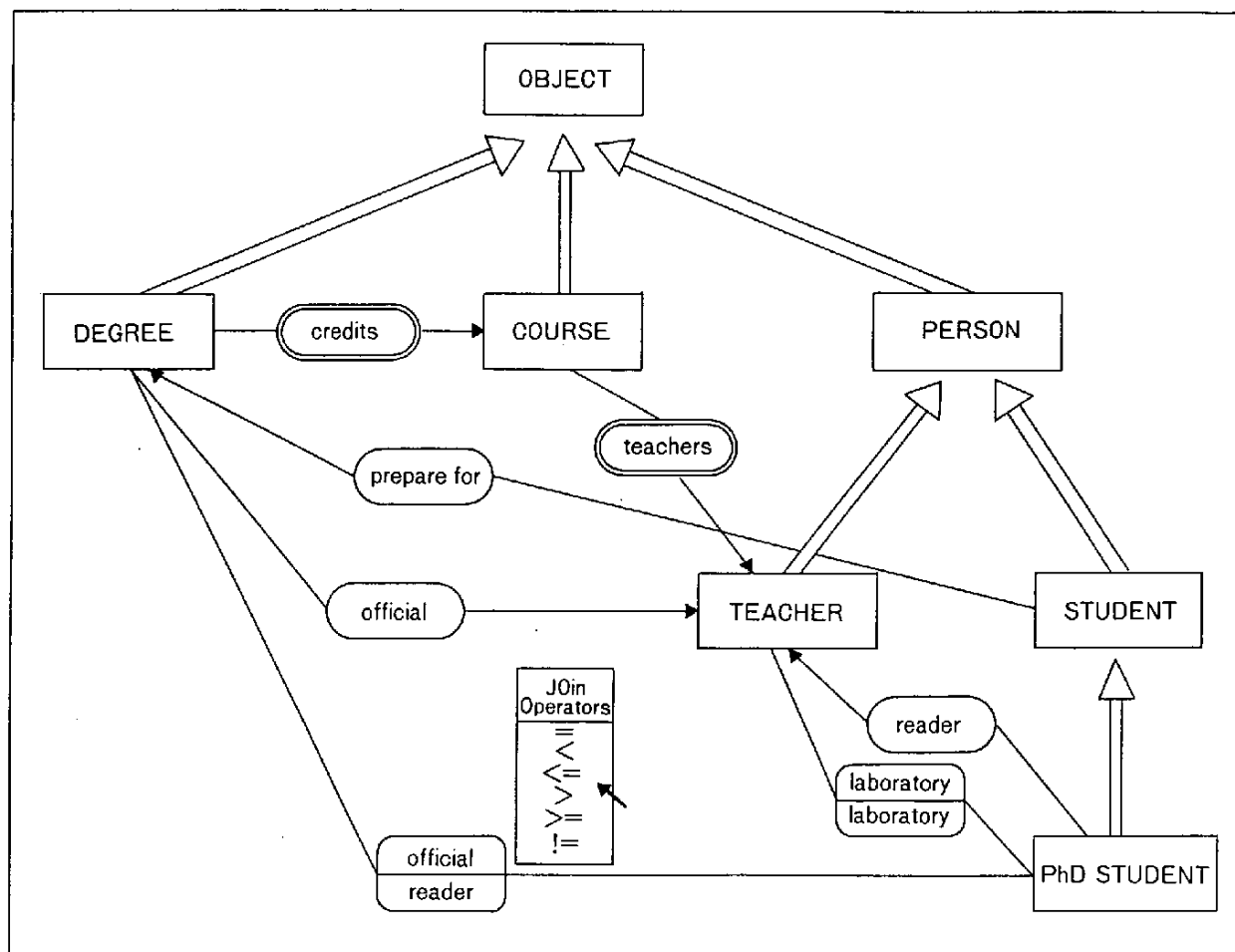
たとえば、ユーザはTEACHERにおいてメソッドselection () を起動し、“Lewis”および“Smith”へのオブジェクトのセットを削減する。DEGREEには職員が“Lewis”および“Smith”である学位だけが、COURSEにはこの学位の単位である講座だけが格納される。

②他と共存可能なドメインのリンクの選択

他と共存可能なドメインのリンクを選択すると、二つのオブジェクト・クラス間の結合を明示的に行うことが可能となる。この結合が行われると、結合の述語を満たす各クラスのオブジェクトの連結によりオブジェクトのセットが得られる。テンポラリー・クラスはシステムによって生成される。テンポラリー・クラスのインテンションは、結合された二つのクラスの属性とメソッドから構成される。またテンポラリー・クラスの

インタフェースは、結合された二つのクラスのインタフェースで記述されるメソッドから構成される。

クラス DEGREE、PhD_STUDENT 間でユーザがリンク “official,reader” を選択している例について考えてみよう。この選択により、ユーザが結合演算子を指定する演算子ウィンドウが突然現れる。

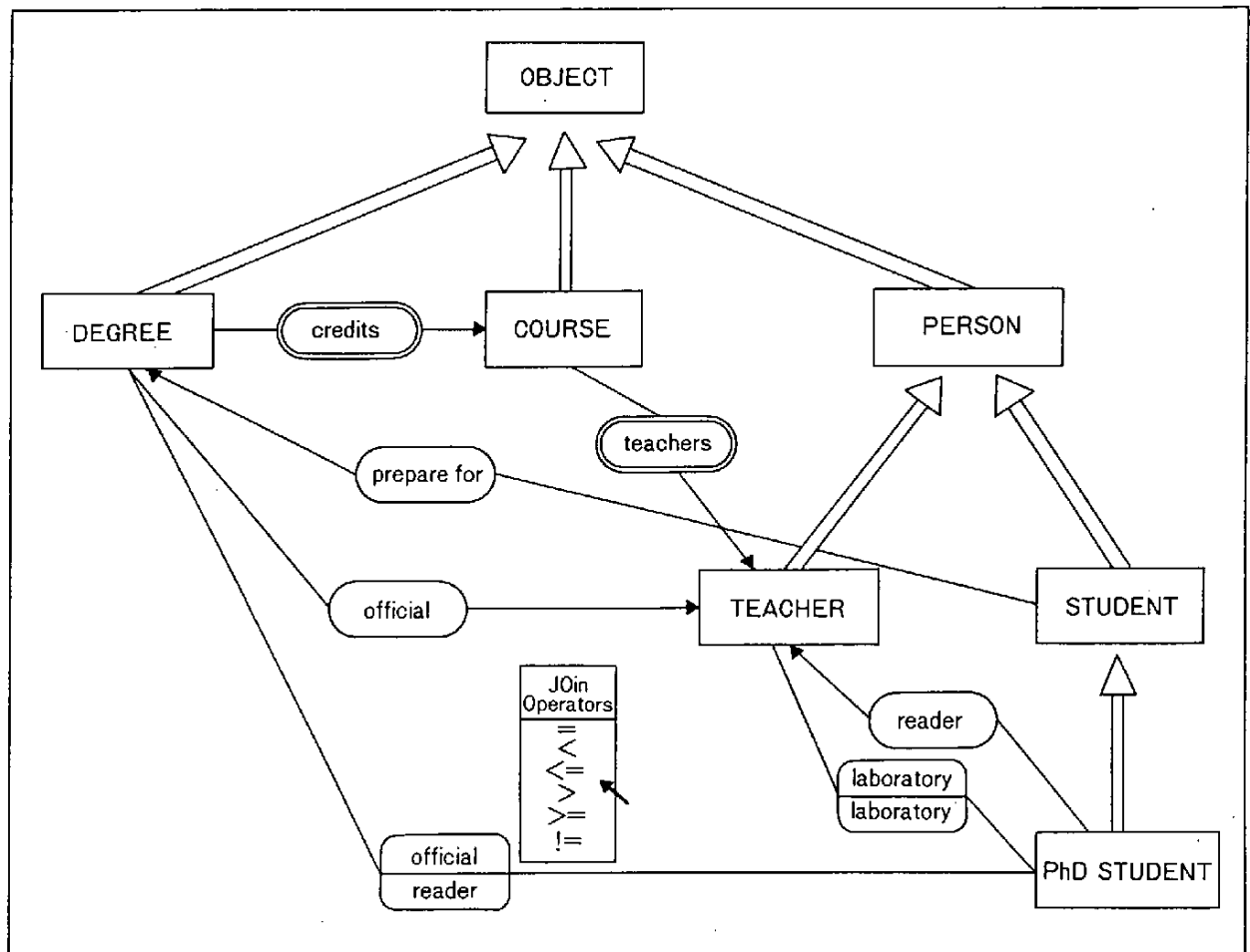


ユーザは、クラス DEGREE と PhD_STUDENT において結合結果を見ることができる。クラス DEGREE と PhD_STUDENT は、これらの各属性と関連するメソッドのプロジェクトによって得られる。

③複合オブジェクトのセットのハンドリング

合成ドメインのリンクと、他と共存可能なドメインのリンクを選択すれば、複合オブジェクトのセット間で結合が可能となる。

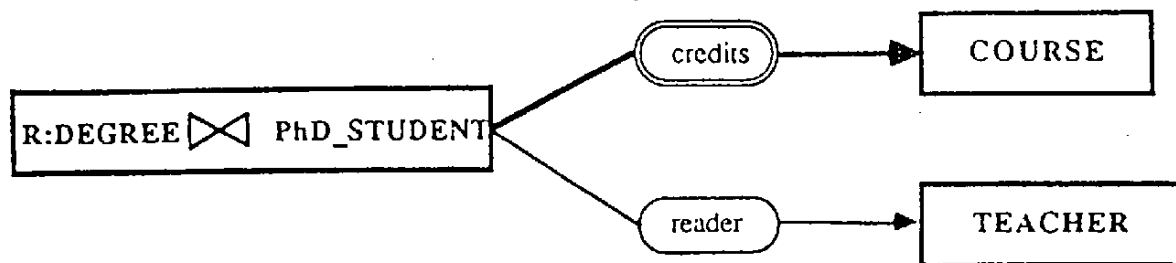
以下の例について考えてみよう。ユーザは合成ドメインのリンク “credits” と “reader”、および他と共存可能なドメインのリンク “official,reader” を選択している。他と共存可能なドメインのリンクは複合オブジェクトの二つのセット間での明示的な結合と対応しており、この複合オブジェクトのセットは、ひとつはクラス DEGREE、COURSE とリンク “credits”、もうひとつはクラス PhD_STUDENT、TEACHER とリンク “reader” で表される。



ユーザは、クラス DEGREE、PhD_STUDENT、COURSE、TEACHER において照会結果を見ることができる。

照会に対して返ってくるものは複合オブジェクトのセットであり、これはグラフによって示される。合成ドメインのリンクは照会グラフのリンク、すなわちリンク “cred-

its” および “reader” と同じである。直接、明示的な結合のなかにある二つのクラスを除けば、グラフのクラスは照会グラフのクラスである。これらのクラスは、結合によって作成されたクラスによって置き換えられる。ここにあるのは、クラス COURSE、TEACHER と、DEGREE、PhD_STUDENT 間での結合によって生まれたクラス R である。この結果は以下のように示される。



出典：

HYPertext INTERFACE FOR
AN OBJECT-ORIENTED DATABASE

E. ANDONOFF - M. CANILLAC
C. MENDIBOURE - G. ZURFLUH

I.R.I.T. & C.E.R.I.S.S.
I.R.I.T. - S.I.G. - Université Paul Sabatier
118 Route de Narbonne - 31062 TOULOUSE Cedex (FRANCE)

4.2 情報検索のための媒体間の操作可能なコード化されたもの

4.2.1. 序

“IR システムの設計に当たって、情報の対象となる集合体をブロージングできるようなツールをユーザーに与えるものとして、考えることにしたい。これらの対象とは、テキストの一部、イメージ、絵、ボイス・レコーディングなどである。”

ここでは、まず、情報検索 (IR) システムに対するインタフェースについて述べる。これは、ユーザーがサーチ・プロセスの演算子を、代替となる媒体 (グラフィックス、自然言語など) におけるコード化されたものに適用できるようにして、ある媒体 (テキスト、フィルム、音楽など) におけるコード化されたものとして提示された情報を、動的サーチ、予約サーチできるようにするものである。サーチ・スペースにおける演算子を適用することは、計算に関するアクティビティーであり、ここで扱っているのは、サーチ、推論、認識などの計算に関するプロセスへの考察を含む、計算に関する見方である。ある意味で、動的サーチはブロージングと同義であるが、様々なタイプのブロージング・アクティビティーを統合するものであると考えられる。例えば、Palay と Fox が、“うまく結合された記録のスペースにおける探求的サーチ”と述べたような目的を持ったブロージングから、Canter らの定義で示されたような、“ユーザーは、興味あるところにぶつかるまで、データが連れていってくれるところどこへでも喜んでいく”、というような、あるいは、Cove と Walsh が、“ブロージングとは、何を求めているかをそれが見つかるまで知らずにいるやり方”と述べたような目的の無いものまで、様々なタイプのブロージング・アクティビティーがある。いずれにしても、後に論じるように、ブロージングは効果的であることがわかってはいるが、IR の性質は、ブロージング・タイプ・システムにおいて、まだ仕様の余地、つまり情報への要求 (“照会”と呼んでもよいが) を構築する余地があることを示している。ゆえに、ユーザーがサーチ・スペースの表示の操作に最初から最後まで従事する “動的サーチ” をここでは意味している。

ここでは、まず、人工知能 (AI) 研究を引き合いに出して、表示、媒体と、ユーザーによる計算上のアクティビティーとの関係を論じる。その後、ある媒体における一つ

のコード化されたものでの操作が、別の媒体における操作を表示する、媒体間の操作可能なコード化されたもの（MIME）について述べる。次に、MIMEがなぜIRで必要とされるかを、関連のある、人とコンピューターの対話（HCI）、および媒体研究に関して論じる。さらに、MIMEベースのインタフェースを動的サーチに使用している、いくつかのIRシステムの概観を提示する。特に、テキストにない情報ベースを持つシステムにおいて、1つの媒体の（上記の意味における）計算に関する制限を、別の媒体でのコード化されたものを操作することで、いかに克服できるかを見る。それから、焦点をIRのためのMIMEの設計および効果の側面にあて、媒体の提示スペース（提示可能なすべてのコード化されたもののセット）と、アクセス・スペース（情報のための可能なすべての仕様のセット）との違いについて論じる。この分析は、HCI研究の対話装置に関連しており、また、“全体的”外部サーチ・スペース表示と“連続的”外部サーチ・スペース表示との区別にも関連している。MIMEを使用した照会に対して、現在“選択ベース”のインタフェースで利用できるものよりさらに合成されたアプローチを採用できるかについても考察する。MIMEベースのIRシステムにおける推論の役割の重要性は、まだ十分に理解されていないが、それについて、学習の反復的側面におけるインテリジェント・チュートリング・システム（ITS）の研究に関連して論じる。最後には、IRシステムにおけるMIMEの使用を提案し、研究の重要な分野の概要について紹介したい。

4.2.2. MIMEとユーザーの計算に関するアクティビティの外面化

コンピューター科学では、表示ともっとも関連のある分野は人工知能（AI）の分野である。AIの初期の時代には、研究者は、形式では異なるが、内容では異なっていない表示の間の計算に関する違いを確立することに関心を注いできた。AIの研究者達は、ダイアグラム形式のような代替となる外部表示（LarkinとSimon、1987;Funt、1980など）に関心を向けてきたが、媒体の役割は、大きくはなかった。研究者達の目的が、機械で処理可能な形式での外部表示の、計算上効果的な側面をとらえることであったからである。この形式は、通常、線形テキスト記述言語の形を意味している。ゆえに、LarkinとSimonの、幾何学定理用ダイアグラム公式化の“永久に向上する”テキスト記述が問題となった。しかし、ダイアグラム表示の有効性の分析において、Larkin

と Simon は、複数媒体のインタフェースの分析と評価に使用できる 2 つの有益な概念を提供している。第一に、コンピューテーション（計算）という概念がある。コンピューテーション（計算）の 3 次元は、サーチ、認識、推論である。コンピューテーション（計算）というこの概念が、第二の有益な概念の要因となる。すなわち、情報上の等価と計算上の等価の区別という概念である。

Larkin と Simon は、情報上等価であり代替できる表示を図式で表すことは、計算上のコストという理由から、もっとも効果的になりうると指摘している。たとえば、図はしばしば近似の対象物の属性を示すので、サーチを減らすことが可能である。認識のプロセスが利用されるので、引き出されるべき推論の量も減少する。幾何学的問題の一部のテキスト記述を例にとってみよう。2 つの平行線が横断線で遮られるとする。この記述によって 8 つの角が暗黙に作り出されるが、その存在を推論しなければならない。一方、ダイアグラムでは、8 つの角をはっきりと表示することができる。ここでは、Larkin と Simon のコンピューテーション（計算）の概念を採用する。この概念は、ユーザーが IR システムと対話する、実際のアクティビティーを引き合いに出すものであるため、ユーザー・コンピューテーションと呼ぶことができる。後述するように、この論文で論じられているインタフェースの様々な側面が、ユーザーがそういったコンピューテーション（計算）を積極的に実行するのを促進することになる。

ここで使用される“表示”という用語は、ユーザーが利用できる媒体の要素（コード化されたもの、など）の特殊な配列を意味することにする。特に、なんらかの形でユーザーが操作できる外部コード化されたものに焦点を当てた場合、こういった操作の実行とその結果が、IR のアクティビティーの一部になりうる。さらに、検索される情報は、必然的に媒体内のコード化されたものにおいて具体化もされる。また、媒体間の操作可能なコード化されたもの、（略して、MIME と呼ぶ）は、外部コード化されたものが表現される媒体（外部媒体）が、必要とされる情報がコード化されている媒体（ベース媒体、または基礎的媒体）と区別され、かつ、ユーザーが外部コード化されたもので実行する操作が制約付きのものなので、操作がベース媒体における結果を引き合いに出すような場合を指す。

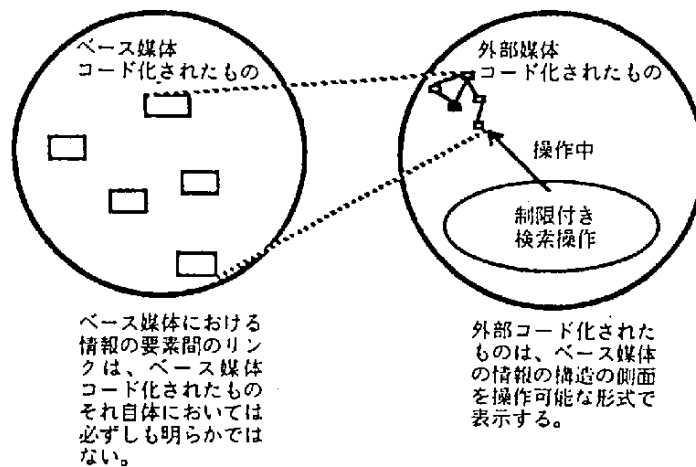


図4-1 IR状況用のMINEの概観

図4-1はMIMEベースのIRの概観を示している。MIMEを使用することで、ユーザーは、ベース媒体で直接実行すると、非常に困難か、不可能であるような計算上のアクティビティーに積極的に従事することができる。1つの与えられた媒体が、いくつかの点において強力な計算上の特徴を持っていたとしても、他の点においては計算上の制限を持っていることもある。

次に、多くの場合テキスト・データベースをベース媒体として持っている様々な既存のIRシステムにおけるMIMEの使用について論じる。しかし、MIMEは論じられている他の2つのシステムにおいても特色になっている。その1つは音楽をベース媒体としており、もう1つはビデオである。上で論じられた計算上の見方に重点が置かれている。

4.2.3. IR用のMIME

ある媒体においてコード化されたものを使って、別の媒体においてコード化されたものを表示するのは、一般的に行われる。たとえば、音楽を表すためのテキストとなった楽譜がある。MIMEベースのコンピュータ・インタフェースを使用することで付け加えられることは、1つのコード化されたものの操作が、他の媒体におけるコード化されたものという形で、結果を引き起こすことが可能であるということである。(たとえば、音楽のスコアのテキスト表示の音符を操作することが、音楽の音、という形で直接的な結果をもたらし、またその逆もある。) さらに、MIMEが適切に設計されて

いるならば、ユーザーが実行する操作は、どれも少なくとも表面的には、十分に納得のいく結果をもたらすことができる。これが、いわゆる“直接操作”インタフェースの根本的哲学である。次のセクションで、MIMEの使用に関わるより複雑な問題を示す前に、まず、なぜMIMEがIRに有益なのか、その理由を論じてみたい。それから、(a) ユーザーになされる計算上の要求及び(b) MIMEが対処するベース媒体の特徴、という観点から、MIMEの既存のIRシステムでの使用を提示し、論じる。

1. IR用になぜMIMEが必要であるか

diSessaによると、インタフェースのもっとも重要な機能の1つは、見えるものと、それを理解することとの間に介在する、目に見えない構造物を取り除くことであるという。同様に、Hutchinsらによれば、“インタフェースは、ユーザーとシステムが、想定されてはいるが、明らかに表現されてはいない世界について会話をかわす言語媒体であり、インタフェースは、ユーザーと、物事が言われている世界との間の言外に示された媒介である。”としている。最近では、多くのものが、いわゆる“直接的操作”インタフェースから成されている(Shneiderman、1983;Hutchinsら、1985;Hutchinsら、1986)。そのようなインタフェースの特徴は、興味ある対象物を継続的にビジュアルに表示し、それに基づいて、ユーザーは物理的操作を行うことができ、その結果はすぐにディスプレイで明らかになるということである。しかし、IR アクティビティーの特質は、興味ある対象物がベース媒体の要素であり(たとえばデータベース内のドキュメントなど)、そういった対象物への物理的操作が常に可能とはいえないようなときに、興味ある対象物を継続的に表示することである。重要な点は、“興味ある対象物”は、データベース中に散らばっているかもしれず、それゆえに“目に見えない構造物”は、ベース媒体自身の固有の特徴であるかもしれないということである。ゆえに、媒体の特殊な要素を直接表示することは、そして認識の範囲を利用することは、可能であるかもしれないが、目に見えない構造物の操作可能な表示、(すなわちサーチ・スペースの表示)を提供しない限り、サーチの範囲を完全に利用することは可能ではないのである。

つまり、IR用の階層メニュー・システムでは、メニューの項目がデータベース検索の際のキー用語となっている。メニューは、コマンドベースのシステムにおける向上を提供するためのものである (Shneiderman, 1982; Howes と Payne, 1989; Allen, 1982)。というのは、とりもなおさず、メニューはコマンドの認識を検索より優先させるが、オプションの奥行き (メニュー・レベルのではない)、幅 (与えられたメニューのオプションの数)、ラベリングが問題を引き起こす可能性がある (Kiger, 1984; Snowberry ら, 1985; Laverson ら, 1987) ような情報検索に対して、メニューがどれほど適切であるかははっきりしていないからである。Landauer (1987) が、情報検索用のメニュー・オプションのラベリングに言及して、“自然のセットを1つ以上の任意の区分に分けても、必ずしも人間の認識に同じように向くカテゴリーを生み出すとは限らないし、またその区分に同じように理解可能な名称を割り当てられるとも限らない”と指摘している。実際、Snowberry らは、サーチ・アクティビティーの際に階層メニューでアクセスする情報検索システムにおいて、ユーザーが経験した困難を示す研究を引用している (すなわち、ユーザーは、わからなくなったサーチに見切りをつけ、木の下層部で代替となるパスを見つけようとする際ではいけないとされているのに、トップレベルまで戻って、ミスをおかし、木の下層部へ進むにつれ、ますます時間がかかってしまう、というものである)。Snowberry らは、ユーザーのサーチの実行を向上させるために、サーチ・スペースの構造についての情報をいかにメニューベースのIRシステムに付け加えられるかを示している。

メニュー・システムは、ユーザーがベース媒体自身においては直接利用することができないサーチ・スペースの構造を表示するものだが、それでもまだ、“遠ざけられた窓”と呼ばれる (Reichman, 1986) ものに似た問題を残している。すなわち、ユーザーは、ディスプレイの“ビジュアル運動量” (Woods) と呼ばれてきたものの認知アカウントを維持する責任をすべて負っているのである。Woods は、平行に提示された方がよいデータが、連続的に提示されたときに起こる、“キーホール現象”と彼が呼ぶものを説明している。Woods によると、キーホール現象は、サーチ型のアクティビティーに関わっている際に“分からなくなる”ことに関連する問題の、第一の要因であるということである。同様に、IRの状況において、Foss (1988) は、ハイパーテキスト

システムでのサーチに関連する2つの問題を説明している。すなわち、“はめ込まれた脱線”（ユーザーが、別の興味ある項目のために、実際のサーチ経路からそれてしまう）と、“美術博物館現象”（興味があり、多種多様なものがたくさんあるために、ユーザーがセッションを終えるときには、何を調査していたのかについてはっきりとした考えがなくなってしまう）である。これらの問題に対するFossの解決法を、次のセクションで論じる。しかし、サーチ・スペースの適切な表示は、単にインタフェースの設計の問題なのではない。連続的調査（テキスト、音楽、フィルムなど）からその情報の内容がもたらされるものの、サーチは直接しなければならない媒体すべての使用に関連する問題なのである。

そういった媒体に関連したもう1つの問題は、媒体における情報の実際の表示と、媒体からの情報に対する要求をユーザーが表現しようとする、または表現できる方法との関係である。CLORISとハーモニースペースシステムについての我々の議論に示されているように、ベース媒体のなかには（フィルムや音楽）、ベース媒体の要素を使用しては表現できない、媒体に関連した情報の側面のあるものがある。特に、特定のタイプのユーザーに関してがそうである（媒体内の情報が表示する領域における初心者など）。こういった場合、以下で検討していくように、情報をベース媒体から代替となる媒体の要素へコード化して、ベース媒体の必要とされる情報についてのユーザーの仕様が表現できるようにする必要がある。これを行う際に、コード化されたものを1つの媒体から別の媒体へ翻訳する能力として、重要なスキルに形体を与えているのである。これは、しばしば、与えられた媒体の専門家以外はできないことである。ゆえに、ユーザーが熟知している媒体にあるか、あるいはユーザーの認識プロセスを引き出す媒体にある、外部コード化されたものを使用すれば、有益である。しかし、外部コード化されたものを設計する際には、媒体間でコード化されたものを翻訳するアクティビティーは、記号の変換を必要とするため。注意を払わなければならない。これは、本質的に、いわゆる“コード化されたものの特徴”原理を言い換えたものである。この原理が主張していることは、メモリーからの情報の検索は、情報が当初そこから得られたところの表示に似た形式のキューを使用することで促進されるというものである（Cardら、1983;WellsとHrywic、1984）。しかし、Whiteらは、理解レベルは、

“媒体の保全を保持することで” 確立できることを示したが（たとえば、ビデオのプレゼンテーションを見て達成された理解レベルを、テスト・プロセスにおいて、口頭の質問だけではなく、プレゼンテーションの画像を使用してテストすることなど）、IRの問題は、媒体からの特定の情報は、（任意のタスクについては、また一定の任意のユーザーの知識レベルに関しては、）その媒体の要素を操作するだけでは表現できない、ということである。この点については、後ほど論じる。

次のセクションでは、主にサーチと認識の計算上の範囲に関した問題に対処するのに MIME を使用している IR システムをいくつか例に挙げる。また、それに続くセクションで考慮されているシステムは、ベース媒体の要素を使用するだけでは可能ではないような（検索用の）情報をユーザーが指定できるようにする MIME を提供するシステムである。

2. 動的サーチのためのサーチ・スペースの具体化

MIME のなかには、テキストベースの IR システムに対するユーザー・インタフェースに使用されているものがある。たとえば、ブックハウスシステム（Pejtersen、1989）は、高分解能カラー・グラフィック・ディスプレイの形で、情報検索環境を表示している。このディスプレイは、図書館の建物としてのシステムを実際に表示するもので、ユーザーは、スクリーンの適当な部分をマウスで選択することで、いろいろな部屋を訪れることができるのである。このシステムにおける興味深い MIME は、もっとも頻繁に使用される索引用語を表示するのにアイコンを使用していることである。（たとえば、キスをしているカップルが“ロマンス”という主題を表すのに使用されている。）Pejtersen は、108 のアイコンで 1000 以上の索引用語が結合されたと述べている。Michard（1982）は、ユーザーがベン図表を直接操作して、ブール・ドキュメント検索照会を表現するシステムを記述している。McMath ら（1989）が記述したシステムでは、IR システム用のシソーラス用語の階層を、ネストされた円の集合体として表示している。ルート用語（たとえば、COMPUTERSCIENCE）は、ラベル付きの大きな円で表示され、その周囲の回りにラベル付きの小さな円が付随している（データ、ハードウェアな

ど)。この小さな円の周囲の回りには、ラベルのない小さな円が付いている。選択した円の用語は、照会の形にすることができる。円を選択した結果、ディスプレイが選択された円をルート・ノードとして書き直される。(また以前にはラベルの付いていなかった関連のある円すべてにラベルが現れる。)

上記のシステムによって対処されている計算上の範囲は、おもにサーチと認識の範囲である。ユーザーは、照会ベースのIRシステムの場合同様、分析的サーチ (Pejtersen, 1989; Campagnoni, 1989)、パラメーター付きサーチとも呼ばれているが (Palay と Fox, 1981)、を強制的に使用しなくてもよい。このサーチでは、ユーザーが、必要とされる項目の記述を媒介 (すなわちコンピュータ) に提示しなければならず、今度はその媒介が実際のサーチ・アクティビティーを実行するのである。MIME は、ユーザー自身が積極的にサーチを行うサーチ・スペースの表示を、ユーザーが利用できるようにするものである。この動的サーチは、また、認識プロセスや、ざっと目を通したり、一瞥で知覚したりする能力をも利用している (Bates, 1986)。Anderson (1985) のような心理学者によると、答えが事実上問題の一部であるような認識の問題では、問題に答えるほうが、問題を思い出すより容易であるという。それは、認識の問題が、問題自身のメモリをサーチするための主題を多くの方法で提供するからであるという。

こういったシステムにおける MIME にはもう 1 つ別の範囲が加えられている。ベース媒体では利用することができない、サーチ環境の一定表示を提供するのである。McMath が発見したように、サーチ・スペースの表示は、ユーザーがベース媒体における情報の構造についての知識を形成する際にも有益な場合がある。システムによっては (たとえば、Foss が記述したもの、1988)、これをもう一歩進めて、ユーザーにユーザー自身のサーチ・アクティビティーの概要を示している。これも、元の媒体 (Foss の場合では、ハイパーテキストシステム) では利用することができないものである。MIME の使用をこの場合、“メタコンピューテーション (メタ計算)” の適用の明示と呼ぶことができる。つまり、表示されたサーチ・スペースは、ユーザー自身の計算によるものだからである。これは、ITS (送信勧誘) 研究において、他の場合では隠れているプロセスの“具体化” (Wenger, 1987) と呼ばれるものに関連している。

これは、ユーザーが実行する動的サーチのための媒体としては、大規模なデータベースであり、テキストでは不適切である。つまり、MIMEの使用によって、テキストベースのIRシステムが要求するユーザーの計算の性質が変わることが有り得るということになる。

3. テキストではないベース媒体を持つ MIME

1) CLORIS.

CLORIS (Conceptual Language Oriented to the Representation of Instructional film Sequences、インストラクショナル・フィルム・シーケンスの表示に適応した概念言語) システム (ParkesとSelf、1990) は、いくつかの点において上で記述したシステムの逆である。というのは、CLORISは概念的グラフ表示から、動いているフィルムの事象のテキスト記述を生成するからである。それから、CLORISではベース媒体はビデオ・マテリアルであり、外部コード化されたものは、“自然”言語の媒体において成されるのである。言語におけるこういった記述は操作可能である。それは、ユーザーがそういった記述を(マウスによって)選択できるからであり、またフィルムの事象の因果関係についての情報を求めるために、ユーザーがそういった記述をあらかじめ定義された質問“フレーム”に編入することができるからである。ここで簡単な例を挙げてみよう。(この簡単な例では、CLORISの複雑さはそれ程に扱われていない。) マイクロメーターでメタルの薄片を計測する前にエンジニアがその薄片を取り上げているときに、CLORIS内で動いているフィルムの上演を中断したとする。“エンジニアがワークピースを取り上げている”というステートメントが、スクリーンに現れるステートメントのうちの1つであろう。ユーザーがこのステートメントを選択すると、そのステートメントがユーザーの尋ねようとして選んだ質問の主題となる(メニューからの質問のタイプの選択に基づいて)。この2つの例が、事象に対する理由、という質問のタイプと、結果としての状態、という質問のタイプである。事象に対する理由という質問のタイプは、この場合では、“なぜ、エンジニアはワークピースを取り上げ

ているのか？”ということの意味し、結果としての状態という質問のタイプは、この文脈では、“エンジニアがワークピースを取り上げたためにどのような事態がもたらされたか？”ということの意味する。さらに、その答えは、可能な場合は、自然言語の形で与えられるが、CLORISは、その間、図解するためにフィルム・シーケンス、あるいはスチルを提示する。前記のIRシステムにおけるMIMEは、特定のユーザーの計算には不適当な、ストアされたテキストの特徴のいくつかに対処するが、一方CLORISMIMEは、動いているフィルムに対して類似的機能を実行するのである。動いているフィルムに提示された事象を理解するには（特に、新しい領域では）、背景となる知識に基づいたかなりのレベルの推論が必要とされる。別の文脈において述べられているように、“絵は質問に答えるものではない。絵の力は、質問に答えるツールとして使用されうるその能力にあるのである”（Smoliar, 1987, p 259）。CLORISMIMEは、フィルムの媒体からの要素を使用したのでは提示することができないような、また完全に答えることができないような、フィルムについての質問を、学習者が尋ねたいと望んでいることもあるという事実を反映しているのである。

2) CLANQUE.

CLORISシステムについてはParkes (1989) により詳細に記述されているが、それを拡張したものが、言語という形で表現された操作可能なコード化されたものによって、画像をさらに表示する。拡張によって、ユーザーは、自然言語のステートメントを選択できるだけでなく、それを実際に制約付きの方法で修正して、その先の画像のための照会を表示するステートメントを生成できるからである。ここでは、これがいかに達成されるかについての技術的な面には触れずに、ユーザーが利用できるシステムの計算上の側面を、例を使用して説明してみたい。このシステムは、CLORISLanguageQuestionEditor、CLORIS言語質問エディター（CLANQUE）といい、これは、ポインティングの使用性と自然言語の力を、言語と画像との関係の性質を利用するような方法で結び付ける、実験的な試みを反映したものである。基本的なアプローチは、ここでもビジュアルなデータベースの動的サーチの促進である。ビデオのデータベースにおいては、ポインティングは対象物への興味を示すのに、非常に効果的な方法である。

対象物がスクリーンに見えている場合、ただ単に物理的にそれを指し示せばよいのである（指、またはペン、マウス装置を使用して、感知可能スクリーンに触れることで）。この対話形式には2つの直接的な欠点がある。（a）そこにはないものは指し示すことができないということと、（b）スクリーン上にあるいくつかの対象物を選択することができるとしても、それらの対象物の間の必要とされる関係を指定するのは難しいということである。（たとえば、自動車とガレージを指し示したとしても、自動車がガレージの中にある、という画像を見たいということをどうやって指定するのか？）また、拡張メニューの規定がないと、後に続く画像に適用したいと思うような制限を、指定するのが難しい。（たとえば、赤い自動車を指し示した後、どうやって“青い自動車を示せ”と指示するのか？）最後に、スクリーン上の対象物を指し示すというのは、指定例の描写を指し示すことであるが、どうやって、その指定例から一般例へ進んでいったらよいのか？（“もう少し別の自動車を示せ”とか、あるいは“違った運搬手段を示せ”など）フィルムの媒体にとどまっている限り、こういった問題を克服するのは困難なように思われる。

CLANQUEのアプローチは、画像自体ではなく、画像のテキスト記述を操作可能な対象物として扱おうとするものである。生成されたテキストの意味あるフレーズが、どうやって言語の正式表示から引き出されたかという説明、ならびにそれらのフレーズのディスプレイ・スクリーン上での所在、を維持することで、システムは、ユーザーが意味論的用語でのテキストを指摘するのを解釈することができる。図4-2は、単純な仮想ピクチャー・データベースに対して一連の照会がどう進展するかの概観であり、一連の画像がステートメントの変更によって、プロセスのそれぞれの段階でどのように指示されるかを示したものである。インタフェースは、ゆえに、言語の強力な特徴のいくつか（たとえば、一般性の様々な程度を表現する能力）を利用している。それは、テキストの単純な物理的操作によって達成できるような方法で、言語の生成プロセスを表示することによって行われるのである。これが、ピクチャー上の操作では表現できないような、ピクチャーに対する照会を表現する能力へとつながるのである。

図4-3は、2つの異なった外部媒体が、コンピューターのコントロールするビデオの

検索の制限のいくつかを克服するために使用された方法をいくつか示す、簡単な概観である。

3) ハーモニースペース

MIMEベースのインタフェースの最後の例では、基本媒体は音楽である。ハーモニースペース (Holland, 1989) も、初心者が音楽的合成スキルを得るのに関連した問題に対処している。このシステムは、精巧なMIMEを使用するもので、ペース媒体は音楽の和音であり (シンセサイザーによって演奏される)、外部コード化されたものは3次元の和音名の配列であり、それが、マウスを使った配列でラインを記述して、“うまく形成された” 和音シーケンスの演奏を引き起こすような方法で組み立てられている。このコード化されたもので、ユーザーが自然に見いだす計算の使用が便利になる。たとえば、直線を認識すること、近似の対象物を認識、発見すること、直線の動きをすること、などである。また、このコード化されたものは、従来のMIME (たとえば、テキストの記譜法) が必要とする推論の量を減少させる。MIMEを使用して音楽の合成に熟練するために、ユーザーが必要とする推論の量およびタイプについては、明らかではない。

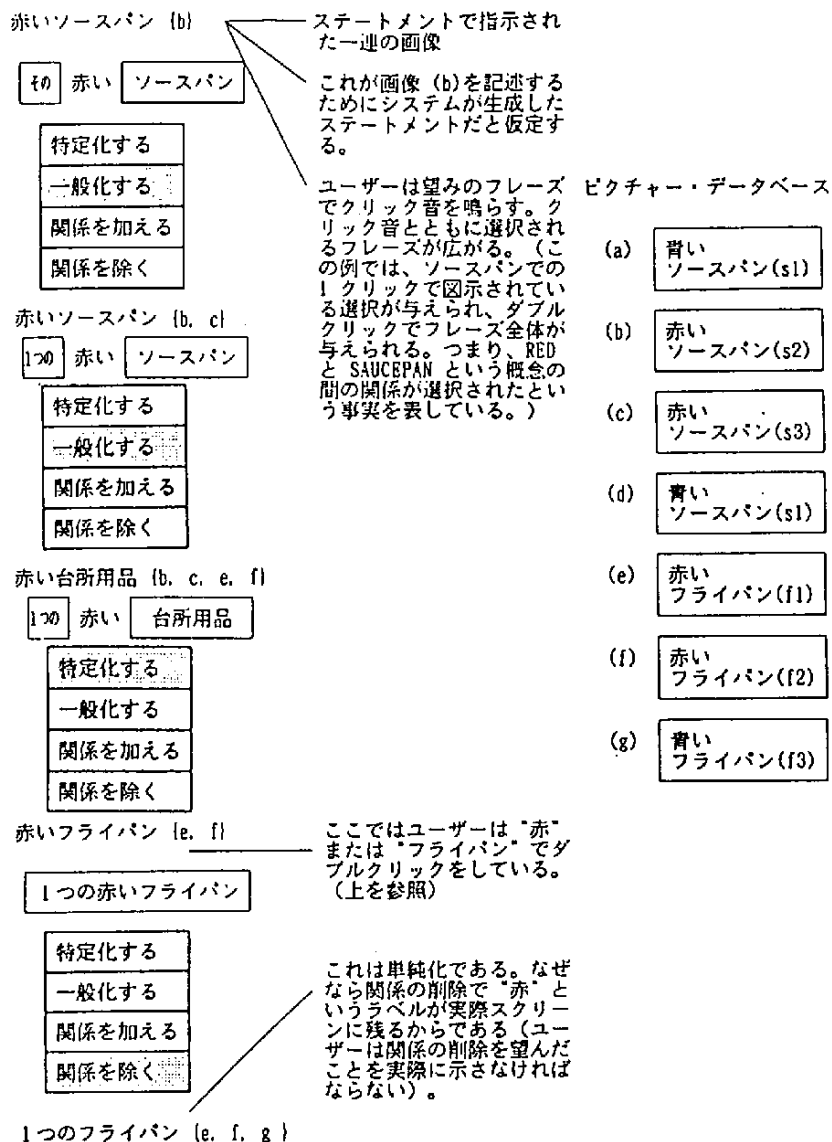
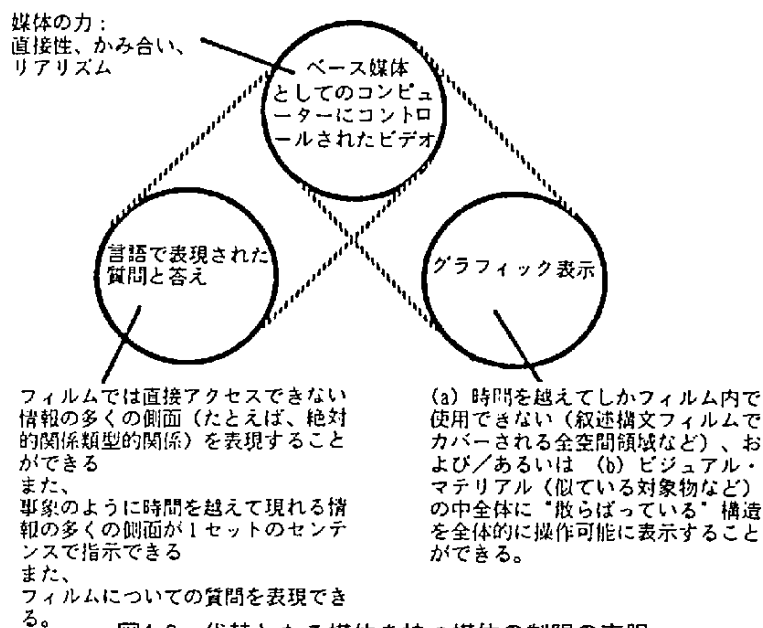


図4-2 画像を指示するための言語の操作



4.2.4. MIMEの設計、および効果についての側面

特定の状況においては、ユーザーにうまく情報を提示できるような媒体でも、ユーザーが要求した情報に対するアクセスを提供するのは、そううまくはいかないかもしれない。(たとえば、従来のオーディオ・カセット・プレーヤーの場合。)媒体には提示スペースとアクセス・スペースの両方がある。媒体の提示スペースは、不特定のユーザーが経験できる媒体の要素がコード化されたものすべてのセットとして、定義することができる。アクセス・スペースは、媒体からの情報に対する要求を提示するためにユーザーが使用できる、コード化されたものすべてのセットである。提示スペース、アクセス・スペース、両者とも、ユーザーが操作でき、あるいは適応させることができるツールを、実績のあるコード化されたものと関連づけている。両者のツールのセットの違いというのは、操作の結果なのである。提示ツールは、すでにユーザーが利用することができる特殊なコード化されたもの、の側面に焦点をあてる手段を提供するものと見なすことができる(たとえば、オーディオ・システムの音量調節の場合など)。しかし、アクセス・ツールの場合は、実行された操作の最終結果、あるいは実際のアクティビティー自体が、媒体からは要求されるが、現在の実績あるコード化されたものにおいては実際に使用可能でないような情報の仕様を表示するのである。

根本的に、MIMEは、1つの媒体の提示スペースが別の媒体のアクセス・スペースとして役立つようにするものである。なぜなら、後者の媒体は、提示という目的については適当であるが、そのコード化されたものは、情報内容の望まれる側面の仕様として使用できるような方法では、直接的には操作できないからである。簡単に言うと、1つの媒体の提示スペースをそれ自身のアクセス・スペースとして使用することは、以下に述べるケースで提唱されたものの、必ずしも可能ではないということである。そのケースとは、“機会均等システム”(RuncimanとThimbleby、1986)や、“相互関係のあるインプットとアウトプット”(Draper、1986)であり、そこでは、システムからのアウトプットが表示するものを、ユーザーがインプットとして再使用できるからである。

1. 全体的、および連続的サーチ・スペース

対話システムの使用に関わる2つの“状態スペース”があるという見方は、新しいものではない。たとえば、Payne (1987、1988) は、彼が呼ぶところの“結合された状態スペース仮説”を提示することで、対話システムの学習可能性を評価する問題に対処している。Payneによると、対話システムのユーザーは、以下の3つについて知的判断を維持しなければならないという。その3つとは、システムで操作することができる外界の可能な状態を表す、「目標スペース」、システムの可能な状態を表す、「装置スペース」、および目標スペースの状態が装置スペースの演算子を適用することでいかに達成できるかを説明する、「両者の間の意味論的マッピング (“つなぎ”）」である。たとえば、1つのパラグラフをカット・ペースト・エディターに置き換えるには、装置スペース演算子で、パラグラフを選択し、新しい位置を示し、選択したパラグラフをそこへ移動させることが必要になる。意味論的マッピングとは、ゆえに、目標を達成する際の、個々の装置演算子の役割についてのユーザーの知的判断なのである。

Payneの分析の1つの特徴は、装置スペースを、演算子が目標ベースのアクティビティに適用されるような状態スペースとして見ていることである。さらに、ベース媒体の情報要素を概念化したものとしてユーザーが操作できるようにするような推論があり、Payneは、そういった推論は、ユーザーにそうした操作を許さない推論よりも学習しやすいと言っている。(Payneによると、カット・ペースト・スクリーン・エディターのほうが、ラインベースのエディターよりずっと学習しやすいのはこのためである。) Payneは、典型的なカット・ペースト・エディターにおいて、ユーザーが利用できる主な操作可能なエンティティは(典型的な複写、および移動型のオペレーションについては)、ストリングであると指摘している。ストリングは、ドキュメントの関連のあるすべての部分を、ユーザーが感知するものとして表示することができる(文字、単語、センテンス、パラグラフ、行でさえも)。しかし、テキスト・エディターでは、我々は実際にベース媒体をインタフェースに持ってくることができるのである。すなわち、ある意味で、アクセス・スペースと提示スペースを同一のものにすることができるのである。前に述べたように、これは、IRインタフェースでは必ずしも可能ではない。

IR インタフェースに関しては、多くの人達がテキストのドキュメントを理解しているのと同程度に、ユーザーがベース媒体の構造的要素を理解していると必ずしも想定することはできない。このため、動的サーチを促進するためには、アクセス（サーチ）スペースを設計する必要がある。アクセス（サーチ）スペースは、既に述べたように、しばしば代替となる媒体内のコード化されたものの使用に関わっている。さらに、IR に関して非常に重要な問題は、どのくらいのアクセス・スペースが、1 回につきユーザーに利用可能であればよいかというものである。サーチ・スペースについての世界的な見解では、IR 用アクセス・スペースの具体化に対する主な 2 つのアプローチが指摘されている。つまり、連続的なものと、全体的なものである。しかし、多くの MIME ベースのシステムでは、2 つを組み合わせたものを使用している。現在導入されている、ハーモニースペースシステムのアクセス・スペースは、全体的であり、アクセス・スペース全体が、1 つのディスプレイとして提示されている。ハイパーテキストシステムでは、連続的アクセス・スペースを表示する傾向にある。ブックハウスシステムは、両者の要素を含んでいる。キー用語を表す複数のアイコンがスクリーン上に現れるが、図書館システムのナビゲーションは、連続するスクリーンで表示されている。（そのそれぞれが、全体的性質をもっており、描写されたシーンの異なった領域が、選択によって、システムの異なった部分へつながっていくのである。）

連続的アクセス・スペース、全体的アクセス・スペースはともに、ユーザーにとって長所と短所がある。全体的アクセス・スペースは、ベース媒体における情報の構造が、ベース媒体自体からは利用することができないようなときに、特に有益である。それゆえに、ハーモニースペースでは、多くの可能な音楽和音のシーケンスから成るベース媒体の提示スペースのコード化されたものが、全体的なグラフィックなのである。全体的アクセス・スペースのもう 1 つの利点は、サーチ環境を常に表示できることである。連続的アクセス・スペースの長所は、サーチ・プロセスのそれぞれの段階で、ベース媒体からより多くの情報を提示することができるということである。ゆえに、

（a）サーチが進んでいく際、その結果を継続的に評価するときにユーザーをサポートすることができ、また、（b）コード化されたものにおける操作と、ベース媒体における結果との関係を、より即座に明らかにすることができるのである。連続的アクセ

ス・スペースの短所は（そのいくつかは、Fossのハイパーテキストシステムの中で論じられており、また“遠ざけられた窓”あるいは“ビジュアル運動量”の問題も先に論じてある）、認知構造図の性質に関連したものである。認知構造図は、“日々の空間環境における現象の相対的位置と属性”（DownsとStea、1973）についての情報をストアしている知的構造と定義されている。連続的アクセス・スペースにおいては、システム内の情報の構造についての認知構造図を構築する負担は、ユーザーが負うことになる。しかし、認知構造図は、“不完全”で“ゆがめられている”（個人の“主観的な幾何学”によって）ものであり、“図式化されている”（すなわち、分類された情報を含んでいる）、“肥大化している”（“飾り”として付け加えられる、おそらく存在しない現象を含んで）ものである（DownsとStea、1973）。さらに、認知構造図がビジュアルなインプットのみから作成されると、ますます不完全な性質を帯びる（DownsとStea、1977）。連続的アクセス・スペースにおけるサーチに関連した問題に対するFossの解決法は、認知構造図の考えられる4つの短所にすべて対処するもので、特筆すべきものである。その方法とは、ユーザー自身のサーチ・アクティビティーの全体的でグラフィックな視野を提供するものである。不完全さ、ゆがみ、誤った肥大化は、完全な視野という外部規定によって対処することができる。外部表示は、指示されたグラフにおいて短くラベル付けされたノードのみが見えている情報を表示する、という意味では図式化されているが、提示スペース要素（すなわち、表示されたカード自体）へアクセスできるよう選択することができるのである。

2. アクセス・スペースにおける計算

“インタフェースの真の問題は、編集、合成、検索に関わるものである。”

（NewellとCard、1985）

計算上、アクセス・スペースを使用した情報の検索は、外部コード化されたものの操作を含んでいる。それは、時には、要求された情報が有するいくつかの特性としてユーザーが指定できるような方法による操作である。これはIRにおいて、常に、アクセス・スペースがテキストの照会言語から構成されているか、あるいは以前に定義され

た何らかの方法でリンクされたページの集合体から構成されているかどうか、といったケースであった。MIMEが、サーチ・スペースの外部化の性質において異なっているように、ユーザーに提供される操作の性質においても異なっていることがある。しかし“操作”とは何を意味するのであろうか？ いくつかのIRシステムでは、操作とは単に、“選択”のことである（ディスプレイの対象物かメニューのオプション上で、マウスをクリックさせて）。もし、メイン・アクセス演算子が選択であり、アクセス・スペースが連続的であるなら、1つのディスプレイから別のディスプレイに進んでいくには、中間ディスプレイを視てみるなど、一連の多くのステップが含まれる（BOOKHOUSEシステムの場合）。しかし、これが悪いことだといっているのではない。Canterらは以下のような発見をした。家の価格と説明についてのページベースのデータベースを使用するユーザーは、あらかじめ定義されているページ間のリンクを使用して、価格の幅等の特徴に基づく情報をナビゲートすることができるのであるが、こういったユーザーは、分析的サーチを使用するユーザーよりも、多くの質問に答えていただけではなく、実際より多くのページにアクセスしていたのである。Canterらによると、これは、リンクされたシステムのユーザーは、データベースをナビゲートする能力に、より自信を持っており、ゆえにより冒険心に富んでいたからであるという。しかし、連続的サーチ・スペースにおける方向づけの問題についての世界的分析では、実際の操作のレベルでは逆の結果もでている。複数の選択のシーケンスを含むサーチは、ユーザーにとっては一回の操作であるかもしれない。ここでも、この知的判断を維持する負担は、ユーザーが負うのである。Buxton (1986) の言葉を言い換えると、ボタンを押すとか、“ポインティング”といった行為は、本質的にサーチ・アクティビティーの原始的な操作である。ユーザーは、サーチ・アクティビティーを、こういった原始的なもののみが構成要素であると概念化してしまうかもしれない。

選択ベースのみのサーチ・インタフェース（および多くのブロージング・タイプのシステム）に欠けているものは、ユーザーが情報のための仕様を合成できるようにするという概念である。もちろん、従来の照会言語ではこれを行うことができる。しかし、会得と使用は困難であると、注意書きがされている。そこで、本質的に照会ベースのシステムの中には、便利な合成用の演算子を使用して、照会システムに“友好的な顔”

を提示しているものもある。ウィンドウを使用するシステムにおける演算子構築をサーチするには、合成的アプローチをとることが多い。たとえば、Godinら、(1989)によって記述されたシステムでは、ユーザーは、1つのウィンドウからキー用語を選択し、それを照会ウィンドウに“置く”ことができる。しかし、このシステムは、動的サーチにも基づいており、選択された照会用語は、最も最近に検索されたドキュメントに関連したものである。CLORISシステムの拡張であるCLANQUEも、照会の公式化に対して合成的アプローチを表している。この場合、自然言語照会の構築は、意味論的に制約付きの編集操作（選択、削除、複写等）を適用して実行することができる。ハーモニースペースシステムでは、合成演算子を使用しており、この場合では、マウスを“引きずる”操作で、和音のシーケンスが演奏されるのである。

3. インタフェース：知られざる量

“検索とはインタフェースである。” (VanRijsbergen, 1989)

前記されているPayneの分析は、彼の考えるシステムにおけるインタフェースが、ベース媒体に関しては比較的明解である、という限りにおいては有益である。ゆえに、我々は、ブックハウスシステムの表面操作の学習しやすさともいえるべきものを評価するのに、Payneの分析を適用することができたのである。しかし、ユーザーが図書館データベースに関して実行されたアクティビティの意味論的判断を提供できたかどうかを決定するには、すなわち、選択したアイコンがどの特定のキー用語を表示したのかについての判断をユーザーが組み立てたかどうかを決定するには、Payneの分析は役に立たないであろう。IR状況において、外部媒体の操作の意味論の学習しやすさが、ベース媒体に関するそういった操作の意味論の学習しやすさを意味するのかが、重要な問題なのである。1つの媒体における制約付きの操作で、別の媒体における結果を照会することができることによって、IRMIMEは情報のための仕様を表示するのである。このことで、Ohlsson (1988) によるコメントが想起される。その趣旨は、ある媒体において、特定のコード化されたもの上で実行された操作が、手続上の正しいステップを示すためだけに制約されているならば、不正確なステップをとる

ことができるような媒体において、ユーザーが表示された手続を実行することができるかどうか尋ねてみるのは妥当であろう、というものである。これは実に有効な観察である（また、特に、HarmonySpaceシステムの場合、適用可能と見ることができるものである）。しかし、IRMIMEに関して、よりいっそう差し迫った問題というのは、ユーザーが、意図したIRアクティビティーを実際に実行しているかどうかを実際に評価することである。インタフェースのみにおけるユーザーのアクティビティーには、“エラー”と指摘されるものはなく、すべての操作が、“意味あること”として制約を受けるからである。

インタフェースでのアクティビティーについての意味論的判断を行うために、IRMIMEのユーザーは、実行された操作（情報の仕様）とその結果（すぐあとに起こるベース媒体のコード化されたもの）との間の関係について、“熟考”してみなければならない。ITS（Intelligent Tutoring Systems、知的チュートリング・システム）社会では、最近、学習の過程での熟考の重要性について多くのことが行われている。学習者が、自分が何を行っているか、あるいは何を行ったかについて熟考するのを奨励し、彼らの、いわゆる“メタ認知”のスキルを開発することが、教授システムの重要なタスクであると見られている（Wenger、1987; Collinsら、1987）。Geometry Tutor（Andersonら、1985）や、AlgebraLand（Brown、1985; CollinsとBrown、1988）のようなITSシステムの中には、学習者のサーチ・パスの検査可能なグラフィック表示を提示しているものもある。しかし、MIMEをIR用に使用して好結果を得ようとする場合、もしユーザーが、操作の意味することはいったい何であるのかを正しく推察しようとするなら、ユーザーが自分の計算上のアクティビティーについて熟考するのは、システムを使用する際の欠くことのできない一部なのである。Norman（1987）が指摘しているように、“インプットとアウトプットの間には、互いに関連し合う距離があり、その距離が、行為がその上で実行される対象物と、アウトプットがその上で知覚される対象物との間の区別を指定するのである”。さらに、この互いに関連し合う距離は、Normanの言葉を借りれば、ユーザーにとって越えることが非常に困難な、“溝”となる場合があるのである。

出典： Manipulable Inter-Medium Encodings for Information Retrieval.

Alan P. Parkes
Department of Computing
University of Lancaster
Lancaster
LA1 4YR

4.3 インテリジェント・マルチメディア情報検索システム

4.3.1. 序

次世代の情報検索技術は、本質的に、分散型マルチメディア・システムによって示される。これらのシステムの主な目的は、複数のデータベースを統合し、エンド・ユーザーによって意識されない効率的なマルチメディア（MM）情報検索サービスをそれらのユーザーに提供することである。

これらのシステムには、オフィス情報システム、CAD／CAMシステム、ならびに医療、軍事、および図書目録の各検索システムなどが含まれる。これらのシステムの主な目的は、テキスト情報、イメージ情報、および音声情報などの異なる媒体の情報を統合して、エンド・ユーザーによって意識されない効率的なマルチメディア情報検索サービスをそれらのユーザーに提供することである。

MM情報管理に対するアプローチは、ハイパーメディア・システムで開発されているが、これらのシステムでは、MMデータは、リンクによって接続されているデータ・ノードから構成されるネットワークに格納される。ノードは、テキスト、イメージ、および音声の各情報を含むことができる。リンクは、通常、現行選択ノードから次の選択のノードのセットを指している「from-to」ノードとして表わされる。これらのシステムでは、ある点において、MM情報検索はナビゲーション・プロセスである。

しかし、MM情報システムの開発で以下のいくつかの課題については検討されていない：

- * 内容探索を含むMM情報への照会をベースにしたアクセス
- * MMと知識ベースのシステムの統合
- * 仮想構造の使用（ビューの概念）

他方、テキスト情報の内容検索の基本的な技法は良く知られている。

イメージ検索の主な問題は、イメージの内容を正確に定義し解釈することの難しさから発生する。大体において、イメージ検索技法は、DBMSまたはイメージの内容がテキスト形式で表わされる情報検索システムに基づいている。他のイメージ検索技法では、イメージが、前もって記述されている特定のドメインに属している必要がある。

テキスト情報とイメージ情報を統合するAMCIRS (A Multimedia Cognitive-based Information Retrieval System - マルチメディア認知ベース情報検索システム) システムは、MMシステム以外のMM情報検索システムへのより知的で高度なアプローチを使用する。AMCIRSに組み込まれている革新部分のいくつかは次の通りである：

* MMと知識ベースのシステムは、認知ネットワーク（以下CNという）の構成によって提供される。一部の知識構造は、IRSと他のシステムで使用されていた。ここでの基本的な目的は、情報検索のプロセスでエンド・ユーザーの振舞いをシミュレートすることである。CNは、MMのオブジェクトの一定のドメインに関する情報を統合するMM構造になっている。このネットワークは、イメージ内容検索のための情報、ユーザー照会の拡張機能と関連情報、照会言語へのサポート、およびユーザー照会改善技法を提供する。CN情報の選択は、ユーザー照会がイニシエートする活性化拡散により行なわれる。

* ここでの照会メカニズムは、ベクトル・モデルを使用するMMオブジェクト内容探索に基づいている。イメージ・オブジェクトの記述を導入することにより、MM DBオブジェクトの内容探索をサポートするモデルが開発されている。ここでは、フラクタル・ジオメトリを使用して、自動イメージ・インデクシングが提供されている。このように、照会ベクトルとインデックス・ベクトルは、フラクタル・オブジェクトとジオメトリック・オブジェクトで拡大し、イメージ内容検索は、フラクタル・オブジェクトとジオメトリック・オブジェクト間の類似性評価に演繹される。

4.3.2. グローバル・マルチメディア・システム・モデル

グローバルMMシステム・モデルは、図4-4に示している。ユーザー照会は、AMCIRS COGNITIVE SUBSYSTEMにより実施される認知ベースのプロセスへの入力を示している。このサブシステムは、CNからの情報を使用し操作しながら、ユーザー・オブジェクトとMM DBオブジェクトとの通信を可能にするインタフェースとして導入される。MM DBオブジェクトには、インデックス・ベクトルの索引が付いている。各インデックス・ベクトルは、関連重み係数を持つテキストとイメージのオブジェクトから構成されている。

これらの重み係数は、MM DB内容のインデックス・ベクトル・オブジェクトの重要度を示している。

他方、ユーザー照会は、CN情報を使って照会ベクトルへ変換される。照会ベクトルは、テキスト・オブジェクト、イメージ・オブジェクト、および（エンド・ユーザーの照会ベクトル・オブジェクトの重要度を示している）重み係数も含んでいる。

検索のプロセスにおいて、対応するインデックスと照会データ間の類似性がしきい値を超える場合、一定のMM DBオブジェクトが選択される。

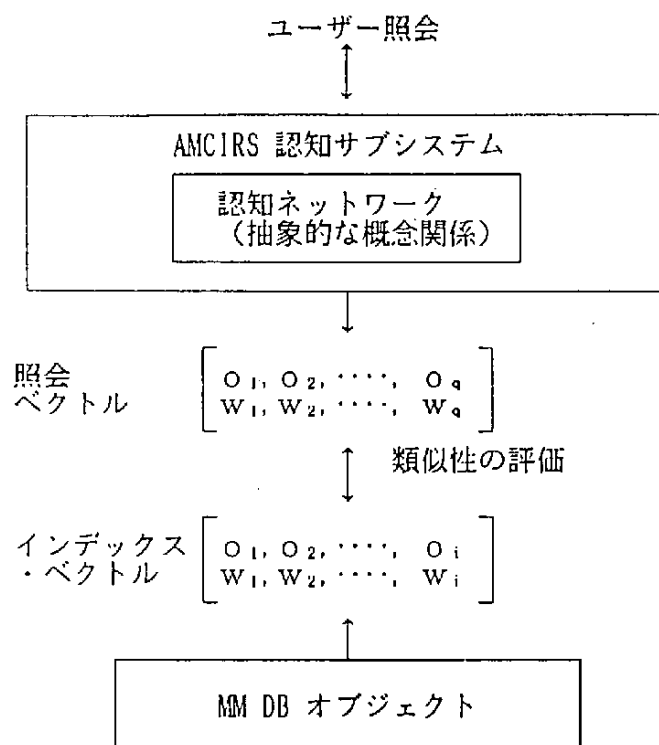


図4-4 グローバルAMCIRSモデル

MM DBオブジェクトのインデクシングのプロセスについて説明しよう。自動テキスト・インデクシングは良く知られている技法である。

すでに、フラクタル・ジオメトリから、イメージ・オブジェクトの記述に効率的な方法が得られることがわかっている。

イメージ・セグメンテーションのプロセスを使って、イメージが、基本的な2つのディメンショナル・エリアと曲線とに分割される。これらのエリアは、ジオメトリック・オブジェクトまたはフラクタル・オブジェクトとして分類される。フラクタル・オブジェクトごとに、反復機能システム (IFS) が計算される。ジオメトリック・オブジェクトの場合、分析的關係と特性点が決められる。任意の開曲線がピクセルの集まりとして表わされる。イメージは、フラクタル・オブジェクトとジオメトリック・オブジェクトの集まりである。

このシステムにおけるイメージ分解のプロセスは、図4-5に示している。

イメージ分解プロセスの主要な部分は、エリアのIFSの決定である。このシステムの場合、それは自動的に行われる。ここで用いているアルゴリズムは、許容可能なイメージ再生を可能にするいくつかの高速ヒューリスティック方式に基づいている。それは、同じ座標系の始まりですべてのエリアを変換し、同じフォームの要素を持つエリアを扱い、対応するIFCを算出する。

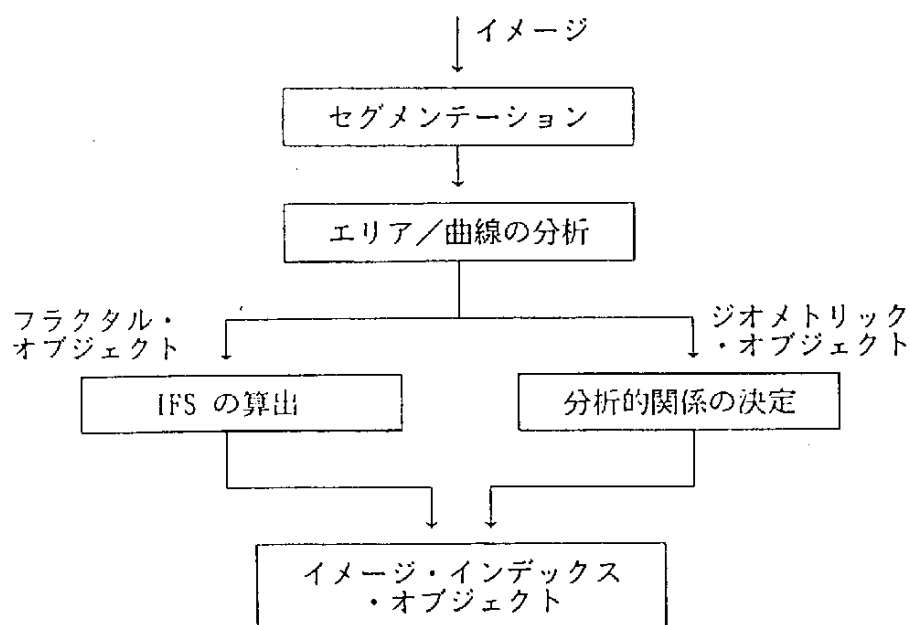


図4-5 イメージ分解のプロセス

たとえば、車のイメージは図4-6に示している。セグメンテーションにより、それは2つのフラクタル・オブジェクト（f01とf02）と3つのジオメトリック・オブジェクト（g01、g02、およびg03）に分割される。

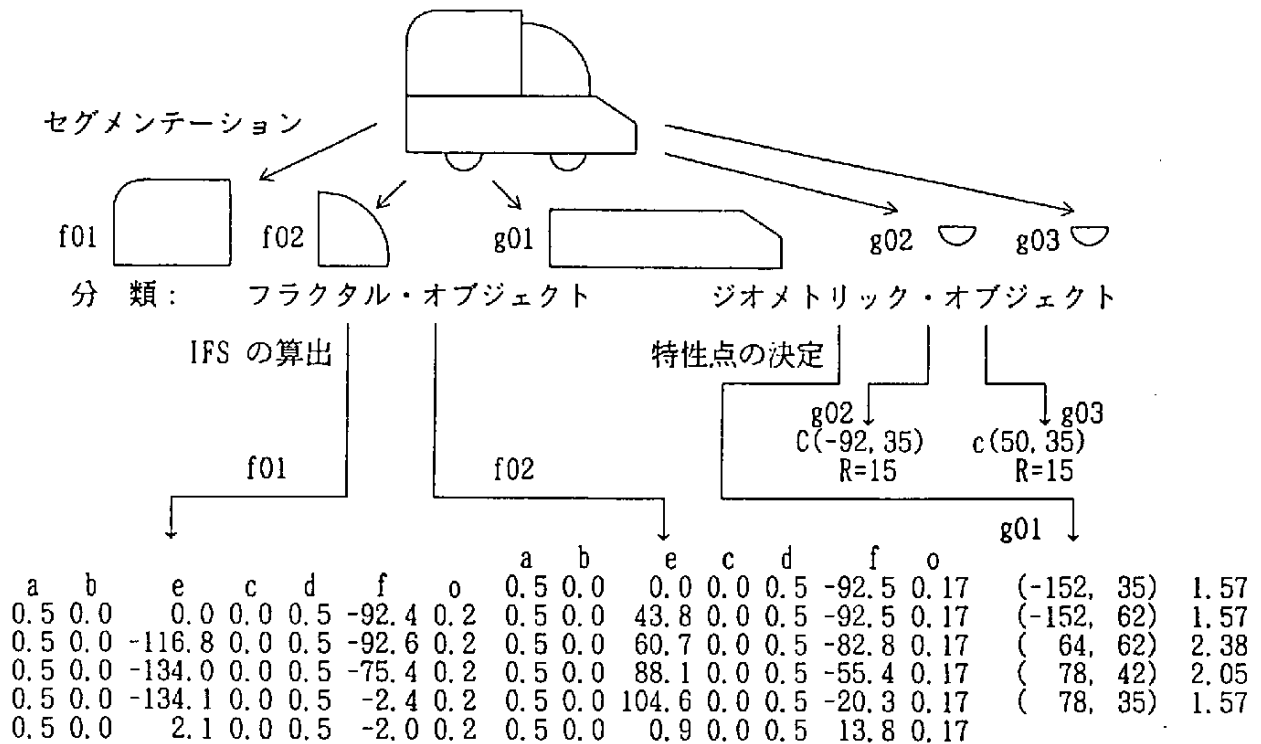


図4-6 イメージ分解のプロセスの例

IFSの算出方法を使って、各フラクタル・オブジェクトは数字のセットとして示される。数字a、b、c、d、e、およびfは、オブジェクトの再構成に使用される関数の係数である。この例では、フラクタル・オブジェクトは5つまたは6つの異なる関数で記述される。数字pは、オブジェクトの構成を首尾良く行なうのに必要な確率である。

g 01は、頂点と関連角度で表わされる多角形である。これらの角度は、多角形エリア間の類似性評価のプロセスで使用される。

ジオメトリック・オブジェクトとして、半円は、その中心と半径で記述される（g 02とg 03）。

イメージは、いったん分解されると、MM DBまたはCNに格納される。分解の方法には多くの時間を要するが、その分解はイメージに対してだけ実施される。他方、イメージ再生のプロセスはかなり速く実施される。

4.3.3. 認知システムにおける情報の表現

認知ネットワーク（CN）は、現実の世界のオブジェクト（概念）とそれらのオブジェクト間の結合に関する若干の考えを実現したものである。実際には、情報検索のプロセスにおけるユーザーの振舞いをシミュレートしようとしている。

CNは、指示したグラフで表わすことのできる知識ベースの構造になっている。グラフのノードは、概念と概念間の弧の関係を示している。

ノードは以下の2つのタイプの情報を含んでいる：

- * テキストの具体化と言われる、概念の可能なテキスト表現。それらの表現は、概念が実際のテキストで生成するいろいろな仕方を示しているテキスト形式を含んでいる。

- * イメージの具体化と言われる、概念の可能なイメージ表現。それらの表現は、概念がイメージで認識できるいろいろな仕方を示しているイメージ形式を含んでいる。これらの形式は、以前に説明したフラクタルな記述とジオメトリックな記述を通して表わされる。

ノード間には次のような2つの関係が存在する：

- * 「IS-A」と「PART-OF」の階層を含んでいる階層指向の関係。

- * 概念間の関係の強さを示している関連性指向の関係。

CNのグラフには、階層や関連を示す弧サイクルはない。

同様に、AIアプリケーションの場合のように、具体的な概念や抽象的な概念に基づいて概念を分類する。このアプローチの場合、各概念ではテキストの具体化が生成するが、具体的な概念の場合にだけイメージの具体化が生成する。

図4-7は、簡単なCNフラグメントを示している。

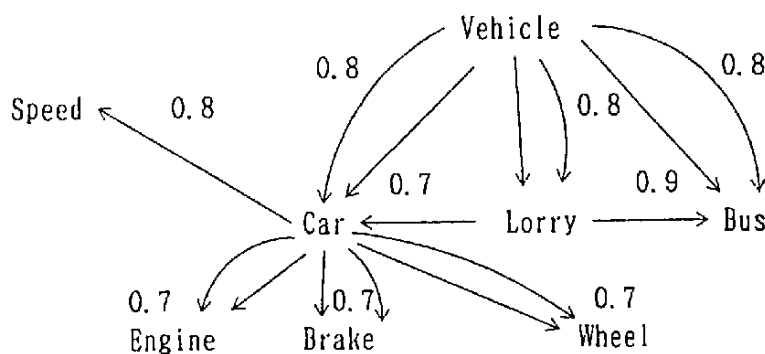


図4-7 簡単なCNフラグメント

関連番号のついている有向弧は、概念間の関連性指向の関係を示している。関連性の度合（DR）と呼ばれるこれらの番号のインターバルは（0、1）となっている。

階層の関係は、番号のついていない有向弧（HR弧）で示される。

たとえば、概念SpeedとCar間の関連性の度合は0.8、LorryとBus間の関連性の度合は0.9である。

概念間の階層の依存関係を表わす場合には「<」と「>」を使用する。たとえば、Lorry
 <Vehicle、Car> Motor というふうに。

CN情報は、以下の4つのデータベースに格納される：テキスト具体化DB (TMDB)、
 イメージ具体化DB (IMDB)、関連性度合DB (DRDB)、および階層DB (HRDB)。こ
 れらのDBは、CN概念に関する最も重要な情報を含んでいる。

図4-8は、概念Carに関するCN情報を含んでいるTMDB、IMDB、DRDB、およびHRDB
 の内容を示している。

TMDBのフラグメントは、関連具体化関連性値を持っている概念「Car」の3つの具
 体化を示している。

Car	...	
Car	Car	1.00
Car	Automobile	1.00
Car	Motor-car	0.90
	...	

TMDB のフラグメント

Car	...	
Car	Speed	0.80
Car	Engine	0.70
Car	Brake	0.70
Car	Wheel	0.70
	...	

DRDB のフラグメント

Car	...	
Car	Vehicle	0.80
Engine	Car	0.70
Brake	Car	0.70
Wheel	Car	0.70
	...	

HRDB のフラグメント

Car

フラクタル・オブジェクト :

CNf01 :

a	b	e	c	d	f	o
0.5	0.0	-102.2	0.0	0.5	-0.08	0.17
0.5	0.0	-128.8	0.0	0.5	11.3	0.17
0.5	0.0	-138.8	0.0	0.5	38.7	0.17
0.5	0.0	133.6	0.0	0.5	24.8	0.17
0.5	0.0	124.0	0.0	0.5	8.2	0.17
0.5	0.0	110.3	0.0	0.5	3.2	0.17

CNf02 : 図 4 の f01 に同じ。

ジオメトリック・オブジェクト

CNg01 : polygon

(-120, 40)	1.40
(-112, 75)	1.65
(110, 75)	1.57
(110, 40)	1.57

CNg02 : 半円

C(-80, 40) ; R=30.0

CNg03 : 半円

C(40, 40) ; R=30.0

関連性 : 1.00

IMDB のフラグメント

図4-8 CNデータベースのフラグメント

所望の関連性を持つ概念を表わしている記述事項が複数ある場合がある。たとえば、最初のイメージの具体化は、具体化関連性1.00を持つ概念Carをグラフで示す。同じ概念は、具体化関連性0.90でも示される。なぜなら、Carというそのグラフ表示はバックサイドからのものとすることができるからである。

DRDBとHRDBのフラグメントは、概念Carに関連するグラフ情報を含んでいる。

CNグラフ情報を使用しながら、CN概念のセットに関するいくつかの関係を紹介しよう。

階層指向の関係は、以下によりCN概念のセットに関して定義された関係Hrである：もし概念Cpから概念Cqへの階層弧の経路がある場合、 $(Cp, Cq) \in Hr$

階層指向の関係は、半順序関係である。

関連性指向の関係は、以下によりCN概念のセットに関して定義された関係Rrである。

もし概念Cpから概念Cqへのあるいは概念Cqから概念Cpへの関連性弧の経路がある場合、 $(Cp, Cq) \in Rr$

たとえば（図4-7を参照）、 $(Vehicle, Brake)$ 、 $(Wheel, Vehicle) \in Rr$ 、しかし、 $(Brake, Speed)$ 、 $(Speed, Engine) \in Rr$ 。

最後の定義では、以下の関数により複数の概念CN間の関連性の度合（関連の強さ）が定義される：

$$DRel(Cp, Cq) = \begin{cases} \frac{\sum_{i=1}^k V_{pq}^i (2 - V_{pq}^i)}{k + 1 - \prod_{i=1}^k V_{pq}^i}, & V_{pq}^i = \prod_{DRs \in V_{path}^i_{pq}} DRs \quad \text{ただし、} \\ & (Cp, Cq) \in Rr \quad \text{の場合} \\ 1 & \text{ただし、} p=q \text{ の場合} \\ 0 & \text{ただし、} (Cp, Cq) \notin Rr \text{ の場合} \end{cases}$$

ここで、 k は、ノード C_p とノード C_q 間の関連性弧の異なる経路の数である。

$V_{path_{pq}^i}$ は、ノード C_p と C_q 間の経路 i 上の弧と関連する DR のセットである。

注：両方向への 2 つのノード間の経路は存在することができない。なぜなら、その存在は我々の CN グラフの定義と矛盾するからである（CN グラフには関連性の弧サイクルがない）。

この関数は重要な特徴を持っている。その値は、2 つのノード間の既存の経路すべてならびにそのような経路の数により異なる。

例：

$$DRel ("Vehicle", "Speed") = \frac{0.8 * 0.8 (2 - 0.8 * 0.8) + 0.8 * 0.9 * 0.8 (2 - 0.8 * 0.9 * 0.8)}{3 - (0.8 * 0.8) (0.8 * 0.9 * 0.8)} = 0.64,$$

$$DRel ("Speed", "Engine") = DRel ("Speed", "Wheel") = 0, \text{ 等。}$$

上記の関数から、システムの重要な特徴が得られる。各新しいユーザーは、新しい概念を所有し、CN にリンクし、システムによる現実の世界に関する認識を（CN の世界観により）深める。

4.3.4. 操 作

AMCIRS システムは、MM 情報への独創的な照会ベース・アクセスを提供する。

内容探索プロセスは、以下のユーザー・フィルタ指定により起動される：

〈filter specification〉 →

〈compare operator〉 1 “list of terms” 〈compare operator〉 2 〈rel〉

ここで、

“list of terms” は、ユーザーの関心事の主語を示すユーザー指定の語、句、または文のリストである。

〈compare operator〉 1 は、階層の点からユーザーの関心事の方向を指示し、標準的な比較演算子のうちの1つとすることができる。

〈compare operator〉 2 は、関連性の点からユーザー関心事の方向を指示する標準的な比較演算子のうちの1つである。

〈rel〉 は、ユーザーが指定する関連性値を示す。

フィルタ指定は、照会ベクトル生成のプロセスを起動する。このプロセスは、CNノードで実施される活性化拡散 [26] 手順から始まる。我々の活性化拡散手順の主なステップと制約は次の通りである：

- * 開始点は、フィルタ指定におけるユーザー項のうちの一部で識別されるCNノードである。実際、CNノードが活性化拡散手順の開始点になるのは、ユーザー指定の項がそのCNノードのテキスト具体化で検出できる場合である。

- * 単一のDRまたはHRリンクにより開始点へ接続されるCNノードすべては、次に起動されるノードである。DRリンクの場合、任意の方向での開始点への経路が存在しなければならない。

+ * CNノードの起動上の重みは以下の値とすることができる：

1) Drel関数により算出された値（セクション3を参照）－ノードの起動がDRリンクを介して行なわれたとき。

2) 1-ノードの起動がHRリンクを介して行なわれたとき。

* DRリンクの場合、プロセスは、Drel関数値が〈rel〉値を超えるまで繰り返される。HRリンクの場合、プロセスは、HRリンクを使用するCNノードすべてが走査されたときに終了する。

* 活性化拡散手順から、DRリンクまたはHRリンクによって起動されたノードの交点で決められた起動CNノードのセットEcn（基本CNセット）が得られる。CNノードが両方のタイプのリンクにより起動された場合、その起動重み値は両方の重み値の小さいほうの値となる。

ユーザーが指定した項としてCarを使用して行なったさまざまなフィルタ指定と図4-7に関係する対応基本CNセットを下に示す。

フィルタ指定	CNサブセット
(<= "Car")	{Car, Engine, Brake, Wheel}
("Car"> 0.7)	{Car, Speed, Vehicle}
(> = "Car"> =0.8	{Car, Vehicle}
(<"Car"> 0.8	0
("Car")	{Car}
(= "Car" 0.7)	{Car, Speed}

基本CNセットは、照会ベクトル生成のプロセスで重要な役割を演じる。実際、照会ベクトルは、このセットのCNノードの概念具体化により生成される。

照会ベクトル重み係数は、CNノードの起動重み値と具体化関連性値との積として算出

される。

生成されたMM照会ベクトルは、MM DBインデックス・ベクトルと比較される。

MM DBオブジェクトの選択は、以下の値に従って実施される：

$$a \sum_i \sum_j W_q(T_i) W_i(T_j) \text{Cort}(T_i, T_j) + b \sum_i \sum_j W_q(I_i) W_i(I_j) \text{Cori}(T_i, T_j)$$

ここで、 a と b は適正な定数である。 T_i 、 T_j 、 I_i 、および I_j は、テキストとイメージのベクトル・オブジェクトである。 W_q と W_j は、それぞれ、照会とインデックスとのベクトル・オブジェクトに関連する重み係数である。最初の加法は、すべてのテキスト・ベクトル・オブジェクトに対して行なわれる。2番目の加法は、すべてのイメージ・ベクトル・オブジェクトに対して行なわれる。

相関関数 $\text{Cort}(\quad)$ と $\text{Cori}(\quad)$ は以下によって定義される：

$T_i = T_j$ の場合に1

$\text{Cort}(T_i, T_j) =$ 持っていない場合に0

関数 $\text{sim}(I_i, I_k)$ は、 $k = j$ の場合に

$\text{Cori}(T_i, T_j) =$ 最大値(tr よりも大きな値)を持っている場合に1

持っていない場合に0

ここで、 tr は類似性のあるレベルを示すしきい値である。

$\text{sim}(\quad)$ は、フラクタル・オブジェクトのIFSフラクタル記述の情報内容に基づく類似性関数である。ジオメトリック・オブジェクトの場合、 $\text{sim}(\quad)$ は特性点と関連角度に基づいている。

4.3.5. 実現上の注意

AMCIRSの実現は完了していなが、現在研究中のAMCIRSプロトタイプはすべてCで書かれており、386IBM/PC上で動作する。

ここでは、インデックス・ベクトルのテキスト部分を作成する際に単一項インデクシング方式 [27] を用いている。最も頻繁に現われる語（ストップ・ワード）や語の接頭辞と接尾辞は排除される。インデックス項の重み係数は以下の式から算出される：

$$w_i = \frac{Tf_i \times Idf_i}{\sum_{\text{vector}} (Tf_i \times Idf_i)^2}$$

ここで：

Tfは、一定のMM文書における項のオカレンスの回数の基準である項出現回数係数である。

Idfは、 $\log(N/n_i)$ として算出される逆項オカレンス回数係数で、NはMM文書の総数、 n_i はiが出現した文書の数である。

現行AMCIRSバージョンは、ジオメトリック・イメージを含んでいるMM文書とともに動作する。これらのイメージは、多角形や円のエリアから構成される二次元イメージを含んでいる。Hough方式 [12] は、多角形の線を検出するのに使用された。さらに、ここでのアルゴリズムは、突き止められた頂点に対応する線と角度との間の交点を突き止める。これらの要素は、インデックス・ベクトルのイメージ部分に含められた。照会言語の実現は、フィルタ指定に基づいている。フィルタ指定のテキスト部分は、項のセットとしてあるいは自然言語で入力できる。簡単なパーサ（構文解析ルーチン）は、概念テキスト具体化としてCNノードに存在するフィルタ指定項をチェックする。CN処理と照会ベクトル生成は、ネットワーク構造に適用される再帰手順により実現される。

テキストとジオメトリック・イメージの類似性関数は、4.3.5で示している式で実現される。非ジオメトリック・エリアのフラクタル記述のインデックスと照会のベクトル要素が確認されているが、この場合の類似性評価の問題は依然として考慮中である。

最初の経験では、AMCIRSは、MM情報検索を首尾良く行えるようにするためには厳密に定義されたCN概念具体化を必要とすることがわかっている。システムが関連情報を検索するプロセスで成功をおさめているかどうかはrecallパラメータとprecisionパラメータで判断できる [25]。recallパラメータとprecisionパラメータは以下の2つのケースで比較されている：

- (1) 照会とインデックスのベクトルはテキスト部分のみから構成されていた。
- (2) 照会とインデックスのベクトルはテキスト部分とイメージ部分とから構成されていた。

最初のケースの場合、科学文書全文からいくつかの部分が抜粋されている。2番目のケースの場合、その同じ抜粋部分に若干のジオメトリック・イメージを追加して拡大されている。

この実験で、2番目のケースにおいて、recallパラメータの値はほぼ常に最初のケースの場合よりも大きく、precisionパラメータの値は両方のケースで同じであることが明らかになっている。

出典：

AN INTELLIGENT MULTIMEDIA INFORMATION
HANDLING SYSTEM

D. Davcev and D. Cakmakov

Faculty of Electrical Eng. & Computer Science
University of Skopje - Yugoslavia
Karpos II, bb, 91000 Skopje

Ⅲ. 今後の課題

1. 技術の進展と応用の拡大についての検討

知的ハイパーメディアあるいはマルチメディアは、今後ますます進展し、様々なプロダクトが紹介されよう。特に、こうした機能を標準装備したPCとかパソコンOS、マルチメディア対応AV機器等については、応用範囲が広いだけに引き続き調査研究を実施していく必要がある。

2. 知的ハイパーメディアに関する技術的課題の検討

複数の利用者が、一つの接点（オブジェクト）を共有して操作する場合のオーバーヘッドの解消、また複雑な大量のデータ構造を高速にアクセスする機構の必要性の増大等、技術的な課題についてさらに検討を加える必要がある。

3. 標準化、知的財産権等の問題

知的ハイパーメディアの進展につれて、様々なメディアの複合利用に関し標準化の対応が急がれる。また、個人によるハイパーメディアの利用が、今後、急速に進むため、著作権等を含む知的財産権に関する法制度の整備等が必要となる。

4. 普及啓蒙

新しい技術については、技術的な論文の紹介あるいは解説等が行われるものの、その利用の効果を分かりやすく解説するなどの努力がなおざりにされがちである。したがって、知的ハイパーメディアの普及啓蒙を図る必要がある。

—— 禁 無 断 転 載 ——

平成 4 年 3 月 発行

発 行 財団法人 データベース振興センター
東京都港区浜松町二丁目 4 番 1 号
世界貿易センタービル 7 階
TEL 03(3459)8581

委託先 株式会社 新世代システムセンター
東京都新宿区富久町 11-5-402
TEL 03(3355)5184

印 刷 株式会社 エーヴィスシステムズ
東京都文京区本郷 2-39-5
TEL 03(3816)4111



