

フレーム理論に基く知識型システム

上野 晴樹 (東京電機大・理工)

1. はじめに

知識型システム (knowledge-based system) は、知識工学の概念に基く応用指向の人工知能システムである。ソフトウェアの構造としてみると、知識ベース (knowledge base) と推論機構 (inference engine) とから成り立っており、この両者が互いに独立なモジュールを構成している。更に、知識ベースは、知識の定義、保存とともに容易に追加・変更のできる機能をもっている。従って、一つの推論機構を用いて、知識ベース内の知識を入れ換えることにより、色々な問題解決システムを作る事が可能となる。しかし実際にはどのような問題解決にも向くという汎用システムは未だ存在しない。比較的汎用性をねらった、いわゆる、知識型システム設計ツールと、特定の応用問題に対して特に強力な能力を持たせることに重点を置いた応用指向型ツールとに分かれる。どっちがより強力であるかということよりも、出来上がったシステムをどんな目的で使うのかという点が重要である。始め応用指向型ツールとして開発してから汎用化の改良を行ったもの (MYCIN → EMYCIN の例など) を、汎用型ツールとして最初から設計されたもの (AGEXPERT など) があるが、いずれも応用指向型での経験に基いたものであり、それをシッポとして残している様である。

ここで知識型システムの基本的特徴は、“システムの能力は推論メカニズムによってではなく、システムが持っている知識の豊富さによって決定される” という考え方である。“知識の中にこそ能力が眠っている (In the knowledge is the power (1))” という考え方をとっている。その問題に特有の知識 (domain specific knowledge) が問題解決のキとなる。従って、知識とは何か、推論の中で知識のはたす役割りは何であるか、エキスパートからどうやって専門知識を引き出すかという問題に加えて、個々の知識をどのような形式で表現するか、これを知識ベースとして如何に管理するか、推論の過程でどのように知識を選択し利用するか、これをどのようにして使い易くかつ効率の良いシステムとして実現するかといった問題が、知識工学としての技術的課題となる。知識型システムはコンピュータ・ソフトウェアとして実現されるはじめて意味をもつからである。この意味で、知識工学は実証的学問であると言える。

さて、従来のAI研究では、汎用で強力な推論メカニズムの研究に大部分の努力がはさわれていたのに対し、知識工学においては、柔軟性に富んだ知識の表現と取り扱い法に重点が置かれている。これまでの研究の経験からプロダクション・システムが最も基本的な知識表現の一つであることが知られている。しかしながらプロダクション・ルールだけで知識全体を表現すると、ルールの数が増加するにつれて、それらの相互関係や全体の構造がつかみにくくなり、推論に力をも一とありで柔軟性にとぼしいという欠点があり、これを解決するためにモデル形成に色々な工夫がなされた。ただし単なる技術的工夫ではなく、人間の記憶や推論の特性をより深く研究して、その本質的なものを構造化して組み込むという種類の工夫である。グラフボード・モデル、フレーム・モデルおよびセマンティック・ネットワークが代表的な汎用性の高いモデル化の工夫である

と思われ。ここに階層モデルを加えてもよい。特に最初の2つのモデルは知識工学の最近の重要な成果と言えよう。特にフレーム・モデルは人間の知識の構造をフレームと呼ばれるデータ構造で統一的に表現しようという試みである。これはここでのメインテーマであり、Minskyのフレーム理論に基いた工夫である。これは推論よりも知識の表現法に重点があり、プロダクション・システムとともに、最も重要な知識工学的技法であると考えられる。これは最近の米国における新しく研究開発された知識型システムのレポートに目を通してみると、その多くがフレーム理論の何らかの応用であったり、あるいはこれから強い影響を受けていることが明らかになる様があるという事実からも言える。今後により強力な理論や技法が提案されるであろうが、フレーム理論は永く残る基本的かつ重要な概念であると思う。

ここでは、Minskyのフレーム理論の概要を極く簡明に紹介し、この理論を何らかの形で応用しているいくつかのフレーム型システム(Frame-based system)をサーベイする。但し、フレーム・モデルを明確に意識して設計されたものはないが、フレーム構造として見えるシステム例(INTERLISPなど)も含まれている。特に、なぜフレーム構造を採用したのかということや、モデルとしてどの様に具体化しているかという点に焦点を絞り、細かい技術的側面にはふれない。原著論文の中のフレームの例示を転写してあるから、比較的具体的なイメージで比較ができると思う。この比較を通して、それぞれの研究者がどのような考え方をしているかという点が多少とも伝われば、このレポートの目的は十分に果たされるが、情報源の不足と考察の不十分さおよび紙面不足により目的が達せられないかどうかわからないが、少なくとも読者の概念形成のトリガーにはなれば幸である。なお、UNITSについては著者が米国滞在中に直接試用したので、簡単な例題COJETを通して、具体的に感触をお伝えしたいと思う。

2. Minskyのフレーム理論の概要

フレーム理論はH. Minskyによって1974年に与えられた(2)。これは、それ以前に提案された概念である彼自身による"microworlds"やNewellの"problem spaces"などのアイデアを基に、一つの統一的理論で体系化したものである。議論が抽象的であること、具体的なデータ構造を示していないこと、色々異なった応用分野について考察していること、人間の心理とのかかわりに重点を置いていること、などにより、難解の論文に属する。彼自身も"この論文は回答よりも多くの疑問を生み出している"と論文の頭で述べている。それにも関わらずこの理論が注目されている理由は、フレーム構造が極めて一般性が高くかつ有用な知識表現の手段となり得ると評価されているからである。多くのこの分野の研究者が"目からウロコが落ちた"という感じをもったのではないかと。なお、この理論を解り易く解説し直したWinston著の本があり、日本語訳も出ている(3)。先ず最も重要なフレームの概念記述の部分をMinskyの論文から引用しよう。

この理論のエッセンスは次の様なことである：人は新しい場面に遭遇したとき(あるいは現在の問題に対する見方を根本的に変えようとするとき)、フレーム(frame)と呼ばれる基本構造を彼の記憶の中から一つ選り出す。これは過去に記憶された一つの枠組み(framework)であり、詳細は必要に応じて現実の問題に

合う様に変えらるようになつてゐる。一つのフレームは、ある種の部屋の中に居る状態とか、子供の誕生日パーティへ出かけるところ、といった様な、型にはまった場面 (stereotyped situation) を表現するためのデータ構造である。各フレームには各種の情報が付加されてゐる。それは、そのフレームの使用に關する情報、次に何が起りようかに關する情報、この予測がはずれたときにとるべき行動に關する情報、などである。一つのフレームは、いくつかのノードとリレーションで構成された一つのネットワークと考えることができる。一つのフレームのトップレベルは固定されてゐる、対象としてゐる場面について常に真である物事を表してゐる。それ以下のレベルには多くのターミナルスロット (slot) をもっており、これはある特定の場面とかデータで必ず埋められなければならない。各スロットには、値の対応づけをするとき満足されるべき条件が定義されてゐる。(対応づけされるもの自身も普通はより小さな部分フレーム (sub-frames) となつてゐる) 単純な条件はマーカーによつて指定され、ターミナルに対応づけられるものも人間であることとか、十分な値をもつたものとか、特定の型の部分フレームを指すポイントであること、という様な要求である。複雑な条件とは、いくつかのターミナルに対応づけられる物事の間の關係を指定する条件である。關係のあるフレームの集まりは、互いにリンクされてフレーム・システムを形成する。重要な行為の結果は、一つのシステム内のフレーム間の変換 (transformations) として現われる。これは、ある種の計算を經濟的に行うのに用いられるたり、強調や注目する点を變えるのに用いられるたり、イメージの効果を計るために用いられる。

視覚の風景解析に於いては、一つのシステムの異つたフレームは、異つた視点からの風景を記述し、一つのフレームから他のフレームへの変換は、場所から場所への移動の効果を表わす。視覚的でない問題におけるフレームにおいては、一つのシステム内のフレーム間の差異は、行為、原因-結果の關係、あるいは概念的視点の変化などを表わす。一つのシステムで異つたフレームが同一のターミナルを共有することができると、これは重要な点であり、異つた視点から集められた情報を統合することが可能となる。

この理論の現象学的能力は、主として予期や他の仮定を含んでよいところから来てゐる。一つのフレームのいくつかのターミナルには、普通、すでにデフォルトで埋められてゐる。従つて、ある場面に関する細かまじした情報を与えなくても、そのフレームは非常に多くの詳細な情報を持つてゐるのである。この方法は、一般的に情報、もっとも似てゐる事例、ロジック的やり方をさける技法、おおよそ有用な一般化を実現する手段、などの表現に広く応用できる。デフォルトはターミナルに緩く付けてあるので、当面する状況により良く合致する様に新しい内容で容易に置き換えることができる。従つて、デフォルトはまた、変数として、あるいは事例による推論の特別なケースとして、あるいは典型的ケースとして有用なものであり、また論理限定子の使用を不用とするためにもよく用いられる。

フレーム・システムは情報検索ネットワークと連結されてゐる。候補となつたフレームが現実の内容に適合できないとき—つまりターミナルで対応づけがマーカーの条件にうまく合致せられないとき—このネットワークが代りのフレームを与える。このフレーム内構造によつても、事実、類似性、おおよそ理解に有用なと

の他の情報 についてこの知識を表現することが可能となる。あるフレームがある状況を表すものとして選ばれ、マッチング・プロセスがそれぞれの場所のマーカ-の条件に合うように各フレームの全てのターミナルに値を代入しようと試みる。このマッチング・プロセスは元のフレーム（これは奮然物せをどう処理するかに関する情報を持つている）に付けられている情報によって、さらにシステムの次の時点におけるゴールによって、制御される。マッチング・プロセスが成功しなかったときここで得られた情報を利用することが重要である。」(2)

人ははじめての土地を旅しているとき、いきなり以前に見たことのある風景である様な印象を受けることが少ない。また解くべき問題が与えられたとき、過去の経験の中から類似のパターンを思い出すことが解決の糸口となる。これは記憶の中から引き出されたフレームであると考えてよいであろう。したがって、フレーム理論は極めて一般性の高い問題を取り扱っていることが理解できよう。

Minskyの論文の中から、立方体の風景分析の例によるフレーム・システムを示

よう。立方体を見たときのイメージは図1に示す様なものである。部分構造AとBは2つの側面を示している。またネットワークから外部へ出ている矢印は、それぞれのターミナルを意味する。図から明らかな様に、フレーム構造は一種のセマンティック・ネットワークである。ここでこの立方体を右へまわると、Aが見えなくなつて代わりにCが現れる。これを図2に示す。

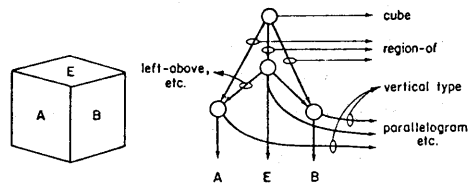


図1 立方体を表すフレーム・システム

図3には、45度ずつ左右に回転させたときのフレーム・システムの関係を示す。各フレームは立方体の異つた視景(perspective)を表していることがわかる。ここで風景分析処理について考えてみよう。立方体を動かす度に全てをはじめから計算し直すのではなく、現れたり消えたりする部分構造に関する知識をうまく管理すれば、経済的に計算できる。(人の場合については)次の様に考えることができよう。フレーム・システムに関する十分な量の情報が長期記憶(long term memory)の中に蓄積されてあり、視界中の風景に合

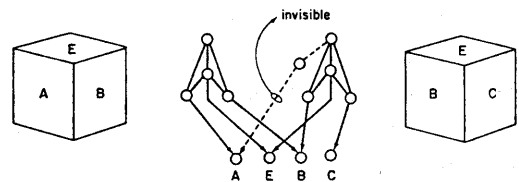


図2 右回転とフレーム・システムの変化

うなものが一だけ取り出されて評価される。もしこれがうまく合致すれば成功となる。うまく合致せず、しかもこれが極めて重要

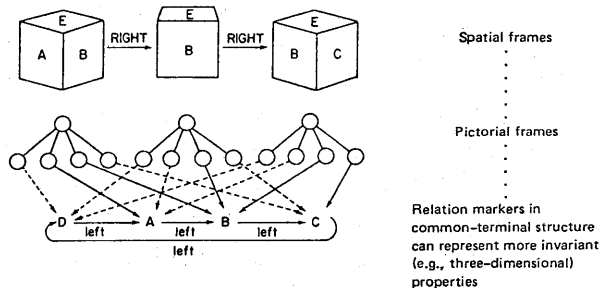


図3 立方体の左右への回転とフレーム・システムの変化

であるにもかかわらず、他のフレームのいづれでもうまくいかなるときは、これまで知らなかったことによる景色の調整をほとんどして、今後利用するために長期記憶の

中に保存することになる。もしこの様が手続きを全ての対象問題に対して実行すれば、大変な量の記憶を必要とする。その代わりに、重要な対象についての特別なシステムをもち、加えて、汎用の“基本形”(basic shapes)となる色々なフレームをもっており、新しい場面に出合ったときには、これらを用いてフレームが構成される。なお、異ったフレームは異った風景を表し、フレーム間のポインタは視点を移動する行為に対応する。各フレームは副フレームへのポインタが付いたターミナルをもつ。視点が変っても物理的に同一の特性を表す場合には、いくつかのフレームが同一のターミナルを共有できる。これは“視点に独立な”情報である。

図4は文献さより引用した、橋の構造を表すフレーム・システムの具体的な表現の例である。フレーム・システム、フレーム、スロット、ターミナルおよびデフォルトがどんな関係にあるかが解り易く示されている。

さて、フレーム型のシステムを実現するときの重要な問題を以下にあげる：

- 予想 (expectation) : どのフレームを最初のフレームとするか、
 - 考案 (elaboration) : くわしく表現するとき如何にサブフレームを選ぶか、
 - 変更 (alteration) : 十分に合致しないとき如何に代りのフレームを発見するか、
 - 新型 (novelty) : どのも合致しないときどうするか、
 - 学習 (learning) : 経験の結果としてどのフレームを修正または生成するか。
- 記憶の中から最適なフレームを選び出す操作が極めて重要であるが、パターン・マッチング技法、クラスタリング理論および similarity network を組み合わせた技法が有用であろう。また、処理の途中でトラブルに出合ったときの手続きも非常に重要であり、matching, excuse, advice および summary の機能が有効であるが詳細は省略する。原論文を読んでいただきたい。

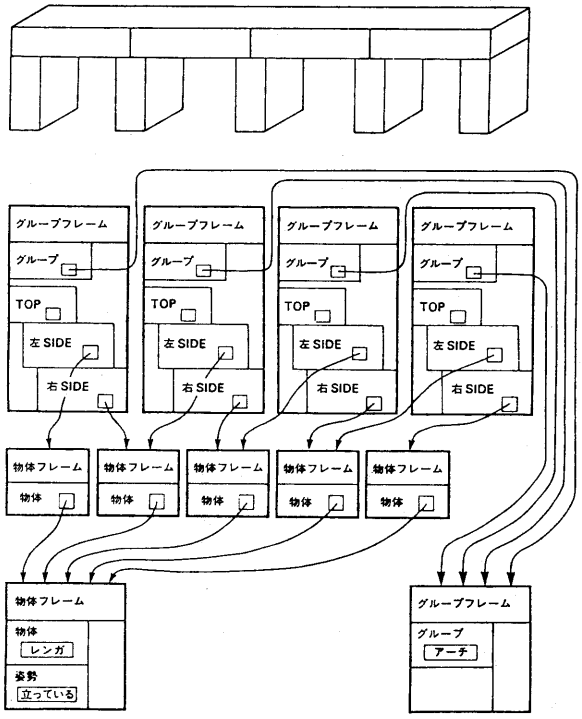


図4 橋を表すフレーム・システムの例(3より)

3. フレーム型システムの例

Minsky のフレーム理論がヒントとなって設計された知識型システムは少くない。ただし、設計者と対象として目的によってかなり実際のシステムは異なっている。共通な点は、対象および知識をフレームの集合として表現し、具体的な情報をそれらのフレームのスロットに持たせるといった表現法のことだけであると言っても過言ではない。ただし、汎用ツールとして設計されたシステムに

は、フレーム理論の特徴を示す共通の技法がいくつか見られる。階層構造とインヘリタンス (inheritance) 機能の組合せ、procedure attachment による各フレーム特有な推論機能の実現などがこれである。推論の進め方に関しては、フレーム構造を採用したことによる利点は大きく、"仮説生成と検証" (hypothesize and test) という高度な概念を data-driven と hypothesis-driven という2種の推論制御手法を組合せて比較的容易に実現できるとし、サブゴールの設定と注目点 (focus of attention) のダイナミックな制御など、システムの柔軟性を高める上で極めて有効となっている。なお、frames を units, prototypes, scripts あるいは schemata などと呼ばれることもある。

3.1 PIP

PIP (The Present Illness Program) は初期の医学エキスパート・システムであり、腎臓病に関する患者病歴を質問-応答形式で収集するコンピュータ・プログラムである。診断論理はもっていない。エキスパートのタリオをシミュレートする目的で設計された(4)。

病気カテゴリは図4に示される様な因果関係のネットワークを形成し、各ノードがフレームで表現されている。一方、所見もまた各フレームとして表されており、属性の取り得る値が記入されている。各病気仮説フレームは、スロットとして主な所見、他の病気との因果関係、評価ルール、スコアリング関数および鑑別診断リンクをもつ。推論は次の様に行われる。先ず利用者がいくつかの所見を与えるとシステムがそれらの所見の属性値を質問し、それらに答える。その結果図5のような所見フレームが生成される。次に与えられた所見を手がかりとして病気仮説フレームを選択し、推論用生成する。ここでこれらの仮説フレームの各スロットの情報を用いて、評価に必要な他の所見を問合せたり、関係のある他の病気仮説フレームについての評価をするなどの処理を次々に実行して行く。実行の結果として、所見のリストと可能性のある病気仮説のリストが重要度とともに得られる。

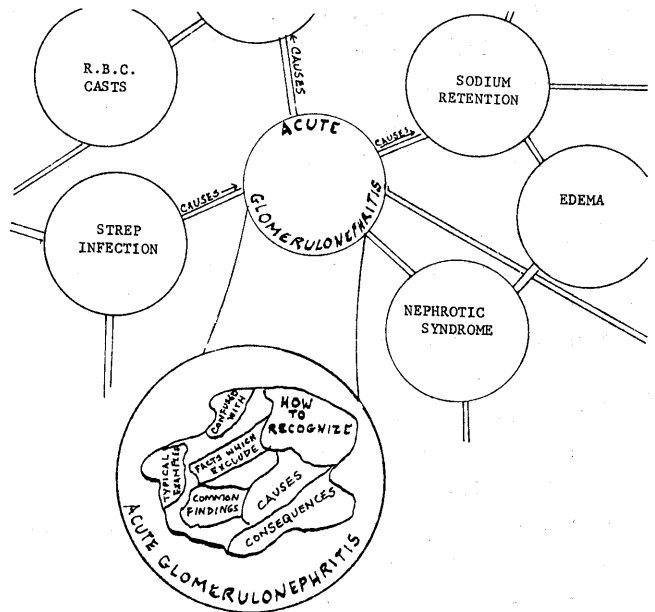


図4 Hypotheses networkの一部

Edema	
LOCATION	: Facial
STATUS	: Present
SEVERITY	: 2+
DURATION-OF-EPISODE	: For-Days
PITTING?	: Non-Pitting
PAIN	: Not-Painful
ERYTHEMATOUS?	: Not-Erythematous
SYMMETRY	: Symmetric
DAILY-TEMPORAL-PATTERN	: Worse-in-the-Morning
RECURRENCE	: First-Time

図5 所見EDEMAのフレームと各スロット

3.2 NUDGE

NUDGEは知識型のスケジュール作成プログラムであり、FRM(フレーム型の知識表現言語)で書かれた応用システムである。例えば、利用者が研究打合せを計画する目的で希望日時や出席者を入力すると、システムは持っている知識を使って確定するために必要な情報の入力も要求し、他のスケジュールとの調整を行うなどを行って、最終的なスケジュールを作成し、必要なメンバーへの連絡も行う。つまり秘書的な作業を代行するプログラムである。(5)

このスケジュールリングのためには、5種類の性格の異った知識(情報)が必要であり、それは作業の種類、メンバー、場所、時間および作業の内容である。これは別々に階層構造のフレーム・システムとして知識ベース内に格納される。この各要素がフレームとして定義されている。図6にフレーム・システムを、図7にその中のPA-MEETINGフレームの構成を示す。この中で例えばス

PA-MEETING	SVALUE	MEETING
AKO	SVALUE	PA-PROJECT
WHY	SDEFAULT	AI-PLAYROOM
WHERE	SDEFAULT	((ON FRIDAY) (FOR 1 HOUR))
WHEN	SDEFAULT	IRA [ROLE: MANAGER]
WHO	SDEFAULT	BRUCE [ROLE: FRL]
		CANDY [ROLE: SEMANTICS]
		MITCH [ROLE: SYNTAX]

図7 PA-MEETINGフレームの構成

ロットAKOは a-kind-ofを意味し、このフレームの直接の上位フレームがMEETINGフレームであることを示す。場所はデフォルトで定義されて

いるから、このミーティングは通常AI-PLAYROOMで用いられることを示す。処理は例えば利用者が、

NUDGE, schedule a meeting with Bruce for next Tuesday.

と指示して開始される。まずこの問題の実行のためフレーム・システムをESTABLISHが生成され、前述の知識ベースを用いて、スケジュールリングを行って行く。フレーム・システムが階層構造にわたっているので、inheritanceの機能を用いて、スワキリと知識を体系的に管理できる。また、IF-NEEDEDやIF-REMOVED等のprocedural attachmentsを使って、そのフレーム特有の処理を容易に指定することができる。

3.3 CENTAUR

この後に開発されたWHEELER(後述)とCENTAURは、典型的なプロダクション・システムであるMYCIN(EMYCIN)の弱点を解決する試みをもって試作された。Framesとproduction rulesとを組み合わせたシステム

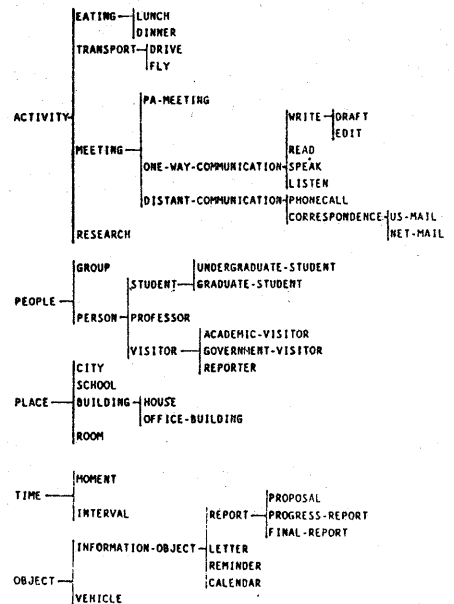


図6 階層形フレーム・システムの定義

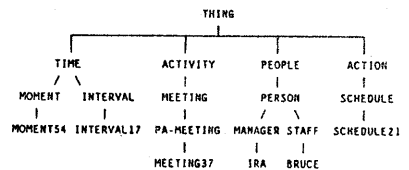


図8 GESTALTのフレーム・システム

である。プロダクション・システムの弱点は、知識が平面的なルールの集合であるので、病気の特徴的なパターンをつかみにくく、推論の順序が単調で柔軟性に欠けることである。このシステムでは、PROTOTYPEと呼ばれるフレーム表現で各病気の仮説を表し、その中のCONTROL SLOTSの情報のよって、順序の制御を指定できるようになっている。更に、フレームに対応して定義されたプロダクション・ルールで、病気特有の論理を表現している。図9と10にこれらの例を示すが、二つの組合せで、知識がすつきりと表現されていることがわかる。なお、この例はMYCIENで書かれたPUFFをCENTAURで書き直したものの一部である。(6)

3.4 WHEEZE

これはCENTAURと同じく、スタンフォード大で開発されたシステムである。ここでは推論を構成する論理的表現単位が図11に示される様がフレームで表されている。図の上部に対応するPUFFのプログ

If: 1) The severity of Obstructive Airways Disease of the patient is less than or greater to mild, and
 2) The number of pack-years smoked is greater than 0, and
 3) The number of years ago that the patient quit smoking is 0
 Then: It is definite (1000) that the following is one of the conclusion statements about this interpretation: Discontinuation of smoking should help relieve the symptoms.

OADwithSmoking:

Manifestation ((OAD-Present 10) (PatientHasSmoked 10) (PatientStillSmoking 10))

SuggestiveOf ((SmokingExacerbatedOAD 5) (SmokingInducedOAD 5))

ComplementaryTo ((OADwithSmoking-None 5))

Certainty 1000

Findings "Discontinuation of smoking should help relieve the symptoms."

HowToDetermineBelief function for computing the minimum of the beliefs of the manifestations

図11 WHEEZEのフレームの例

3.5 UNITS

Unitパッケージ(UNITS)は、フレーム型の知識ベースを表現するための汎用言語である。名前が示す様に、これは(INTERLISPで書かれた)

PROTOTYPE	Obstructive Airways Disease
GENERAL SLOTS --Bookkeeping Information --Pointers to other prototypes (link prototype) --English phrases	Author: Aikins Date: 27-OCT-78 Source: Dr. Fallat Pointers: (degree MILD-OAD) ... (subtype ASTHMA) ... Hypothesis: "There is an interpretation of OAD."
CONTROL SLOTS Control If-Confirmed If-Disproved Action	If-Confirmed: Deduce degree of OAD Deduce subtype of OAD Action: Deduce OAD findings Print OAD findings
COMPONENTS Plausible Values Default Value Possible Error Values Rules Importance of value to this prototype	TOTAL LUNG CAPACITY Plausible Values: >100 Importance: 4 REVERSIBILITY Rules: 19, 21, 22, 25 Importance: 1

図9 PROTOTYPEの例と実際の値の入ったフレームの例(右)

If: 1) A: The mmf/mmf-pred is less than 20, and
 B: The fvc/fvc-pred is greater than 80, or
 2) A: The mmf/mmf-pred is less than 15, and
 B: The fvc/fvc-pred is less than 80
 Then: 1) There is evidence that the degree of OAD is severe, and
 2) One of the OAD findings is: Low mid-expiratory flow is consistent with severe airway obstruction.

図10 プロダクション・ルールの例

クション・ルールが示してある。(7) 図にあるスロットのうち、ManifestationおよびSuggestiveOfおよびComplementaryToはそれぞれ特有の評価機能をもつとともに、右側に記入されている各フレームへのポインタでもある。数値は重要度を意味し、評価のとき利用される。またMYCIENで不明瞭であった信頼係数の評価方法も、HowToDetermineBeliefスロットで、フレームごとに指定できる。推論の制御は、次の評価対象を決定する各時点で、評価の候補をプライオリティ値をつけてリストにしたagendaと呼ばれる記録の中から選定するという技法により、data-drivenおよびgoal-drivenの両方を実現し、医師のそれに近いダイナミックな推論が表現できたと述べている。(詳細は文献7を参照)

コマンドのパッケージである。強力な *inheritance* 指定機能と組合せられた階層型のフレーム・システム、豊富なデータ・タイプとデフォルト指定、および極めて汎用的な *procedural attachment* などにより、非常に汎用性の高い強力な知識表現システムを実現している。汎用性はこのシステムが特定の推論機構を持っていないことからも来ている。従って *UNITS* のユーザは、推論の戦略も設計しなければならないという大きな負担があるが、逆に言えばどんなシステムでも設計できるという柔軟性がある。attached procedure としてどんな *LISP* 関数でも書け、しかもこの中で *UNITS* のコマンドをも利用できるのも、これを使ってどんな強力な推論も実現できる。これはまた、明瞭な方針を持たないユーザは混乱に陥入ることも意味する。詳細は文献 8 をゆずり、ここでは極く簡単な例題 *CONET* (*Coagulation network*) を通して、システムの機能の一部を述べる。

CONET は 5 つの 2 進木ノードからなる簡単な *binary-branching logic* で、7 つの結論および 1 つの中間結論をもっている。これを実行するためのフレーム・システムの定義を図 12 に示す。図に示される様に、結論、所見、論理、および *LISP* 関数のサブ・トリーで構成されている。*LISP PROC* サブ・トリーには、任意のフレームの中で定義された *attached procedure* が全て自動的に定義される。図 13 に、*LOGIC* フレームの構成を示す。フレームの各スロットは、スロット名、インヘリタンス機能指定、データタイプ指定およびデータ値の組であるが、データタイプとして、他のフレームへのポインタ (*UNIT*)、*attached procedure* (*LISP*)、*ATOM*、テキストなど 13 の型のいずれかが指定でき、これとデフォルトを組合せることにより、極めて柔軟性に富んだフレームも定義できる。

図 14 に *attached procedure EVALUATE* の例を示す。下線を著したものが *UNITS* のコマンドの例である。

この例は簡単過ぎるし、*UNITS* 用の例題としてもふさわしくはないが、誰

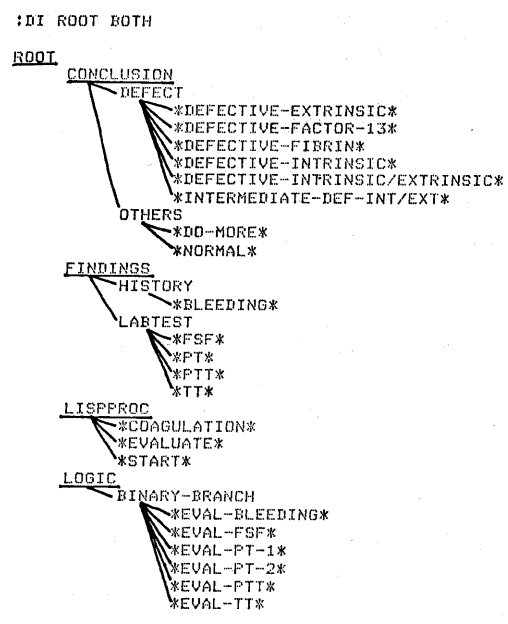


図 12 CONET の階層フレーム・システム

```

UE:EDIT LOGIC
EDIT: !PR ALL
DESCR: (U) from ROOT <DESCR>
SCHEMA NODE FOR A BRANCHING LOGIC
MODIFIER: (U) from ROOT <MODIFIER> LINDBERG
CREATOR: (U) from ROOT <CREATOR> LINDBERG
MODIFIED: (U) from ROOT <MODIFIED> 12-Mar-80 07:44:10
CREATED: (U) from ROOT <CREATED> 13-Nov-79 08:11:18
TCHILD: (U) *Top* <UNIT>
FCHILD: (U) *Top* <UNIT>
DOCUMENT: (U) *Top* <TEXT> DOCUMENT OF THIS UNIT
BRANCHING: (U) *Top* <LISP>
PARENT: (U) *Top* <UNIT>
TCONC: (U) *Top* <UNIT>
FCONC: (U) *Top* <UNIT>
FINDING: (U) *Top* <UNIT>
  
```

EDIT: DO
(Leaving EDIT of unit LOGIC)

図 13 LOGIC フレームの内容

にでも解り易いからシステムのしくみも知る手がかりにはなろう。しかしLUNITSは極めて豊富なコマンドをもっているから、Long term memoryとして問題解決の知識を持たせ、実際の推論のときには、その問題自体の処理の為に、これとは別にShort term memoryとしてのフレームシステムも構築しながら推論を進めるといったやり方ができる。更し、procedural attachmentとデフォルトなどの組合せを使つて、各フレームが“私は何をするべきか、どうすべきかを知っている”ように設計し、推論機構の役割は単に適切な順序で適切なフレームも指定するだけではないという、いわゆるobject-centered controlも可能とならう。但し、コマンドが多過ぎてシステムが大き過ぎるといった批判もある。

4 まとめ

紙面の制約で十分に考察できなかったが、フレーム理論のすぐれた点と、設計者によって具体的なフレーム・システムの見方により差思があることはお解りいただけたと思う。

文献

- 1) E. Feigenbaum; The Art of Artificial Intelligence - Themes and Case Studies of Knowledge Engineering, Proc. 5th IJCAI (1977)
- 2) M. Minsky; A Framework for Representing Knowledge, in P. Winston (ed.) The Psychology of Computer Vision, McGraw-Hill (1975)
- 3) P. Winston; Representing Knowledge in Frames, in Artificial Intelligence, Addison-Wesley (1977), 人工知能(長尾, 白井訳), 培風館
- 4) P. Szolovits and S. Pauker; Research on a Medical Consultation System for Taking the Present Illness, Proc. 3rd Illinois Conference on Medical Information Systems (1976)
- 5) I. Goldonam; NUDGE, A Knowledge-Based Scheduling Program, Proc. 5th IJCAI
- 6) J. Aikins; Prototypes and Production Rules: An Approach to Knowledge Representation for Hypothesis Formation, Proc. 6th IJCAI (1979)
- 7) D. Smith and E. Clayton; A Frame-Based Production System Architecture, Proc 1st AAAI (1980)
- 8) M. Stefik; An Examination of a Frame-Structured Representation System, Proc. 6th IJCAI
- 9) H. Pople; The Formation of Composite Hypotheses in Diagnostic Problem Solving - an Exercise in Synthetic Reasoning, Proc. 6th IJCAI (1979)
- 10) N. Mantim, et al; Knowledge Base Management for Experiment Planning in Molecular Genetics, Proc. 5th IJCAI (1977)

```
(EVALUATE
  LAMBDA (UNIT)
    (* Attached procedure *)
    (* edited: "14-Nov-79 14:29" *)

  (PROG (ANS)
    AGAIN
      (PRINTSLOT 'QUESTION UNIT)
      (ANS_(READ))
      (if (EQUAL ANS '?')
        then (GO HELP)
        elseif (EQUAL ANS 'Y')
        then (RETURN ANS)
        elseif (EQUAL ANS 'N')
        then (RETURN ANS)
        else (GO ERROR))
      HELP (PRINTSLOT 'HELP UNIT)
      (GO AGAIN)
    ERROR
      (WRITE "Input error. Try again!")
      (GO AGAIN))
```

図14 Attached procedureの例