ICOT Technical Memorandum: TM-0577

TM-0577

問題解決モデルと知識表現
－昭和62年度 知識システムシェル・WG・問題解決モデルと知識表現サブワーキング・グループ報告書－

KSS・WG－KRP・SWG

December, 1988

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

昭和６３年１２月２６日

ＩＣＯＴ　Ｔｅｃｈｎｉｃａｌ　Ｍｅｍｏｒａｎｄｕｍ

ＴＭ－５７７

『問題解決モデルと知識表現』

－ 昭和６２年度知識システムシェル・ワーキンググループ・
問題解決モデルと知識表現サブワーキング・グループ報告書 －

ＫＳＳ－ＷＧ・ＫＲＰ－ＳＷＧ
（主査：小林　重信「東京工業大学」）

編集：竹之内　博夫，坂根　清和
（ＩＣＯＴ　第５研究室）

目次

まえがき

昭和６２年度のＩＣＯＴ知識システム・シェル・ワーキング・グループ（ＫＳＳ－ＷＧ）は次世代知識システム構築のための５つの要素技術（仮説推論，問題解決モデルと知識表現，分散協調問題解決，知識獲得支援，設計型エキスパート・システム）についてサブワーキンググループ（ＳＷＧ）を開催する形式で運営された。本報告書はこの内の問題解決モデルと知識表現ＳＷＧの担当分野について報告する。（主査：小林重信助教授[東京工業大学] ）

本報告書は，９回にわたるＫＲＰ－ＳＷＧにおける研究発表・討論に基づいて，各委員・オブザーバが(1) 問題解決モデル および (2) 知識獲得と学習の２つの観点から各自の研究内容のまとめ・問題提起の形で執筆した活動報告書を編集したものである。

本報告書がＡＩ研究における次世代知識システム・シェル研究に対する１つの提言として役立てば幸いである。

１９８８年１２月２６日

編集者　坂根 清和

1　活動報告総論

KRP-SWG 主査　　小林 重信　（東工大）

(1) はじめに

分類や診断など解析型問題の分野ではエキスパートシステムを構築するための方法論がかなり確立され、方法論に基づく有用なツールも利用可能になってきている。これに対し計画や設計など合成型問題の分野では方法論の整備が遅れており、試行錯誤的・場当たり的なアプローチがなされているのが現状である。このような現状認識がKRP/SWGを組織化する契機になっている。

合成型問題では、システムの特性が与えられたとき、これを実現するようなシステムの構造と構成要素の特性を決定することが要請される。合成型問題に対する基本的接近法は、生成検査法(Generate and Test)に求められるが、組合せ的爆発の問題をどう克服するかが本質的課題である。

合成型問題へ接近するうえでいくつかの視点が考えられる。

第1の視点は、合成型問題をシステムの構造と構成要素の関係としてとらえる立場である。システムの構造と構成要素の関係に着目すれば、合成型問題はいくつかのクラスに分類可能であり、各クラスごとに類型的タスク(Generic Tasks)を抽出することが可能と考えられる。

第2の視点は、合成型問題を高次推論問題として定式化する立場である。すなわち仮説推論や類推、協調型推論などを武器として接近しようとするものであり、工学的観点から高次推論を整備することにより、合成型問題への強力な手がかりを得ることが期待できる。

第3の視点は、合成型問題を機能的仕様から操作的仕様に変換するプロセスととらえる立場である。“説明に基づく一般化”(Explanation Based Generalization：以下EBGと略す)の理論および手法は領域理論、操作性規範および事例に基づいて機能的目標を操作的レベルに展開し、これを一般化することにより、知識の洗練化を行うもので、合成型問題に対する方法論を確立するうえで有望な基盤技術として着目される。

本報告は、以上のような現状認識および視点をベースとして、昭和62年7月より昭和63年3月までに行われたKRP/SWGの活動内容と成果をまとめたものである。

(2) 活動方針

KRP/SWGは次の方針に基づいて活動を展開することとした。

1)合成型問題をシステム構造と構成要素特性の関係として定式化したうえで、いくつかのクラスに分類し、各クラスごとに類型的タスクを抽出し、問題解決モデルを構築する。

2)合成型問題に対する問題解決機能を整備するために、類推や協調型推論などの高次推

論を基盤技術として確立する。

3)学習の新しい方法論であるEBGの現状分析を行い、理論的および手法的基礎を固め、合成型問題への利用可能性を検討する。

4)会議は３回／２月、計１０回程度とする。

(3) 活動内容

KRP/SWGは９回の会合を通じて次のような活動を行った。

第１回　昭和62年7月17日

KRP/SWGの活動方針を決定した。

第２回　昭和62年8月20日

1)片井（京大）：設計プロセスのモデル化

Mostowの文献を題材に、設計プロセスをモデル化するうえでの６つの視点およびモデル化を行ううえでのアイデアを考察した。

2)寺野（電力中研）：類型的タスク

Chandrasekaranが提唱する類型的タスクとその応用を紹介したうえで、本アプローチの有用性と限界を議論した。

3)小林（東工大）：問題解決と類型的タスク

合成型問題をシステム構造と構成要素特性の関係に着目して３つのクラスに分類し、各クラスごとに基本的接近法を示し、これに基づく類型的タスクを抽出した。

第３回　昭和62年9月11日

1)竹之内（ICOT）：仮想知識ベース検索モデル

エキスパートシステムの構築を容易にするという問題解決的観点から仮想知識ベース検索モデルの構想を述べた。

2)橋木（京大）：説明による学習と類推による学習

設計支援における学習の重要性を指摘し、説明による学習と類推による学習の比較検討を行い、設計問題への利用可能性を考察した。

3)新田（電総研）：法律における推論

特許法におけるシステム構築の経験に基づいて、法律知識の表現モデルおよび高次推論機能の利用可能性を論じた。

第４回　昭和62年10月2日

1)小野（広大）：目標間の関係に着目した分散協調型問題解決

複数エージェントの目標間の関係を利用した分散協調型問題解決のモデルを提案し、応用例としてタクシー網の分散協調制御を紹介した。

2)末田（東芝）：設計／計画エキスパートシステムの事例分析

画像処理エキスパートシステムおよびプリント基板設計工程計画エキスパートシステムを事例として類型的タスクの観点から分析を行った。

3)石田（NTT）：知識ベース構築に適した表現モデル

kEE、ART、Knowledge Craftの３つのシステム構築ツールのフレームシステムについて、機能の比較分析を行い、各ツールの特徴と問題点をまとめた。

第５回　昭和62年11月5日

1)寺野（電力中研）：エキスパートシステムにおける階層的決定法の利用

AHP(Analytic Hierarchy Process)を適用したエキスパートシステム開発の研究事例を紹介した。

2)山村（東工大）：類推による問題解決

類推を問題解決の観点から考察し、類推に階層的問題解決を導入することにより、マクロを生成する機能が実現されることを示した。

第６回　昭和62年11月27日

EBGに基づく最新の論文７篇のサーベイを行った。

第７回　昭和62年12月27日

前回に引続き、EBGに基づく最新の論文８篇のサーベイを行った。

第８回　昭和62年2月2日

第６回および第７回のサーベイに基づいて、EBGの総括を行った。

1)山村・小林（東工大）：工学的システムにおける学習問題へのEBLによる接近

工学的応用の観点からSBG(Similarity Based Generalization)とEBGをそれぞれ単独に用いる際の問題点を指摘し、両手法を融合した学習を提案した。

2)樋木（京大）：EBL-知識整理と学習の観点から－

EBGが従来のシステム工学的方法に代って情報整理手法として使えることを指摘した。またEBGとSBG、類推、事例推論の関係を示した。

3)寺野（電力中研）：EBLの応用について

EBGの応用領域は典型例からの一般化が要請され、問題解決の効率化が望まれ、知識コンパイルが効かないことを特徴とすることを指摘した。

4)新田（電総研）：EBGの法律への適用

判例へのEBGの適用により、判決の中から一般的な法律解釈を抽出することができ、

法律概念の具体化への応用可能性を指摘した。

5)片井（京大）：設計物に対する説明と理解

設計物の働きに対する因果機械論的説明と目的論的説明を結びつけるうえで機能系統図の役割とその導出を論じた。

第９回　昭和62年3月3日

1)石田（NTT）：EBGの課題

EBGの課題について、特に、Incomplete Theory Problemが工学的観点からは重要であることを指摘し、階層関係の視覚化への応用の構想を示した。

2)小野（広大）："説明の説明"に基づくEBGについて

従来のEBGでは単純かつ表層的な一般化しか行われないことの問題点を指摘し、説明の理解に基づくEBGの方法を提案した。

3)末田（東芝）：EBLの画像処理への応用

画像処理エキスパートシステムEXPLAINにEBGを導入することにより、知識獲得機能を強化する構想を示した。

4)SWG活動の総括

９回にわたったKRP/SWGの活動を総括するために、研究成果をResearch Memoとして残すこととし、報告書の構成と役割分担を決めた。

(4) 活動成果

KRP/SWGの活動成果は次のように要約される。

1)合成型問題における類型的タスクおよび機能分析について

①合成型問題をシステム構造と構成要素特性の関係によって、クラス１（システム構造、構成要素特性ともに未知）、クラス２（システム構造が未知、構成要素特性は既知）、クラス３（システム構造は既知、構成要素特性が未知）の３つに分類し、類型的タスクが明らかでなかったクラス１およびクラス２の合成型問題に対し、それぞれ"設計事例の手直し"および"トップダウン精密化"に基づく類型的タスクを提案した。[小林]

②画像処理エキスパートシステムおよびプリント基板設計工程計画エキスパートシステムを取り上げ、類型的タスクの観点から事例分析を行い、類型的タスクの枠組みの有用性を確認した。[末田]

③設計物の働きに対する説明は、実体集合空間－価値集合空間の両極間を「因果機械論的説明」と「目的論的説明」を通して結びつけることによりなされるとして、このような説

明構造を可視化する手段として機能系統図の有用性を指摘した。精密な機能系統図を導出するうえで、公理論的方法、機能推論、Consolidationと併せて、EGBの適用が有用であることを示した。［片井］

2)問題解決と高次推論について

①分散協調型問題解決を「物理的に分散され、疎に結合された知識源の集りによる協調的な問題解決である」と定義して、分散協調型問題解決モデルとしての黒板モデルおよび契約ネットモデルの枠組み上の問題点を指摘し、新しい分散協調型問題解決モデルとしてエージェントの目標間の関係を陽に利用した方法を提案し、タクシー網の分散協調制御への応用を行った。［小野］

②問題解決型類推による有用性を向上させるためには、BASE問題の構造分析を利用して、類推によって生成される副目標の任意性を制限することが必要であることを指摘し、類比における不完全さを抽象化としてとらえなおし、類推と階層的問題解決を融合する新しい枠組みを提案した。［山村・小林］

③商用ES構築ツールについて、ルールの特徴はよく知られているが、フレームの特徴があまり知られていないとして、KEE、ART、KCの３つのツールのフレーム機能について比較分析を行い、各ツールに共通して、表現された知識のセマンティックスの保証がないこと、継承機能に問題があること、言語機能が直交していないこと、を指摘した。［石田］

④特許法のコンサルテーションシステムの構築および法律ES研究会への参加を通して得られた知見に基づいて、法律推論における推論の形態とその特殊性、法律ES構築の際の問題点およびオブジェクト指向の考え方を導入した法律ESの構築を提案した。［新田］

⑤階層化意思決定法として知られるAHP(Analytic Hierarchy Process)のESへの適用可能性を検討し、実際にAHPを組み込んだESの開発経験を紹介し、AHPの有用性とその限界を明らかにした。［寺野］

3)EBGの知識獲得および工学的応用への可能性について

EBGはSBG（類似性に基づく一般化）を超える極めて有望な枠組みであるとの認識のもとに、AAAI-87およびIJCAI-87の関連論文をサーベイし、さらに知識獲得問題や合成型問題解決への利用可能性を検討した。

①設計支援における学習の重要性を指摘し、EBGとSBG、類推、事例推論との関係を明らかにして、設計問題への利用可能性を考察した。さらにEBGは従来のシステム工学的方法に代って情報整理手法としても使えることを指摘した。［椹木］

②工学的応用の観点からSBGとEBGをそれぞれ単独に用いる際の問題点を指摘し、両手法を融合した学習方法を提案した。工学的応用では操作性規範が重要であるとして、操作性を保存する一般化関係を用いて学習させることにより、探索の効率が促進されることが可能なことを示した。［山村・小林］

③従来のEBGでは単純かつ表層的な一般化しか行われないことの問題点を指摘し、説明の理解に基づくEBGの方法として、“説明の説明に基づくEBG”を提案した。［小野］

④EBGの適切な応用領域は典型例からの一般化が要請され、問題解決の効率化が望まれ、かつ知識ンバイルが効かないことを特徴にもつ分野であることを指摘した。［寺野］

⑤EBGの課題について、特に、Incomplete Theory Problemが工学的観点からは重要であることを指摘し、ワークステーション上でのネットワークの視覚表示への応用の構想を示した。［石田］

⑥画像処理エキスパートシステムEXPLAINにおける知識獲得機能を強化するために、EBGを導入する構想を示した。［末田］

(5)おわりに

エキスパートシステムの研究開発の進展に伴い、ユーザの関心はつぎのようにシフトしつつある。

1)実装レベルでの知識表現からタスクレベルでの問題解決へ

2)解析型問題への接近から合成型問題へ接近へ

3)KEによる知識獲得支援から学習システムによる知識獲得へ

これらの問題を解くキーワードは、類型的タスク、高次推論と問題解決、知識獲得と学習であり、KRP/SWGはこれらを中心に活動を進めてきた。特に、新しい学習の理論であるEBGについては、KRP/SWG活動の時間の半分以上を投入した。

EBGが注目されるのはEBGによって初めて学習の中に工学的視点が導入されたことにある。すなわち、機能的目標概念、操作性規範、領域理論の導入により、学習を問題解決システムとしてとらえることが可能になり、工学的応用の可能性が拓けてきたといえる。

SWGの活動期間が約８ヶ月と限られていたため、活動がピークを迎えようとしている時期に解散せざるをえないことは残念なことであるが、来年度以降のWGにKRP/SWGの成果が発展的に継承されることを期待して結びとする。

2　問題解決モデル

２．１　法律における問題解決モデル

エキスパートシステムの構築技法の発展に伴い、多くの分野でエキスパートシステムが開発されるようになってきた。我が国では法律分野のエキスパートシステム（法律ＥＳ）はやっと開発が開始された段階であるが、アメリカにおいては古くから法律ＥＳのプロジェクトが存在していた。本稿では法律における問題解決モデルの必要性とその１つの案を述べる。

基本的な法律ＥＳ

初歩の法律ＥＳでは、条文をそのまま論理式に変換して知識ベースに格納し、問題解決は論理式の定理証明の形で行われるものが多かった。質問したい事柄を定理として考え、定理が知識ベースの論理式から証明できれば適法であり、証明できなければ違法であると判断される。条文の論理式への変換については例えば以下の点が問題となり、さまざまな工夫がされてきた。

①条文は単独では解釈できず、他の条文と整合をとる必要があることがある。すべての条文との整合をとると前提条件の部分が膨大なものとなることがある。

②表現の基礎となる概念（用語）の抽象度のレベルが条文によって異なる。同じ用語でも条文によって解釈が異なることがある。

③法律には構造があり、すべての条文が同格ではない。

④時間やデフォールトの推論が要求されることがある。

⑤条文にはいくつかの範疇があり、論理式に変換しにくいものがある。

条文の単純な論理式化は直感的であると同時に実現が容易であり、いくつかの法律の診断問題にはこれで対処できる。しかし、現実に高度な問題解決を行わせるにはこの手法ではすぐ限界に達してしまうことがわかってきた。条文を論理式に変換する操作がすでに法解釈の問題に関わっているし、法律の専門家の行う問題解決は定理証明で行われるような三段論法ではなく、法律原理を用いた法的推論が行われているのである。専門家の行う問題解決モデルの構築なしに、法律ＥＳを構築しても大きな発展は望めない。以下では法律の問題解決モデルについて考察する。

法律体系と法原則

我が国においては法律は実定法の条文として日本語で表現されている。条文に表われる文章は抽象的な概念（用語）で書かれており、これを現実の事件にあてはめて要件が満たされているかを判断するのが専門家としての裁判官の役割である。例えば、「権利の侵害」という用語に対して、どのような行為が侵害行為を形成するかについては裁判官によって異なる判断がなされる可能性がある。この判断の基準になっているのが「何が法であるべ

きか」という原則（法原則）である[1]。これには、法の目的（立法趣旨）や社会通念や国の政策などが含まれる。時代の変遷に伴なって法原則も変化していく。従って、条文は時代の変遷とともに異なった肉付けがされて、動的に解釈され適用されることになる。法原則の詳細は専門家によって異なるため、法の解釈において主観的要素が含まれることは避けられないが、その基本的な部分は社会の合意によって形成されているものと考えられ、法律上の判断には他の法律の解釈や判例や他の諸条件との整合性が要求される。

このように法原則は問題解決を行うにあたり、適切な条文がないときや条文の条件を満たすかどうかの判断が困難な場合に用いられる知識である。専門家の経験的知識であり、かつ、社会的に容認されている原則であると言える。しかし、法原則は実定法のように確立したものではなく抽象的な概念である。また、「公共の利益」と「基本的人権の尊重」のように法原則同士が矛盾を生じることとなり、「公平の原則」という別の法原則で調整する必要があるなど複雑な要素がある。

次に「法的推論」について考える。法的推論においては、まず、事実の認定が必要である。裁判における事実認定には裁判官の経験的な知識と状況判断がものを言う。事実が認定されると関連の条文が選定され推論がなされる。この場合、条文の前提条件にあたる部分にこの事件の事実が該当するかを判断しなければならない。前提条件は通常は抽象的な記述がなされているので、具体的事件と比較する場合には、前提条件をこの認定された事実レベルに合致するように枠を設定する（具体化する）ことによって行われる。これを「解釈」という。解釈がなされ、前提条件が認定事実と合致したときに法が適用される。このように、法的推論においては大前提（条文）の範囲に小前提（認定事実）が入るか否かの判断において枠の設定という三段論法と異なる要素が含まれている。

ＥＳにおける法原則の取込み

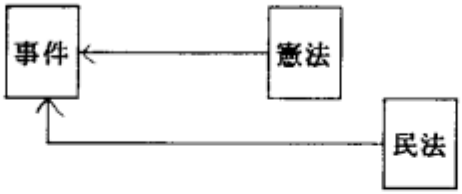
前に述べたように法原則は実定法と異なる抽象的な概念であって、一般的にはこれを知識ベースに取込むのは困難である。しかしながら、法原則の一部は憲法のような上位法に実定法として以下のように表わされているのであって、これを知識ベース化することは可能である。

憲法11条　国民はすべての基本的人権の享有を妨げられない。

憲法13条　すべて国民は個人として尊重される。生命、自由及び幸福追求に対する国民の権利については、公共の福祉に反しない限り、立法その他の国政の上で、最大の尊重を必要とする。

実際の法の運用にあたっては、前述のように適用される条文を法原則によって動的に解釈し、問題解決に用いることとなる。しかし、法律ＥＳにおいては、原文を問題に応じて動的に解釈することは困難であり、事件解決にあたっては、民法のような下位法と憲法のような上位法を２重に適用することによって法原則を適用することが現実的である。すなわち、同一事件について民法と憲法を並列に適用し、いずれにも抵触しなければ適法と判

断するのである。この場合、憲法と民法のそれぞれについて、どのような条件のもとではどのような判断がされるかについての知識が用意されていなければならない。



法律における問題解決モデル

法律ＥＳにおいては法原則を用いた動的な問題解決は困難である。実際には、法原則を必要とするようなＥＳは必ずしも多くはなく、事前に法解釈が（静的に）定まり、ある程度の判断基準が経験的に定まっているものが多い。しかし、このような場合でも知識ベースには条文のみならず、解釈や運用のための知識を十分に補ったものでなければ問題解決には役立たないことは言うまでもない。その場合、解釈や運用のための知識をできるだけ原条文から分離して知識ベースを構成したほうが、知識が明確に整理され、判例や法改正などによる知識ベースの更新が容易になる。モデル構築のため、法律ＥＳの機能を考察する。

Ⅰ．法律ＥＳの機能

法律ＥＳの機能は、条文から直接答えを引出せるものと、事例問題に法を適用するものに大別できる。前者は、法律に関する質問に応答を行うものであり、具体的な事例がないために回答は法律の条文にある範囲の抽象的なものにとどまる。このタイプの質問には、条文を１つのデータベースと考えてキーワードから直接に関連条文を引出すもの

EX.「～に関する条文を列挙せよ」
「条文～の例外規定はあるか」

と、条文間の論理的な帰結を必要とするもの

EX.「～が成立すれば常に～ができるか」
「～ができるための条件を列挙せよ」

に分けることができる。

後者の事例問題解決機能は、事例問題に対して実際に生じる法律効果を調べたり、違法な出来事を検出したりして法律診断や予測、助言を行うものである。前者と異なり、事例問題では法律に直接表われない用語が用いられるために、どのような条件が満足されればどんな判断がされるかという知識が必要となる。

Ⅱ．法律の解釈

法律の知識ベース化のために条文をあらかじめ解釈しておかなくてはならない。法原則を用いた動的な法律解釈は困難なため、条文の使われ方をあらかじめ想定した解釈を行う必要がある。解釈が不必要に詳しくなると知識が複雑になるだけであって問題解決に役立

たないし、解釈が粗いと実効ある問題解決ができないことになる。あらかじめ問題の解決に必要な解釈のレベルを決めなければならない。

法律解釈にあたって以下の点を考慮しないと法律は形式的には矛盾を持っていることになる。①立法時にはその法律の適用する範囲が前提として決まっているはずである。しかしながら、条文にはその適用範囲が明確にはでてこない。その結果、条文の内容だけを取り出すと、いろいろの法律の条文が矛盾を生じることになる。一方の法律が他方の特別法であるならば、「特別法は一般法に優先する」というメタルールにより矛盾を解決できる。しかしながら、このような関係にない法律の間の調整や、「この規定は強行規定である（矛盾がおきたときはこの規定が優先する）」というような運用上の規則（法原則に関係する）の抽出は困難なことがある。②自然言語で記述されているため、常識的に考えれば問題が生じないときには、細かい適用条件が省略されることがある。また、解釈の余地が何通りもあることがある。

①の解決には、適用範囲を前提条件として補いながら条文を解釈するか、または、適用範囲を個々の条文の解釈には含ませないで、問題解決時にどの適用範囲の問題かを判定し（例えば、公害訴訟か、刑事事件か）、その世界から適用できる法律のみを選定してから「特別法は一般法に優先する」などのメタルールによって法を適用する。

②の解決には、専門家の助けを借りて、省略されている条件を補わなくてはならない。専門家は法解釈のための知識（法原則、法律用語の意味、自然言語の解釈、．．）を用いて法律を解釈するため、形式的に矛盾を生じても問題はないのである。例えば「Ａならば Ｂである」「Ｃならば Ｂでない」という矛盾する２つの規定があるとき、一方が原則的規定、他方が例外的規定であるならば「例外は原則に優先する」というメタルールによって解決できる。

さて、法律ＥＳの知識ベースとしては、このような条文の解釈だけでは十分ではない。法律に現れる用語は日常生活に使われる用語の意味と異なることがあるし、その語彙も法律用語ははるかに少ない。事例問題を日常用語で表現できるためには、日常用語と法律用語の辞書と「どのような条件が整えば、法律的にどのような行為とみなされるか」を決めるための知識が必要となる。

また、効率の良い問題解決を実現するためには、法律を事前に整理し、体系化しておくことが必要である。条文の文言にこだわらず、関連するキーワードからユーザの意図を推定し適切な応答をするためには、キーワード間の意味に関係した知識を要する。ある程度の論理的帰結を必要とする質問に効率的に答えるには、条文を論理式として部分計算しておく必要がある。

さらに法律適用の知識が必要である。解釈された条文には、原則的なルールしか与えられておらず、法律診断の目的には十分かもしれないが、「今後、どうしたら有利か」という戦略的な知識を必要とする問題には対応できないし、①に示したような適用世界の判断もできない。戦略的知識は条文には現れないが、この知識の質が専門家の実力に関係する

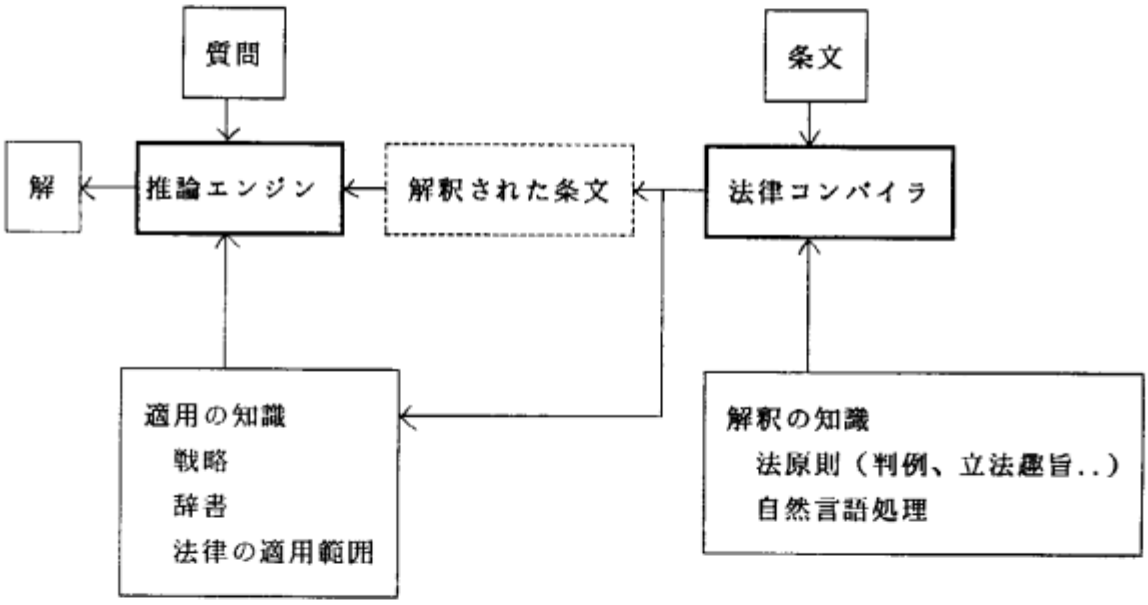
のである。

Ⅲ．法律の問題解決モデル

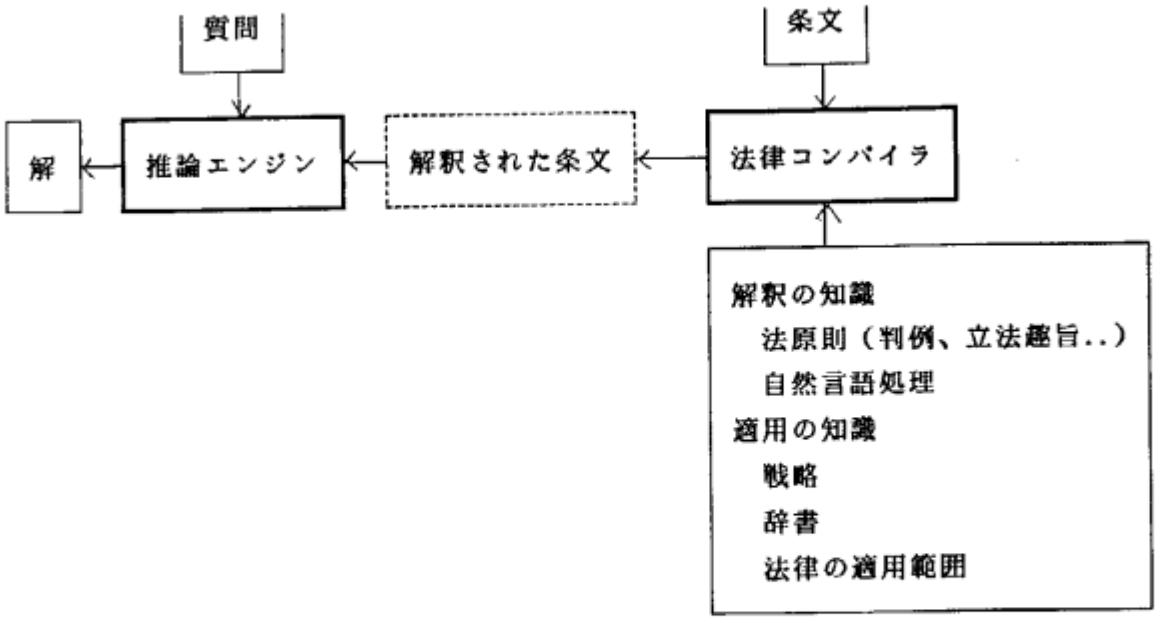
以上の考察から、法律ＥＳの実現には条文そのもの以外に「解釈の知識」と「適用の知識」とその他の付加的知識が必要である。ここで「解釈の知識」とは、立法趣旨や判例などから導かれる法原則と、法律解釈のための基礎知識および自然言語としての常識的知識を言う。「適用の知識」とは関連条文選択のための戦略的知識と、法律用語と日常用語を対応させるための知識と、法律の適用範囲に関する知識を言う。両者は完全に独立な知識ではなく、適用の知識は広い意味で解釈の知識と考えることができる。

完全に自動化されたシステムをモデル化すると、①部分コンパイル方式と②完全コンパイル方式と③インタプリタ方式に大別できる。

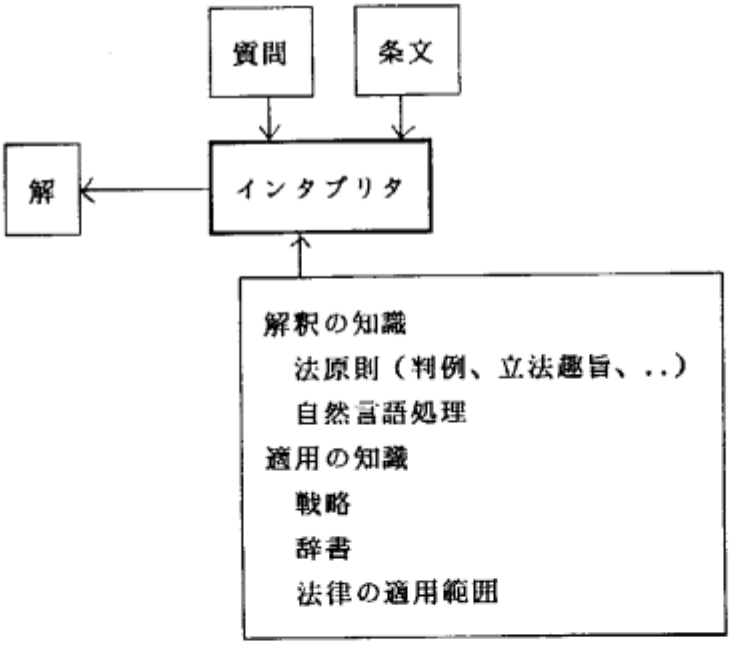
①は、法律を「解釈の知識」を用いて静的に解釈し、その解釈を使用して問題を解決するものである。解釈された条文の一部と一般的な法律適用の知識は一体となって「適用の知識」を構成する。適用の知識の一部は推論エンジンに対する制御の知識（メタ知識）としてはたらく。



②は、「解釈の知識」と「適用の知識」により条文を完全にコンパイルして知識ベースを構成するものである。適用の知識は個々の条文の解釈の中に埋めこまれるため、個々の条文の表現は原文に比べて複雑になることが多い。推論エンジンの負担は小さく、法律のための特別な機能は不要である。



③は、条文が知識ベースであり、「解釈の知識」「適用の知識」を用いて、条文を動的に解釈しながら問題を解決するものである。現実の専門家の法律解釈／適用は①と③のモデルの併用であると考えることができる。



現状の法律ＥＳでは専門家自身が①か②のコンパイラとなって、法を静的に解釈して、

整合をとり、体系的に整理したものを知識表現するのが現実的である。法を動的に解釈するモデル（法律解釈のdeep model)については3.5でふれる。

オブジェクトによる問題解決モデル

前述のように部分コンパイル方式では専門家が事前に条文を解釈し、解釈された条文賭適用の知識からなる２つの知識ベースを構築する。この方式をオブジェクト指向の言語（例えばＥＳＰ）で実現する場合の方法について考える。適用の知識として以下のものを考える。

(1)条文間の関係

「条文間の関係」とは、①法律Ｘが法律Ｙと一般法／特別法の関係にある、②条文Ａは条文Ｂの例外にあたる、③条文Ａと条文ＢはＡＮＤ／ＯＲの関係にあるなどの関係を言う。①のように法律ＸがＹの特別法である場合、Ｘの条文とＹの条文が矛盾したときには、Ｘの条文が優先される。また、Ｘに規定してない事柄でＹに規定した条文があればＹの条文が用いられる。従ってＹはＸのデフォールトの知識として考えることができる。②のように条文ＡがＢの例外にあたるときには、矛盾が生じたときにはＡが優先される。ＡとＢが例外関係にあるかどうかは条文の書き方（EX.　Ｂの規定にもかかわらず～である）から判断できることもあるが、専門家によらないとわからないものがある。③のように条文ＡとＢがＡＮＤの関係にあるとは、両方の前提条件を満たさなければ結論を導けないことであり、ＯＲの関係にあるとは、一方の条件を満たせば結論を導けることを言う。

(2)条文の使い方

「条文の使い方」とは、①条件Ｃの下では条文ＡとＢを適用すると有利である、②条件Ｃの下では法律Ｘが適用できる、③キーワードＤの関連条文はＡとＢとＣである、などの知識を言う。①のような戦略的知識は専門家の経験的知識を反映したものである。②の知識は法律の適用範囲の知識に対応したものである。

(3)辞書

用語の辞書として、権利には物権と債権がある、自然人と法人は行為能力者であるなどの概念間の関係が含まれる。

専門家の頭の中には、条文そのものだけでなく、これらの知識が整理されているはずであり、専門家はこれらの知識を用いて問題を迅速に解決しているのである。従って、

解釈された条文＝原条文をその使われ方を想定して論理的に解釈したもの

適用の知識　　＝専門家の頭の中で整理された条文に関する知識

と対応させれば、現実の専門家の思考に近い方法で問題解決が可能となる。

まず、用語の辞書をクラスオブジェクトとして構成する。１つの用語を１つのクラスに対応させ、概念間の関係（概念ＡとＢが上位／下位の関係にある、法律ＸとＹが一般法／特別法の関係にあるなど）をクラス間のリンクで表現する。

次に、専門家によってコンパイルされた知識（個々の条文の解釈）をクラスオブジェクトで表現する。１つの条文が１つのクラスに対応する。その条文に対する質問の応答はメソッドとして記述する。条文の記述は辞書にある用語のみを用いて記述する。例えば「ＡまたはＢならばＣである」という条文については、

```
inq(In,Out) :- (check(In,A);check(In,B)),Out=C.
conf(In,Out) :- check(In,C),(check(In,B);check(In,A)),Out=yes.
```

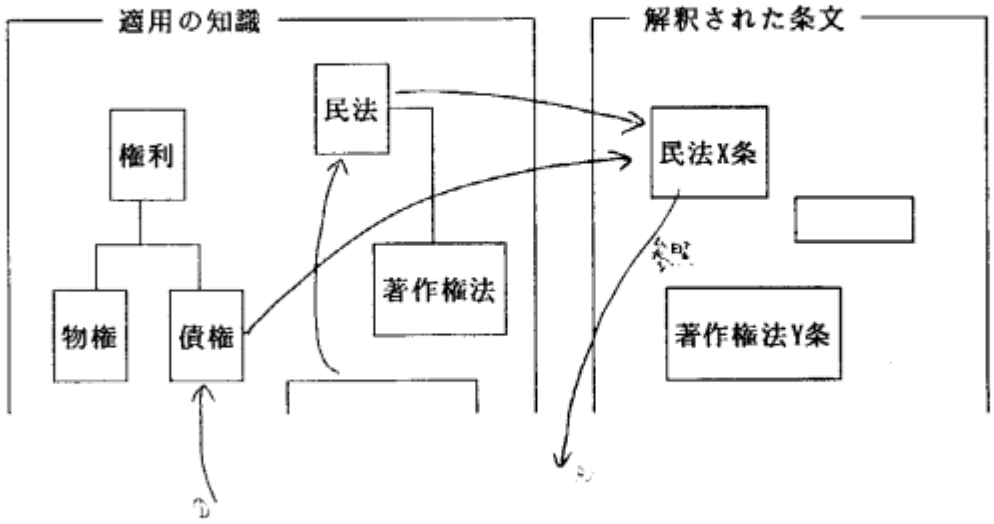
などのように記述される（記述は正確なものではない）が、この場合Ａ、Ｂ、Ｃは辞書になくてはならない。Ａが成立していることのチェックが失敗したとき、Ａが別の概念αの下位概念であるときには、αが成立しているか否かがチェックされる。

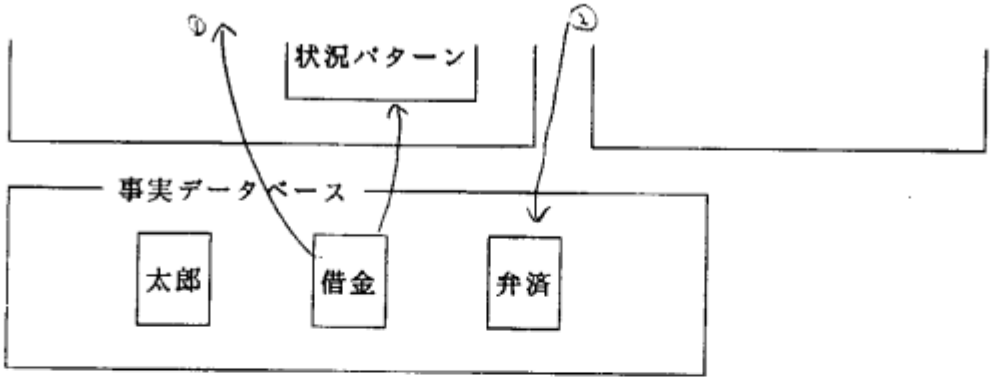
上記の(1)(2)のような条文検索のための知識の一部は、クラスの中でメソッドとして手続的に記述される。例えば、状況Ｘでは条文Ａが原則的規定、Ｂが例外的規定であるときには、Ｘというクラスのメソッドとして

```
refer(Cond) :- refer(Cond,B),!.
refer(Cond) :- refer(Cond,A).
```

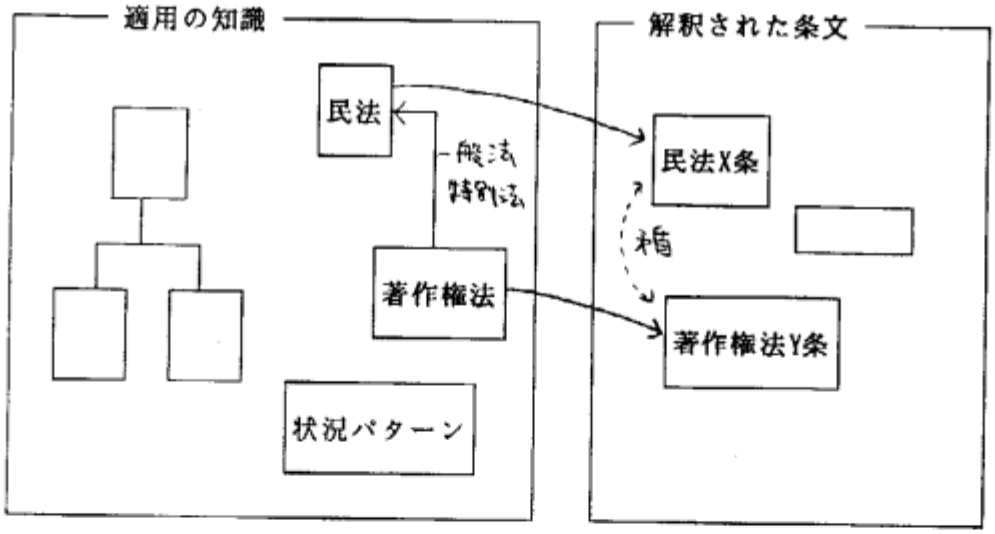
のように記述される。

問題を解決する手順は以下のようである。まず、事例を事実データベースに記述する。事例はインスタンスオブジェクトの集合として表わされる（個々のインスタンスに対応したクラスオブジェクトは必ず知識ベースの辞書に含まれていなければならない）。人や物や権利などの概念や行為はそれぞれ１つのインスタンスとして表わされる。ある行為の法律的効果を調べるには、その行為にチェックメッセージを送ると、その行為のメソッドに書かれている手続的知識を用いて関連条文を検索し、それと同時にその状況に対応したクラスのメソッドにより、適応する法律を決定する。関連する条文と適応する法律が合致したものが選択される。（下図で１つの四角が１つのオブジェクトに相当する）。選択された条文が満たされているか否かは事実データベースのオブジェクトの内部状態で判断される。





条文間に矛盾があったときは、適用の知識の中のメソッドにおいて条文の優先度を手続的に埋めこむことによって対処することができる。また、前述したように法律のチェックにおいて直接の関連条文だけでなくその上位法を自動的に参照する機能を付加すれば、法原則を用いた推論がある程度可能となる。特別法と一般法の間の矛盾については以下のように適用の知識の中で優先度を示すことができる。同様に、条約の効力が憲法によって国内法に優先することや、外国人については法例（という名の法律）によってどの国の法律が適用されるかが決定されることなどは、適用の知識において法律間の関係を利用して占めされる。

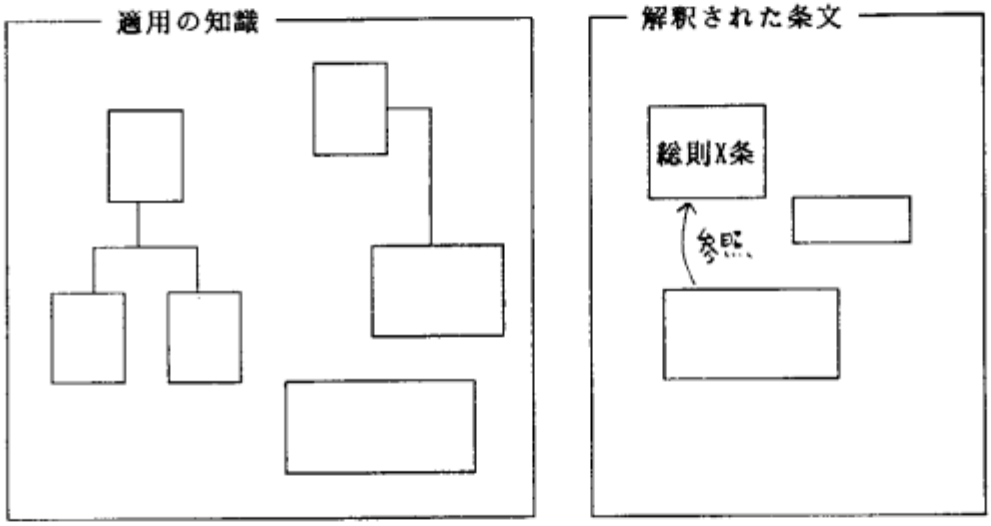


条文の検索問題においては、適用の知識のキーワード間のリンクをたどることによって関連条文を網羅することができる。この場合、条文の間の関係を適用の知識内で十分に整理しておく必要がある。また、解答パターンがあらかじめ定型化しているもの（EX.権利関する質問は一般に性質、主体、客体、発生、消滅、変更、効力などの項目に整理して答えれば良い）については下図のようにオブジェクトとして適用の知識に用意しておき、必

要に応じて呼出すことになる。例えば物権と債権の相違について答えるには、テンプレートを利用して物権と債権から必要な情報を抽出し、その内容を比較すれば良い。

権利関係テンプレート	物権	債権
性質	性質	性質
主体	主体	主体
・・・	・・・	・・・

法律の中には、その最初の部分に「総則」とよばれるいくつかの条文がある。これは、その法律を解釈するための指針や重要語句の定義などを与えるものである。解釈の指針となる部分は、専門家が法律を解釈するためのものであって、他の条文の解釈の中に埋めこまれるものがあり、単独では知識表現しにくいものがある。重要語句の定義については下図のように定義を直接参照し、その定義に従って事実データベースを参照する。



まとめ

法律の問題解決モデルについて考察した。法的推論を行うためには条文以外の知識が必要である。この知識を専門家がどのように利用しているかは、専門家自身も十分に認識していないのではないかと思われる。法律にはさまざまな種類があり、法律的な議論をしなくとも、面倒な計算式や拘束条件のチェックだけで処理できる法律がある。このような法律においてはＥＳが容易に構築でき、しかも、実用性があるものが作りやすい。しかし、法律ＥＳの発展のためには問題解決モデルの研究は最も重要な課題である。

参考文献

[1]井上：「現代法」、NHK市民大学叢書
[2]新田：「法律の知識表現に関する一考察」、情報処理学会35回全国大会

２．２　画像処理における問題解決モデル

２．２．１

（１）推論方式の決定方法の現状と問題点

エキスパートシステムの概要がある程度明確になると、次は知識をどのような構造で体系化していけば良いか、また目的とする出力を得るには、どのような推論方式で行なえば良いのか検討を行なう。しかし、現状のアプローチは使用する言語やツールからくる制約から知識表現、推論方式が決まる場合が多い。これは、システム工学的見地からは本末転倒のアプローチである。このようなアプローチをせざるを得ない理由として以下のことが考えられる。

1)推論方式を決めるための判断基準が無い

2)言語、ツールの知識表現、推論方式が画一化されているため、トップダウン的なアプローチをしても最終的にはほぼ一義的に決まってしまう。

基本的には1)に起因するところが大である。なぜならば、言語，ツールが画一化している理由も、どのような知識表現、推論方式を提供すればよいかを分析する時に1)の問題に直面するからである。

推論方式決定のための判断基準とはシステムの要求仕様が与えられた時、それをサブ問題に分解するためのテンプレートが存在することである。このように推論方式に裏付けされたサブ問題の関係を示すテンプレートがあることにより「Ａ問題はＡ１問題とＡ２問題との複合問題であると考えられる。従ってＡ１問題においては、このような推論方式があり……」というように、問題を構造化することができる。このように推論方式のイメージを与えることが、知識表現，推論方式の決定段階において有効である。

以上のように問題を定式化し、パターン化していく研究の成果は、最近ようやく顕在化しはじめている。その代表的なものに Generic Tasks[Chandrasekaran 86].[Broan 86], Heuristic Classification[Clancey 85], Ontological Analysis[Alexander 86]などが提案されている。また、国内においても色々な研究が成され始めてきている。[JIPDEC 87]

（２）設計問題における問題解決モデルの特徴

設計問題は、B.Chandrasekaranによると３つのクラスに分けられている。[Chandrasekaran 86]

Class 1 ：設計対象の構成要素すら分からないレベル

Class 2 ：設計対象の構成要素は分かっているが、まだ手本となる事例がないレベル

Class 3 ：設計対象の構成要素は分かっており、設計の各段階において手本があるレベル

この分類に従って企業内における設計業務を見てみると、多くのものがClass 3 のいわゆる「ルーチン設計」に含まれる事がわかる。

また、問題解決パターンでみると選択型問題と合成型問題の両方を包含した問題解決パターンになっている。簡単な設計問題では、設計の目標、制約条件等から前例を選択するという選択問題のみで問題解決が行なえる。しかし、多くの問題では、前例を少し修正・変更し、目的とする対象の設計を行う。この修正・変更を行うパラメータをどのように設定すれば良いかが問題となる。つまり、パラメータの値の組み合わせと、その段階の目標、制約条件等から、目的に合った解を得る合成問題に帰着する。

設計問題における他の特徴は、各段階において、様々な実行タスクが介在することである。この介在は次の様な時に発生する。

・パラメータ設定のための処理
・推論結果の検証ための処理

以上のような特徴を設計問題は有していると考えられる。

２．２．２　画像処理エキスパートシステム(EXPAIN)の概要［末田86］

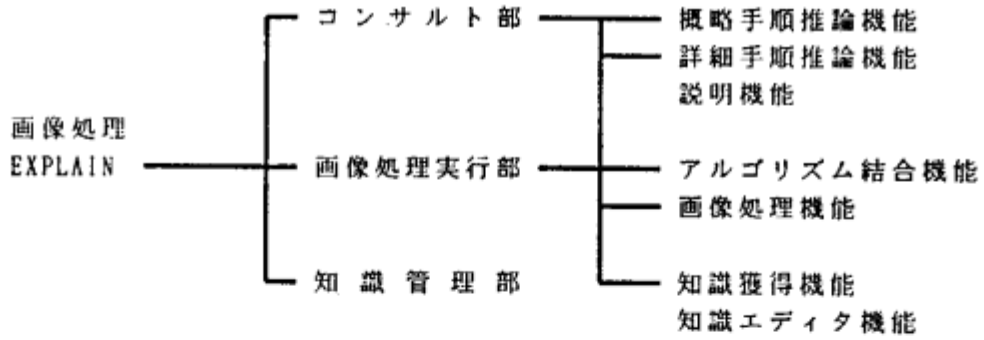
（１）システムの目的

画像処理の初心者が、抽象的処理要求（例：画像切り出し、画像復元etc ）と、対象画像属性（例：ノイズの有無、画像の濃淡、etc ）を与えることにより、抽象的処理要求を満す画像を得るための画像処理コマンド列を抽出するものであり、画像処理アルゴリズム開発の生産性を向上させることを目的としたシステムである。

注）（EXPLAIN:EXpert system of image Processing Logic Algorithm INference)

（２）　システム構成

ＥＸＰＬＡＩＮ以下の構成になっている。



＜コンサルト部＞

(a) 概略手順推論機能：原画像の画像属性を入力することにより、その画像に適した概略手順を推論する、いわゆるメタ推論を行う。これは次に行う詳細な手順を推論する際に不必要な推論を抑え推論効率を高める役割を果たしている。

(b) 詳細手順推論機能：概略の手順に沿って詳細な手順を導出するための推論である。システムは、推論に必要な画像属性を利用者に問い合わせながらアルゴリズムを選び出す。選択されたアルゴリズムは画像処理実行部で実行され、その処理効果は利用者に問い合される。この際の処理効果は、適用されたルールに記述されており、問合せの結果、処理画像が提示された効果を満たしていれば、次の手順の推論を行い、満たしていない場合は、処理を前の段階に戻し別の手順を選択するために再推論を行う。

(c) 説明機能：手順を導出した理由や、ユーザに対してシステムが問い合わせている根拠について説明するものである。また現在の処理の位置付け、推論トレースもユーザは必要に応じて見ることができる。

＜画像処理実行部＞

(a) アルゴリズム結合機能：画像処理を実行するために必要な情報を収集し、推論過程で実行される画像処理の件属性を連続性が保たれるように実行環境を整える。つまり、以前に実行した処理結果値との自動パラメタ結合や、推論時バックトラックが発生した際の処理対象画像や各処理情報の復元などを行う機能である。

(b) 画像処理機能：画像処理を実行する部分であり、アルゴリズム群からなっている。

＜知識管理部＞

(a) 知識獲得機能：知識獲得の方法は、専門家に画像処理を実行させ、システムは、その試行錯誤を過程をトレースし、最終的に絞られた手順の着眼点を専門家との質疑応答方式によって得ることにより、知識を生成していく方式をとっている。

(b) 知識エディタ機能：すでに格納されている知識の修正、削除を行うものであり、知識として表現されている属性との整合性をとるために、メニュー方式で編集する。

２．２．３　ＥＸＰＬＡＩＮの構造

ＥＸＰＬＡＩＮを例にとり、その構造を分析してみる、ここで分析方式として、「知的情報処理システムに関する調査研究報告書」［JIPDEC87］で述べられている分析方法（以下「小林方式」と言う）と、B.Chandrasekaranが提唱するGeneric tasks の１タスクである" Hierarchical Derign by Plan Selection and Refinment"（計画選択・精密化による段層設計）方式の２方式に、基づいて検討を加える事にする。

(1) 小林方式による分析

a)　小林方式の概要

小林方式によると、問題を構成する要素として「システム構造」、「構成要素特性」、「システム特性」の３つに分け、これらがgiven か unknownかで問題のタイプが決まるとしている。

解析問題	…システム構造	(given)
	構成要素特性	(given)
	システム特性	(unknown)
合成型問題：クラス1	…システム構造	(unknown)
	構成要素特性	(unknown)
	システム特性	(given)
合成型問題：クラス2	…システム構造	(unknown)
	構成要素特性	(given)
	システム特性	(given)
合成型問題：クラス3	…システム構造	(given)
	構成要素特性	(unknown)
	システム特性	(given)

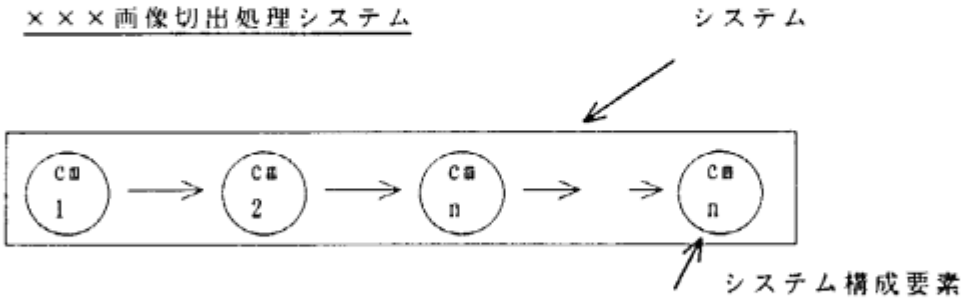
合成型問題（クラス 3）が Chandrasekaran の階層的設計のgeneric task （ルーチン設計）対応する。

b) EXPLAINの特徴

システムの構造　(unknown)
構成要素特性　(unknown)　→　合成クラス1
システム特性　(given)

＜目標システム＞

×××画像切出処理システム



システムの構造　→コマンドの並び
構成要素　→コマンド群
システム特性　→原画像を入力すると、対象物が切出されて表示される。

このようにEXPLAIN が合成クラス1 にもかかわらず、解けているのは、構成要素特性を導出する際の順列が結果的にシステムの構造になっているためである。従って本質的には、合成クラス3 の問題と、とらえる事ができる。

c) 小林方式による基本タスクとの対応

基本タスク	本問題における表現	実現機能
1.構成要素とその関連の表現	・構成要素は、画像処理モジュールである。 ・構成要素の関連 <関連1> 構成要素の並びが手順に相当する。 <関連2> 処理コマンドのパラメータの入出力関連	・手順の知識表現 ・コマンドの知識表現
2.設計問題の階層的表現	・処理モジュールに階層構造がある。 ・階層的な構造は陽には表現していない。	・ルールセット（モジュール化） ・トップダウン精密化機能
3.代替案の自動生成	・仮説ゴールを検証するめにサブゴール（サブ仮説）を立てる。仮説が否定されると他の候補を選ぶ。	・世界管理（コンテクスト管理） ・仮説推論
4.部分システムの評価	・処理結果を表示し、評価は、オペレータに委ねる	・ユーザ対話処理機能 ・結果表示機能
5.上位レベルの知的バックトラック	・バックトラックは行っているが、知的バックトラックはしていない縦型探索におけるバックトラック	・世界管理（コンテクスト管理） ・仮説推論 ・（ATMS）

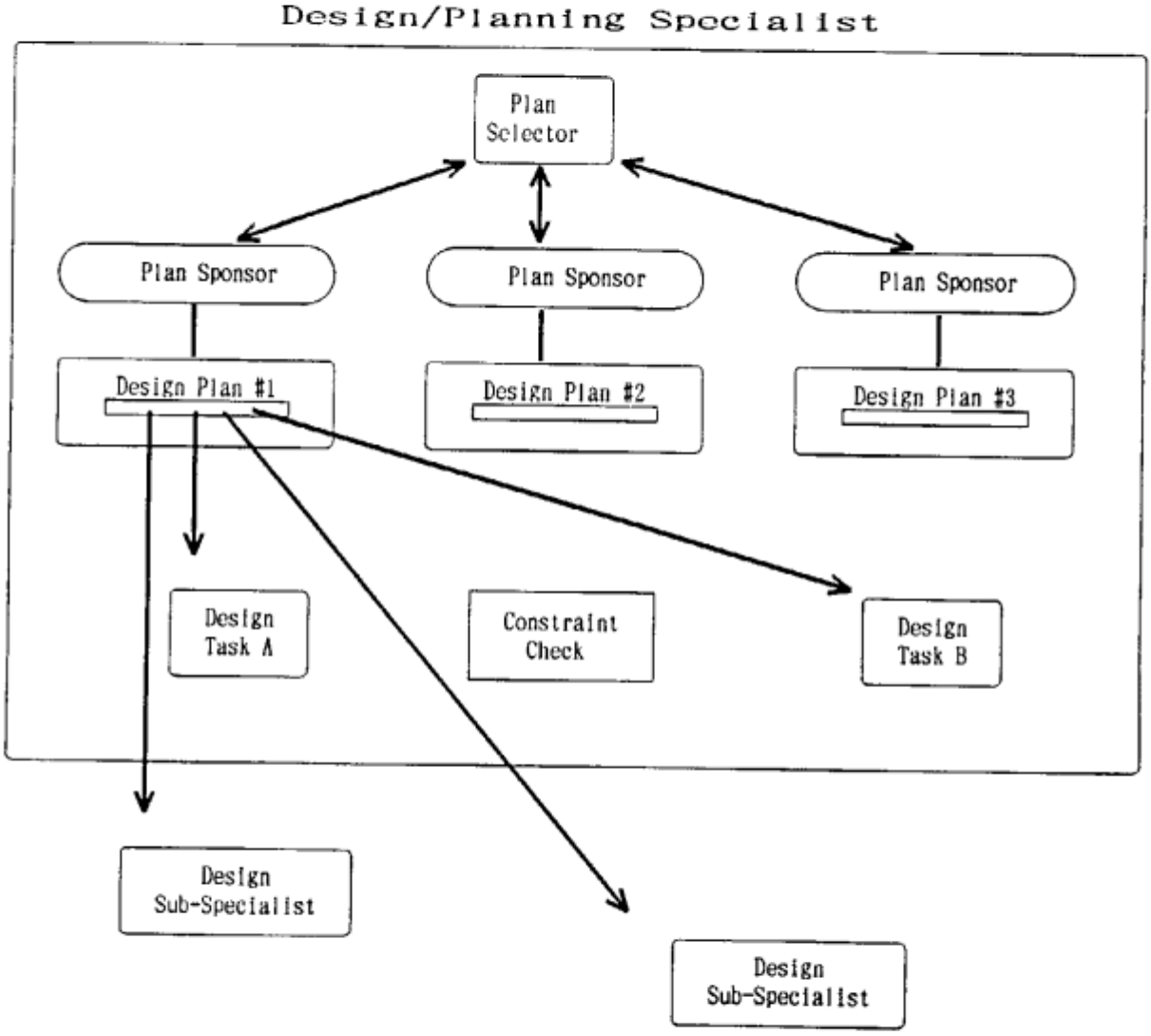
２．２．３　設計型問題解決機能の一考察

（１）基本タスクの考え方

基　本　タ　ス　ク	本問題における表現	実　現　機　能
6.設計事例の検索と利用	・手順の知識そのものが事例を対象としている。 ・事例はルールセットになっている。	・Constraintによるパターン ・マッチ
7.並列的問題解決	・取り扱っていない。	—

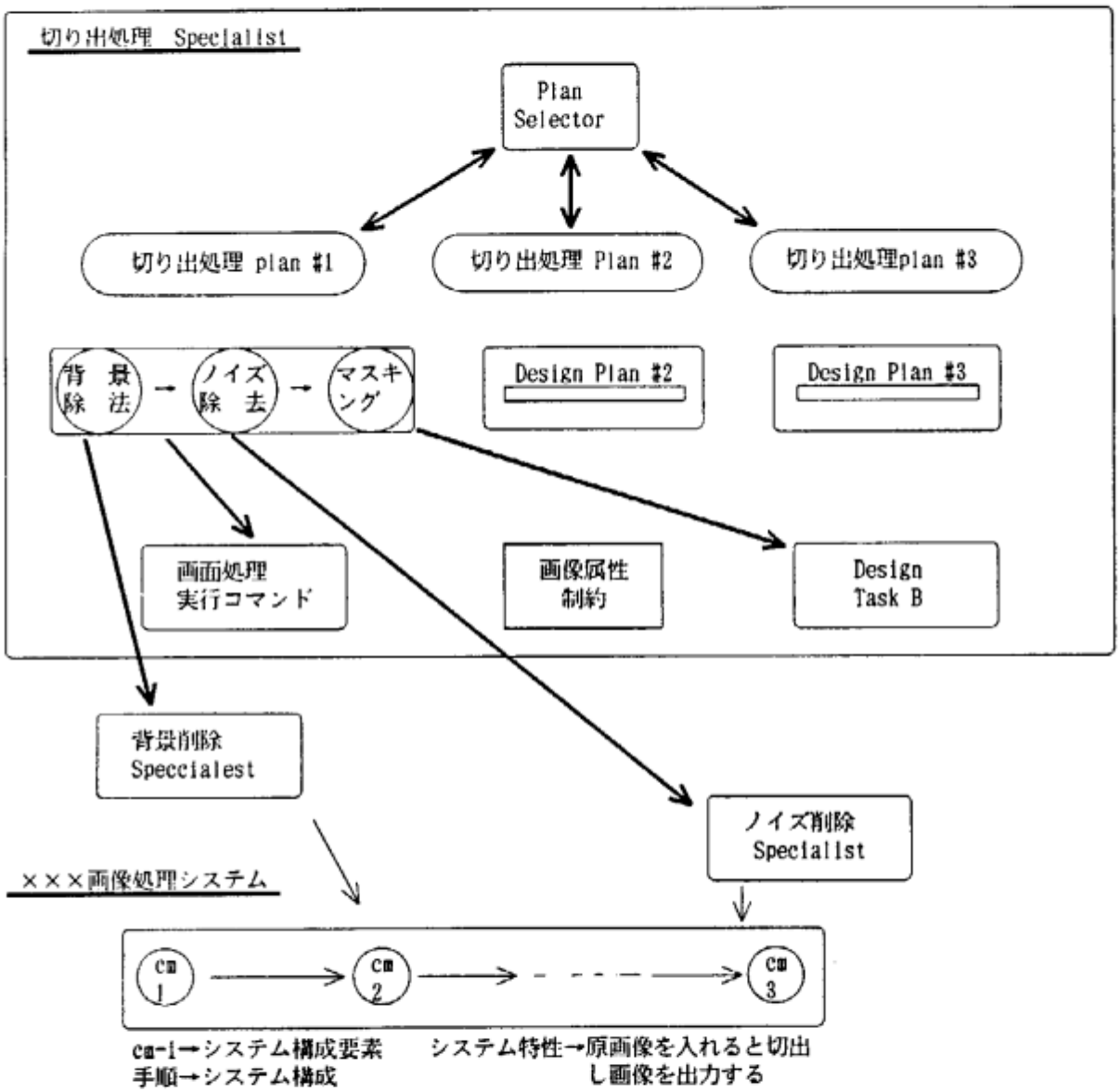
（２）Chandrasekaran　方式による分析

（１）計画選択・精密化による階層設計とは、計画選択・精密化による階層設計方式と図式化すると以下の様になる。



（２）EXPLAIN への適用

上記タスク構造をEXPLAIN適用すると以下のようになる



ルーチン設計という概念でとらえると、本タスクへの親和性は、かなり良いものであるとえる。

２．２．３　設計型問題解決機能の一考察

（１）基本タスクの考え型

"計画・選択・精密化による階層設計" タスクを1 つのgeneric タスクにするには、少し大きすぎる。なぜならば、このレベルの情報をエキスパートシステム設計者に与えたとしても、推論方式、知識構造等を設計するうえでは、その枠組みが抽象的過ぎるからである。そこで、もう少しプリミティップなレベルの機能をモジュールと考え、これを基本タスクと呼ぶ事にする。
基本タスクであるための要件は以下の点である。

1)機能的に独立している事

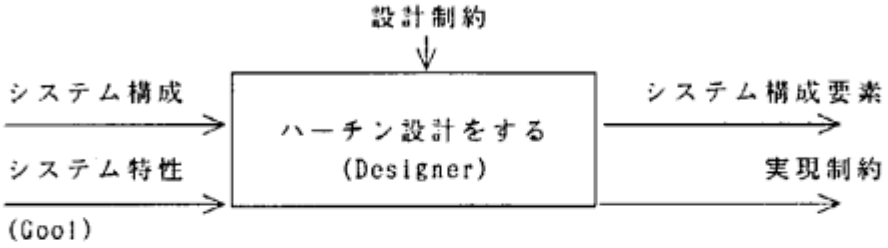
2)他の問題においても共通して使える事

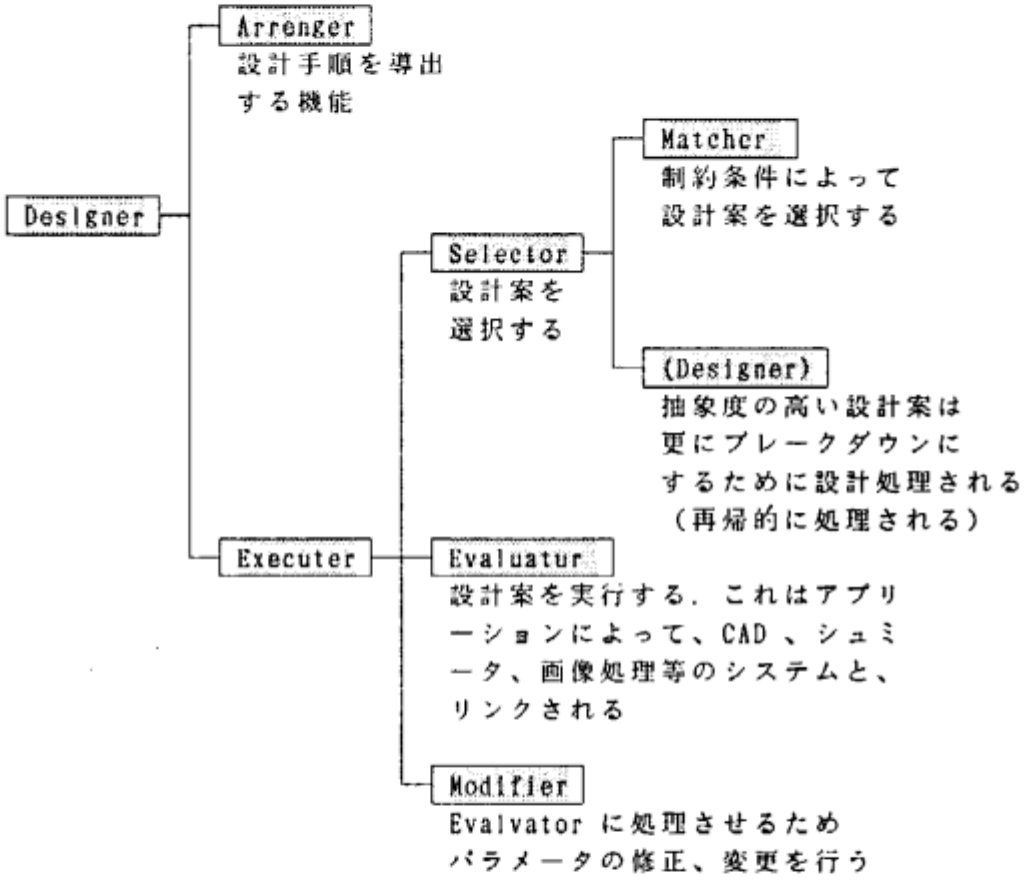
3)あまり抽象的過ぎず、かつ、あまり具象的過ぎない事

3)に関してはそのレベルの線引きは非常に難しいが機能分析した結果、ブレークダウンが2 ～3 レベル程度が目安になると考えられる。

（２）設計問題機能分析

設計問題と機能分析を決案の事例を以下に示す





この様に機能分析を行い、問題を定式化することによって、エキスパートシステム開発者に有効なテンプレートを伝えることが、期待できる。

［参考文献］

[Chandrasekaran 86]

B.Chandrasekaran: Generic Tasks in knowledge Based Reasoning :High-Level Building Blocks for Expert System Design. IEEE EXPERT. vol.1. no.3. pp.23-30.Fall 1986

[Brown 86]

D.C.Brown and B.Chandrasekaran:Knowledge and Contorol for a mechanical design Expert System. IEEE COMPUTER. vol.19.no.7.pp.92-100.July 1986

[Clancey 85]

William J.Clancey: Heuristic classification. Artificial Intelligence.

[Alexander 86]

James H.Alexander: Knowledge Level engineering :Ontological Analysis.AAAI '86.pp.963-968

[JIPDEC 87]

知的情報処理システムに関する調査報告書－知識システム開発方法論－ICOT-IPDEC AI センター（昭 62)

[末田 86] (1977)

末田，屋：画像パッケージ利用におけるプログラム設計支援システム．
日経データプロ．pp.1 35-154(1986)

2.3 目標間の関係に着目した分散協調問題解決

広島大学 工学部
小野 典彦

1. はじめに

われわれの周囲には単独で解決するよりは集団で解決した方がはるかに効果的に解決できる問題が数多く存在する．このように問題の大きさや難しさを解消することを目的として集団で解決される問題とは別に，問題の性質上，必然的に集団で解決せざるを得ない問題も少なくない．

分散協調問題解決とは，このように協調し合う知識システムの集団を採用することによって，単独の知識システムでは解決が困難なあるいは事実上不可能な問題を効果的に解決することを目指した研究分野といえよう．

分散協調問題解決には主としてつぎの2つの流れがある．

(1) 分散的あるいは並列的な問題解決：複数の知識システムにより問題解決を並列的あるいは分散的に行なわせることによって，単独の知識システムでは達成し得ないような信頼性および高速性をもった知識システムを構築すること，すなわち，たとえ単独の知識システムにより解決可能な問題であっても，それをあえて分散的あるいは並列的に解決することによって，飛躍的な問題解決能力をもった知識システムを構築することを目指した研究である．

(2) 分散的環境下での協調的な問題解決：複数の知識システムが地理的に分散されており，それらが協調し合わないかぎり，それら全体に課されている問題は解決し得ないという制約の下での分散的かつ協調的な問題解決機構を確立すること，すなわち，種々の制約から，地理的に分散された知識システムの集団によって解決せざるを得ない問題を効果的に解決するための機構を確立することを目指した研究である．

前者のような研究が要請されていることはいうまでもない．実際，分散協調問題解決に関する研究の多くは，知識システムの高性能化を主たる目的として展開されている．しかし，これは知識システムよりはむしろ計算機科学における分散処理や並列処理に関する研究と考えた方がよい．知識システムに関する研究という立場からは，後者のように問題解決を分散的環境下で展開せざるを得ないという制約の下での問題解決機構を確立していく

べきであろう．

本報告では，このような立場から特に分散的環境下での協調的な問題解決に焦点をあて，これまでに試みられてきた分散協調問題解決の限界を浮き彫りにすると共に，それらを解消するための方策の１つとして，分散された知識システムがもつ目標群の間の関係に着目することを提案する．

２．分散協調問題解決とは

分散協調問題解決とはいかなる型の問題解決なのかあらためて定義しておこう．ここでは，議論を分散的環境下での問題解決にしぼるため，Smith［Smith et al. 1981; Smith 1984］にしたがってつぎのように定義する．

分散協調問題解決とは**分散され**（decentralized），**疎に結合された**（loosely coupled）**知識源**(knowledge source)の集団による**協調的な**(cooperative)問題解決である．ここで，知識源とは何らかの知識表現に基づいて記述された知識システムであって，異なるプロセッサ上に実現されているものとする．これらが協調するのはどの知識源も全体の問題を解決するために必要な情報や情報処理能力をもたないためである．これらが分散されているとはこれらの間に大域的な制御や大域的なデータ格納場所が存在しないことをいう．これらが疎に結合されているとは各知識源が時間の大部分を通信よりも計算に費やすことをいう．

このような分散協調問題解決は随所で行なわれている．例として，タクシー会社の業務活動，すなわち，その本部とタクシーの集団とが通信を介して分散的かつ合目的に展開している業務活動を考えてみよう．この活動は，この会社の本部およびタクシーの各々を知識源とみなすことによって，分散協調問題解決と考えることができる．

実際，これらの知識源は地理的に分散されているばかりでなく，単独ではこの会社の業務活動全体を達成することはできない．本部にとって根本的に欠けている能力とは，いうまでもなく利用者をその目的地に向けて運搬し，報酬を得るという能力である．この他，特定のルートにおける事故，工事，渋滞などの有無，あちこちに設置されている乗車場における利用者の濃度など，大局的な状況を分析するために必要な局所的情報を収集する能力にも欠けている．

一方，タクシーは，単独ではこの会社に依頼される業務活動の全体を達成することはできない．電話予約に対するサービスなどは特に難しい．これは本部と分散されたタクシーが共同することではじめて達成し得る業務活動である．また，大局的な状況分析能力に欠けるため，配車の不足している乗車場や現在の目的地にいたる最適なルートなどを単独で

知ることは難しい．

さて，上のような定義に加えて，知識源は目標に基づいて活動すると仮定する．すなわち，知識源は，複数の目標をもっており，それらのいくつかに焦点をあて，それらを同時に達成し得る詳細なプラン（行動系列）を生成し，それを実行するものとする．特に，知識源がそれ自身の目標を達成するための活動（それが実行する行動系列）を**個人的活動**，他の知識源の目標を達成しやすくするための活動を**協調的活動**，さらに他の知識源の目標を達成し難くするための活動を**敵対的活動**と呼ぶ．また，協調的および敵対的な活動をあわせて**社会的活動**と呼ぶ．

3．分散協調問題解決のための諸モデルとその問題点

分散協調問題解決を主たる目的として数多くの計算モデルが提案されている．しかし，これらは著者らが目指すような分散協調問題解決のための枠組みとはいえない．

例えば，黒板モデル［Nii 1986］やオブジェクト指向並列計算モデル［Yonezawa et al. 1986］は，分散的環境下での問題解決よりはむしろ分散的あるいは並列的な問題解決を目指したものと考えた方がよい．

一方，契約ネット［Smith et al. 1981］のように分散された知識源の間の通信規約を規定した枠組みは，知識源の間の協調的活動をモデル化するためのインタフェースを定めたものに過ぎないことから，つぎのような問題点をもつ．

(1) **個人的活動と協調的活動を統一的に扱えない**：これらの枠組みでは，知識源の各々が，その個人的活動と協調的活動とを別々の枠組みの下で行なうことを意図している．しかし，個人的活動と協調的活動とを別の機構に基づいて展開するのは望ましくない．一般にこれらの活動には重複が認められるのが普通であって，それらを別々に行なうのは冗長だからである．例えば，タクシーが“本部より依頼された業務を請う”という活動は，本部の目標を達成しやすくするための協調的活動であると同時に，タクシーがこれを依頼される以前からもっていた“利用者を獲得する”という目標を達成するための個人的活動でもあることに注意されたい．

(2) **協調的活動を行なうべき状況を認識しえない**：これらのモデルでは，各知識源が，他と協調すべき状況をいかにして表現し，それに照合する状況をいかにして認識したらよいかは規定していない．すなわち，知識源は，他の知識源から陽に協調的活動を依頼されないかぎり，協調すべき状況を認識しえない．

(3) **敵対的活動を扱えない**：知識源の間の敵対的活動は扱えない．また，協調的活動の場合と同様，知識源はそれが敵対的活動を行なうべき状況を認識しえない．

(4) **通信系統の完全性を仮定している**：これらのモデルでは知識源の間の通信が完全であることを仮定している［Genesereth 1987］．しかし，通信が途絶した場合でも，各

知識源が状況の変化を認識することができ，そのような変化に対して通信できない他の知識源がいかに行動するかを推測できるならば，知識源は互いに合目的に振る舞うことができる．これには知識源が共通の行動原理にしたがって活動するか［Ishii et al. 1987］，各知識源が他の知識源の行動原理を徹底的に獲得しておけばよい．

4．分散協調問題解決のための新たなモデルを求めて

4．1 新たなモデルへの要求仕様

将来の分散協調問題解決システムは，前節で列挙した諸問題を解決し得るモデルに基づいて構築すべきであろう．そのようなモデルとは，つぎの質問に対して明確な説明を与え得るものでなくてはならない．

(Q-1) **社会的活動の必要性をいかにして認識するか**：すなわち，知識源は個人的活動，協調的活動，敵対的活動の間の切り替えを行なうべき状況を何に基づいて表現し，それをいかにして認識すべきかという問題である．

(Q-2) **社会的活動をいかにして計画するか**：すなわち，社会的活動の必要性を認識したなら，その状況にふさわしい活動の計画をいかにして組み立てるべきかという問題である．

(Q-3) **個人的活動と社会的活動をいかにして統括するか**：すなわち，社会的活動のみならず個人的活動についても，その必要性を認識し，それを計画するための基礎となり得る概念とは何かという問題である．

(Q-4) **他の知識源の活動をいかにして推測するか**：すなわち，通信系統が完全であろうとなかろうと，何に基づいて他の知識源の活動を推測すべきかという問題である．

4．2 目標間の関係と分散協調問題解決

(Q-1) ～ (Q-4) の問題に対する１つの方策は，知識源がもつ目標間の関係［Wilensky 1981, 1983; Carbonell 1981］，すなわち，つぎのような関係に着目することであろう［Ono et al. 1987］．

(1) **目標の重複**（goal overlap）：単独の知識源がもつ目標の間の関係であって，それらを別々に達成するよりは同時に達成する方が容易であることをいう．例えば，あるタクシーが特定の乗車場に向かう際に燃料が不足していることに気がつき，途中にあるガソリンスタンドに寄ったとしよう．これなどは“乗車場に向かう”という目標と“燃料を補給する”という目標の間の重複を利用した例である．

(2) **目標の一致**（goal concord）：異なる知識源がもつ目標の間の関係であって，これらは知識源の各々が単独で達成するよりは互いに共同することによって同時に達成する方が容易であることをいう．一般に，電話予約を受けたタクシー会社の本部と利用者を待

ち続けているタクシーの各々は互いに一致した目標をもつことになる．

(3) 目標の衝突（goal conflict）：単独の知識源がもつ目標の間の関係であって，それらの一方を達成しようとすると，他方を達成することが難しくなることをいう．例えば，一日の業務の終盤にさしかかったタクシーは“利用者を獲得する”という目標と“他のタクシーの業務を妨害しない”という衝突し合う目標をもつことになる．

(4) 目標の競争（goal competition）：異なる知識源がもつ目標の関係であって，知識源の一方が目標を達成しようとすると，他方が目標を達成することが難しくなることをいう．例えば，同じ乗車場に向かっているタクシーの各々は互いに競争し合う目標をもつことになる．

これらの関係に着目するならば，(Q-1) ～ (Q-3) はつぎのように説明される．

(A-1) 知識源が社会的活動の必要性を認識するのは，それ自身の目標と他の知識源の目標との間に一致あるいは競争を認識した場合である．すなわち，①相手との間で協調的活動を実行し合うことによって，重要かつ単独では達成が難しい目標を互いに達成し得ること，あるいは②敵対的活動を実行しなければ，相手側の敵対的活動によって，重要な目標が干渉を受け得ることを認識した場合である．

(A-2) 他の知識源との間に目標間の一致あるいは競争を認識した知識源は目標間の一致を生かすことを目標として，あるいはそれらの間の競争を解消することを目標として社会的活動を計画する．すなわち，①お互いの目標を同時に達成するのに必要な活動，あるいは②相手の敵対的活動を不可能とするに十分な活動を組み立てようとする．

(A-3) 知識源の個人的活動は，それ自身の目標の間の重複や衝突に着目することによって，その必要性が認識され，計画される［Wilensky 1983］．すなわち，個人的活動を社会的活動と同様の枠組みの下で扱うことができる．

(A-4) 他の知識源の活動は，相手が種々の状況の下で発生し得る目標についての知識および相手のプランニング知識（相手がいかにして目標間の関係を認識するか，さらにその関係に基づき相手がいかにしてプランニングを行うかという知識）を互いに共有し合うことによって推測される．

5．目標間の関係に着目した分散協調問題解決の試み

著者らは，目標間の関係に着目して分散協調問題解決を行なう知識源の枠組み DAISY（a Framework of Decentralized AI Systems）を設計すると共に，タクシー会社における特定のタクシーの活動をこの枠組みによりモデル化し，その活動のシミュレータおよび説明系を試作している［Ono et al. 1987］．

この知識源はつぎの構成要素よりなる（図1）．

(1) Situation Space (SS)：知識源が置かれた世界および知識源の内部状態を記録するためのデータ格納場所である．

(2) Planning Space (PS)：知識源の目標およびそれを達成するためのプランを階層的に展開し，記録するためのデータ格納場所である．PS には，後に述べるプランニング戦略や他の知識源の目標なども記録される．

(3) Goal Detector (GD)：SS の内容を常時監視しており，知識源が達成すべき目標を認識すると，それを PS に記録する．GD は SS の内容の変化に反応して発火するプロダクションルールの集合として実現されている．

(4) Goal Guesser (GG)：SS の内容を常時監視しており，他の知識源がどの目標を達成するためにどのプランを実行しているのかを推測し，それを PS に記録する．

(5) Meta Plan Proposer (MP)：PS 上に記録される目標の間の関係に着目して，プランニング戦略を生成し，それを PS に記録する．

(6) Plan Proposer (PP)：PS に記録されたプランニング戦略にしたがい，詳細プランを作成し，それを PS に記録する．

(7) Plan Executor (PE)：PS に記録された詳細プランを実行する．すなわち，詳細プランにしたがい，SS1を更新する．

(8) Knowledge Base (KB)：知識源のプランニング知識の格納場所である．

著者らがシミュレーションを試みているのは，図2のような格子状の交通網が張りめぐらされた地域を拠点とするタクシーによる個人的および協調的な活動である．図において，点線および（R1 などの）名称の施された経路は路上からの乗車率が高い経路を示し，②，③などの数値は経路を通過するのに要する時間的なコストを示す（これらは時間帯に応じて変化する．また，外部より動的に変更することもできる）．

このシミュレーションは，上のような知識源の枠組みに基づいてモデル化された特定のタクシーの活動だけを対象としたものであって，通常，このタクシーは個人的活動だけを行なう．しかし，シミュレータとのインタフェースを通してこのタクシーの Situation Space の内容を書き換え，それが本部や他のタクシーと協調すべき状況を設定してやると，このタクシーはそれを認識し，その状況にふさわしい協調的活動の計画を組み立てようとする．

このシミュレータに特徴的なことは，タクシーの個人的活動とはそれが単独で行なうタクシー業務だけではないということである．このタクシーはきわめて個人的な目標も発生し，それらの達成を試みるようにモデル化されている．例えば，このタクシーは食事すべき時間になれば食事をとろうとするし，銀行や郵便局へ寄るといった用件があればそれらが閉ってしまう前にそれを済ませようとする．

このタクシーによるプランニング過程の概略を示すために，図2における格子点 (2,3) にて利用者を降ろしたばかりのタクシーの活動をトレースしてみよう（図3）．

まず，このタクシーの Goal Detector は新たに“G1：タクシー業務を遂行する”という目標を発生し，それを Planning Space に記録する．目標によっては，その達成の前に予め達成しておかなければならない前提条件となる目標が付随していることがある．G1 はこのような目標の1つであって，“G2：利用者を獲得する”という目標が前提条件として付随している．Goal Detector は G1 のような目標を Planning Space に記録する場合には，その前提となる G2 もまた Planning Space に記録するように設計されている．

さて，このタクシーは“G3：手紙を出す”および“G4：のどの渇きをいやす”という2つの個人的な目標を既に発生し，Planning Space に記録していたとしよう．このように Planning Space の内容が更新されると，Meta Plan Proposer が呼び出され，Plan Proposer はその実行を待たされる．このとき，Meta Plan Proposer は複数の目標があることを認識してつぎのように振る舞う．

(a) **目標の発生および推測**：Situation Space の内容を常時監視している Goal Detector および Goal Guesser の各々が，知識源が新たに達成すべき目標および他の知識源がもつと推測される目標を認識し，それを Planning Space に記録する．

(b) **主目標の設定**：現在の状況（Situation Space および Planning Space の内容）に応じて，まず第一に達成すべき目標を主目標として選択する．主目標の選択は，Situation Space の内容（この場合には，“のどの渇き”の程度，“手紙を出す”という用件の重要度，現在の時刻など）に依存するわけであるが，ここでは G2 を選択したとしよう（G2 の達成前に G1 が選択されることはないことに注意されたい）．

(c) **目標間の関係の認識**：主目標と一致または重複する可能性のある目標群を探索する．この場合には G3 と G4 との重複の可能性を認識する．これらの目標が G2 と重複し得ることは，それらを達成するためのプランがいずれも“空車状態でタクシーを移動する（Ptrans［Schank et al. 1977］）”という行動を含むことに着目して認識される．

(d) **プランニング戦略の生成**：認識された目標間の関係に基づき，プランニング戦略（ここでは，“G2 を主目標として，可能ならば G3 や G4 を達成せよ”なる戦略）を生成し，それを Planning Space に記録した後，Plan Proposer に制御を移す．

ついで，Plan Proposer さらには Plan Executor がつぎのように振る舞う．

(e) **詳細プランの生成**：Plan Proposer は，この戦略に基づき，いわゆる“生成検査”の手法を用いて，G2，G3，G4 を同時に達成し得る詳細プランを生成し，それを

Planning Space に記録し，Plan Executor に制御を移す．

(f) 詳細プランの実行：Plan Executor は生成された詳細プランを実行する（この結果，Situation Space の内容が更新される）．

なお，この例題に限った表現ではあるが，プランニング戦略は図4に示すような5項組で表わされており，試作したシミュレータはつぎのような詳細プラン（タクシーが採用すべき経路）を生成する．

(2,3)→(2,2)→(3,2)→(4,2)→郵便局A→(4,4)
→(5,4)→コンビニエンスストアB→乗車場E

さて，協調的活動の場合もこのような個人的活動と同様にして計画される．例えば，Plan Executor がこのプランを実行している間に，本部より“利用者からの電話予約”を知らせる放送があったとしよう．

この場合には Goal Gucsser が，この放送は“契約［Smith et al. 1981］により利用者を割り当てる”という（本部が採用した）プランの一部であること，さらに，そのプランが本部の目標“G5：利用者をタクシーに割り当てる”を達成するためのものであることを推測して，その旨 Planning Space に記録する（図3）．

するとやはり Meta Plan Proposer が呼び出され，G2 と G5 との一致を認識して，Plan Proposer に対してそれらを同時に達成できるかどうか調べるよう提案する．これらの一致は G2 および G5 がそれぞれ“利用者を得る（Atrans［Schank et al. 1977］）”および“利用者を与える（Atrans）”を一部とするプランをもつことから認識される．

Plan Proposer はこの提案にしたがって，G2 と G5 とを同時に達成し得るプラン“契約により利用者を獲得する”を見い出し，それを現在 Plan Executor が実行中のプランと比較し，前者の方が適当と判断すればプランを修正する．

このように目標間の関係に着目することは分散協調問題解決への新たな接近法を提供する．しかし，この方策を採用するためには，目標間の関係をいかにして認識するかという未解決の問題［Mostow 1985］をはじめ，解決すべき問題が山積していることもまた事実である．ここではこのうち特に DAISY に関連するものを挙げておこう．

前節で述べた（A-1）～（A-4）に基づく分散協調問題解決の枠組みである DAISY の問題点は，一般に知識源の各々がこれらにしたがって活動しているだけでは，①それらが施すべき協調的活動を網羅しているとはいえないこと，および②場合によってはそれらの活動が必ずしも協調的とはいえないことである．

例えば，知識源が協調すべき状況とはそれらが目標の一致を認識したときだけに限られるわけではなく，問題によってはむしろ知識源が他との間に目標の競争を認識したときこそ，それらが協調すべき状況になっているからである（これらの知識源が競争している目

標を交互に達成するために譲歩し合うことは協調的活動と考えてよいであろう）．

一方，知識源の各々が常に目標の一致を生かすように活動を組み立てていたとしても，それらが協調し合っているとはいえないことがある．例えば，タクシー会社の本部が利用者からの電話予約を知らせる放送を施したとしよう．この放送に対して，利用者を獲得しようとしているタクシーの各々が，本部との間に目標の一致を認識したからといって短絡的に殺到していたのでは，本部とこれらのタクシーとが協調し合っているとはいえない．

以上のように，DAISY においては，①知識源が施し得る協調的活動は限られていること，さらには②後者のような問題点を回避するために，目標に付随するプランの各々について，それが適用可能となる条件を予め記述しておかなくてはならないことなど，いくつかの制約があることを最後に付け加えておく（なお，これらについては，目標をクラス分けすると共に，他の知識源の目標のうち，ある特定のクラスの目標を達成し易くするための活動だけをその知識源に対する協調的活動として定義することによって少なくとも部分的には解消できるものと考えている）．

6．むすび

本報告では，分散協調問題解決とは分散的環境の下で行なうことを義務付けられた協調的な問題解決であるとの立場から，従来の研究の問題点を述べると共に，それを克服するための接近法として，知識源の各々がそれらの目標間の関係に着目しながら分散的に問題解決を行なうための枠組みを提案した．

既に述べたように分散協調問題解決に関する研究は，従来，分散された知識システムの間の通信，同期および負荷分散といった計算機科学における並列処理や分散処理に関する研究の延長としてとられてきた感があり，分散された知識システムが知的に協調し合うための基礎的な問題，例えば，本報告でも言及したような，知識システムが協調的活動を施すべき状況とは何に基づいて認識され，認識された状況にふさわしい協調的活動とは何に基づいて計画されるべきかといった問題を追及した研究は少ない．

しかし，分散的な環境下におかれた知識システムの集団を採用することによって飛躍的な問題解決能力をもった分散的かつ協調的な知識システムを実装しようとするならばこのような研究に基礎をおく分散協調問題解決機構を確立していくことが不可欠であろう．

参考文献

[Carbonell 1981] Carbonell, J.: POLITICS. in Schank, R.C., and Riesbeck, C.K. (eds.), Inside Computer Understanding: Five Programs Plus Miniatures, Lawrence Erlbaum Associates, Inc. (1981).

[Genesereth et al. 1987] Genesereth, R.M., Ginsberg, M.L., and Rosenschein, J.S.: Cooperation without Communication, Proc. of the 10th IJCAI, pp.51-57 (1987).

[Ishii et al. 1987] 石井威望, 廣瀬通孝: 分散と協調, 計測と制御, Vol. 26, No.1, pp.2-10 (1987).

[Mostow 1985] Mostow, J.: Toward Better Models of the Design Process, The AI Magazine, Vol.6, No.3, pp.44-57 (1985).

[Nii 1986] Nii, H.P.: Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, The AI Magazine, Vol.7, No.2, pp.38-53 (1986).

[Ono et al. 1986] Ono, N., and Kobayashi, S.: A Parallel Object-Oriented Model for Journal Editing Processes, Control - Theory and Advanced Technology, Vol.2, No.3, Special Issue on Expert Systems and Fuzzy Control, pp.497-516 (1986).

[Ono et al. 1987] 小野典彦, 平松健司, 翁長健治: 目標間の関係に基づいた分散協調問題解決, 日本ソフトウェア科学会第4回大会論文集, pp.323-326 (1987).

[Schank et al. 1977] Schank, R.C., and Abelson, R.: Scripts, Plans, Goals, and Understanding, Lawrence Erlbaum (1977).

[Smith et al. 1981] Smith, R.G., and Davis, R.: Frameworks for Cooperation in Distributed Problem Solving, IEEE Trans. on Systems, Man, and Cybernetics, SMC-11-1, Special Issue on Distributed Problem Solving, pp. 61-70 (1981).

[Smith 1984] Smith, R.G.: Report on the 1984 Distributed Artificial Intelligence Workshop, The AI Magazine, Vol.5, No.3, pp.234-243 (1984).

[Yonezawa et al. 1986] Yonezawa, A., and Tokoro, M. (eds.): Object-Oriented Concurrent Programming, MIT Press (1986).

[Wilensky, 1981] Wilensky, R.: Meta-Planning: Representing and Using Knowledge About Planning in Problem Solving and Natural Language Understanding, Cognitive Science 5, pp.197-233 (1981).

[Wilensky 1983] Wilensky, R.: Planning and Understanding, Addison-Wesley (1983).

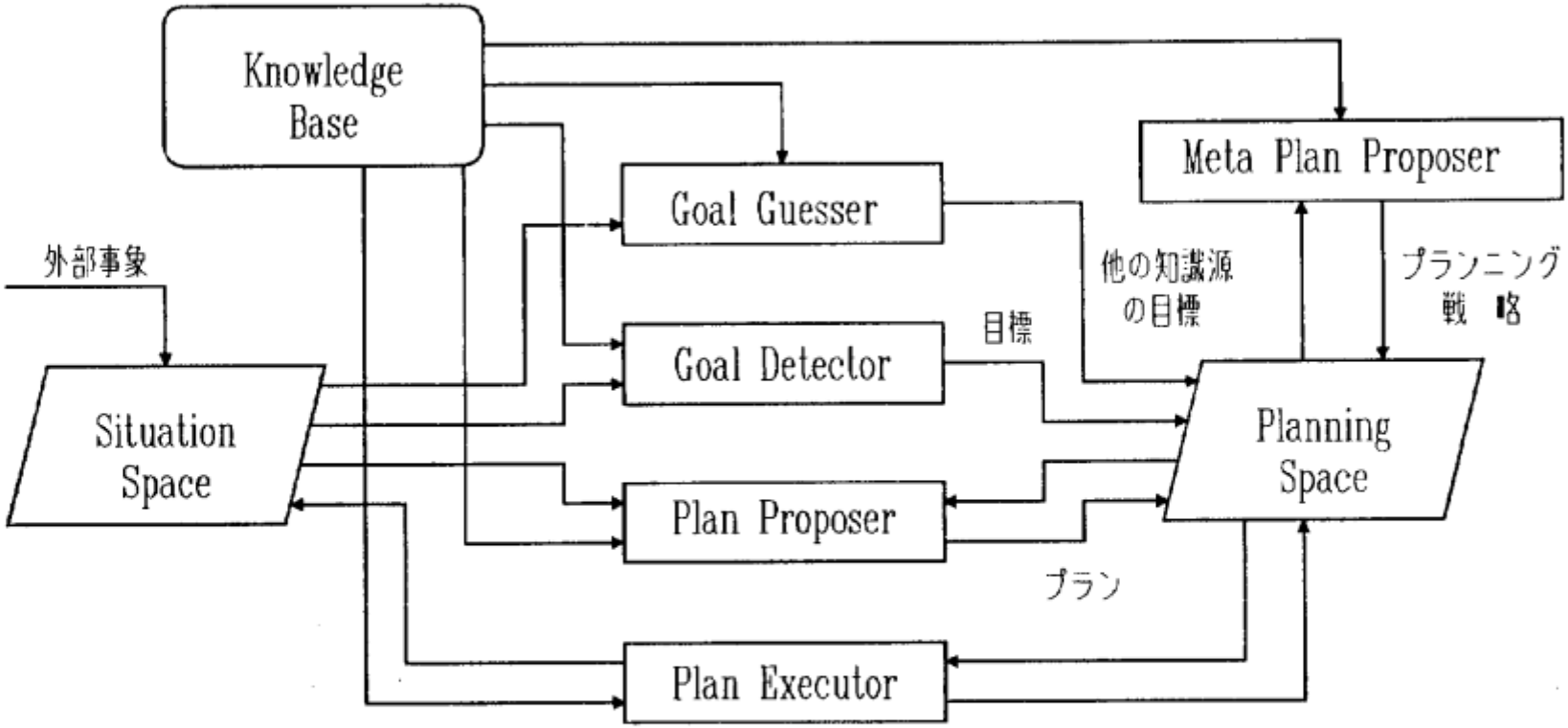

Knowledge Base
Meta Plan Proposer
Goal Guesser
外部事象
他の知識源の目標
プランニング戦略
目標
Goal Detector
Situation Space
Planning Space
Plan Proposer
プラン
Plan Executor

図1　知識源の枠組み

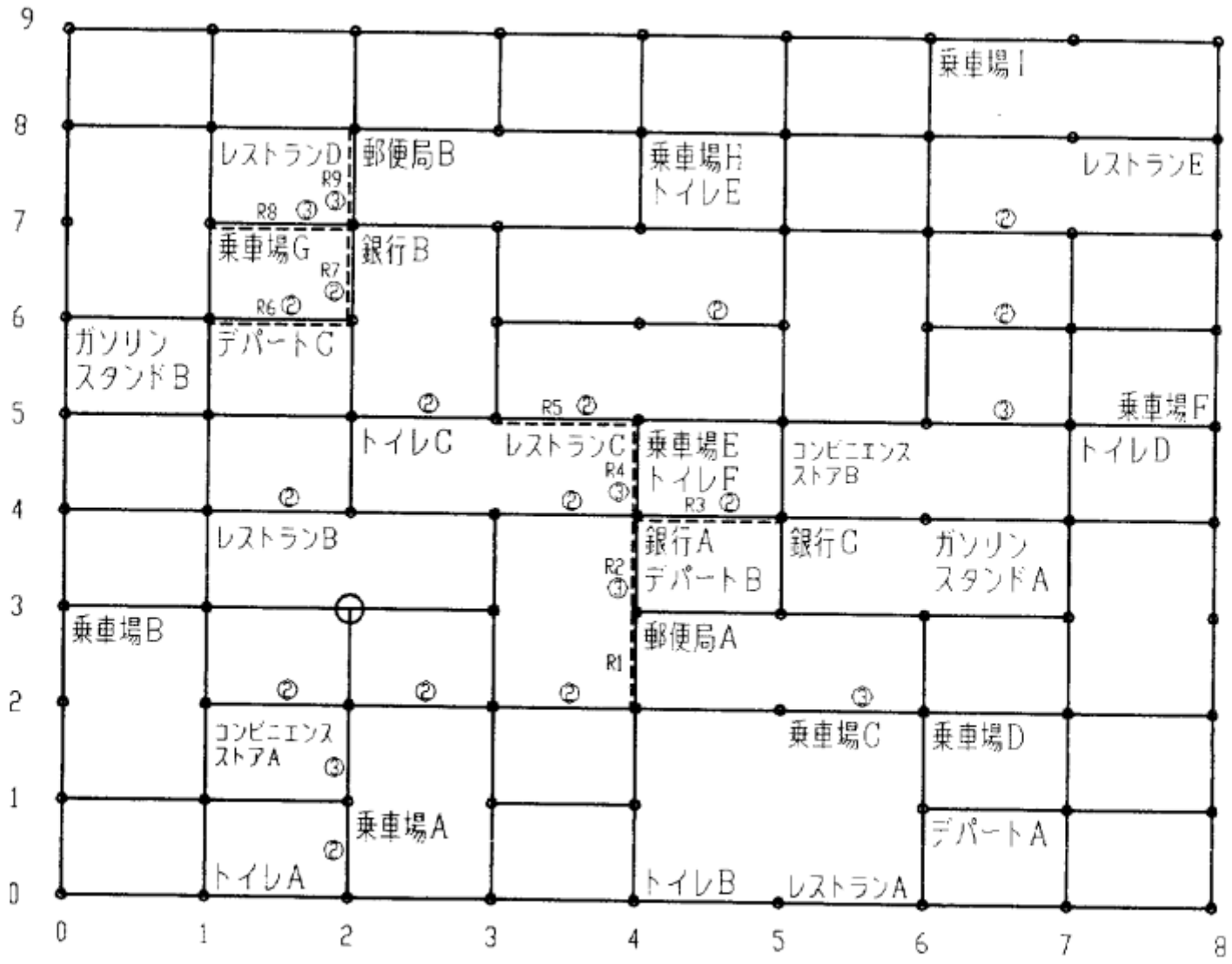


図2　タクシーの活動拠点

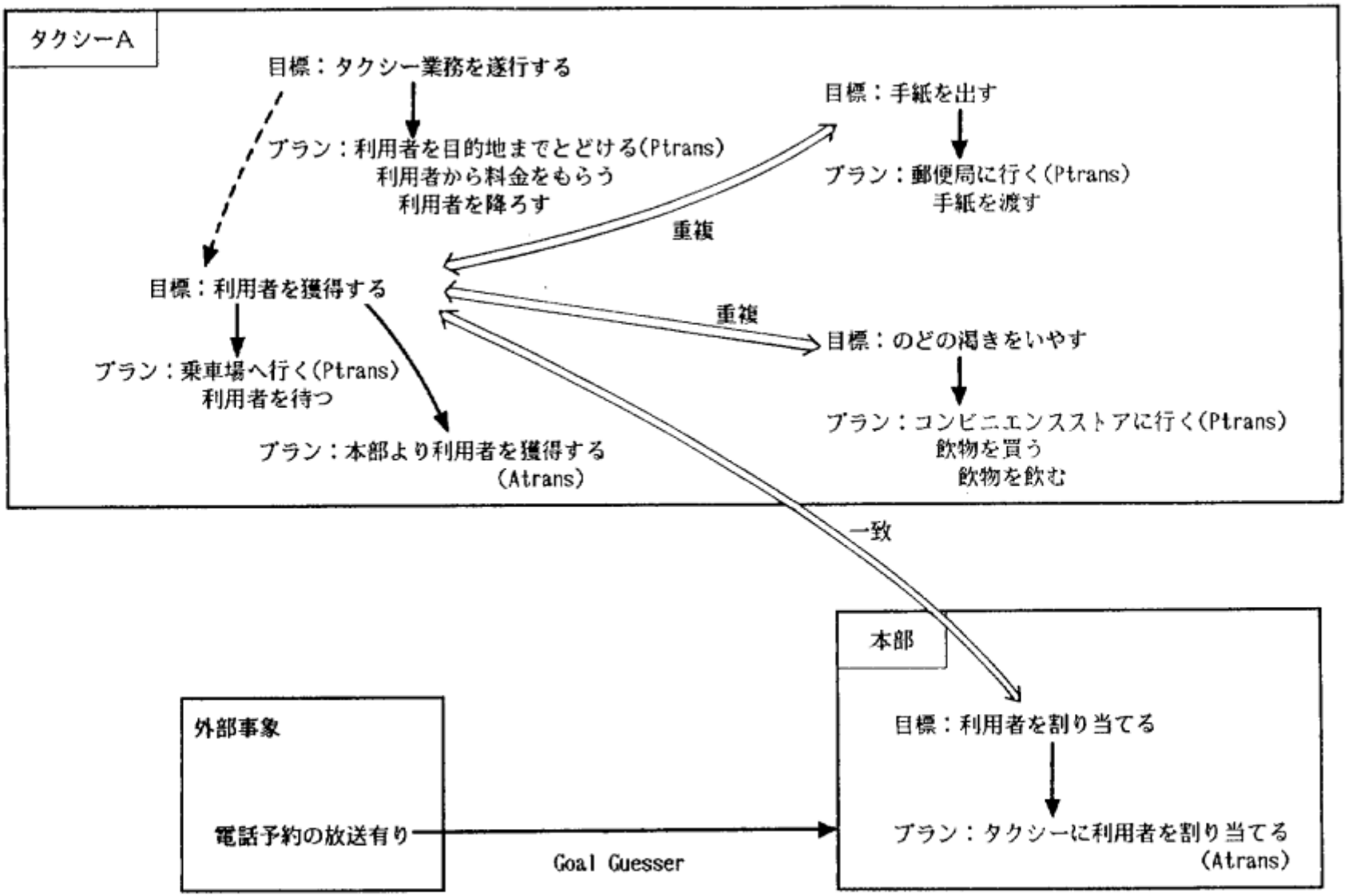


図3 タクシーによるプランニング過程

プランニング戦略　　5項組〈主目標，プラン，報酬，手続き，評価〉

主目標：選択すべき主目標
プラン：主目標を達成するために採用すべきプラン
報　酬：生成された詳細プランのそれぞれを評価するための情報
（つぎの3項組〈目標，プラン，評点〉のリスト
目　標：主目標と同時に達成可能な目標
プラン：この目標を達成するために採用すべきプラン
評　点：この目標を達成した場合の利得（整数値））
手続き：詳細プランを生成するための手続き
評　価：詳細プランを比較するための評価関数

例えば，“G2 を主目標として，可能なら G3 および G4 の達成を試みよ”というプランニング戦略はつぎのようになる．

〈主目標 = G2,
プラン =“駐車場に行く”,
報　酬 = [〈G3,“郵便局に行き，手紙を渡す”, 3〉,
〈G4,“ｺﾝﾋﾞﾆｴﾝｽｽﾄｱ に行き，飲物を買い，飲物を飲む”, 4〉],
手続き =“全数探索”,
評　価 =“評点の和が大きく，時間コストの小さいものを優先する”〉

図4　プランニング戦略の表現

2.4 エキスパートシステムにおける階層化意思決定法の利用

寺野　隆雄
（財）電力中央研究所

1．はじめに

最近、あらゆる分野においてエキスパートシステムの開発例は非常に増加しており[Waterman 1985]，[Hayes-Roth 1984]，ルール、フレームなどの基本的な人工知能手法もごく普通に用いられるようになっている。そして、解釈・診断などの分析型の問題を解くための、経験則に基づく"浅い"推論処理手法には、理論的な課題はほとんどなくなっているようにみえる。ところが、比較的単純な分類型エキスパートシステムでも、プロトタイプの開発までは比較的容易だが、それを実用システムにまで拡張するのは一般に非常に困難なのが実情である。その理由のひとつは、問題解決に伴って現れる不確実な情報をどのように取り扱うかが、特に実用上、明確になっていないことにある。

不確実な情報をエキスパートシステムで取り扱う手法としては、よく知られているとおり、確実性係数 CF [Shortliffe 1976]、それを発展させたDempster & Shafer理論[Shafer 1976]、ファジイ論理[Suurikagaku 1987]などがある。これらの理論は、いったん知識が整理され、計算機上に格納されてしまえば、その後の推論処理にある程度の合理性を与えるものである。しかし、いわゆる"浅い"エキスパートシステムで中心的な役割をはたす経験的知識（の断片）に割り当てるべき不確実な情報（の値）をどのように定めれば良いかについては、これらの理論からは直接答えられない。

本節では[Terano 1987-2]に基づいて、分析型エキスパートシステムにおいて不確実性をともなう経験則を処理するための新しい手法を提案する。この手法は、通常エキスパートシステムで利用されるフレーム型知識表現、ルールによる推論処理に、非ベイズ型意思決定手法のひとつである階層化意思決定法（ＡＨＰ：Analytic Hierarchy Process）[Saaty 1980] を組合せたものである。本手法の特長は、推論処理で用いられる経験則に、比較的容易にある種の"合理性"を与えることを可能とし、同時に、意思決定手法であるＡＨＰの持つ欠点－適切な階層構造を構成する難しさと、加法的価値関数の能力の限界－を改善していることにある。

以下に本節の内容を示す。まず、準備として、§２でエキスパートシステムにおける不確実な知識の処理手法の特徴を述べ、§３でＡＨＰの基本的な考え方を紹介する。ついで、§４において、提案する手法について論ずる。また、本手法は、筆者等がすでに発表したエキスパートシステム[Terano 1986]，[Terano1987-1]で採用した方法を改良・発展させたものである。そこで、§５では、本手法が、実際の例ではどのように用いられているかを紹介し、本手法に関連する研究について考察する。

2．エキスパートシステムにおける不確実な情報の処理

従来型のシステムでは、一般に、知識はプログラムと数値データという形で表現される。すなわち、まず、与えられた問題を適当な手法で離散的に近似する。そして、その処理に必要な知識をアルゴリズムとデータ構造という形で定式化し、最終的にプログラムの形で記述する。その時に、与えられた問題は、離散化するという意味と適当なアルゴリズムを用いるという意味とから二重に近似される。ただし、従来型システムの場合は、問題を定式化する際に十分な近似がなされるので、問題を解くにあたって、すなわち、プログラムを動かすにあたっては、不確実な情報を意識して扱う必要性はほとんどなくなる。

それに対し、エキスパートシステムにおける問題解決では、従来型システムにはみられない近似がなされる。エキスパートシステム手法の特徴は、従来ならプログラムに埋め込まれるはずの知識をプログラムの外に出して、陽に記号的に記述することである。しかし、専門家の問題解決知識を網羅的に記述し、その適用条件をすべて明らかにすることは非常に困難である。そのため、一般には、知識を人工知能の概念を使って記述した時点で、自然にもとの問題は何らかの形で近似される。

また、記述された知識は定性的であり、さらに、現在の知識処理技術の未熟さから、知識の正しい解釈や処理手順を推論機構で完全に与えられないことも多い。したがって、すくなくとも知識を問題解決に利用する場合は、その適用方法も（専門家の知識の利用方法に比べると）近似的なものにならざるをえない。

エキスパートシステムで知識にあいまいさを含んだまま推論処理を行う場合は、このような二種類の近似を計算機上で扱うことを意味する。したがって、エキスパートシステムにおいて不確実な情報を扱う意義は次の二点にあると考えられる。

・第１の役割：専門家の経験的知識を表現するときに生ずる不確実件の近似

これは、上で述べた問題解決知識を近似することに伴って生ずる不確実件である。知識の収集・記述にあたっては、一般に、適用可能な知識表現法に知識を強引に変換することが必要である。そこで、対象とする事象の因果関係を明確にするのが難しい場合には、ルール形式は制約をゆるめる目的で、経験則に基づくルールにＣＦなどの確からしさの尺度を導入するという方法がしばしば用いられる。また、入力データのもつ不確実性も、このようなデータの存在によって、問題解決知識を充実させる必要が生ずることから、この役割に含めることができる。

・第２の役割：計算機上に蓄えた知識に対する推論処理の効率化

たとえば、ルールシステムにおいて推論処理中で変化するＣＦ値は、競合解消のひとつの手段として利用される。また、ファジイ値を用いた処理も、通常のルールシステムで生ずる多くの推論ステップを省略するために有用とされる。したがって、この意味では、これらの手法は専門知識の近似という役割よりも、推論処理の効率化に役立っていると考えることができる。

本報告で提案する手法は、分析型の意志決定問題に適用可能であり、上の第１の役割の改善をめざしている。また、Bhatnagar,R.K., Kanal, L.N.は、不確実な情報の処理が必要なケースを①情報が部分的な場合、②情報に信頼性が欠ける場合、③表現言語が不完全な場合、④複数の情報が互いに相いれない場合、の四種類に分類している[Bhatnager 1986]が、彼等の立場からは、本手法は、③，④の意味での不確実性を扱うものと考えることができる。

３．階層的意思決定法（ＡＨＰ）の概要

ＡＨＰは複数の項目間の重要度の比を一対比較して得たデータをもとに、各項目のウェイトを決定する手法で、 もともと、複雑な意志決定問題を簡便に解く手法としてSaaty,T.L.によって開発された[Saaty 1980]。我国では最近、オペレーションズ・リサーチ分野において、新エネルギー・システムの評価、商品計画などさまざまな応用が試みられている[OR 1986]。

ＡＨＰでは、まず与えられた問題（意思決定の対象）を階層的な構造をもつ部分問題に分解して整理し、下位レベルの部分問題の解、すなわち部分構造の評価値から線形の評価関数を計算して、ひとつ上のレベルの（部分）問題の解としていく。そして、これを各階層で繰返すことで全体の解を得る（図１）。この点からみると、分類問題を階層的に整理して解く知識処理の問題解決手法と似た面が存在する。

ＡＨＰの特徴的な点は、線形評価関数の定め方にある。すなわち、ＡＨＰでは、階層的に表現した各項目の評価値が線形関数（加法的価値関数という$\Sigma W_i * V_i$ の形の関数、ここでW_iは下位の部分問題に対する評価用のウェイトを、V_iは下位部分問題i の評価値を示す）で表わされることを仮定する。ＡＨＰでは、複数の項目間の重要度の比を一対比較して得たデータをもとに、各項目のウェイトW_iを決定する。すなわち、このウェイトW_iを定めるのに項目間の一対比較行列を利用する。

この原理は次のようなものである。例えば、S_1, S_2, S_3の評価すべき項目がある場合、その重要度（ウェイト）がW_1, W_2, W_3であったとする。この時、各重要度の比に関する一対比較の値 $(W_1/W_2, W_2/W_3, W_3/W_1)$を意志決定者が与える。一対比較は、通常、定性的な判断に基づいて行われ、５段階(5,3,1,1/3,1/5)または７段階(7,5,3,1,1/3,1/5,1/7)で行うのが適切とされている。ここで、7 は項目iは項目jより "大変優れている" 、5

は“優れている”、3は“やや優れている”、1は“同程度”、…、の意味を表わす。一対比較が完全に合理的であれば、その結果得られる重要度比の行列（一対比較行列）と各ウェイトW_i (i=1,…,3)との間には、次の式が成立するはずである。

$$\begin{pmatrix} 1 & W_1/W_2 & W_1/W_3 \\ W_2/W_1 & 1 & W_2/W_3 \\ W_3/W_1 & W_3/W_2 & 1 \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix} = 3 * \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix}$$

ＡＨＰは、この式を逆に利用して、与えられた一対比較行列の固有値、固有ベクトルを求める。これによって、n*(n-1)/2 個の一対比較値から n個の項目のウェイトW_iを得る手法である（通常は、$\Sigma W_i = 1$となるように、ウェイトW_iを正規化する）。上式が成立していれば、一対比較行列はランク１で数値的な安定性もよく、比較項目数n に等しい固有値をもつ。そこで、比較の結果実際に得られた一対比較行列の最大固有値の大きさと nとを比べることで、ウェイトW_iの整合性を調べることができる。これには、通常、Consistency Index, CI= (固有値－n)/(n-1) ならびに Consistency Ratio, CR = CI/CI* が用いられる。ただし、CI*はまったくランダムな一対比較行列から実験的にえれるCIのことであり、たとえば、文献 [Saaty 1980]に表の形で与えられている。CI, CR とも 0.1 以下ならば、ウェイトの整合性がとれていると判定するのが普通である。

ＡＨＰがうまく適用できるポイントは、一対比較を階層的に繰り返し実施できるよう、対象システムの情報を適切に階層構造の形に整理することにある。そして、適切な階層構造のもとで、（専門家によって）適切な一対比較がなされる限り、得られたウェイトは、専門家の経験則をよく表わしていると考えることができる。

一方、ＡＨＰの欠点は、まず、意思決定の前提として前述した加法的価値関数の存在を仮定していることである。また、項目間の整合性のみに着目してウェイトを決定しているために、従属性の強い項目をそうでない項目と互いに比較すると正しいウェイトが得られないことがあげられる。

これらのＡＨＰの欠点は、ＡＨＰの手法の単純さとうらはらな関係にあり、実際の分類問題に適用する場合には、さらに、別の手法を取入れて評価方式をより柔軟なものにしていく必要も生ずる。次章で提案する手法は、ＡＨＰによる意思決定の過程に、エキスパートシステムで一般的なフレーム型の知識表現手法とルールによる推論処理とを加えることによって、その限界を改善したものと解釈することもできる。

４．フレーム／ルール型知識表現とＡＨＰとを融合した推論処理手法

４．１．　手　法　の　概　要

図２に、フレーム／ルール表現とＡＨＰとを融合させてエキスパートシステムを開発する手順を示す。以下、この手順にしたがって手法の概要を説明する。

（Step １）解候補の数え上げ

分類問題の解候補をすべて数え上げる。最終的な解候補を得るのが困難な場合には、２章で述べた中間仮説の導入や階層的な知識の記述方法などを利用して適当なグループ化を行う。ここでまとめた解候補は、次ステップ以降で詳細化・厳密化することができるので、この段階では、解候補を逃さないことに留意する。

（Step 2)解候補・解の性質の階層的な整理

フレームの概念を利用して知識を表現し、階層構造を作成する。このプロセスは、４章で述べたＡＨＰの適用において最初に階層構造を作成する作業と類似している。すなわち、フレームの階層構造の各部分は一対比較の対象と捉えることができる。そこで、ＡＨＰの概念・手法が容易に扱えるように、一対比較できるよう対象システムを適切な階層構造に整理するために、フレームの全体－部分の関係(part-of関係）を利用する。

part-of 関係をもつフレーム・システムはフレームの概念をサポートするエキスパートシステム開発ツールには標準として組込まれており、また、みずから、この機能をプログラムすることも容易である。ただし、階層構造を整理する際、問題解決の視点を

ばしば変更することが頻繁に生じる場合がある。そのような場合は、視点を明確にするという立場から柔軟なフレームシステムを利用することが有用である。これには、たとえば、当所で開発したKJ法に基づく知識整理支援システム CONSIST [Sinohara 1986] の機能が適切と考えられる。

(Step 3) AHPによる部分問題の評価

フレーム中で互いに比較可能な基本的な項目にウェイトを与え、それに基づいた評価を行う。項目間で一対比較を用い、AHPによって各項目の相対的な重要度を決定する。ここで得られる重要度は専門家の経験則を陰に表現したものであり、知識の確からしさを数値化したものと考えられる。従来の知識処理手法では、このような情報は、CF値もしくはファジイ理論のメンバーシップ関数の形で表現されていた。このウェイトは、階層の下位に位置する部分問題の評価値を計算するのに使われる。

また、加法的価値関数のみによる項目の評価に加え、評価を上位の階層へ合成する過程で、評価値に対してそれを評価するための任意の演算（たとえば Min/Max演算など）を許す。ある階層に含まれる項目群を、たとえばMin/Max演算で評価することは、その階層の価値を（個々の項目にとらわれずに）全体的に評価することに相当する。したがって、複数の情報から仮説が支持される場合にしばしば用いられるCF値の複合操作（Combination）に相当する操作が、このような演算の拡張によって可能となる。

(Step 4) プロダクション・ルールの記述

AHPに基づく評価手法ではうまく表現できない問題解決知識、もしくは、AHPで用いる評価方法より厳密に意味を表現できる知識を、プロダクション・ルールの形式で記述する。これは、専門家の経験則を陰に表現したAHPによる重要度と対象的に、専門家の知識を陽に表現したものとなる。不確実な情報に与える重要度は、すでにAHPによって定められているので、ここで記述したルールには、CFなどの不確実性を表す尺度を付加する必要はない。

(Step 5) プロダクション・ルールの適用

(Step 3)で得られる中間的な評価値に基づいて上位レベルでの判断を下すのに、(Step 4)で記述したプロダクション・ルールを適用して推論処理を行う。すなわち、ルール中で使われる（あいまい性を含む）述語を、下位レベルのAHPによる評価値に対応させることによって、不確実な推論処理を可能とする。CFやファジイなどの概念を使うと、推論処理で正しい結論を導けるようにするまでに、CFやメンバーシップ関数の調整に手間取るのが普通であるが、AHPによれば、各項目のウェイトに専門家の経験則が生かされ、また、（同じ階層の中では）ウェイトの間の整合性が保証されるので、このような調整の手間をほとんどなくすることができる。

AHP，フレーム、ルールを用いる分類問題を解く諸方式と、本報告の方式の特徴をおおまかに整理すれば表1のようになる。これからもわかるとおり、本方式は、フレームによる経験的知識の記述のしやすさ、ならびに、AHPによる解候補間の評価の整合の取れやすさを表現手法と、AHPという意思決定法とを組合せているために、小規模な分類問題を扱う場合でもシステムがおおがかりになるという欠点をもっている。

(step 1) 分類問題として定式化できるように解候補をすべて数え上げる

(step 2) フレーム表現の持つ階層的な分類機能を利用して解の候補の集合、ならびに、それらに関する性質を整理する。

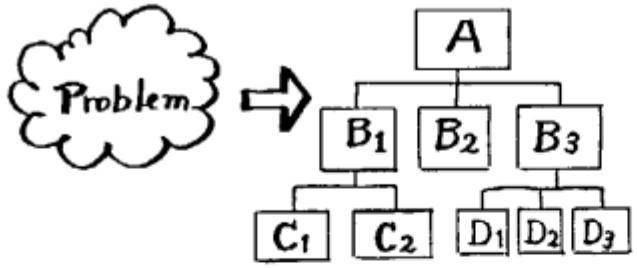
(step 3) 階層的に整理した情報のうち、同じ階層にはいる項目の重要性をＡＨＰを用いて評価し、各項目のウェイトを定める。

(step 4) 分類問題を解くための経験的知識をルールで記述する。

(step 5) ルール中の述語とＡＨＰで定めた評価項目を対応させ、推論処理中で利用する。

図２．フレーム・ルールとＡＨＰとを組合せた知識処理手法の適用手順

① 部分問題への階層的な分解



② 加法的価値関数による評価

$$E(A) := \sum_i w_i E(B_i)$$

一対比較による重み決定

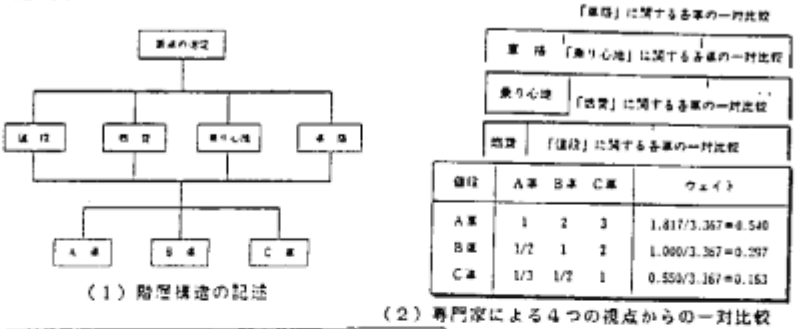
図１．　ＡＨＰの原理

表１．分類型問題を解くための知識処理の諸手法の特徴

	長所	短所
フレーム	対象システムの知識を階層的に整理しやすい	不確実な経験的知識を記述しにくい
ルール	経験的知識を記述しやすい	不確実な経験的知識を整合させにくい
ＡＨＰ	解候補間の評価を整合させやすい	適切な知識の階層構造を得るのが難しい
本手法	対象システムの知識を整理しやすい 経験的知識を記述しやすい 解候補間の評価を整合させやすい	小規模な問題を扱う場合でも複雑なシステムになる

４．２．単純な例題への適用

本手法を単純な例で示す。ＡＨＰの入門書［Tone 1985］に記述のある自動車の選択問題を考える。これは、図３（1）に示すように、Ａ，Ｂ，Ｃの３種類の車の選定を行うのに、値段、燃費、乗り心地、車格の４つの視点を用いるというものである。この問題はごく小規模であるために、表１にまとめた他の手法でも容易に解くことができ、また、本手法を適用すると必要以上に複雑になるが、手法の概要を例で示すためには適当である。



（1）階層構造の記述

（2）専門家による４つの視点からの一対比較

対	値段 (2.1)	燃費 (2.2)	乗り心地 (2.3)	車格 (2.4)
値段 (2.1)	1	3	5	7
燃費 (2.2)	1/3	1	5	7
乗り心地 (2.3)	1/5	1/5	1	3
車格 (2.4)	1/7	1/7	1/3	1

（3）利用者による一対比較

図３．ＡＨＰによる車種選定（［Tone 1986］に基づく）

ＡＨＰの手法のみによってこの問題を解く場合は、まず、４つの視点を階層構造として整理しなくてはならない。この過程がＡＨＰを適用する場合に一番難しいものとなるが、車種選定問題は小規模なため、すぐに、図３(1) のような階層構造を得ることができる。次に、Ａ，Ｂ，Ｃの３種類の車（代替案）が値段、燃費、乗り心地、車格の点でそれぞれどのような（相対的な）重要度をもつかを一対比較法で与える(図３ (2))。より規模の大きな難しい問題の場合は、この一対比較行列の作成は、高度の専門知識をもつ専門家が行うこととなる。一方、車種の選定のために、値段、燃費、乗り心地、車格の各評価項目のどれを重視するかという問題は、車の選択を行う利用者の問題である。そこで、利用者の視点から、各項目の重要度を、やはり、一対比較法で与える(図３ (3))。最終結果は、これら代替案・評価項目のウェイトつき平均の大小によって得られ、このウェイトの場合、Ｂ，Ａ，Ｃの順に望ましい順序となる。

この問題をフレーム／ルールのみを使って解くと典型的な分類問題になり、これは図４のように表現できる。ただし、ここでは、エキスパートシステムによる問題解決の特性を明確なものとするために、もとの自動車選択問題を一部拡張し、乗り心地の評価を行うには、さらに、高速時、低速時の評価が必要なものとしている。車の概念の枠組みを与えるフレーム（プロトタイプフレーム）には、値段、燃費、乗り心地、車格の項目を設け、Ａ，Ｂ，Ｃの３種類の車に対応する情報を投入したフレーム（インスタンスフレーム）を作る。そして、各インスタンスフレームの基本的な項目に定性的な値を絶対評価値として専門家が与えておく(図４ (1))。

ここで、フレームの階層の表現が、記述の上では、図２のＡＨＰによる階層の表現と上位・下位の記述が逆転していることに注目されたい。ＡＨＰに限らず、意思決定手法を用いて階層を定義した場合と、知識処理技術を用いて階層を定義した場合では、このような（記述上での）階層の逆転がしばしばみられることが知られている[Henrion 1987]。知識を記述する際には、同じ構造を別の観点から整理しているので、注意が必要である。

より上位の項目の値を決定するには通常ルールを利用する(図４ (2) (a))。これらのルールの適用によって乗り心地の評価値が定まる。一般に、複雑なシステムを対象にする場合は、図４ (2) (a)のような種類のフレームの項目値を定めるための知識が膨大なものとなる。

さらに、最終的な意思決定のために、値段、燃費、乗り心地、車格の４つの視点から、たとえば、図４ (2) (b)のような経験的なルールをやはり専門家が記述する。この時、一般に、適切な答を出すためには、ＣＦ値の調整が必要となる。すなわち、経験則が陰に表現されるＣＦの値を調整することで、複数の仮説の組合せを処理したり、ルールの競合解消を行う必要が生ずる。利用者がシステムを使う時には、値段、燃費、乗り心地、車格のうちどれを重視するかという情報を入力し、ルールを起動して推論を行う。

（１）自動車フレーム

```
If ＊車の"乗り心地", "高速時"＝"良い"かつ"低速時"＝"良い"
                then ＊車の"乗り心地":="良い" [CF 0.9]
If ＊車の"乗り心地", "高速時"＝"良い"かつ"低速時"＝"悪い"
                then ＊車の"乗り心地":="普通" [CF 0.7]
If ＊車の"乗り心地", "高速時"＝"悪い"かつ"低速時"＝"悪い"
                then ＊車の"乗り心地":="悪い" [CF 0.9]
        ……
          ……
If "燃費"を重視する   then ＊車の"燃費"の"良い"＊車に決定 [CF 0.8]
If "値段"を重視する   then ＊車の"値段"の"安い"＊車に決定 [CF 0.9]
If "乗り心地"と"車格"の両方を重視する
   then ＊車の"乗り心地"と"車格"とが"悪い"でない＊車に決定 [CF 0.6]
        ……
          ……
            ……
```

（ａ）

（ｂ）

（２）選定用のルール

図４．フレーム／ルールによる車種選定

は、目な示速1)す分H行ム項的に高～価部A較ー本対(2)、(0評の)比(図レ基相ら)た値を定に対ルフのつまま価地決様一一車ムか図。評心思同なル動一的、るなり意とうの自レ量はい的乗(2よ様。フ定値て対、が図の同るス、なれ絶て者(3)とンた的さのし用はらに夕得量現家そ利にの図うスて定表門。を断図、ようンっのも専いか判、はのイ行らで、なるのばに下各をれ値にはす分え定以。較こなとで視部と決とる比。的ごの重のた思くき対る性車も度こは意解で一あ定各た程、値のを述のので、、しのと価位題記車分てはは定どる評上問でC部いて決をす対、の形，るづいでら。相てこのBな基つPちるのしで(1)，異ににHどすこそ法らA，とル目Aのと。方らに(1)ブ項、目のるる。た図と(4)ーのは項もきき し、ご4テ地れのるでで案様目図換心こつすがが提同項が変り、2断とととがこの乗がの判ここる。節例家こトのるこ)るるす本た門、イ時い、てすめ用、ベ専りェ速てはし用定利に述、あウ低えきと適らを次にはでた、与と題をか上値値し時をる問P列(4))

、決ル述実、き価思一記でで評意ルで形分がのなくルた部とル的づーれるこベ分基ルとれうレ部に、のさ行位)則で性求を下る験と合要定、よ経こ整が決きにはるに性思お者でせ軽軟意て用定合手柔たし利決組をくい現(思に定づ用表と意切決基を定の適思にルム決ルを意則ーテのベ法の験ルス値レ手で経なシ価位のルの易象評上方べで容対なii)両レルので的、位べ更ム対、に下レ変ー相とう、位、レ)とよる上く、フるこのぎ、すたや(i)よるこすたや(i)にいい。りまし、家用るな、解は門をあにき理徴専Pに雑でて。特(Hと繁がっる。の、Aこはととな法ちはるてこにに手わにいいる家う本なと用てす門よす定をし施専る

プロトタイプ
フレーム
自動車フレーム
- 値段（ウェイト　評価）：
- 燃費（ウェイト　評価）：
- 乗り心地（ウェイト　評価）：
  - 高速時（評価）：
  - 低速時（評価）
- 車格（ウェイト　評価）：

インスタンス
フレーム

	A車	B車	C車
値段	(0.54 安い)	(0.30 中)	(0.16 高い)
燃費	(0.11 悪い)	(0.74 良い)	(0.15 悪い)
乗り心地	(0.54 良い)	(0.16 悪い)	(0.30 普通)
高速時	('0.95' 良い)	('0.20' 悪い)	('0.54' 良い)
低速時	('0.55' 良い)	('0.24' 悪い)	('0.45' 普通)
車格	(0.20 悪い)	(0.40 普通)	(0.40 普通)

（１）自動車フレーム

良い：	0.5～1.0
普通：	0.25～0.5
悪い：	0.0～0.25
安い：	0.5～1.0
中　：	0.25～0.5
高い：	0.0～0.25

（２）ウェイトの解釈テーブル

"乗り心地"のウェイト付け

	低速	高速	ウェイト
低速	1	4	0.8
高速	1/4	1	0.2

"乗り心地"の評価：＝
0.8*"低速時の値"＋0.2*"高速時の値"

（３）乗り心地の一対比較と評価

If "燃費"を重視する then ＊車の"燃費"が"良い"＊車に決定
If "値段"を重視する then ＊車の"値段"が"良い"＊車に決定
If "乗り心地"と"車格"の両方を重視する
then ＊車の"乗り心地"が"悪い"でなくかつ"車格""悪い"でない＊車に決定
……
……
……

（４）選定のためのルール

図５ フレーム／ルールとＡＨＰによる車種選定

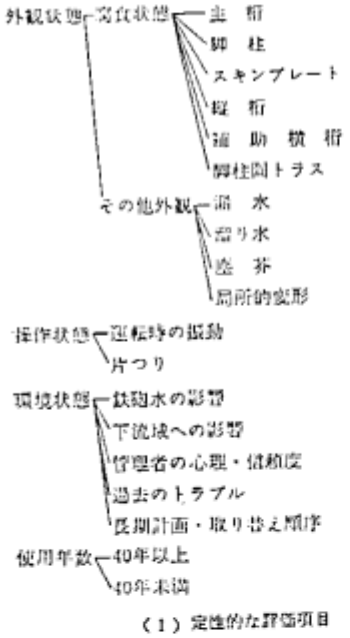
5．本手法の適用例と関連する研究の考察

本節では、本手法の有効性を実問題への適用例で示し、ついで、関連する研究について考察する。ここで示す2つの例のうちひとつは、本手法をほぼ忠実に適用したエキスパートシステムの例であり、もうひとつはフレームシステムの柔軟性を使ってＡＨＰ適用の高度化をはかった例である。

5．1．ダムゲートの寿命診断エキスパートシステム［Terano 1986］

本システム（DGDA; Dam Gate Diagnosing Advisor）は、水力発電所のダムゲート（水門）の診断を目的として、当所で開発中のエキスパートシステムである。これは、既存のデータベース、有限要素法による簡易構造解析プログラムのフロントエンド・システムとして利用することを前提としており、対象ダムゲートを、構造力学的視点とそれ以外の経験的な視点から診断し、その寿命予測を行う機能をもつ。本論文で述べた手法が DGDA で使われているのは、経験的知識に基づいて外観状態（各部材の腐食状況など）操作状態；環境状態；経過年数（これらはすべて定性的な評価項目である）の観点からダムゲートの診断・評価を行なう部分である。

これら重要性を評価すべき定性的な項目は、ダムゲート知識ベースのフレーム構造と一致した形で階層的に整理することができる。この階層は、エキスパートシステムの設計初期段階で自然に導入された構造である。最下位の項目は、経過年数を除き、それぞれ 0,1,2,3 （"良い" から "悪い" までに対応する）の値（スコア）をもつ。DGDAでは、これらの項目を総合的に評価するために、ＡＨＰを適用して各項目のウェイトを決定した。各基本項目のスコアとＡＨＰで定めたウェイトとを利用して得た評価値は"良い" から "悪い" までに対応する4段階に離散化される。そして経験則に基づくルールにおいて、「外観状態が" 悪い" 」などの用語で参照される（図6）。



（1）定性的な評価項目

（2）定性的データの評価方式

	項　目	評価点	ウエイト	評価結果
外観評価	主桁	0,1,2,3	0.25	Σ(評価点×ウエイト)
	脚柱	0,1,2,3	0.25	
	スキンプレート	0,1,2,3	0.10	=0.0～1.0 良い
	縦桁	0,1,2,3	0.05	=1.0～1.5 概ね良い
	補助横桁	0,1,2,3	0.05	=1.5～2.0 良くない
	脚柱間トラス	0,1,2,3	0.05	=2.0～3.0 かなり悪い
	漏水	0,1,2,3	0.10	
	錆水	0,1,2,3	0.05	
	変形	0,1,2,3	0.05	
	局所的変形	0,1,2,3	0.05	

……
……

（3）　診断ルールの例

- IF　外観状態が "良く"，操作状態が良くなく，経過年数＜40年
  THEN 現状のまま使用可
- IF　外観状態が "かなり悪く"，操作状態が "概ね良く"，環境状態が "良く"，経過年数＜40年
  THEN 補修が必要

図6．ダムゲートの寿命診断システムにおける推論方式［Terano 1986］

この例では、対象システムの構成情報を固定したフレームとして記述しておき、下位レベルの評価をＡＨＰで、上位レベルの評価をルールによって行っている。なお、DGDAには、KEE[Fikes 1985]で記述したプロトタイプバージョンとEshell[Fujitsu 1985]

で記述した実用化バージョンとが存在するが、どちらのフレームシステムでも本手法は容易に実現されており、この手法の汎用性を示している。

5．2．エキスパートシステム開発ツールの評価 [Terano 1987-1]

エキスパートシステム開発ツール（シェルという）を評価するためには、①知識表現や推論機構などのシェルの機能、②システム開発に使われる開発者インタフェース、③システムの利用者インタフェース、④他言語や他システムとの連係をはかるシステムインタフェース、⑤シェルの性能（以上シーズ側の視点）、⑥システムの利用目的、⑦システムで扱うタスク（以上ニーズ側の視点）など多くの視点がある。さらに、これらの視点に基づいて、各シェルの評価項目を決定できたとしても、これらの評価項目は互いに独立なものではない。たとえば、ルールのもつ機能などは、知識表現／推論機能にも、シェルの性能にも関係しうる。

したがって、これら評価項目情報を利用し適切な評価を行うためには、各項目を扱う視点を目的に応じて切り替えることが必要である。そこで、我々は、これらの評価項目を柔軟なフレームシステムをもつ知識整理支援システム CONSIST [Sinohara 1986]を利用して取りまとた。CONSIST では、基本的な知識表現として、フレームモデルにおいてスロット間にクラス階層を導入したものを採用しているので、リンクの意味付けを自由に指定することが可能である。また、同時に、視点を変更するという（人手では困難な）作業を非常に容易に実行することができる。

個々のシェルの評価を行うためには、適当な視点からフレーム構造を整理し、その上でＡＨＰを適用する。その手順は次のとおりである。

(1) ニーズ側から評価の視点を決定し、その評価項目を選び出す；
(2) ニーズ側で選んだ評価項目に関連する項目を、シーズ側の視点から選択する；
(3) 上で選択した項目について、比較対象となるシェルの間で一対比較を行い、ＡＨＰを適用して各シェルの評価値を決定する（図6）。

ただし、この問題に対しては、少なくとも現在のところ、意思決定にはＡＨＰの手法のみを利用している。ここで、ルールによる推論処理を採用しなかった理由は、シェル評価用の知識にはあまり明確なものがないため、これを経験的なルールの形式にまとめることができなかったことによる。

フレーム・システムの機能が豊富であれば、我々の提案した手法は、この例に見られるように、さまざまな視点から柔軟な意思決定を行うことを可能とするものである。

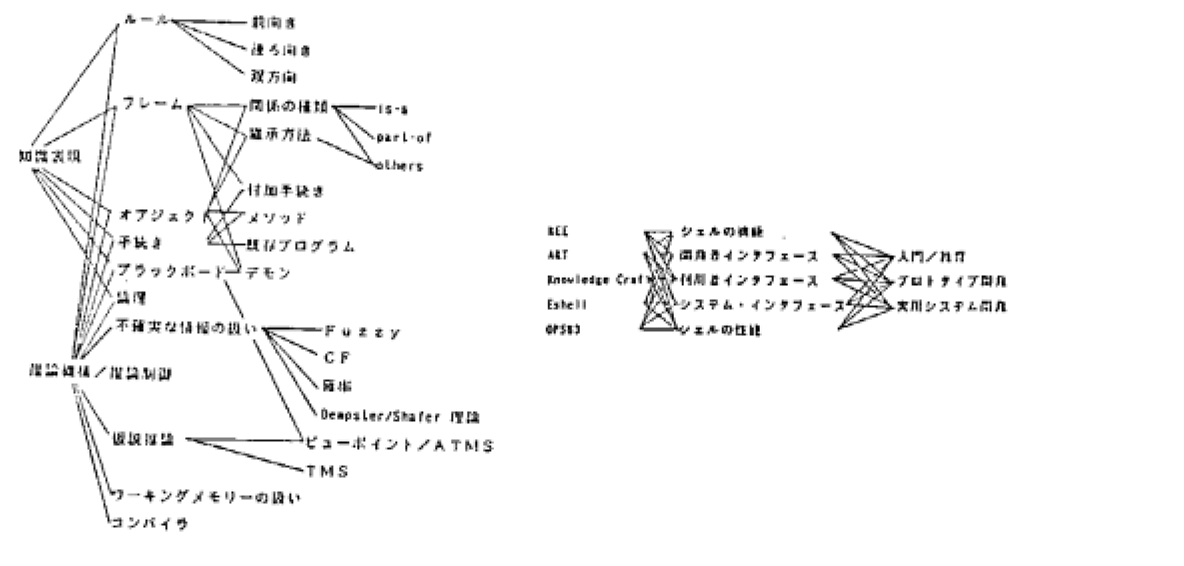


図6. シェルの機能の評価項目の構成例 [Terano 1987-1]

５．３．関連する研究

本報告で提案した手法は、前述したエキスパートシステム開発研究の中から生まれたものであるが、これに関連する研究としては次のようなものがある。

（１）エキスパートシステムの評価 [Liebowitz 1986]

エキスパートシステムの性能を、変更容易性、利用容易性、費用／効果など複数の視点から評価するためにＡＨＰを利用した報告である。この報告では、[Saaty 1980]にあるような典型的なアプローチでＡＨＰを利用しているのみで、手法自体に知識工学的な考察は見られない。

（２）意思決定手法を利用したルールの正当化 [Langlotz 1986]

MYCIN に含まれる経験則に基づくルールの正当性を検証するために、決定木による意思決定分析を適用する。これは、意思決定用ルールのひとつを詳細に分析して、そのルールの正当性を示したもので、分析手法としては一般的な枠組みを与えている。ただし、このような手法は非常に手間がかかるために、これによって知識ベース全体を分析するのはほとんど不可能である。それに対して、我々の方法は、ルールに対する評価基準を適当に定めることさえできれば、このような分析にも簡便に利用できるものと考えられる。

（３）エキスパートシステム手法と意思決定手法の比較 [Henrion 1987]

小規模な、リンゴ木の病害診断システムをエキスパートシステム手法(KEE)と決定木を使う意思決定手法とで実現し、２つの手法の評価を行う。両手法とも、機能的にはほぼ同等のシステムが実現されている。開発初期段階に作られる病害診断の階層構造は両者とも類似していると報告しているが、これは、意思決定作業をシステム化するのに必要な対象の階層化に、フレームシステムが有用であるという、我々の主張と一致するものである。。ただし、この報告は、両手法の比較検討を述べたのみで、両手法の融合をめざしたものではない。

（４）ＣＳＲＬにおける不確実な情報の扱い [Chandrasekaren 1986]

Chandrasekarenは、彼の提案したGeneric Tasksの枠組みの中で、分類問題向きのツールＣＳＲＬを開発した。この特徴は、問題の階層構造を整理する階層的分類と不確実な情報を扱う仮説照合のタスクをもつことである。ＣＳＲＬでは、下位のエージェントにわたすことで処理が進行し解決した問題の解に確実度をつけて上位のエージェントにわたすことで処理が進行していく。彼の枠組みでは、この値を定める手法は与えられていないが、階層的分類のタスクをフレームの階層構造に対応させ、仮説照合のタスクをＡＨＰに対応させれば、我々の手法によって、より整合性の高い推論処理が実行できるものと考えられる。

（５）診療圏予測システムに対する適用 [Ohhashi 1987]

合成型エキスパートシステムにＡＨＰを適用したものである。ルールによる推論処理とＡＨＰと組合せるという意味では、我々が提案した手法に近い。ただし、この報告では、ルールをＡＨＰのウェイトを決定するために利用する、すなわち、一対比較を行うために利用しており、意思決定には用いていない。これは、上位レベルの意思決定にこそ専門家に理解しやすいルール表現をもちいるべきであるという我々の考え方とは異なるものである。

（６）知識獲得支援システムAQUINAS での利用 [Boose 1987]

Boose, J.H. 等は、認知科学における人間の概念形成理論であるPersonal Construct Theory という概念を利用して、知識を階層的に分類することで、知識獲得作業を支援するシステムAQUINAS を開発している。その中で、彼等は不確実な知識を取扱う手法として、ＣＦ、ファジィ、ＡＨＰの３つを採用している。具体的にはどのような手法を使っているのかは論文には記述がないが、本報告で提案した手法も知識を適切に整理して、近似するという意味からは、知識獲得の問題とも関連が深い。

6. おわりに

本節では、分析型エキスパートシステム向きの、不確実性をともなう知識を処理するための新しい手法を提案した。この手法は、従来から利用されてきた、フレーム型知識表現、ルールによる推論方式と非ベイズ型意思決定手法のひとつであるＡＨＰとを組合せていることが特長であり、広い範囲の分析型意思決定問題に適用できるものである。ただし、本手法は、"浅い"エキスパートシステムの性能、および開発の手間を改善するものであり、より"深い"システムに適用するには、今後の一層の研究開発が必要である。そのためには、知識工学的手法とさまざまな意思決定手法／システム工学的手法[Kobayashi 1986]との関連性を明確し、両者を適切に組合せることが重要である。

参考文献

[Bhatnager 1986] Bhatnagar, R.K., and Kanal, L.N.: Handling Uncertain Information: A Review of Numeric and Non-numeric Methods. (in Kanal, L.N., and Lemmer, J.F. (eds.): "Uncertainty in Artificial Intelligence." North-Holland, pp. 3-32, 1986.)

[Boose 1987] Boose, J.H., and Bradshaw, J.M.: Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge-Acquisition Workbench for Knowledge-Based Systems. Int. J. Man-Machine Studies, Vol.26, pp.3-28 (1987).

[Chandrasekaran 1986] Chandrasekaran, B.: Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. IEEE Expert, Vol.1, No.3, pp.23-30 (Fall 1986).

[Fikes 1985] Fikes, R., and Kehler, T.P.: The Role of Frame-Based Representation in Reasoning. Comm. ACM, Vol.28, No. 9, pp.904-920 (1985).

[French 1986] French, S.: "Decision Theory - An Introduction to the Mathematics of Rationality.", Ellis Horwood, 1986.

[Fujitsu 1985] 富士通株式会社：" Ｅｓｈｅｌｌ解説書（エキスパート・システム構築ソフトウェア）．" 99AR-3030-1, 1985.

[Hayes-Roth 1983] Hayes-Roth, F., Waterman, D.A., and Lenat, D. B. (eds.): "Building Expert Systems." Addison-Wesley, 1983.

[Henrion 1987] Henrion, M., and Cooley, D. R.: An Experimental Comparison of Knowledge Engineering for Expert Systems and for Decision Analysis. Proc. 6th AAAI, pp. 471-476 (1987).

[Kobayashi 1986] 小林重信：" 知識工学．" 昭晃堂，１９８６．

[Langlotz 1986] Langlotz, C.P., Shortliffe, E.H., and Fagan, L.M.: Using Decision Theory to Justify Heuristics. Proc. 5th AAAI, pp.215-219 (1986).

[Liebowitz 1986] Liebowitz, J.: Useful Approach for Evaluating Expert Systems. Expert Systems, Vol. 3, No. 2, pp. 86-96 (1986).

[Ohhashi 1987] 大橋昭南（他）：ＡＨＰを応用したExpert System（診療圏予測システム）．人工知能学会全国大会（第１回）論文集，pp. 331-334, １９８７年６月．

[OR 1986] オペレーションズ・リサーチ：" 特集：ＡＨＰ（階層化意志決定法）．" Vol.31, No. 8, (1986).

[Saaty 1980] Saaty, T.L.: "The Analytic Hierarchy Process.", McGrow-Hill, 1980.

[Shafer 1976] Shafer, G.: "A Mathematical Theory of Evidence." Princeton-University Press, 1976.

[Shortliffe 1976] Shortliffe, E. H.: "Computer-Based Medical Consultation: MYCIN." Elsevier, 1976.

[Sinohara 1986] 篠原靖志，寺野隆雄：" 関係の階層化を利用した知識ベース構築支援システム．" 電子通信学会技術研究報告（人工知能と知識処理研究会資料） AI86-32 (1986年12月)．

[Suurikagaku 1987] 数理科学："特集：ファジイ理論とその応用." No. 284 (1987年２

月).
[Terano 1986] 寺野隆雄（他）："ダムゲート診断エキスパート・システムとＡＩＰ." in [OR 1986], pp.500-504 (1986).
[Terano 1987-1] 寺野隆雄：" エキスパート・システム開発ツールの一評価方法．" 人工　知能学会全国大会（第１回）論文集, pp. 257-260, １９８７年６月.
[Terano 1987-2] 寺野隆雄：" エキスパートシステムにおける階層化意思決定法の利用." 情報処理学会人工知能システムの枠組みシンポジウム, pp 55-64, 1987.
[Tone 1986] 刀根薫：" ゲーム感覚意思決定法．" 日科技連, １９８６.
[Waterman 1985] Waterman, D.A.:"A Guide to Expert Systems." Addison-Wesley, 1985.

２．５　仮想知識ベース検索モデル（ＩＣＯＴ　竹之内）

２．５．１　仮想知識ベース検索モデルのねらい

エキスパートシステムを構築するためのアプローチの１つとして，仮想知識ベース検索モデルの考え方を提案する。このモデルは，任意のエキスパートシステムを，問題解決の根拠となるすべての事実群を提供する仮想的な知識ベース（仮想知識ベース）と，仮想知識ベースが提供する事実群を用いて問題解決を行う問題解決プログラムの，２つの構成要素によって構築しようとするものである。その主たるねらいを以下に示す。

①　システム開発並びに保守の容易化

エキスパートシステムの開発や保守を，仮想知識ベースと問題解決プログラムとの２階層に分けて行えるようにし，システム開発並びに保守の容易化を図る。

②　異なるシステム間での知識の共用

仮想事実ベースを問題解決プログラムから独立させることによって，同一の仮想知識ベースを異なるエキスパートシステムの間で共通的に利用可能とする。

２．５．２　仮想知識ベース

仮想知識ベースは，演繹知識ベースの１種であり，それが提供する知識のすべての中身が実体として存在している訳ではないが，ユーザから検索要求があれば該当する知識を演繹して応答することができる，仮想的な知識ベースである。

仮想知識ベースの基本となる知識構造モデルとしては，階層モジュール構造モデルを採用する。これは，仮想知識ベースに含まれ得るすべての知識をモジュール化し，モジュール間の階層的な上下関係によって知識構造を表わすためのモデルである。これらの各々のモジュールを知識モジュールと呼ぶ。

また，階層モジュール構造モデルの基本構造は木構造とし，単一の木構造では表わせないような構造は，複数の木構造の集合および重畳によって表現するものとする。図１に，仮想知識ベースの階層構造の表現例を示す。

知識モジュール間の階層的な上下関係は知識の検索径路を与えるものであり，知識の検索は，木構造を根から葉の方向にトップダウンに辿ることによって行われる。

km11 ○

km21 ○　　km22 ○

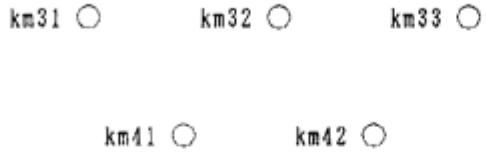


図1　仮想知識ベースの階層構造モデル

2．5．3　知識モジュールの構成

仮想知識ベースを構成する各々の知識モジュールは，次の3種類の知識から成る。なお，知識モジュールに含まれるのは，仮想知識ベースのユーザから直接検索可能な知識群のみとし，例えばある仮想的な知識を導出する際にだけ用いられるルールなど，ユーザから直接検索されない知識は，知識モジュールには含まないものとする。

① 実知識　：　知識の実体が存在している知識。これには，システムの外部から陽に与えられる知識群と，システムが導出によって得た知識群とがある。

② 仮想知識　：　知識が実体としては存在しないが，導出によって得られる可能性のある知識。

③ 要約知識　：　知識モジュールに含まれ得る実知識と仮想知識のすべてを要約するもの。すなわち，ある知識Kが知識モジュール内に含まれる可能性がある場合には，そのKは要約知識上で真でなければならない。

2．5．4　仮想知識の導出

知識モジュールに含まれ得る知識の内，仮想知識に対しては，それを導出するための戦略を与える必要がある。この知識の導出戦略は，ある知識が如何なる知識モジュールに対して如何なる推論手続を適用することによって得られるかを示す，知識フロー図によって表わされる。推論手続には，前向き推論，後ろ向き推論などいくつかの組込み手続が提供され，またユーザ定義の推論手続を組み込むことも可能である。

問題解決フロー図は，図2に示すように，知識モジュールを○で，推論戦略を□で表わし，知識の導出関係にある知識モジュールと推論戦略の間を→で結んだグラフとして記述される。ただし，→の向きは知識の論理的な流れの方向を表わすものであり，必ずしも推論の方向とは一致しない。

図2において，FC および BC は，それぞれ前向き推論戦略および後ろ向き推論戦略を表わすしており，いずれも事実モジュールとルールモジュールの2つの知識モジュー

ルから事実型の知識群を演繹する，システム標準の推論戦略である。

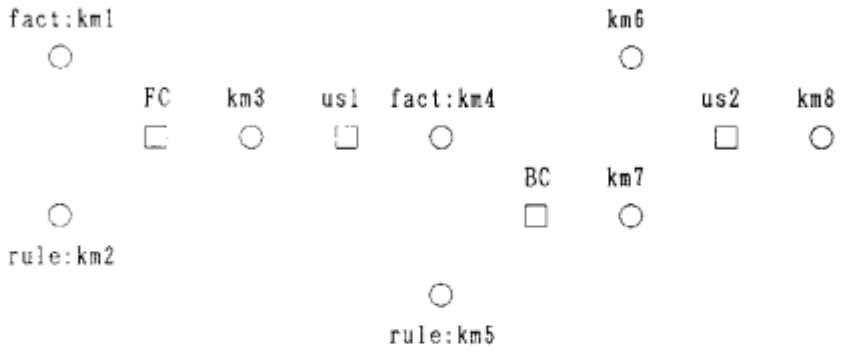


図２　問題解決フロー図の例

仮想知識の導出は，基本的に要求駆動の形で行われる。図２の例で，仮想知識ベースのユーザから"km8" に対する知識の検索要求が出されると，その時点で"km6" と"km7" に対して"us2" の適用が試みられることになり，これによって，"km7" の検索が要求されることになるため，"km4" と"km5" に対して"BC"の適用が試みられる。ただし，前向きの推論手続の出力を利用する推論手続については，その前向きの推論手続の適用を先行させるため，図２の例の場合，推論手続の適用順序は，

us2 → BC → FC → us1

となる。

２．５．５　仮想知識ベースの検索

仮想知識ベースの検索要求は，Prolog と同様に，ゴール節の形で与えられるものとする。ただし，知識モジュール限定子を用いることによって，知識モジュールの検索範囲を指定できるものとする。すなわち，ゴール節の記述形式は次のようなものとなる。

[（知識モジュール限定子）.]（ゴール節）.

ここで，知識モジュール限定子は，ある特定の知識モジュールまたは知識モジュール群を指定するためのものであり，知識モジュール名を上位モジュールから下位モジュールの方向に "." で連結することによって特定の知識モジュールを指示することができる。また，最後の部分を変数にすればそれ以前で指示されている特定の知識モジュールのすべての下位モジュールを指示することもできる。

以下は，図１の階層構造に対する知識モジュール限定子の記述例である。

① km11
② km11.km22.km33

③ km11._
④ km11.km22.X

またゴール節の記述例を以下に示す。

(a) age(taro,X).
(b) legal_inheritor(taro,X).
(c) ancestor(taro,X),blood.same_blood_type(taro,X).

仮想知識ベースの検索は，基本的に，トップダウンの形で行われる。以下に，基本的な検索の手順を示す。

(1) まず始めに，知識モジュール限定子によって指定された知識モジュールの範囲の中から，知識モジュールの要約知識を用いて検索すべき知識を含むであろう知識モジュールの候補の選択を行う。このとき，候補が複数存在していれば，何らかの戦略によって候補を絞り込んだり優先順位の判定を行って，最終的に１つの候補だけを選択する。

(2) 次に，選択された知識モジュールの候補について，実際の知識の検索を行う。ただし，選択された知識モジュールが下位の知識モジュール群から成っている場合には，(1) と同様にして下位の知識モジュール群からの候補の選択を行い，選択した知識モジュールについて知識の検索を行う。

(3) 上記を実行した結果，知識の検索に失敗した場合には，バックトラッキングを行って，外の知識モジュールの候補について知識の検索を試みる。

２．５．６　知識表現言語

知識表現言語は，仮想知識ベースの構造と各知識モジュールの知識内容を記述する知識モジュール定義部と，各知識モジュールの知識の導出戦略を記述する知識導出戦略定義部の２つから成る。この内，知識導出戦略記述部については，前述のように知識導出フロー図を用いて図式的に知識を表現する方法を検討している。したがって，ここでは知識モジュール定義部についてその概要を示す。

知識モジュール定義部では，知識モジュールの名前，その知識モジュールに含まれる実知識，並びに要約知識の記述を行う。知識モジュールの基本的な定義形式を以下に示す。

```
module:〈知識モジュール名〉.
  [ include:〈要約知識〉 ]
  [〈実知識の並び〉 ]
end_module: [〈知識モジュール名〉 ].
```

ここで，〈実知識の並び〉の部分には，下位の知識モジュールの定義を含めることができ，それによって知識モジュールの階層構造を記述することができる。

知識モジュールを構成する実知識には，事実的知識と規則的知識の２種類がある。

事実的知識は，基本的に事実節として表現される。ただし，フレームのように構造をもった知識は，その構造を表わすような特別な形の事実節（これをフレーム節と呼ぶ）によって表現される。以下に事実的知識の表現例を示す。

```
①  age(taro,20).
②  father(taro,ichiro).
③  #human(name:taro,age:20,father:ichiro, …… ).
```

上記の内，③はフレーム節の例である。フレーム節は，述語名の先頭が "#" で始まる事実節であり，述語名はフレーム名を表わしている。また，述語の引数は

〈スロット名〉:〈値〉

という形式で記述され，引数の記述順序は任意であるし，省略も可能である。すなわち，フレーム節の識別子を "id" とすれば，フレーム節は各引数に対して

〈フレーム名〉(id ,〈スロット名〉,〈値〉).

なる事実節を記述したものと論理的に等価である。

規則的知識は，基本的にルール節として表現される。ルール節は，通常のプロダクションルールとは異なり，結論部には高々１個のアトムしか記述することができない。以下に，ルール節の記述例を示す。

```
④ mortal(X) :- human(X).
⑤ ancestor(X,Y) :- parent(X,Z) , ancestor(Z,Y).
⑥ legal.inheritor(X,Y) :- blood.ancestor(X,Y).
```

上記のように，ルール節の記述形式は，基本的に Prolog と同様である。ただし，カットおよび assert や retract は使用できない。また，逆に，⑥の例のように，項に知識モジュール限定子が付加できるように拡張が施されている。この知識モジュール限定子を使用することによって，参照すべき知識モジュールの範囲を限定したり，演繹される知識の有効範囲を限定することができる。

なお，ルール節は，可能な推論の規則を表わすものであり，結論部に記述されるアトムは演繹される事実を示すものでなければならない。したがって，”もし 〜 ならば 〜 せよ” というような戦略的な知識は，規則的知識として記述するのではなく，これとは別に記述する必要がある。

要約知識の記述形式は，基本的に知識モジュールの事実節に対してメタ変数を導入したものである。メタ変数とは，その値がある知識であるような変数のことである。メタ変数は，"@" で始まる任意の英数字列によって表わされる。以下に，要約知識の記述例を示す。

イ．age(taro,20).
ロ．age(@name,@age).
ハ．@attribute(taro,@value).

上記の例の内，イは，知識モジュールに含まれる知識そのものを記述しているものである。またロは，"age" を述語名とする２引数の事実節で表わされるような知識が含まれることを示し，ハは，"taro" を第１引数とする２引数の事実節で表わされる知識が含まれることを示している。

なお，要約知識は，単にそれによって示された形式をもつ知識がその知識モジュール内に含まれ得る可能性を表わすものであり，該当する知識が本当にその知識モジュールに含まれることを保証するものではない。

仮想知識ベースの階層構造において上下関係にある知識モジュール間では，下位の知識モジュールは上位の知識モジュールから知識を継承することができる。なお，知識モジュール限定子を用いることによって，上位の知識モジュール内で，継承させるべき知識の範囲を限定したり，継承先の知識モジュールの範囲を限定するといった制御が可能である。

知識モジュール定義において，（知識モジュール名）の部分が知識モジュール限定子として記述されている場合には，そのモジュールのすべての知識は知識モジュール限定子によって指定されたすべての知識モジュールの共有知識となる。したがって，例えば知識モジュール "km" のすべての知識をそのすべての下位知識モジュールに継承させる場合には，知識モジュール限定子を "km._" （ "_" は匿名変数）とすればよいし，知識を１つの知識モジュール "km1" だけに継承させるのであれば，その知識モジュール名を連結したものすなわち "km.km1" なる知識モジュール限定子を記述すればよい。

また，知識モジュール限定子を，知識モジュールに含まれる個々の知識に付加した場合には，それが付加された各々の知識の単位で知識の継承が行われることになる。したがって，各々の知識の単位に知識の継承先を規定することができる。なお，知識モジュール限定子が付されていない知識は，下位の知識モジュールに継承されない。

２．５．７　今後の検討課題

今後の検討課題としては，次のようなものがある。

(1) 問題解決戦略についてのメタ知識の導入

(2) 構造型知識に対する推論方式

(3) 問題解決フロー図の記述仕様の詳細検討

(4) 問題解決フロー図作成用グラフィックインタフェース

以　上

2.6 AIﾂｰﾙにおける知識表現　（NTT　石田）

2.6.1 概要

本節では商用に供されているｴｷｽﾊﾟｰﾄｼｽﾃﾑ構築用ﾂｰﾙにおける知識表現，特にﾃﾞｰﾀの表現を概観する。これはﾙｰﾙの表現に関しては比較的よく知られているのに対し，ﾃﾞｰﾀの表現に言及した報告が極めて少ないためである．調査対象は以下の3ｼｽﾃﾑである。

① KEE:　unit
② ART:　schema
③ Knowledge Craft (KC):　schema

調査はARTを除いてはﾏﾆｭｱﾙを対象として行った。ARTに関しては、symbolics上で機能の確認を行っている。調査項目を以下に示す。

① 基本構成に相違があるか。
② System Defined Relationとしてどのようなものが用意されているか。
③ User Defined Relationは定義可能か。その際、
　(a) 関係が成立するための制約条件(Restriction)をいかに指定するか。
　(b) 関係による値の継承機能(Inheritance)をいかに指定するか。

上記の項目はいずれも、ﾌﾚｰﾑ機能だけを対象としたもので、ﾙｰﾙやﾏﾝﾏｼﾝｲﾝﾀﾌｪｰｽ等は調査の対象外である。従って、必ずしも調査した範囲での機能の豊かさが、ﾂｰﾙの良し悪しを決定するものではない。調査結果の概要を以下にまとめる。

① 全般に『ﾂｰﾙは知識表現に必要な言語機能とﾒｶﾆｽﾞﾑを提供するもので、ﾂｰﾙを用いていかに表現するか、また表現された知識のｾﾏﾝﾃｨｸｽの保証は利用者の責任である』という考え方が支配的である。（意味表現に非常に敏感な利用者は、これらのﾂｰﾙをﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語だと割り切って考えた方が精神衛生に良い。）

② KCとARTはよく似ている。ﾌﾚｰﾑ機能に関しては、ARTはKCのｻﾌﾞｾｯﾄと考えてよい。KEEとKCとは以下の点で異なる。
　(a) KEEはunit中心（object指向）である。例えば、is-a関係によって遺伝するslot(member slot)と遺伝しないslot(own slot)が存在するが、同名のslotでも、あるunitではmember slotであり、別のunitではown slotであるというように、unit毎にslotの遺伝の有無が決定される。またslotがとり得る値の制約条件はunit毎に異なる条件を指定できる。
　(b) KC､ARTはslot中心（relationはslot）である。即ち、意味ﾈｯﾄﾜｰｸの考え方に近い。例えばis-a関係によって、遺伝するslotと、遺伝しないslotはslotの定義に従って決定される。どのschemaでもこのslotの性質は変わることはない。またslotがとり得る値の制約条件は、あくまでもslot毎に指定するものであって、schemaが異なっても、同じ制約条件が適用される。

③ 全般に言語機能が直交していない。特に、user defined relationに継承機能を導入した、KC,ARTにこの傾向が強い。これは、これらのﾂｰﾙがたとえﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語であると考えたとしても、まだ未成熟な段階にあることを示している。

2.6.2 基本構成

図1、2に、KEEとKCの基本構成を示す。両ｼｽﾃﾑ共にschema(unit)-slot-valueの階層を基本としている点では共通であるが、meta-informationをどのように書くかという点で方式が分かれる。

① KEE slot-facet-valueという階層をとっており、1つのslotに対し、複数の情報が記入できる。図3にKEEのfacetを示す。valuc-facetはslotの値を保持する。他のfacetはslotのmeta-informationを保持するためのものである。

② KC schema,slot,valueにそれぞれ、meta-informationを記述するためのmeta-schema,meta-slot,meta-valueを付加できる。（特にmeta-slotをfacetと呼んでおり、この点ではKEEと呼称が一致している。）
meta-informationとしては、以下のものが記述できる。
- 値が、how,when,where,why,生成されたか。
- 値の制約条件

ARTでは特にmeta-informationという考え方は強く打ち出されていない。しかし、slot(relation)の制約条件等は、各slot-schema（図8参照）に記述されるので、これがmeta-informationを保持していると考えることができる。

2.6.3 System Defined Relation

3ｼｽﾃﾑ共に、is-a,instance-of（ｼｽﾃﾑ毎に呼称は異なる）をｻﾎﾟｰﾄしている。しかし、is-a、instance-ofによって継承される情報と、されない情報を表現する方式がKEEとART,KCとは大きく異っている。以下では、KEEとARTを例にとってその相違を説明する。

(1) KEE

KEEでは、parent unitからchild unitへ値が継承する。parent-childの関係としては、以下の2種がある。
①subclass：is-aに相当する。
②member　：instance-ofに相当する。
一方、slotには以下の2種があり、parent-child関係によって継承されるか否かが異なっている。
①member slot：classのmemberに共通の情報を格納するためのslot。継承される。
②own slot　：そのunit固有のslot。継承されない。
図4はKEEにおける継承を表わしている。関係subclassによって継承されたmember slotはmember slotのままであるが、関係memberによって継承されたmember slotはown slotに変化する。（KEEではmemberのmenber

を作ることは、禁止されていない。このﾒｶﾆｽﾞﾑによって、memberのmembcrへ値が継承されることが防止されている。）

(2) ART

ARTやKCのslotにはmember slot,own slotの区別はない。is-a,instance-ofによって継承される情報と継承されてはならない情報（例えば、そのclassに含まれるinstanceの個数）はschemaを分けて記述する。以下図5に示すARTの方式をもとに説明する。

①is-a,instance-ofの階層とsubset-of,elemeat-ofの階層を区別して考える。継承はis-a,instance-ofの階層についてのみ生じる。そこで、継承されるべき情報は、is-a,instance-ofの階層に位置するschemaに格納する。継承されてはならない情報は、subset-of,element-ofの階層に格納する。

②両方の階層を橋わたしするのが、関係prototypc-ofである。例えば図5で、movie-stars-prototype-0は、集合movie-starsのprototype（代表元）であり、集合の代表的な値（例えば、schema「人間」のslot「手」の代表的な値は「2本」）が格納される。

(3) 両方式の比較

KEEはobject指向言語の側面が強い。member-slotは、instance-variableに、own-slotはclass-variableに対応する。ATR,KCは、意味ﾈｯﾄﾜｰｸからの影響が色濃くでている。機能的には、両方式ともほぼ同等である。

2.6.4 User defined Relation

利用者による関係の定義は、ART,KCのｾｰﾙｽﾎﾟｲﾝﾄの1つである。利用者は関係が成立するための制約条件(restriction)と関係による継承の規則を定義できる。KEEでは利用者による関係の定義という考え方はない。しかし、slot毎に値の制約条件を記述することはできるので、これらの機能に関しては比較できる範囲でART,KCと対比して議論する。

2.6.4.1 制約条件の指定

制約条件は、どのような場合に関係が成立し得るかを規定するものである。

(1) 制約条件の指定単位

ART,KCではslot(relation)毎に制約条件を指定する。一方KEEでは、unit毎に異なる制約条件を同名のslotに対して指定できる。即ち、ART,KCではslot(relation)はglobalなものだと考えられている。schema毎に、その性質が変わることはありえない。一方KEEでは同名のslotでもunit毎にその意味が変化してよい。object指向言語では、同名のmethodはobject毎に異なる役割を演じる。KEEのslotはこの性質を引き継いでいると考えることができる。

(2) 制約条件の種類

KEE、ART、KCともほぼ同様である。

①domain／range

KCではdomain／rangeを共に指定できる。KEEではslotのとり得る値(value class)を指定できる。図6 にKEEのvalue classの例を示す。 KCのdomain／range指定も形式は異なるが、ほぼ同様の制約を記述することができる。ARTではdomain／rangeの記述はできない。

②cardinarity

KEE、KCではslotに格納可能な値の個数のmaxとminを指定できる。ARTではsingleとmaltipleの区別があるだけである。

(3) 制約条件の遺伝

KEEではparent unitのslotで指定された制約条件が、child unitのslotに遺伝する。maltiple inheritanceの時には制約条件のintersectionがとられる。ART、KCでは、slot名が同じであればどのschemaでも制約条件が同じであるので、遺伝という考え方はない。

2.6.4.2 継承機能の指定

各ﾂｰﾙの継承機能の指定方法を以下の2種の観点から整理する。

①関係によって、何がどう継承されるのか？

②関係は他のどのような関係から導びかれるのか？

①と②は直交する機能ではない。①によって関係がinheritされる場合には、①と②は同じ機能を逆の観点から述べているにすぎない。

(1) 関係によって何がどうinheritされるのか？

①ARTの関係の階層を図7に示す。ARTではinh-relationのinstanceである関係（例えば、is-a、instance-of、それにuser-defined-relation）によって、継承が生じる。inh-relationによって指定slotが継承されるかどうかは、そのslotのslot-schemaのslot-how、slot-whatによって指定できる。図5.8にARTのslot-schemaを示しておく。（ARTでは関係毎に、どのslotを継承するかを指定することはできない。）

②KCでは、関係毎に固有のinheritance-specを書くことによって、ARTに比べてよりきめ細かな継承の制御が可能となっている。inheritance-specの階層を図9に、その例を図10に示す。inclusion spec、exclusion specによって、どのslotが継承されるか／ されないか、map spec、elaboration spec、introduction specによってどのようにslotやvalueが変化して継承されるかを指定できる。

③KEEではuser-defined-relationという考え方がなく、従って関係固有の継承を定義することはできない。3節で述べたように、parent-child間、（即ち、subclass、memberの関係）で継承が生じる。継承の制御は、継承される側のunitの各slotに、override（継承値にそのunitの値が上書きされる）union（継承値とそのunitの値の双方が有効となる）unique（継承値をうけつけない）等を指定することによって行う。KC、ARTが関係毎に、継承機能を定義したのに対し、KEEではunit毎に継承の働きを制御する点が異なる。

(2) 関係は他のどのような関係から導かれるか？

①ART、KCでは関係のtransitivityを指定できる。図11に、ARTの例を示す。repeat、step等を使って関係をproceduralに辿り、新たな関係を導く機能である。
②ARTではさらに関係生成時に、他の関係を同時に生成するnew-relation定義機能がある。図12にこの例を示す。この機能は①と重複が著しい。
③KEEでは上記の機能はない。

参考文献

[KEE 85] KEE 2.1 Software Developmeut System Reference Manual, Intelli Corp. (1985)
[ART 86] ART 2.0 Reference Manual, Inference Corporation (1986)
[KC 85] Knowledge Craft 2.0 Technical Manual, Carnegie Group (1985)

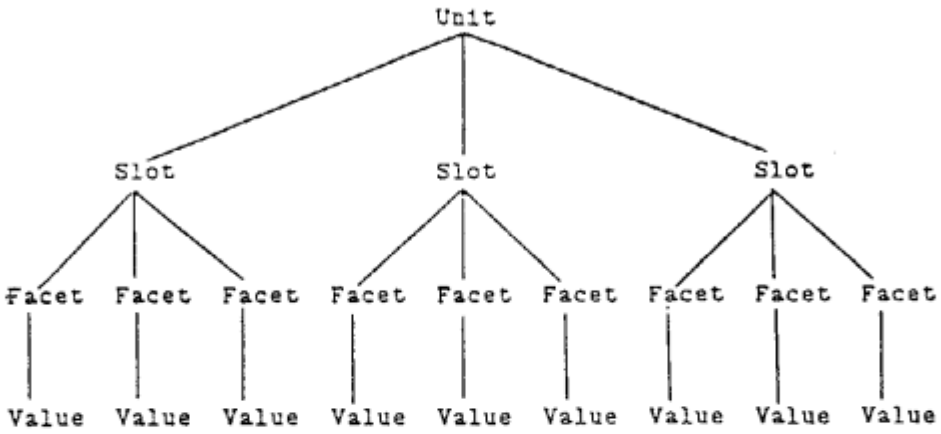
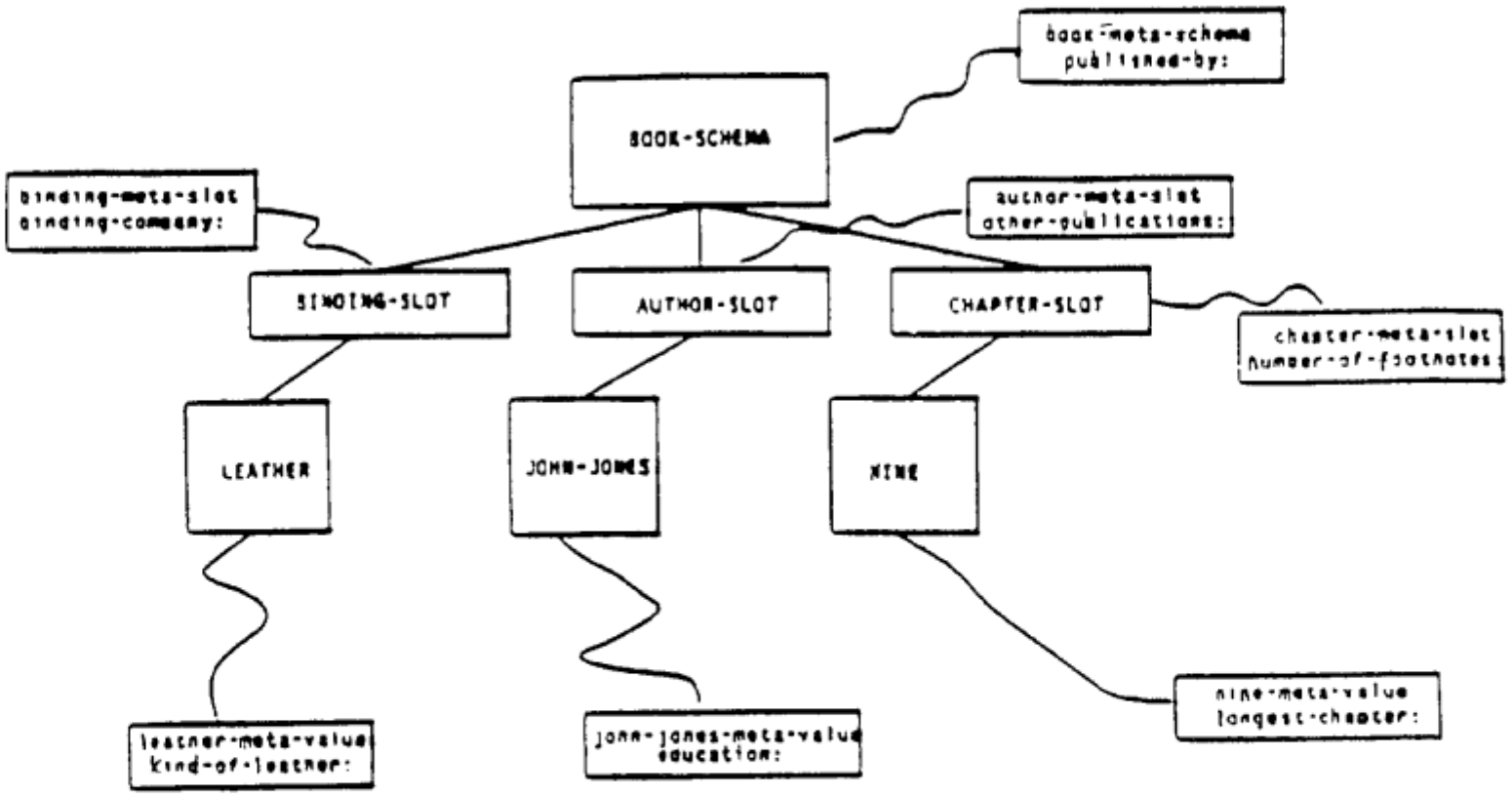


図 1 KEEの基本構成

- Comment – An optional textual description.
- Inheritance role – This specifies how the slot will inherit attributes from units above it in the hierarchy.
- Value class -- This facet specifies what kinds of values will appear in the value facet.
- Cardinality max -- This specifies the maximum number of values that can appear in the value facet.
- Cardinality min -- This specifies the minimum number of values that can appear in the value facet.
- Value -- This is where the value or values of the slot are stored.

図 3 KEEのFacet



This diagram depicts the stucture
of a schema that has meta-information
associated with all its slots and values.

図　2　ＫＣの基本構成

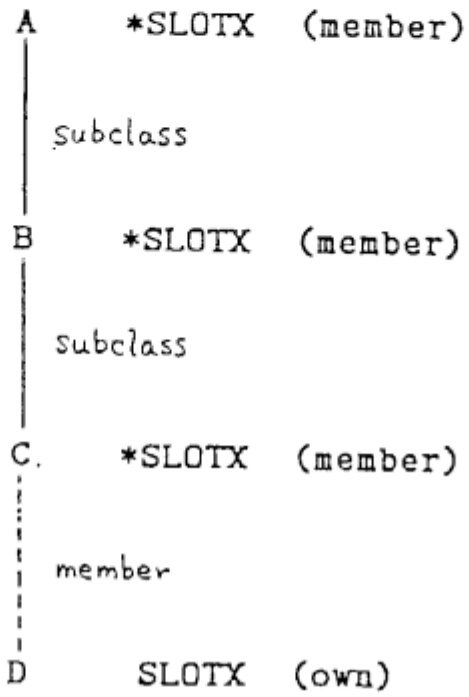
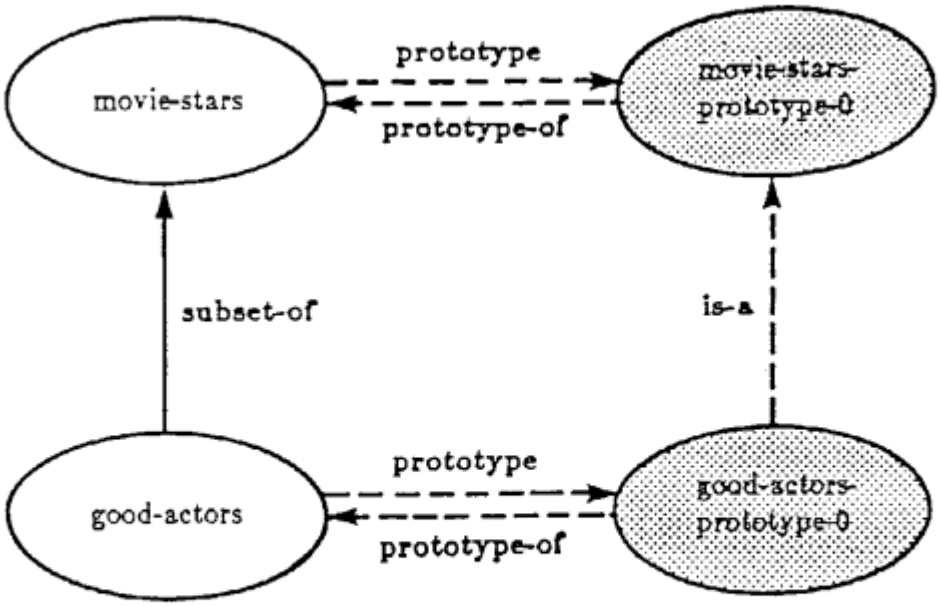
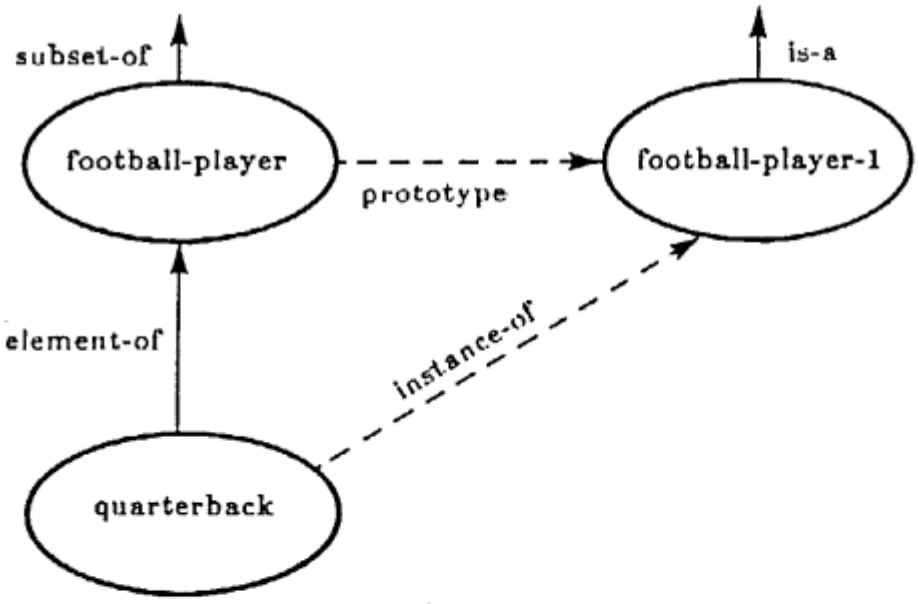

A
*SLOTX (member)
subclass
B
*SLOTX (member)
subclass
C
*SLOTX (member)
member
D
SLOTX (own)

図　4　KEEの継承機能



(1)



(2)

図　5　ARTの継承機能

<class unit>		indicates the class of which the value must be a member unit.
<KEE interval>		indicates an interval of any ordered type.
(ONE.OF	<any object>..)	indicates members. The value must be one (or more!) of the objects.
(NOT.ONE.OF	<any object>..)	indicates members which are not allowed. The value cannot be one of the objects.
(LIST.OF	<set>)	produces the set of all lists which have as elements members of <set>. <set> can be any value class operator, as with INTERSECTION, UNION and NOT.IN.
(NOT.IN	<class>....)	indicates classes which are not allowed.
(UNION	<class>....)	indicates union of existing classes. The value can be a member of any of the value class specifications.
(INTERSECTION	<class>....)	indicates intersection of other classes. The value must be a member of all the value class specifications.
(SUBCLASS.OF	<class unit>)	indicates the set of all subclasses of <class unit>.
(MEMBERP	<function>)	indicates a membership predicate.
(MEMBER.OF	<class unit>)	indicates the set of all members of <class unit>.
(UNIVERSE)		indicates the universal set. (Default)
(EMPTY)		indicates the empty set.

<class> can be any value class specification.

図 6 KEE の value class

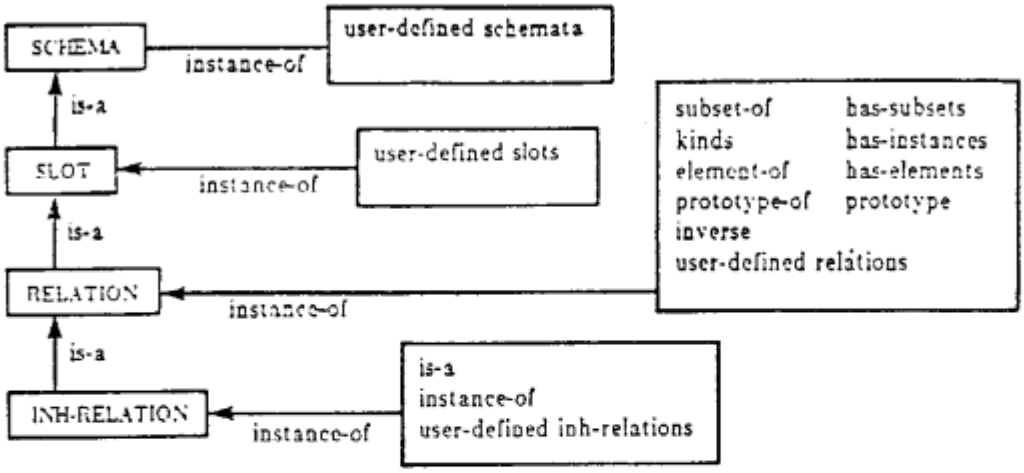


図 7 ARTの関係の階層

```
(defschema Slot
    (slot-input-output (The ?slot of
        ?schema is ?value))
    (slot-output)
    (slot-input)
    (slot-how  definite)
    (slot-what  share-value)
    (slot-how-many  single-value)
    (slot-multiple  prompt))
```

Each primitive slot controls a specific characteristic of slot behavior:

- **Slot-input-output** establishes a natural language syntax for the slot, to be used for both input and output.
- **Slot-input** establishes a natural language syntax for input only. More than one is permitted.
- **Slot-output** establishes a natural language syntax for output only. Only one is permitted.
- **Slot-how** determines whether the slot is inherited definitely or hypothetically.
- **Slot-what** offers a special mechanism for inheriting a "template" of the slot. It can also be used to inhibit or enable direct inheritance.
- **Slot-how-many** determines whether a slot can hold one value or many values.
- **Slot-multiple** determines what ART will do when multiple parents are trying to pass different values to a single-valued slot in a child schema.

図 8 ARTの Slot Schema

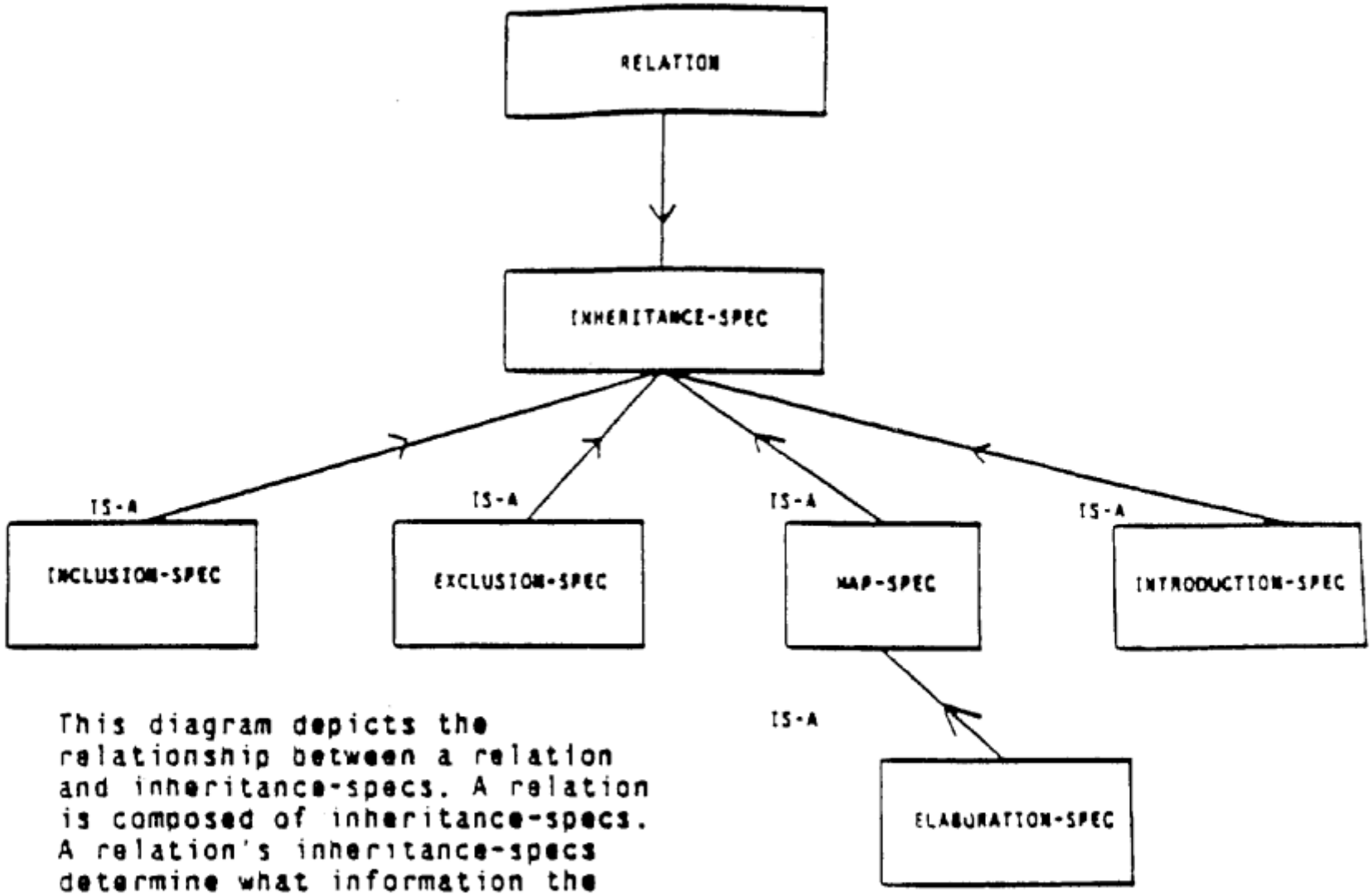
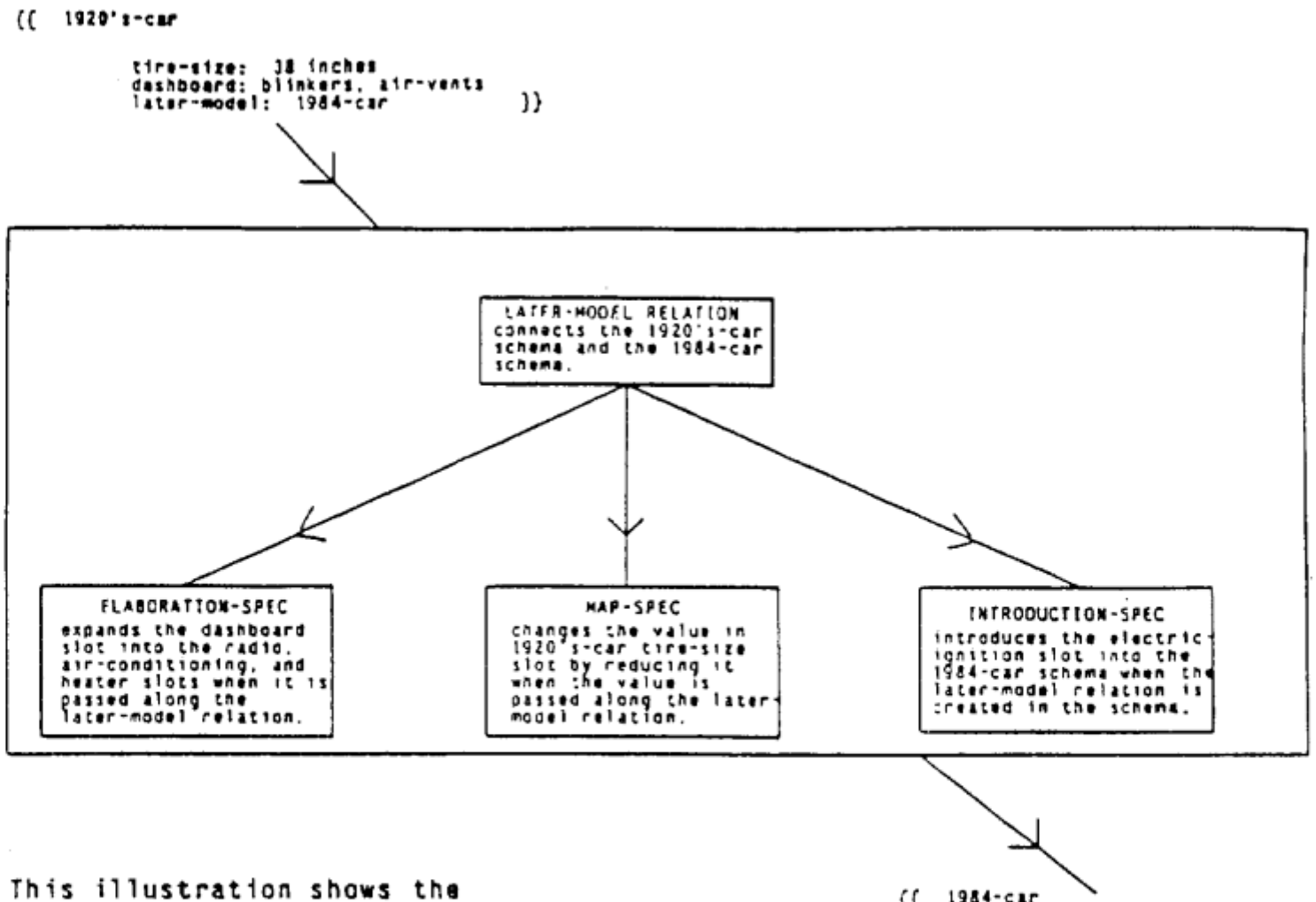
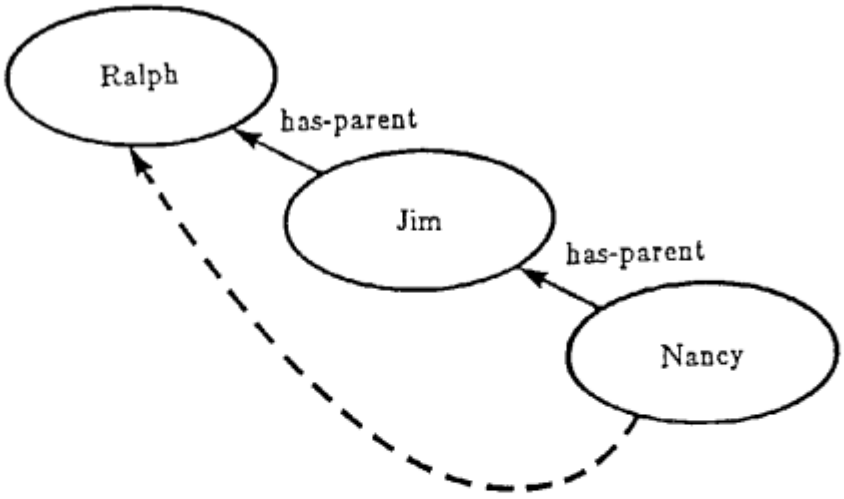


図 9 KC の Inheritance Spec



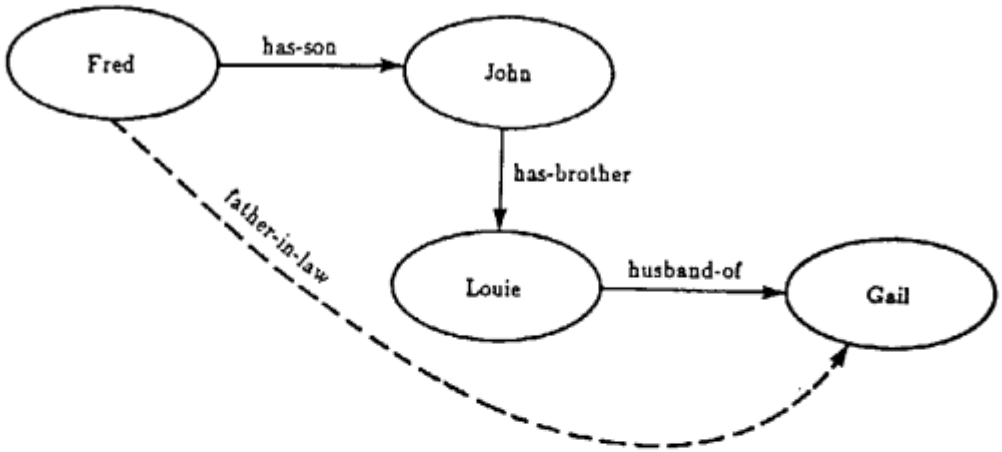
This illustration shows the structure of the later-model relation. The later-model is composed of three inheritance-specs that change the information being passed from the 1920's-car schema to the 1984-car schema.

図 10 KCの継承例




```
(defschema has-grandparent
    (instance-of relation)
    (inverse grandparent-of)
    (transitivity
        (repeat (step has-parent $) 2 2)))
```

図 11 ARTの関係のtransitivity




```
(defschema has-son
    (instance-of relation)
    (new-relations
    (father-in-law (?domain)
        (?range has-brother husband-of))))
```

図 12 ARTのnew-relation

3　知識獲得と学習

３－１ 問題解決と知識学習に関する一般的考察

京都大学工学部精密工学科
椹木 哲夫

3-1-1 はじめに

従来より、機械学習の定義・分類としては、学習プロセスで重要な機能を担う「推論」の視点から類推学習・帰納学習・演えき学習の分類がなされており、各分類のもとで、個別のアルゴリズム開発が独立に進められてきた。最近の機械学習の動向を眺めるとき、これらの手法が互いに組み合わされ融合されたシステム化の方向に向かうと共に、この学習システム自体の目的も、単なる知識ベース構築のための知識獲得支援のための方法論から problem solving や reasoning のプロセスと不可分なものになりつつある。また、より現実的な観点のもと、従来の多数事例からの学習に代わり、少数・単一事例からの学習へと移行し、バージョン空間を探索する際の経験則（バイアス）の受動的な繰り返し適用による「発見型学習」から、学習対象についての領域固有知識に基づき能動的な仮説設定や知識の更新を通して学習事例を説明づけようとする「解釈型学習」（理解に基づく学習）へと移行してきている。そこでは、これまでの画一的な知識表現のもとでの帰納的な一般化パターンの導出に対して、個々の事例に関する、構造、機能、目的概念に関する知識の多重表現を基盤とし、所与または設定された仮説からの演えき推論による知識の変換（knowledge transfer)をも含めた一般化を目指すものである。即ち、多数事例からの平均操作的な学習形態から、個々の事例のもつ「個性」を保持した上での一般化であること、また、一連の閉じた（完結した）procedure としてではなく、ひとつひとつの事例の処理に伴う既有知識の再組織化や同化等、動的な更新を意識した適応的な拡張性の２点が近年の機械学習の特徴として要約されよう。本稿では、これら近年の機械学習の動向について、「説明に基づく学習」について述べる。特に、Mitchell らに代表される「証明木生成に基づく説明学習」（以下、ＥＢＬ（Explanation-Based Learning）と呼ぶ）と、Schank 等に代表される認知科学的アプローチとしての「事例記憶に基づく説明学習」の両者の考え方について概説する。さらにこれら学習手法の概念的背景並びに従来手法との関連性について、類推（analogy）、類似性に基づく学習―ＳＢＬ（Similarity-Based Learning）、事例推論（Case-Based Reasoning）の諸点からまとめる。

3-1-2 類推と因果知識

類推の定義としては、

『いくつかの対象の間に類似性（類比）を発見し、その類比により、一方で成立する事実や関係などの知識を他方の対象に対して変換することにより、問題解決の手がかりを得たり、未知の事実を予測推定する推論方式の一種。』（原口[1]）

として与えられる。このような類推機能を、計算機上で実現するためには、与えられた表層的な属性集合表現に基づく、機械的な類似性比較だけでは不十分であって、その背後にある機能・目的まで遡った上で、何が等価であり、何が重要か、

といった評価基準を導入した類似性尺度が必要になる。また、人間が実際に類推を行っている場合には、既知の多数の関係や属性の中のごく一部の特定のものだけが写像され、かつその際、既知のどの事実・関係が、未知のどの事実・関係に写されるかを直感的に捉えているが、このような写像され易い関係とされにくいものとを分ける基準、さらには、これら写像の対応関係の組合せ的爆発を回避するための手段が明らかにされねばならない。

Gentner[2] は、認知心理学の立場からその構造写像（structure-mapping）理論のなかで、類推の本質は個々の独立した事実を写像するのではなく、個々の事実の間に存在する関係を写すことにあるとし、因果性など高階の関係で結ばれている関係が類推によって写像され易いとするシステム性の原則（systematicity）、さらに対応する事実間の表層的・構造的な類似性が高いほど写像され易いとする透明性の原則（transparency）を唱え、子供の物語再生実験により実証している。

Winston[3,4] は、「類似性は因果関係を保持する。即ち、似た状況は似た結末を生じやすい」という立場に立ち、対応関係のついた事実群がある時、既知事例（precedent）内で成立する因果関係が写像されることによって、未知事例（exercise）上での関係の存在を推論するシステムを構築している。ただし、Winston のいう因果関係とは、あくまで general なものとして確立された「因果則」とみなすよりは、たまたま既知事例の中で成立している因果的に関係づけられるものとして捉えるのが妥当であり、因果性概念の内包（intesion）ではなく外延（extension）として見なすことができる（従って、既知事例中の因果関係からの演えき推論は行っていない）。

Russell[5,6] は、既知事例の中に、多数の関係・属性がある場合に、その中のどれが類推の対象になるかについて、外部から explicit に与えるべきだという観点にたち、決定ルール（determination rule）を導入している。ただしここでのルールは、Winston の因果関係よりは普遍的な定義になっているものの、「因果則」（strong causal theory）ではなく、より弱い（weak な）意味での因果的関連づけであるとしており、前者の知識の存在を前提とするＥＢＬ（後述）とは異なるものとして位置づけている。

以上の考え方は、類推を進める上で、その質の違いこそあれ、いずれもある種の高階の因果的構造の存在を既知事例上に仮定したり与えることが必須であることについて共通した考え方である。しかしながら上述のような因果的構造が、既知事例の上で explicit に見いだしたり、与えたり出来ない場合に、この構造を domain-specific な知識に基づいて、システムに自ずから推論、生成させるという考え方が、以下で述べるＥＢＬの考え方である。目的に応じて、類推の対象として焦点化される属性が変わる、即ち、既知事例中の属性同士が因果的に関係付けられていることが、表層的・構造的な表現だけからは見いだすことが不可能で、更に上位の機能・目標概念まで遡らねば明らかにならない場合に、これら異なる知識表現の間の関係を規定した領域固有知識に基づいて、「説明（explanation）」として生成する。

3-1-3 証明木生成に基づく説明学習（ＥＢＬ）

3-1-3-1 ＳＢＬとその限界

帰納学習の典型的な問題として古くから研究が進められてきたものに「（複数）事例からの学習」がある。ここでのタスクは、目標概念に対する正事例を説明し、負事例を除外するような一般化記述を決定することである。ここでは、学習事例の正負の判別が学習システム外部から教師により与えられる場合（教師つき学習・概念獲得）と、学習システムが与えられた事例群をいくつかのクラスに分類していく途上で、該当クラスの事例を正事例、それ以外の事例を負事例とみなして自律的に振り分けていく場合（教師なし学習・概念クラスタリング）の２種類の学習手法が存在する。

これらの学習手法に共通するのは、一階述語論理・セマンティックネットワーク等により表現された事例間の類似関係に基づいて一般化が進行する点にある。即ち、同じクラスに属する事例の間で共有される特徴は出来るだけ保存し、相違する特徴についてはこれらを無視するか、違いを包括するような概念への統合により一般化を行う。通常、このような一般化操作は、必要最小限の一般化（表現の具象性を最大限保存した一般化）を基本操作として、これを再帰的に繰り返すことによって実行される。その結果、各事例固有の具体的記述から、与えられた同一クラスに属するすべての事例のみならず数多くの未観測事例をも包含するような過剰に一般化された記述にわたるまで、さまざまの一般化レベルの記述が生成される。学習システムの目的は、このようなレベルの両極を下界・上界として、その間に展開される探索空間（バージョン空間）の中から、概念の「適切さ」を規定した評価基準を満足するような一般化記述を決定することになる。このようなバージョン空間の探索には、実際の応用的観点からは、画一的な一般化操作では効率が悪く、このため類似性に基づく学習の現実的な応用を考える際には、以下のような制約が課せられる。

①バージョン空間が出来るだけ小さいこと
②学習すべき概念と、訓練事例が同一の知識・シンタックスにより表現されること
③バージョン空間探索の際の経験的・発見的バイアス・評価関数がインプリメントしやすい形で与えられること

これらの制約を満足させるため、従来の類似性のみに根拠を置く機械学習においては、まず①については、原則として disjunction を認めない、②については、学習概念は簡潔明瞭なものに限る（その結果与えられる正事例は互いに良く似たもの同士であり、かつ負事例からは容易に区別されるようなものにならざるをえない）、そして③の制約に対しては conjunction や disjunction の項数の少ないものに対する優先権を与える、ことにより上記制約を満足させてきた。

しかしながら、本来、概念の「自然な」知識表現はその目的に応じて変わったものになってくる。例えば、ロボットビジョンで扱うような、効率的な認識を目

的とする場合には、構造的（structural）な表現が、またプラニングを目的とする場合には機能的（functional）な表現が、シミュレーションや予測を行うためには挙動的（behavioral）記述が自然な知識表現であると考えられる。

学習プロセスを考える場合、このような知識表現の多重性は、より顕著なものになってくる。即ち、訓練事例の知識表現（Environmental Representation）、獲得されるべき学習概念を定義する知識表現（Learning Representation）、またこの学習概念を実際に新たな事例に対して適用をする際の知識表現（Performance Representation）が考えられるが、これらは互いに異なったものとなるのが通例である。

3-1-3-2 ＥＢＬのプロセスと特徴

ＥＢＬ[7]は、従来の帰納学習手法の多くが、多数の事例からのバージョン空間の絞り込みを、経験的なバイアス・ヒューリスティックスに基づいて行ってきたのに対して、既有の domain-specific な知識（domain theory）と、外部から与えられた学習すべき概念定義（goal concept）に基づいて、単一事例（training example）からの一般化学習を進めるための手法である。即ち、ある事例が何故に、学習概念の一事例となり得るのかについての説明（explanation）（＝論理的証明木）を、既有知識に基づいて生成する。この説明構造から一般化記述を導出するには、modified goal regression アルゴリズムが用いられる。これは、その事例における変数の束縛関係を goal concept から事例を表現する属性の集合（通常、この表現は、goal concept の表現に用いられたものとは異なる知識表現で、operationality criteria としてアプリオリに規定される）に逆に辿ることによってとらえ、その結果、goal concept が成立するために本質的に必要な属性間関係だけをとりだすことになる。即ち、得られる一般化記述は、goal concept の十分条件の一つに過ぎないが、抽象的な定義として与えられた goal concept の内容を、事例表現に用いられた表現力豊かな、操作化の可能な operationality criteria を満足する記述により再表現されることになる。（図１）

ＥＢＬの特徴としては以下のようにまとめられる。

・事例を表現する属性の中には、与えられた goal concept、即ち学習の目的に応じて関連性をもつものやもたないものが含まれる。ＥＢＬでは、説明を生成・解析することによって、すべての属性間の組合せを考慮することなく、本質的に必要な属性だけにフィルタリング・焦点化を行う。

・goal concept を満足するすべての事例を記述する一般化表現を導出するものではない。goal concept の十分条件の一つに過ぎず、かつそれは与えられた事例に大きな制約を受けた表現である。しかしながら、このような訓練事例への依存性が、逆に非現実的な実際の環境下では観測されるはずのないような一般化表現の生成を回避せしめている。

・生成される概念の generality のレベルは、domain theory の generarity に依存して決まる。また学習の performance は、domain theory に新たな知識が学

習付加されるに従って向上する。
・生成されるものは、全く新たな概念の獲得ではない。あくまで既知のこと（domain theory）から、演えき的に導かれる結果をコンパイルした事例記述の再表現にすぎない。

3-1-4 ＥＢＬの指針

3-1-4-1 多元的知識表現――知識の転移（knowledge transfer）――

知識獲得・知識整理のシステム化の要請のもと、ＩＳＭやＤＥＭＡＴＥＬ、ＡＨＰさらにはＫＪ法等、元来、悪構造を呈する問題複合体を対象とした意思決定支援、モデル構築支援のためのシステムズ・アプローチとして発展を遂げてきたシステム工学手法の適用が近年盛んに報告されている。即ち、専門家の有している専門的知識も、このような問題複合体同様、直接的にはルールやフレーム等の良構造知識への変換が困難な悪構造を呈しており、専門家の知識構造を随時、これらの手法を用いて構造化・可視化することにより、専門家自身の自らの有している知識に対し再認識を促し、インタビューやアンケートによる知識獲得のプロセスをシステマティックかつスムーズに進行させるという点に、これらの手法の導入の狙いがある。ただし、これらの構造化モデリングの手法は、あくまで専門家（決定者）の直感的判断を刺激したり、知識内容（問題対象）の理解・把握を促進させることが目的であるが、より創造的な知識の獲得支援手法として、自ずから限界が生ずる。

このような限界の第一の理由は、構造化モデリングの名が示す通り、対象のもつ構造を表現するには長けているものの、その意味内容（セマンティクス）を扱うための枠組みが手法の中には一切組み込まれていないこと、第二に、このような構造情報は、通常複数種類のものが相互に有機的に関係付られて意味を持っているにも拘らず、構造化モデリングを適用するには、これら構造関係のいずれか一つに絞り込まねば効力を発揮するものではないことがあげられる。特に、知識獲得のプロセスの支援を目指すならば、画一的な知識表現では不十分であり、多元的な知識表現が不可欠であることは言うまでもない。これら多元的知識表現を統一的に扱うことが出来ると同時に、これらの異なる表現の間の縦割的な橋渡しを実現するための手法が望まれることになる。このような機能が満足されてこそ、画一的な照合操作による事例の検索から、多面的な視点にたった情報整理・概念レベルでの知的検索が可能になると考えられる。

一方、問題解決に要請される知識については、決して一様なフラットな知識空間を呈しているわけではなく、知識のもつ領域特殊性には以下のような違いが見られる。即ち、通常頻繁に出くわすような問題解決過程においては、外部から与えられた問題記述表現から、問題解決プロセスに指向した内的表現に変換するための知識、また解決手順については、人間の持っている経験的知識が手続き化され、これが文脈に応じて働くわけであるが、これらはいずれも問題領域に特殊な知識である。これに対し、未知の問題に出会ったときに、経験的な手続きが利用できない場合には、試行錯誤や手段－目的分析、仮説生成テストなど、ごく弱い

一般的方法に頼りながらなんとか対応することができる。この際に必要となる知識は、上述のような領域特殊知識を獲得したり修正したりするためのメタ認知的知識、あるいは情報の記憶・因果性分析・チャンク化手続きを規定した知識のように領域に独立なものである。このように問題解決過程の深層には、様々なレベルでの領域特殊性をもつ知識が介入しているにも拘らず、熟練者から得られる訓練事例では、これらが適用された結果として、ある単一の記述表現に従って、表層レベルで表出されたものであり、事例からの問題解決知識の獲得を困難なものにしている。従って、事例からの知識獲得のためには、まずこの事例の中からこれらの上述のような各種の知識が働いている部分を認識し、切り出すことがその前提となるが、この点においてＥＢＬの果たす役割が考えられる。

ＥＢＬによって生成される知識は、訓練事例に大きく依存した一般化記述であり、goal concept として定義された学習概念の十分条件に過ぎないものである。しかしながらこれらの特徴は、上述のような専門家からの知識抽出や事例からの問題解決知識獲得のプロセスを支援する上で、以下のような点で興味ある示唆を与えるものである。

- goal concept 記述レベル － operationality criteria 記述レベルの違いに見られるような多元的知識表現間のリンケージ手法
- goal concept としての学習概念の定義の仕方に応じて、本質的・非無関連な情報の取捨選択を説明構造の生成を通して自動的に行うことにより、多面的な視点からの情報整理、いわば knowledge-based filtering としての機能。

3-1-4-2 マクロ生成――知識のコンパイル（knowledge compilation）――

エキスパートシステム構築の気運の高まりと共に、問題解決過程ならびに問題解決に必要となる知識に関する研究が進められてきている。そこでのテーマは、

(1) 問題を理解するための知識
(2) 問題を解くための知識
(3) 問題を解くことによって知識がどう変化するのか

について、初等幾何学や初等物理学の問題を対象とした実験から議論されてきた。(1) の知識は、問題文を解釈して内的表現をつくりだす際に必要となる知識で、スキーマ構造で代表されるような宣言的知識およびその構造化の度合が問題理解に要するコストと密接な関係をもつことが報告されている。そして、実際に問題を解くにはこのような内的表現を操作し、変換して目標状態を実現するための手続き的知識・方略知識が必要となる。これら(1)(2)の知識について、熟達者と初心者の間の決定的な違いは、情報のチャンク化である。即ち、与えられた情報をどのくらいまとめて理解できるか、また方略の場合には、いくつもの手続きがいつも同じ順序で、自動的に実行されるのが熟達者のもつ方略構造の特徴である。

このような知識の構成（composition）と手続き化（proceduralization）、即ち、既有の宣言的知識を新たな手続き的知識に置換し、解釈のための知識の適用の繰り返しが、一連の手続きとしてまとめ上げられ、自動的・半無意識的な適用へと移行する知識の編纂（knowledge compilation）のプロセスに関する研究は、学習知識による問題解決効率の向上が求められるような問題領域においては、ますます重要性を帯びてくるものと考えられる。

また（3）の知識構造のダイナミクスについては、Rumelhart and Norman[8] が、熟達化と学習過程との関連において、スキーマの増大（accretion）、調整（tuning）そして再構造化（restructuring）のプロセスとして説明づけている。「増大」は、新たな知識が既有知識との相互作用なしに加法的に付加されていく場合、「調整」は、ある問題に固有に得られた知識の適切性を向上させるべく、より広範な領域への適用のための一般化、そして逆に知識適用の的確さを増すための特殊化を伴うプロセスである。その結果、既有知識構造の修正や新規知識構造の生成に至る場合が、再構造化のプロセスであるとし、熟達化に伴って、「増大」から「調整」、「再構造化」の段階を経て徐々に学習過程が進行していくことを示している。

以上述べたような熟達者のもつ二面性、即ち、比較的単純な構造をもつ定型的な知識や技能に基づいて情報を自動的・半無意識的に処理することのできる定型的熟達者（routine experts）としての側面、一方、豊かな内容を含み高度の構造化された知識と柔軟で適応的な技能を習得した適応的熟達者（adaptive experts）としての側面は、ＥＢＬの knowledge filter による情報の取捨選択とコンパイルのフェーズ、そして知識の一般化のフェーズに各々対応していると考えることができる（但し、「再構造化」のプロセスはＥＢＬにおいては言及されていない）。

3-1-4-3 ＥＢＬの認知科学的側面――理解による学習――

類推の項でも述べたように、学習において知識の果たす役割は大きい。行動主義心理学の流れをみても、学習の進行を個体の経験のみから達成されるものと考える試行錯誤的学習理論の考え方から、個体－環境間の相互作用を重視した刺激抽出理論、さらに個体の内的状態が学習に関係していることを主張した確率的反応学習理論と展開してきた。これが1970年代に至って、記憶・情報研究の進行とともに、この内的状態がそのまま「知識」の状態として捉え直されることになる。Ausubel の有意味受容学習（meaningful reception learning）の考え方における先行オーガナイザに見られるように、人々が新しい情報を理解するためには既に獲得した知識や概念が必要であること、学習課題が既有の学習者の知識構造にでたらめでなく、また丸暗記としてでなく関係付られたときに、真に意味のある学習が成立する、いわば「認識的理解を通しての学習」という考え方が広く認められている（このような考え方は既に1930年代の Bertlett のスキーマ理論に見られる）。ここでの学習の特徴をまとめると、

- 変化する知識を中心にした行為と認識のサイクル
- 学習者の持っている知識を使って、ものごとをよりよく理解・認識するために、学習材料に積極的に働きかけ、相互交渉を経て、まとまりのある解釈を生成する過程
- 学習材料が潜在的に学習者にとって有意味なものでなくてはならない、逆に学習者なりの「意味」を見いださせるための、課題内容に特殊な知識が不可欠であること
- 新しい領域の情報に初めて接するときには、学習者は適切なスキーマ持っているわけではないが、適切なスキーマが育つようなスキーマの「もと」になる情報、いわば学習材料と認知構造とを橋渡しするような情報を与え、それによって適切なスキーマをつくりだす

このような既有知識構造の存在は、学習時に重要でない属性は捨て、重要な属性のみを残すという「注視点」の設定、さらには学習者の視点に応じた学習内容の変化といった、人間の学習、特に熟達者の学習プロセスの前提となるものである。従って、既有知識構造としての domain-theory の存在、説明生成による verification のチェックを付加した学習プロセス、換言すれば「理解による学習」のプロセス、さらには説明生成を通しての訓練事例（学習課題）からの情報の取捨選択のプロセス等、ＥＢＬの学習手法は、人間の学習形態を忠実に模擬した機械学習の枠組みであるといえる。

3-1-3-4 ＥＢＬの課題

前節のように、人間の学習プロセスとのアナロジーとしてＥＢＬを捉えるとき、そこでは未だ解決されていないいくつかの問題点が存在する。

- 知識は新たな知識獲得のために使われることによっても変化するし、逆に使われないでいることによっても変化する。このような観点から、domain theory 内部の知識要素間の相互同化や強度変化を含めた知識構造全体の変化等、そのダイナミクスを考慮していく必要がある。
- 既有の domain theory のみをつなぎあわせて、説明生成を行うのではなく、より一般的なメタ認知的知識との関連が扱われなければならない。すなわち、「説明出来ない」と気付くこと、そのときどうすれば「説明できるようになるか」を学習システム自ら能動的に探索にいけるだけの、学習課題に対する能動的な働きかけと自律的な問題提起の能力の具備が不可欠となる。
- 一般に理解・解釈の問題になると、goal concept からのトップダウンな説明生成といったような「説明の脈絡」が明示的に与えられることは希であって、どの要素とどの要素の間に説明付を行うかを自律的に見いだしていくプロセス、いわば「説明生成の制御」が最重要課題となる。このためには、ＳＢＬのように事例の集積を通して、その相関を見いだし説明の脈絡

の候補として効率化をはかる方法、あるいは説明を要するような「異常な」属性であることを検出するための予期駆動型推論の定式化が要請される。

これらの課題に答えるべく、次節ではより認知科学的な立場にたった「事例記憶に基づく説明学習」について、そして次々節では、上述のようなＥＢＬの課題を克服すべく、既存の学習手法と融合した学習形態の事例について概説する。

3-1-5 事例記憶に基づく説明学習

3-1-5-1 説明と理解および学習

Schank らは、従来より自然言語処理を手掛け、人間の言語理解のモデルとして、ＣＤ理論やスクリプト理論を発表し、それまでの自然言語研究の大半が入力文の構文解析や、主語・述語、係り受けの分析に主眼がおかれていた自然言語研究に、意味的な曖昧性や文脈理解のための知識が必須であることを提案、その後の自然言語研究のみならず、画像・信号の解釈・理解の分野においても多大な影響を残した。その後、彼の研究はプラン（plan）・ゴール（goal）・ＭＯＰ（memory organizing packet）へと進み、固定化されライブラリー化された当初のスクリプトの考え方から、より大局的かつ柔軟な情報処理を可能にするべく、高階の知識構造の導入に視点が移ってきた。以上のような考え方は、彼の弟子達により、学習のテーマへと展開されていくことになるが、そこでは Mitchell や Michalski に代表される論理的な機械学習のアプローチに対して、心理学のスキーマ理論に根拠をおきつつ、動的に変化する人間の記憶構造（知識構造）、特に意味記憶・エピソード記憶の構造に根ざした独自のアプローチとして展開を見せている。彼の最近の著書では、説明と理解そして学習の間の関連性について、興味ある提案を行っている[9]。

入力された文章に対して、それが位置づけられるべき既存の記憶構造が見いだされたとき、理解されたことになる。換言すれば、それが適切なコンテキストのもとに置かれたとき、その意味が見いだされた（make sense）とみなせる。これに対し、入力された文章が、予期された結果とくいちがう、即ち、適切なコンテキストが存在しない場合、あるいは決定できないときには、何故にそのような違いが生じたのかを理由付ける説明が必要になり、ここでの説明生成こそが、スクリプトタイプの理解に代わって真に深い理解へと導く、創造的な学習を達成するものである。ここで説明の果たす役割は、理解の失敗、即ち予測の失敗からの回復であり、表層的な特徴からだけでは結び付けようのないような過去の類似事例を想起するためのインデックスとして機能することになる。

説明のパターンとしては、

(1) Canned Explanation
(2) Explain-Away Explanation
(3) Additive Explanation

の三種類に分類している。第一の Canned Explanation は、既知のことを対話の相手に伝えるだけの行為であるが、このようなタイプの説明は、説明が与えられる時点で既に説明者の内部に用意されていたものであり、説明を行うことによって説明者自身にはなんら新たな創造は望まれない。第二のタイプの説明は、説明対象が一事例と見なせるような、既知の仮説世界を見いだすことを意味するが、説明を生成することによって理解システム自身が知識構造の変容を加えられるというものではない。これに対し、第三のタイプの説明は、前述のように既有知識の中に、入力に該当するものが見いだせない場合に、ある一連の手段を講じてその理由付を行い、より深い理解に到達しようとするもので、説明生成を通じて以前には未知であったことが新たに付加される。そしてこのタイプの説明こそが学習と密接な関連をもつものである。

Schank はこれらの説明の分類を、説明が要請されるニーズに対応させて、図2に示すような次元上で捉えている。この様な次元は、外部供与的なニーズ（左）から、内部発生的ニーズ（右）に亙って各々のタイプが対応し、各説明タイプには各々の説明生成のための一定の手続きが存在する。これらの手続きは、左から右に進につれてより複雑なものになり、生成の困難さも増すことになるが、逆に学習システムにとってはより有意味な説明となる。

以上の説明タイプの分類からも明らかなように、学習は、いくつかの事例間の類似性・差異点を認識するような受動的な情報処理では達成され得ないし、また教師から新たな知識を教えられるだけであっても単なる暗記学習に（rote learning）にとどまる。学習者の側に、学習対象を理解するだけの知識が最小限必要であるし、それに加えて、学習対象に対する能動的な働きかけがあってこそ、はじめて creative な学習が成立する。このような能動的な働きかけとは、

（1）予期機能
（2）内的な目標設定の機能
（3）知識修正機能

の三点を意味する。説明プロセスを起動するには、入力が既有知識に該当しないことを見極める、いわば異常（anomaly）を認識することが必要になる。しかしながらこのようなアノマリーは、あらかじめ過去の経験から常套的な世界モデルが予期構造としてつくりあげられていることが前提であり、これへの参照なしには、何が正常で何が異常であるかを識別できない。そしてこのような予期の失敗がトリガーとなって、何が説明付られねばならないのかについて、説明目標の設定が外部から与えられることなく、内部発生的になされる。この目標設定に駆動されて、ライブラリー化された説明パターンの中から整合するものを検索してくることになるが、ここで必ずしもダイレクトにそのようなパターンが見いだされるとは限らない。そこで第三の機能が要請される。この機能は、設定された目標に対して、知識要素間の変換を能動的に試み、既有知識への適応性を見いだすことになる。たとえば、類似の機能を有したり、類似の環境下に置かれた対象、あるい

は類似の挙動を呈する対象への置換といった類推に準じた操作がこれに該当する。こうして入力された情報が説明付られると、ここの至るまでに施した一連の変換操作に基づき、許容される一般化を行って既有知識内部に適切に位置づけるとともに、既有知識の再構造化を図って、来るべき入力情報に備えることになる。

図2は、以上のアルゴリズムに対応する学習（説明）システムのアーキテクチャーである。説明を生成するためには、まず、入力に対して、過去の経験を想起して比較し新たな知識を創り出すような柔軟で知的な記憶構造が要請される。また、初めは説明出来なかった事柄や誤りに正しく対応するために、自分自身をたえず修正していくための機能が不可欠となる。即ち、受動的な知識適用に対して、既有知識を用いながら、予期による能動的な学習対象に対する処理、そしてたとえ該当する項目が見いだせなくとも、メタ認知的知識を駆使して、適用可能性を見いだす。そして、一旦説明がつけば、この新たな知識を、単に加法的に付け加えていくのみならず、既有知識構造の再組織化を行って、来るべき状況に対応していくというものである。

3-1-5-2 事例推論（Case-Based Reasoning）

上述のような事例記憶に基づく説明学習に関連するものとして、事例推論（Case-Based Reasoning）がある。Reasoning ではあるが本質的には2つの事例からの学習である。即ち、未知の事例が与えられると、これを解釈認識した上で、多数の事例データの中から、参照すべき特定の事例が検索される（図3）。この際、検索を効率化するために事例間の類似性を測る尺度や、事例群の間の組織化・カテゴライゼーション、そのもとでのインデックスのメカニズムやカテゴリー中の最も典型的な事例の選定の方法が要求される。参照事例が決定されると、この2つの事例が本質的に等価なものとして照合するか否かが評価される。ここで、システム内に既有の領域固有知識を用いた説明生成が試みられる。説明自身から自律的な知識の生成は原則として行われず、照合の手段および事例のインデックスとして用いられる。生成に失敗した場合には、照合に必要な知識を外部の教師に要求し、rote learning の形で領域固有な知識が集積される。両事例の照合が十分なものである場合には、これらをＳＢＬに基づいて一般化し、未知事例への適用が行われるとともに、事例群の間での記憶の分化（specialization）が行われる。照合が知識に基づいて評価されるところに特徴を有し、知識の動的な更新が、照合の performance や、事例の generality に影響を与えることになる。

診断問題[10]を始め、裁判の判決裁定問題[11]等にも適用が試みられている。

3-1-6 ＥＢＬの拡張事例

3-1-6-1 ＥＢＬにおける説明生成制御のためのＳＢＬの活用

Lebowitz[12] は、ＳＢＬ、ＥＢＬの両者を有効的に融合した UNIMEM という名前のシステムを構築している。UNIMEM は米議会における各州代表議員による数多くの議案の採決投票の内容を記録したデータに対する意味づけ・解釈を生成するシステムである（図4）。

まずこれらのデータはＳＢＬに基づいて、共通属性項目群を一つの概念単位としてまとめたさまざまの一般化レベルにおけるスキーマ階層構造を組織化する[13]。各スキーマを構成する属性は、同一の親スキーマをもつ他のスキーマ属性との比較により、そのスキーマに固有の属性項目、即ちその属性の値がわかれば他の属性値を容易に推定させるような predictive な属性項目、また逆に他のスキーマに数多く共有されているような、non-predictive な属性項目が決定される。システムは前者の属性項目を原因、後者を結果とする因果関係知識を生成する。ただし、この知識はＳＢＬに基づくことから、あくまで仮想的なものであって実証（verification）はなされていない。即ち、個々のデータ表現で定義された属性項目集合が真にそのような因果関係を記述するに十分なものであった保証はないし、また、predictiveな属性項目がデータの間での単なる偶然の一致の結果として得られた可能性も存在する。

そこでシステムは、ＥＢＬの手法を用い、予め常識的なレベルで与えられた小数の domain-specific な知識に基づき、ＳＢＬで生成された仮想的な因果関係の間の説明生成を試みることにより、知識の実証にとりかかる。説明が生成できない場合には、ＳＢＬの結果及び属性項目の定義にフィードバックをかけることにより、実証が行えるまで繰り返す。この結果、実証が終了した場合には、この知識を自らの domain-specific な知識として付加することにより、自己成長していくとともに、その後の説明生成をより確実なものとしていく。

このように、潜在的にノイズを含み得る多数の事例群に対してＳＢＬが適用されることにより、来るべきＥＢＬの説明生成のガイドラインを生成する。そして、既有知識のもと、ボトムアップに準自律的に生成された説明構造のマクロ構造からの修正をトップダウンに受けることにより、一貫したデータの解釈に向かって収束していくことになる。

4-2　識別問題へのＥＢＬの適用

Bergadano[14]らは様々なnoisyな変形を伴う対象の識別の問題に対し、ＥＢＬを有効に活用している。彼らのシステムは、典型的な識別規則を蓄えた知識ベース及び対象のとりうる変形規則を納めた知識ベースの２種の異なる知識を有する（図６）。まず、既有の識別規則のダイレクトな適用では、そのままでは識別できないような変形された事例が、その正しい識別結果と共に教師により与えられる。システムは、既有の変形規則を用い、この事例が具体的にどの様な変形のプロセスを経れば、現在の識別規則の範囲内で正しい識別が可能になるかについての説明を生成する。これは、各変形の過渡状態をstate、変形規則をoperatorとする状態空間として定義されている。説明生成が成功すると、そのプロセスをコンパイルした規則が、そのような変形を伴う事例識別のための新たな規則として知識ベースに追加される。

システムは、このようにして次々に与えられる事例についての説明の履歴を記録しており、ある数の事例の処理が終了すると、これらの説明（＝変形の系列）に対してＳＢＬを適用することにより、今度は変形規則の方の知識の帰納学習を

実行する。即ち、ある一連の変形を使った結果旨く識別に至った場合を正事例、使わなくても識別に至った場合及び使ったがためにうまく識別できなかった場合を負事例とするような事例学習として、状態空間の探索を効率化するための変形系列を帰納学習し、変形規則の知識ベースに追加・修正していく。

3-1-7 むすび

本稿では、機械学習における類推学習を中心に、従来から進められていきた事例間の類似性に基づく帰納学習の限界と問題点をまとめ、近年注目を集めている説明に基づく学習、事例推論の果たす役割を総括した。

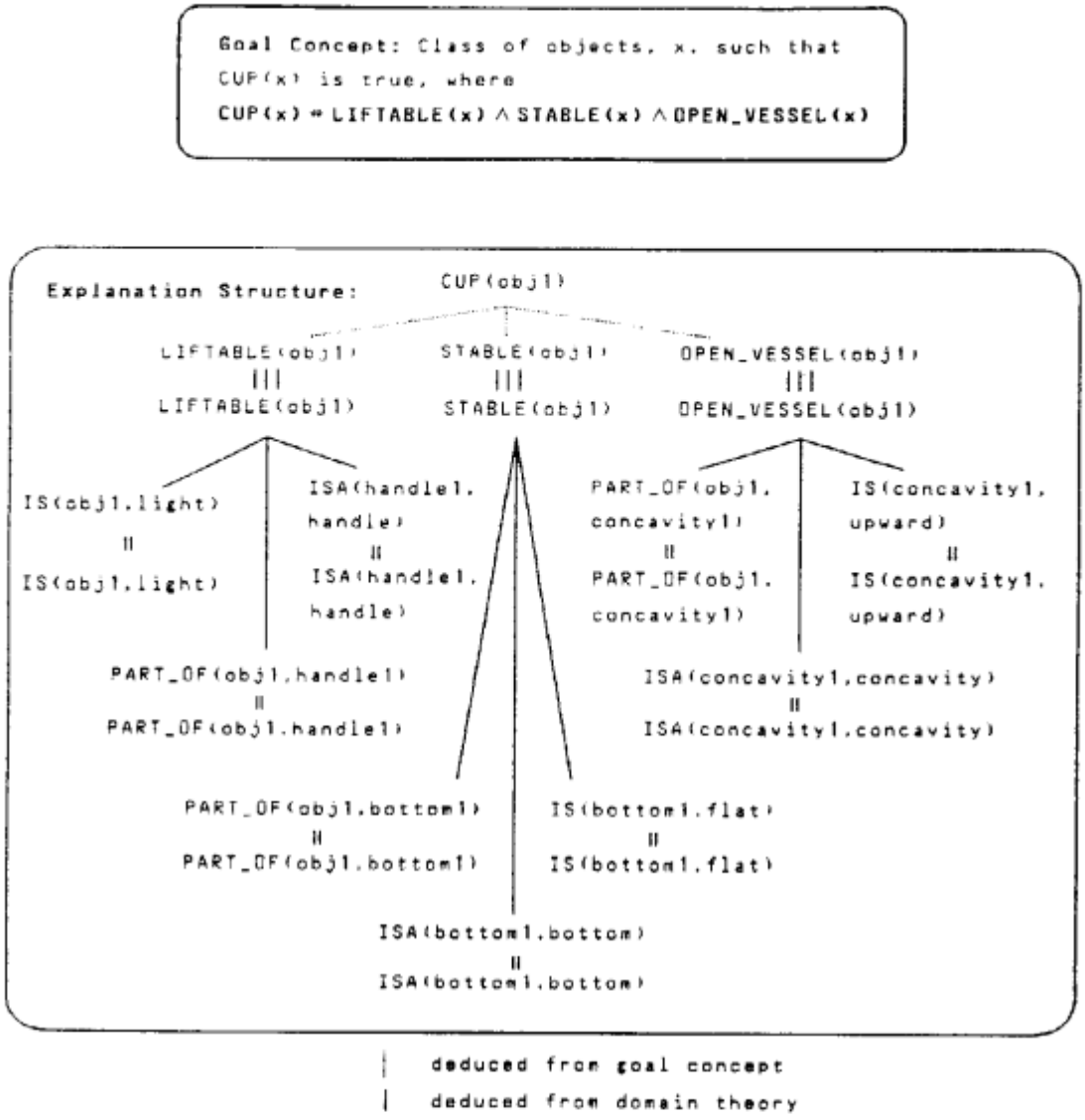
最後に、以上の議論に基づき、今後の展開として予測される方向性をまとめる。

- いずれかの学習形態を選択して適用するよりも、前節で述べたように問題対象
- 事例の数や質に合わせて、演えき・帰納・類推各種の学習形態を融合したシステム化が要請される。
- すべからくすべての問題対象に適用可能な学習アルゴリズムの開発から、より現実的な展開として、学習対象を限定し、領域に固有な知識並びに問題解決手法に大きく依存した学習システムへの移行。このことは、特定の応用を目指したタスク指向的な学習システムの研究を意味し、知識シェル構築にむけて問題解決タスクの類型化を行った generic task 研究[15]と不可分な展開が予想される。
- 小数の特定事例を介在させた学習の展開は、専門家からの一般的な経験則の提示を前提とした従来の知識獲得の手法に対し、現実的かつ具体的なイメージを抱かせ易い。この意味で学習プロセスの対話性の向上、すなわち専門家ーナレッジ・エンジニア間でのインタビュー等の知識獲得のプロセスをシステマティックに支援するツール開発への可能性を潜在させている。
- 多重表現に基づく学習の手法は、事例として表層的に表に現れ出る以外の背後にある高次知識、例えば、専門家がタスクの実行中には殆ど意識せずに用いているが、その実行結果には顕著に現れているような「技能（skill）」の獲得や、問題解決のための戦略やメタ知識の生成など、従来残されていた課題の学習可能性に示唆を与えるものである。

【参考文献】

1) 原口：類推の機械化について、淵監修「知識の学習メカニズム」共立出版、第５章（1986）
2) Gentner, D. et al.: Systematicity and Surface Similarity in the Development of Analogy, Cognitive Science, 10, 277-300 (1986)
3) Winston, P.H.: Learning and reasoning by analogy, Communication of ACM, 23-12, 689-703 (1980)
4) Winston, P.H.: Learning new principles from precedents and exercises, Artificial Intelligence, 19, 3, 321-350 (1982)

5) Russell, S.J.: Analogy and Single-Instance Generalization, Proc. of Int. Workshop on Machine Learning, 390-397 (1987)
6) Davis, T.R. and Russell, S.J.: A logical approach to reasoning by analogy, IJCAI87, 264-270.
7) Mitchell, T.M.et al.: Explanation-based generaization: a unifying view, Machine Learning, 1, 1, 47-80 (1986)
8) Rumelhart, D.E. and Norman, D.A.: Accretion, tuning, and restructuring: Three models of learning, In Cotton, J.W. and Klatzky, R.L. "Semantic factors in cognition," Lawrence Erlbaum Assoc. 37-53 (1978).
9) Schank, R.:Explanation patterns: Understanding mechanically and creatively, Lawrence Erlbaum Assoc. (1986)
10) Bareiss, E.R. et al.:Protos: An exemplar-based learning apprentice, Proc. of Int. Workshop on Machine Learning, 12-23 (1987)
11) Bain, W.M. : A case-basedreasoning system for subjective assessment. Proc. of AAAI86, 523-527.
12) Lebowitz, M: Integrated Learning: Controlling Explanation, Cognitive Science, 10, 219-240 (1986)
13) Lebowitz, M: Generalization from natural language text, Cognitive Science, 7-1, 1-40 (1983)
14) Bergadano, F. et al. : Integrating EBL and SBL approachs to knowledge base refinement, in Ras, Z.W. and Zemankova, M. (eds.), "Methodologies for intelligent systems", Elsevier Science Pub. Co. Inc. (1987)
15) Chandrasekaran, B.: Towards a taxonomy of problem solving types, AI Magazine, 4, 9-17 (1983).



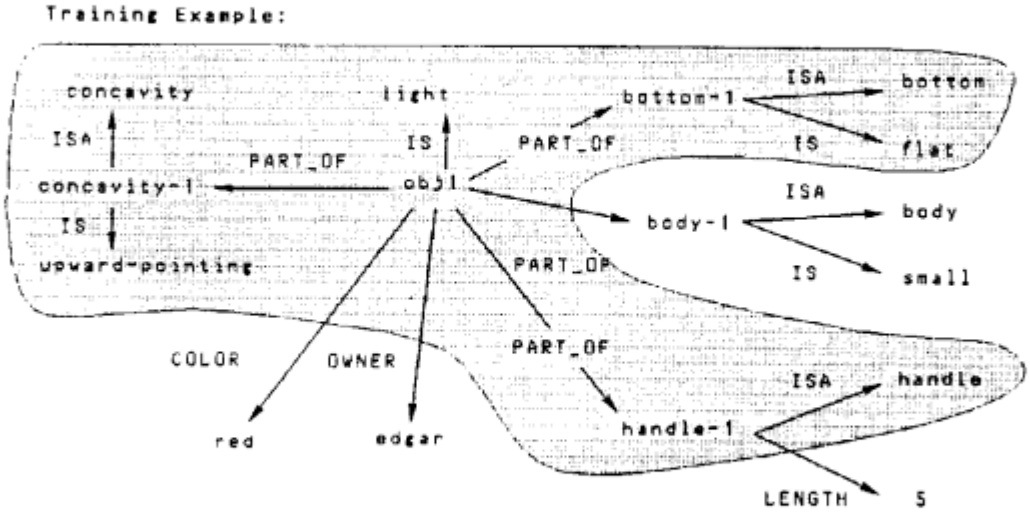


図1　証明木生成に基づく説明学習（modified from reference 7)）

Domain Theory:

$IS(x, light) \wedge PART\_OF(x, y) \wedge ISA(y, handle) \rightarrow LIFTABLE(x)$

$PART\_OF(x, y) \wedge ISA(y, bottom) \wedge IS(y, flat) \rightarrow STABLE(x)$

$PART\_OF(x, y) \wedge ISA(y, concavity) \wedge IS(y, upward\_pointing) \rightarrow OPEN\_VESSEL(x)$

... ...

Acquired Concept:

$IS(y, bottom) \wedge PART\_OF(y, x) \wedge IS(y, flat) \wedge ISA(z, handle) \wedge PART\_OF(z, x) \wedge IS(z, light) \wedge ISA(w, concavity) \wedge PART\_OF(w, x) \wedge IS(w, upward\_pointing) \rightarrow CUP(x)$

図 1 (continued)

Explanation Needs	external	respond	emphasize	remedy failure
Explanation Types	Explain-away	Canned	Analogical	Additive

表1　説明のニーズと説明タイプ（from reference 9)）

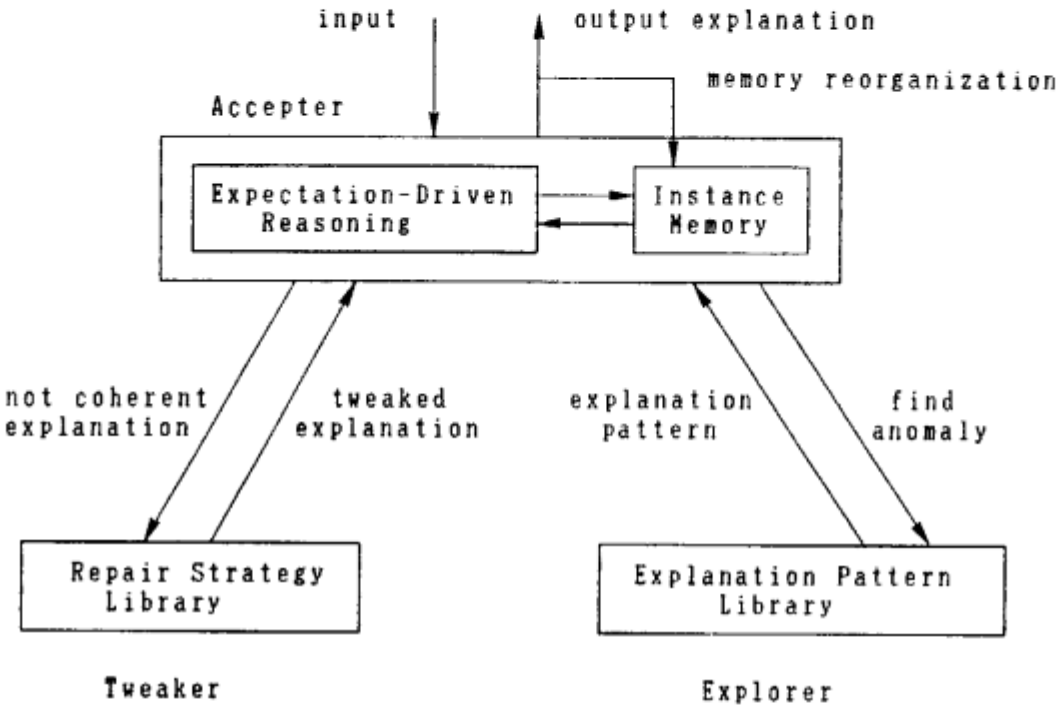


図２　事例記憶に基づく説明学習（modified from reference 9))

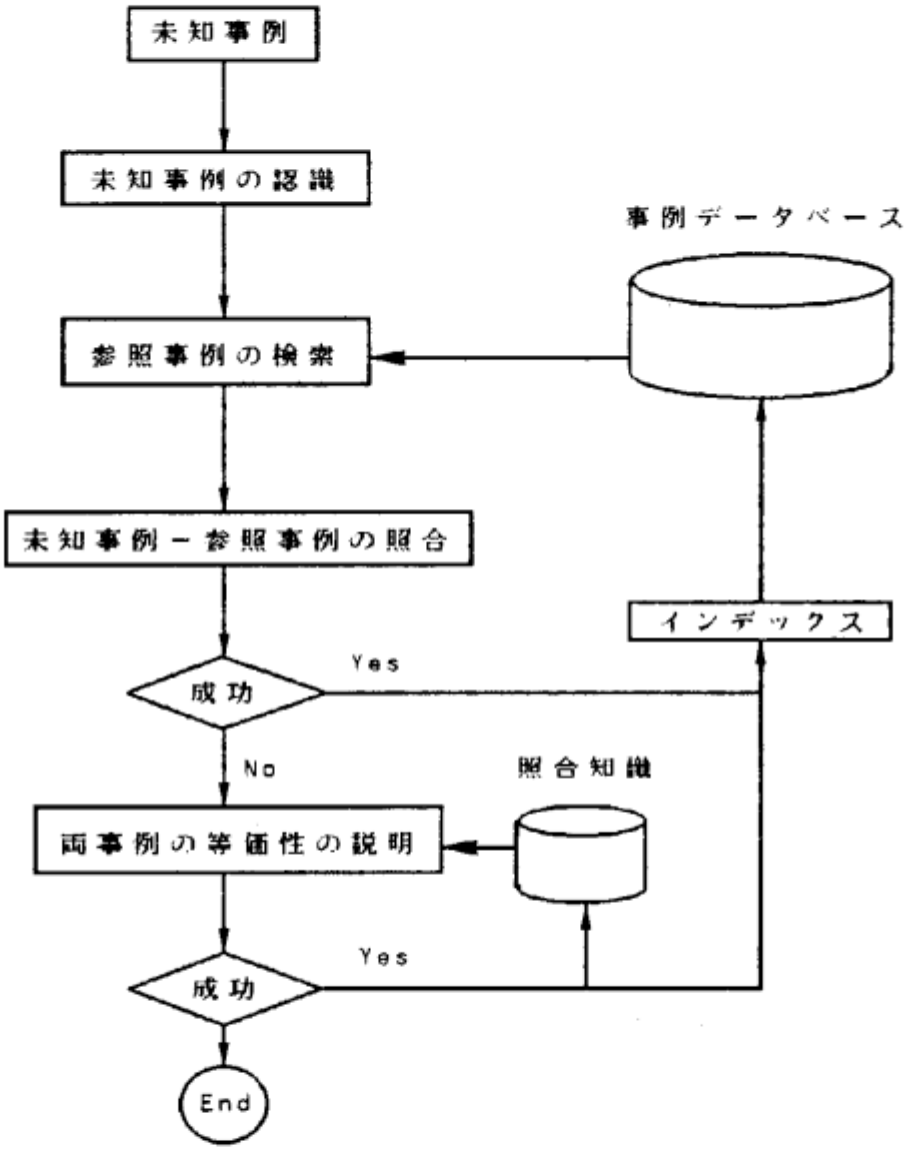


図３　事例推論の処理フロー

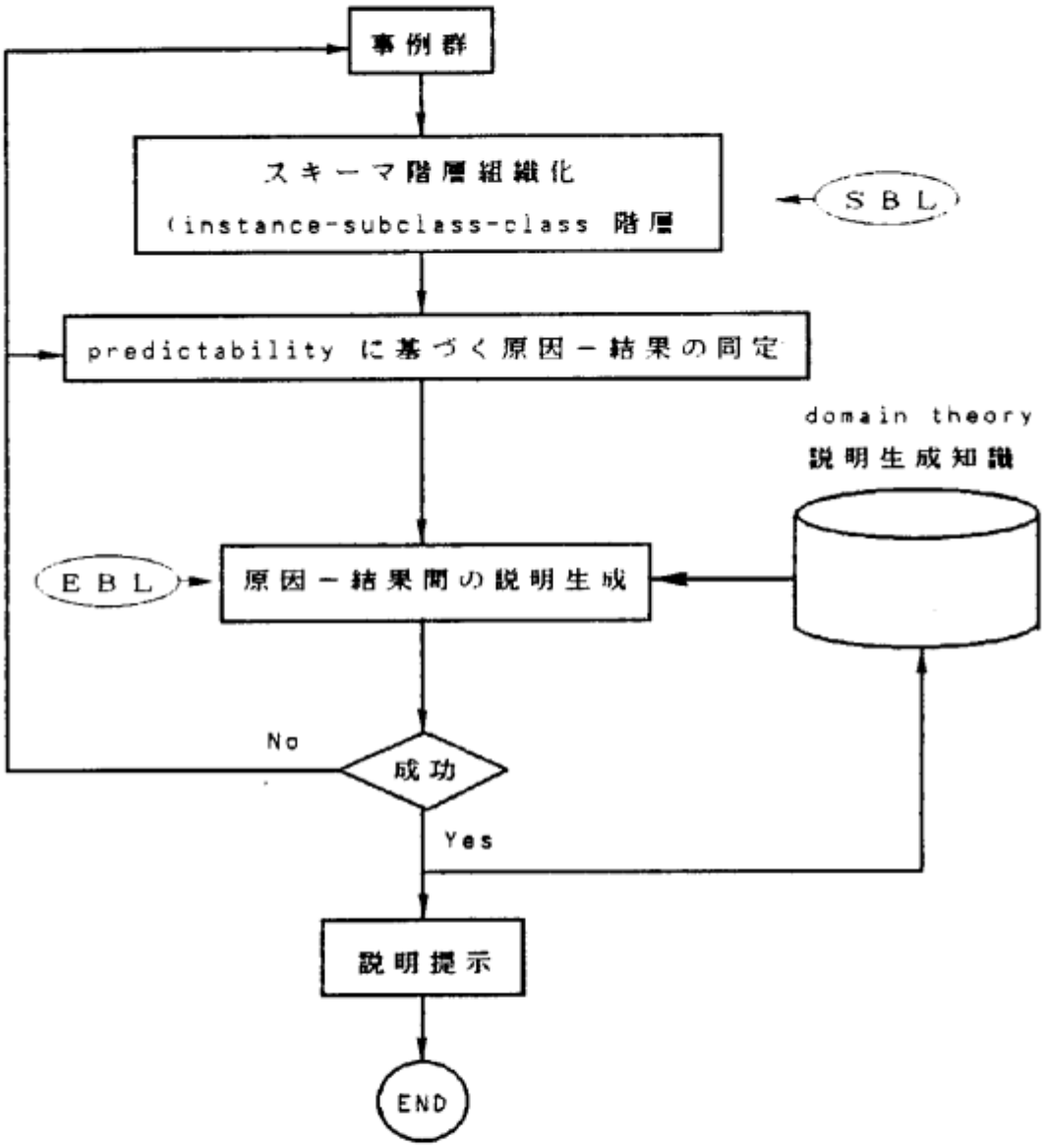


図４　Lebowits によるＳＢＬとＥＢＬの融合事例（１）

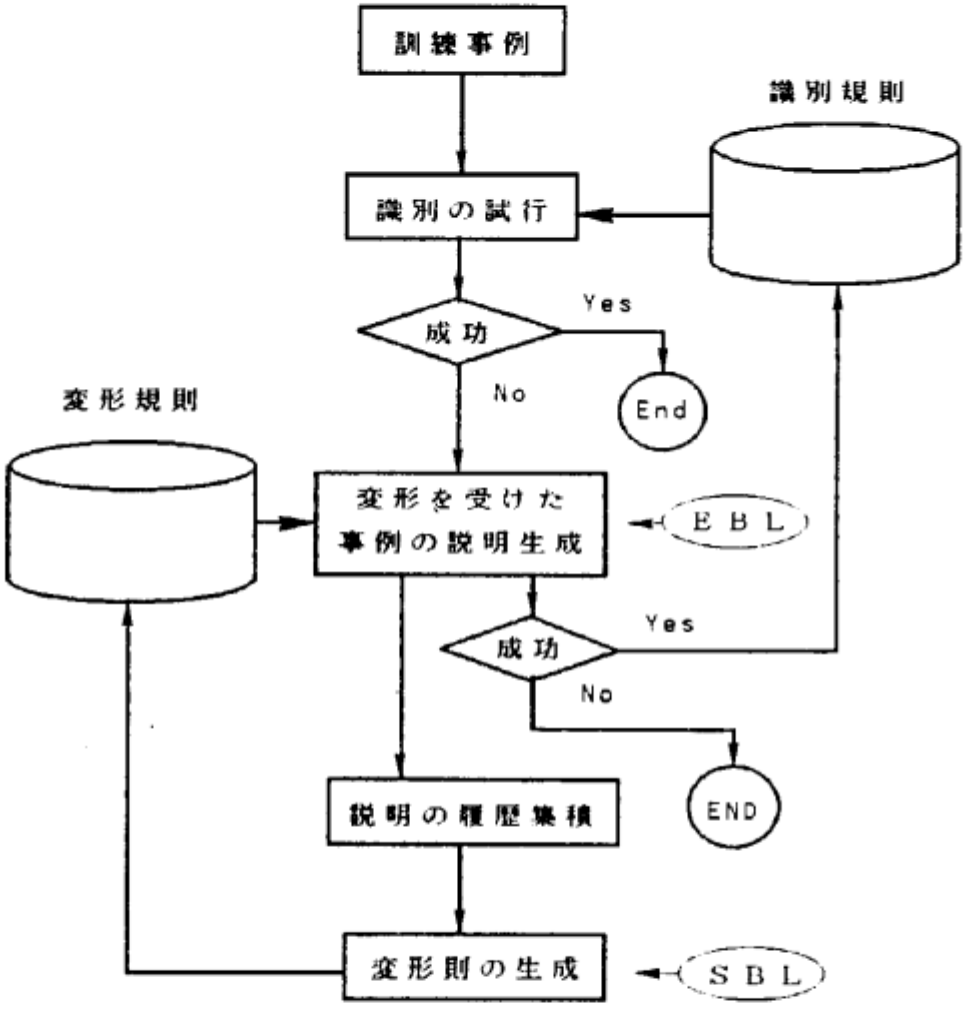


図５　Bergadano etal. によるＳＢＬとＥＢＬの融合事例（２）

3.2 Explanation-Based Generalization の課題（NTT 石田）

本節ではEBGの課題と，それぞれの課題に関する研究状況をまとめる．なお，EBGの課題は下記文献より抽出したものである．課題に関する研究状況はICOT-KSS-KRP-WGでのｻｰﾍﾞｲを基にまとめたものである．

T.M.Mitchell, R.M.Keller, S.T.Kedar-Cabelli: Explanation-Based Generalization: A Unifying View, Machine Learning Vol.1, No.1, pp.47-80, 1986.

予め，用語の定義を行っておく．

- terget concept: 概念の記述．
- training example: target conceptのinstance．
- domain theory: 問題領域の規則．
- EBG: 1つの例からの演繹的学習．
  domain theoryを用いてtraining exampleがtarget conceptのinstanceであることを示す説明を生成する手法．

3.2.1 Imperfect theory problem

3.2.1.1 Mitchell et al. の問題提起

EBGはdomain theoryが完全であることが前提である．言い替えると，学習される知識は既存の知識から演繹的に導かれる．しかし，一般にはこのように強力なtheoryが初めから存在すると仮定することは現実的でない．従って，EBGの主要な研究課題は不完全なdomain theoryを使って一般化を行なう手法，及び学習の過程で不完全なdomain theoryを改善することである．この問題はいくつかのｸﾗｽに分解できる．

1) Incomplete Theory Problem
株価予測や天気予報のように，training examplcを説明する十分なdomain theoryがない場合がこれにあたる．しかし，不完全な(しかも定量的でなく定性的な)theoryであっても，有用なおよその説明（plausible explanation）を生成することができる．不完全なtheoryを利用/改善する手法が重要となる．

2) Intractable Theory Problem
完全ではあるが計算量的に手に負えないtheoryがこれにあたる．例えばﾁｪｽの場合，ｹﾞｰﾑのﾙｰﾙは完全なtheoryであるが，それによって与えられた手が良いか悪いかを説明することはできない．人間はこのような場合，より抽象的で近似的なtheoryを作り出している．このような近似的，抽象的なtheoryは説明からの学習を行なう上で十分に有効であると考えられる．抽象的なtheoryを構築する手法と，そのtheoryをどの様な場合に安全に適用できるかを判断する手法は今後の課題である．

3) Inconsistent Theory Problem
Theoryがinconsistentな結果を導く場合がこれにあたる．例えばSAFE-TO-STACK問題で，ﾃｰﾌﾞﾙの重さを計算するのにﾃﾞﾌｫﾙﾄを使った場合と，質量と体積から求めな

おした場合とで，異なる値が導かれることがある．また，複数のtraining exampleが矛盾するtermを用いて説明され，結果をmergeすることが困難になる場合も考えられる．ﾃﾞﾌｫﾙﾄの概念が多くの応用領域で用いられていることを考えると，この問題は重要な研究課題であるといえる．

3.2.1.2 その後の研究状況

1) R.J. Doyle: Constructing and Refining Causal Explanations from an Inconsistent Domain Theory, AAAI-86.

・Common-sense levelのdomain theoryを用いて説明を作成し，failure-drivenで説明をrefineする方法が述べられている．抽象度の高いtheoryを使った説明は誤った推論を導くことが多いからである．

・例えば，causal mechanismに関する階層が以下のようになっているとする．

```
causal mechanism---propagation---flow---material flow
                                  |      |
                                  |      --light transmission
                                  --mechanical coupling
```

この時，ﾊﾞｽﾀﾌﾞ（material flow）の因果ﾓﾃﾞﾙを抽象度の高いcausal mechanismから得る場合や，ｶﾒﾗ（light transmission）の因果ﾓﾃﾞﾙに適用する場合などが例として考察されている．

・誤りの発見された説明は以下の方法を用いて修正される．
①異なった状況で説明をreinstantiateする．
②抽象度の低い階層のdomain thcoryで説明を再構成する．

2) 他にも，failure-drivenの学習が以下の文献で報告されている．

S.A. Chien: Extending Explanation-based Learning: Failure-driven Schema Refinement, Proc. 3rd CAIA, 1987.

K.J. Hammond: Learning to Anticipate and Avoid Planning Problems through the Explanation of Failures

3.2.2 Combining explanation-based and similarity-based methods

3.2.2.1 Mitchell et al. の問題提起

EBGは単一の例から概念を演繹的に推論し，SBG（similarity-based generalization）は複数の例から概念を帰納的に推論する．この両者を組み合わせることは，特に不完全なtheoryしか得られない場合に有効である．以下の研究がある．

1) Lebowitz(1985)はUNIMEMｼｽﾃﾑで両者の結合を図っている．UNIMEMではまず，議会の投票履歴ﾃﾞｰﾀﾍﾞｰｽから規則を抽出する．即ち，まず帰納的な手法により規則の候補を獲得する．しかし，ﾃﾞｰﾀﾍﾞｰｽがﾉｲｼﾞｰであるため，獲得された規則は必ずしも有効ではない．そこで，次にｼｽﾃﾑはdomain theoryを使ってそれらの候補を検証する．

2) Lebowitzのｱﾌﾟﾛｰﾁと逆に，まずtraining exampleに対してEBGを適用し，次にSBGを用いて得られた結果を組み合わせるｱﾌﾟﾛｰﾁもある．例えば，ﾊﾞｰｼﾞｮﾝ空間の手法を使って複数の例のEBGの結果を統合する方法が考えられる．すなわちpositiveな例からのEBGはﾊﾞｰｼﾞｮﾝ空間の境界をgeneralizeするのに使え，逆にnegativeな例からの一般化はﾊﾞｰｼﾞｮﾝ空間の境界をspecializeするのに使える．この手法は有効な概念を早く学習できるという利点をEBGから得ている．

3) Kedar-Cabelli(1984, 1985)は複数の例からのEBGの結果を統合する別の手法を提案している．この手法(Purpose-Directed Analogy)では，ある例を，類似した別の例のｱﾅﾛｼﾞｰで説明し，次に両者の説明を一般的な概念を作り出すために統合する．

3.2.2.2 その後の研究状況

1) P. Bergadano, A. Giordana and L. Saitta: Integrating EBL and SBL Approaches to Knowledge Base Refinement, Proc. of 2nd International Symposium on Methodologies for Intelligent Systems, 1987.

SBGとEBGを組み合わせることによって，domain theoryをrefineする．例として，変形が多い物体をclassifyする問題を取り上げている．ｼﾅﾘｵは以下のとおり．

- 図形を判別するclassification ruleが与えられている．このﾙｰﾙは典型的なinstanceを判別する能力を持つ．
- 今，変形して判別できないinstanceが現れたとする．但し，instanceがどの様に変形するかを記述した変形規則（deformation theory）は既知とする．
- instanceがどの様に変形したかを，変形規則を用いて説明させることによって，変形したinstanceの判別を行う．（EBGの利用）
- しかし，変形規則は通常，条件がgeneralすぎる．そこで，複数の適用事例から変形規則が適用できる条件を一般化により獲得し，これによって変形規則をよりspecificなものとする．（SBGの利用）

3.2.3 Formulating generalization tasks

3.2.3.1 Mitchell et al. の問題提起

EBGをどの様に利用すれば，ｼｽﾃﾑ全体の効率改善が行えるかが課題である．即ち，実行ｼｽﾃﾑ（performance system）がEBGをどのように取り込むかということが重要である．またこれに関連して，EBGの入力（goal concept, domain theory, operationality criterion）をどう決定（formulate）するか，実行ｼｽﾃﾑと学習

ｼｽﾃﾑとの間でどのような情報を交換すればよいか等も課題である．これまでの研究としては以下のものがある．

1) SOARは実行ｼｽﾃﾑがｻﾌﾞｺﾞｰﾙを解くたびにそれを再利用するための条件を一般化しようとする．SOARはEBGと密接に関連するimplicit generalizationと呼ばれる手法を利用している．

2) Kellerのcontextual learningでは，「問題解決の効率を向上させる」というtop-level goalと探索手法の記述から，「枝刈り用のﾌｨﾙﾀを作成する」ためのsub-goalを導き出す．このﾌｨﾙﾀは解に到達するために'useful'な問題解決ｽﾃｯﾌﾟだけを実行させるためのものである．

3) Kedar-Cabelli(1985)も同じ問題を取り扱っている．そこではCUP等の物（goal concept）の定義をその使用目的（喉の渇きをいやす等）から導き出すことが提案されている．

3.2.3.2 その後の研究状況

1) R.M. Keller: Defining Operationality for Explanation-Based Learning, AAAI-87.

・operationalityの定義を単に'efficient instance recognition'とするのではなくより厳密に定義しようとしている．Kellerによれば，EBGによって得られたconcept descriptionがoperationalであるためには，以下の2点が重要である．

Usability： performance systemによって利用可能である．
Utility： performance systemの効率を向上させる．

・これまでに提案されたEBGｼｽﾃﾑにおいて，operationalityがどのように評価されているかを以下のの3種の尺度を用いて分類している．

Granularity： operationalityをその有／無(binary)で評価するか，程度(continuous)で評価するか
Variability： operationalityが時間によって変化するか
Certainty： ｼｽﾃﾑ全体の効率向上に本当に寄与するか

		Variability	
		static	dynamic
Granulality	binary	Winston	GENESIS SOAR LEX2
	continuous	PRODIGY	MetaLEX

MetaLEXだけがCertaintyを保証する．

2) P.S. Rosenbloom and J.E. Laird: Mapping Explanation-Based Generalization onto Soar, AAAI-86

・SOARとEBGとの対応関係を述べている．SOARの知識獲得（chunkingと呼ばれる）は以下のとおり．

・SOARはgoalが設定されると，発火できる限りのﾌﾟﾛﾀﾞｸｼｮﾝﾙｰﾙを並列発火する．（これをelaboration phaseと呼ぶ．）発火するﾙｰﾙが無くなると，WM（ﾜｰｷﾝｸﾞﾒﾓﾘ）を調べ，operatorを選択しWMを更新する．（これをdecision procedureと呼ぶ．SOARはelaboration phaseとdecision procedureを繰り返して，問題解決を行う．）この時，operatorを一意に決定するのに十分な情報が得られない場合には（この状態はimpassと呼ばれる），subgoalを設定しその問題解決にはいる．

・SOARの学習には以下の2とおりがある．
①適用可能なoperatorが複数個存在する場合，どのoperatorがpreferableかを示す制御知識を問題解決の経験から獲得する．
②subgoalを達成し終わると，そこでの問題解決の経験から新たなﾙｰﾙを生成し，将来の問題解決に利用可能とする．このﾙｰﾙの条件部はsubgoal設定時のWMの内，問題解決に関連するものであり，動作部はsubgoal達成中に生成されたWMの内容である．得られたﾙｰﾙには一般化が行なわれる．即ち，共通のconstantは共通の変数で，異なるconstantは異なる変数で置き換えられる．

・SOARとEBGの相違は以下の点にある．
①一般化（変数化）の方法が異なる．
②SOARは，常にWM内の述語を基に説明を構成する．既にchunkされた中間述語を使うことがない．
③EBGでは，制御知識を得るためにgeneral interpretive ruleの導入が必要だが，SOARでは同一のchunkingの枠組みで行える．

・SOARで獲得されるﾏｸﾛは一般に条件数が多くﾏｸﾛの適用可否を判定するｺｽﾄが大きい場合が多い．下記文献は，SOARで実行中に獲得されるﾏｸﾛ（ﾙｰﾙ）を最適化する試みを述べている．

D.J. Scales: Efficient Matching Algorithms for the SOAR/OPS5 Production System, Stanford University, STAN-CS-86-1124, 1986.

3) S. Minton and J.G. Carbonell: Strategiesfor Learning Search Control Rules: An Explanation-based Approach, IJCAI-87.

・PRODIGYはexplanation-based specialization(EBS)を用いて，種々の現象(solution, failures, goal-interactions等)から問題解決の性能を向上させるための戦略を学習する．

・EBSがこの問題に適するのは，①問題解決のtraceから説明へのﾏｯﾋﾟﾝｸﾞはtop-downに行った方が効率がよい，及び，②この応用ではground-level explanation

が構成できない場合があるからである.

・EBSはproblem solverの実行終了後に，あるいは実行中に起動される.
control ruleは，training example（探索経路）がなぜterget conceptを満足したかを説明することによって得られる．terget conceptには以下の種類がある.
SUCCEEDS　制御の選択が結果的に成功を導いた場合．PREFERﾙｰﾙを導く.
FAILS　　選択が1つも成功を導かなかった場合．REJECTﾙｰﾙを導く.
SOLE-ALTERNATIVE　他の選択がすべて失敗した場合．SELECTﾙｰﾙを導く.
GOAL-INTERACTION　goal間に相互干渉が存在した場合．PREFERﾙｰﾙを導く.

・Control ruleは，どの候補をSELECT，REJECT，PREFERするかを示すﾙｰﾙである.
まずSELECTﾙｰﾙが適用され候補集合のｻﾌﾞｾｯﾄを選択する．もし適用可能なSELECTﾙｰﾙが存在しなければ，全ての候補が選択される．次に，REJECTﾙｰﾙによって候補集合がﾌｨﾙﾀされ，PREFERﾙｰﾙによって最も良いものが選択される.

3.2.4　その他

EBGのｲﾝﾌﾟﾘﾒﾝﾃｰｼｮﾝがいくつかの文献で報告されている.

以下ではEBGがPrologの4個のclauseで記述されている.
S. T. Kedar-Cabelli and L. T. Moccarty: Explanation-Based Generalization as Resolution Theorem Proving, International Workshop on Machine Learning, 1987.

以下ではEBGのいくつかのｱﾙｺﾞﾘｽﾞﾑが比較検討されている.
R. J. Moorney and S. W. Bennett: A Domain Independent Explanation-based Generalizer, AAAI-86.

以下ではMRS logic programming systemを用いたEBGの実現が報告されている.
H. Hirsh: Explanation-Based Generalization in a Logic-Programming Environment, IJCAI-87.

3．3　操作性規範に基づくＳＢＬとＥＢＬの統合化

東京工業大学　総合理工　　山村雅幸・小林重信

要　旨

本研究の目的は，問題解決システムにおける学習，特に知識洗練化のための新しい理論的枠組みを提案することにある．学習は適切な仮説を探索する探索問題としてとらえられる．従来の学習研究を生成－検査法に当てはめると，それらの多くは仮説の生成に重点を置き，学習結果を利用する問題解決システムとの関連において適切な仮説を検査することを無視してきた．このことは学習の応用を困難とする主要な原因であると考えられる．ＥＢＬにおいて提案された操作性規範は，問題解決システムとの関連において仮説の適切さを検査することを明確化した概念である．本研究では，操作性規範に基づいてＳＢＬとＥＢＬを統合化し，操作化問題という枠組みを提案する．操作化問題における研究課題は，知識変換（生成）と操作性規範（検査）の関係を調べ，効果的な探索方法を追及することにある．操作化問題の研究例として，論理プログラム上にＥＢＬを形式化し，後戻りの除去という典型的な操作性規範とＥＢＧの関係について理論的に考察する．

１．はじめに

本章では，問題解決システムにおける学習を生成－検査法に基づく探索問題としてとらえ，従来のＳＢＬとＥＢＬ研究を俯瞰して，学習研究に操作性規範を導入することの必要性について述べる．

1.1.　問題解決システムにおける学習

本研究では，システム内部の知識と外界の観察からの推論によって，与えられた問題を解決する工学的な問題解決システムを対象とする．以下，このようなシステムを単に問題解決システムという．

問題解決システムにおける学習とは，システム内部の知識，外界の観察，過去の問題解決の経験に基づいて現在の知識を変換し，問題解決の“能力の向上”を図ることである．能力の向上は，不完全な知識の補填，非効率な知識の洗練等の広い意味をもつが，本研究では特に後者の知識洗練化に注目する．

学習は，例から生成される仮説の候補から適切な仮説を探索する探索問題としてとらえられる[Mitchell 86]．生成される仮説は広大な探索空間を形成する．このため，学習における研究課題は，探索の制限に役立つ情報をいかに特徴付け，効果的な探索機構を構成できるかにある．

探索問題としての学習を生成－検査法に当てはめると，与えられた記述からの仮説の生成と，生成した仮説の適切さの検査に分離される．探索の制限に役立つ情報は，仮説の適切さの検査に対応する．学習における研究課題は，生成される仮説の適切さをいかに特徴付け，効果的な探索機構を構成できるかと言い換えることができる．

問題解決システムにおける学習では，生成される仮説の適切さの判定は問題固有の多様な規範からなる．このため，次節から述べるように従来の接近法は研究の枠組みとして不適切な面をもつと思われる．

1.2. 類似性に基づく学習（ＳＢＬ）

類似性に基づく学習（Similarity-Based Learning：ＳＢＬ）は多数のデータによって探索をガイドするデータ強調的方法である．ＳＢＬの枠組みを図1-1に示す[Mitchell 84]．ＳＢＬは練習例からの仮説の生成と，仮説の整合性の検査からなる．

問題解決システムにおける学習にとって，仮説の整合性は必要ではあるが十分ではないことに注意されたい．このため，従来のＳＢＬ研究の多くでは適切な仮説を生成するために問題固有のバイアスや経験則を必要とする．

整合的な仮説の空間は，仮説の整合性を保存する一般化関係のもとで範囲をなす．これをバージョン空間という．バージョン空間の維持によって探索は経済化され，その収束状況は学習の進行状況の目安となる．バージョン空間の収束は，一般化関係の設定と練習例の量に関係する．一般化関係における帰納的飛躍の程度をバイアスという．バージョン空間は，強いバイアスの下では少数の練習例で素早く収束し，弱いバイアスの下では収束までに大量の練習例を必要とする．練習例のあらゆる部分集合が記述可能な最も弱いバイアスの下では，バージョン空間の上・下限は常に暗記にとどまる．

期待される帰納的飛躍を引起こす適切なバイアスは各問題に固有である．問題解決システムにおける学習へのバージョン空間法の適用上の困難は，各問題に専用のバイアスを用意しなければならないという非柔軟性にある．また，“オッカムの剃刀”のような経験則によって仮説の生成をガイドする方法にも同様の困難がある．経験則は問題固有の浅い知識であって，適用範囲や意味が不透明だからである．

1.3. 説明に基づく学習（ＥＢＬ）

説明に基づく学習（Explanation-Based Learning：ＥＢＬ）は少数の例と領域知識によって探索をガイドする知識強調的方法である．ＥＢＬの枠組みを図1-2に示す[Mitchell 86]．ＥＢＬは単一の例の説明構造の一般化の生成と，操作性規範による検査からなる．操作性規範（operationality criterion）は，ＳＢＬにおいてバイアスや経験則に陰に組込まれていた仮説の適切さに関する規範を明示した概念である．

従来のＥＢＬ研究の多くは，より“高級な”一般化の生成法を追及し，操作性の保証の問題を重視しない．例えば“操作的な述語”が与えられることを前提とする．このため，例の増加にしたがってマクロの数が単調に増加し，マクロの想起のコストによって実行が非効率的になるような，従来のマクロ化の機能をもつシステムにおける困難を継承する．このことは，操作的な述語は本来期待される操作性規範を実装の都合で“翻訳”した結果に他ならない．

一般化の生成法のみを追及する研究は，前節で述べたＳＢＬの困難をも継承する．一般化の充実は探索空間の拡張をもたらし，期待される一般化が生成されるためには，バイアスや経験則が必要となるからである．

操作性規範を重視した研究にMeta-LEX[Keller 87]がある．Kellerは図1-3に示すように操作性規範を問題解決システムとの関連において定義した．しかしながら，そこでは，効率というきわめて抽象的な概念を規範とするために，操作性は過去の例をすべて解決しなおすことによってのみ保証される．操作性の保証問題の解決は，一般に困難であって，十分な考察が必要な課題である．

1.4. ＳＢＬとＥＢＬの統合化

以上の考察は次のようにまとめられる．

1) ＳＢＬは，効果的な探索のためにバージョン空間法の概念をもつが，仮説の適切さを陽に検査する規範をもたない．

2) ＥＢＬは，仮説の適切さを陽に検査する操作性規範の概念をもつが，操作性の保証問題は困難な課題である．

本研究の目的は，操作性規範を軸としてＳＢＬとＥＢＬの相補的な利点を統合した枠組み（操作化問題）を提案することにある．そこでは，明確に定義された操作性を維持しながら効果的に仮説を探索する機構が追及される．

以下，２章では操作化問題を提案する．３章では操作化問題の研究例として論理プログラム上にＥＢＬを形式化し，後戻りの除去という操作性規範と最小ＥＢＧの関係について理論的に考察する．

2．操作化問題の提案

本章では，操作化問題の枠組みを提案し，操作性のバージョン空間の概念を提案する．また，ＳＢＬとＥＢＬを操作化問題として見直し，拡張を試みる．

2.1. 操作化問題

操作化の概念は[Mostow 83]において提案された学習のひとつの形態である．そこでは探索戦略に関する抽象的な“助言”を操作的な作用素系列に変換するシステムについて述べられているが，問題解決システムにおける学習の一般的な枠組みとしては不十分である．

本研究で提案する操作化問題の枠組みを図2-1～2に示す．図2-2中太枠はシステムの境界，細枠はデータ，または機構を表わす．太枠の入れ子はデータのスコープを表わす．次にそれぞれの構成要素について説明する．

1)　問題解決器

問題解決器は，領域理論と外界の観察からの推論によって与えられた問題を解決し，証明を解答する．問題解決器は記述システムと実行システムからなる．

記述システムは問題の形式的記述言語である．領域理論や例や目標を記述する文法と，解（証明）を得るための推論を規定する．例えば，述語論理，状態計算等がこれに相当する．従来の記述システムの多くでは，推論は非決定的に定義され，問題解決の実行には実行システム（インタープリタ）による探索を必要とする．

実行システムは推論を実行する機械である推論制御知識に基づいて，記述システムによって非決定的に与えられた解の探索を“実行可能な”推論行動の列に写像する．例えば，Prolog，OPS5等のプログラム言語がこれに相当する．

いわゆる“知識”は記述システムにおける形式的な記述と，実行システムにおける推論制御の総和からなる（図3-2点線枠）．また，過去の経験には，過去の例と解の記述だけではなく，実行の履歴が含まれる．

2)　知識変換手法

知識変換手法は，現在の知識と過去の経験から，新しい知識を生成する手法である．例えばＳＢＬやＥＢＬにおける仮説の生成がこれに相当する．

3)　操作性規範

操作性規範は，知識変換手法によって生成された新しい知識の適切さを判定する規範である．例えば，知識洗練化における「良い解をより早く」といった要請がこれに相当する．

操作化問題とは，これらの構成要素の下で，領域知識と過去の経験が与えられたときに，操作性規範を満足するような新しい知識を求める探索問題である．操作化問題における研究課題は，問題領域と，知識変換（生成），操作性規範（検査）の関係を調べ，より効果的な探索機構を追及することにある．

実行システムの設定は操作性規範に相対的であることに注意されたい．もし，操作性規範が特定の計算機における効率を要請するならば，実行システムは言語の実装にまで立入った詳細な設定となるであろう．またもし，操作性規範が記述システムにおける性質のみに関わるならば，実行システムは不必要である．

2.2.　操作性のバージョン空間

操作化問題の枠組みは，ＥＢＬにおける説明に基づく一般化の生成を，一般の知識変換に拡張したものである．知識変換と操作性規範の組み合せは自由度が大きいために，操作性の保証の問題の解決にはなんらかの手掛りが必要である．

この問題への接近法としてＳＢＬにおけるバージョン空間の概念の応用を提案する．バージョン空間法は生成と検査の関係を利用した効果的な探索法の好例と考えられるからである．

ＳＢＬにおける一般化関係には，ある仮説に対して，より一般的な仮説が練習例と整合し，より特殊な仮説がやはり練習例と整合するとき，その仮説は必ず練習例と整合する性質がある．このため，バージョン空間は仮説の空間で範囲をなし，上限・下限によって空間を代表させることができる．このような性質を操作性の保存とよぶ．

一般に，操作性規範と一般化関係の間に操作性の保存が成立するとは限らない．例えば，解答の美しさが問われるような操作性規範と一般化関係とは無関係であろう．従って，操作性の保存は，操作化問題にとって重要な性質であると思われる．

操作性の保存，およびバージョン空間を定義する．

【定義 2.1】　操作性の保存

操作性規範Ωに対して，知識変換の空間Γ上の順序≲が次の条件を満たすとき，≲はΩを保存するという．

$$\forall g_1 \in \Gamma,\ g_1 \lesssim g_2 \lesssim g_3$$
$$\Omega(g_1) \wedge \Omega(g_3) \rightarrow \Omega(g_2).$$ ■

【定義 2.2】　操作性のバージョン空間

知識変換の空間Γ上の順序≲が操作性規範Ωを保存するとき，Ωを満たすΓの要素からなる空間をバージョン空間という． ■

論理プログラムにおけるＥＢＬでは，一般化は後戻り除去の操作性を保存しバージョン空間法が利用できることがわかっている（３章）．

2.3. 操作化問題としてのＳＢＬとＥＢＬの拡張形

操作化問題はＳＢＬとＥＢＬの相補的な利点を統合化した枠組みである．そこでは，ＥＢＬにおける操作性規範が一般化され，ＳＢＬにおけるバージョン空間の概念の応用によって効果的な知識変換機構が追及される．操作化問題としてＳＢＬとＥＢＬをとらえなおすことによって，１章で述べた困難の克服が期待される．

操作化問題としてのＳＢＬには，バイアスの変換までを知識変換手法に含め，バイアスの適切さを操作性規範に含めるような拡張が考えられる．このことによって，仮説の空間の探索と同時に，問題固有の適切なバイアスを探索するような機構が実現されることが期待される．

バイアスの変換について，[Utgoff 84]におけるSTABBの研究がある．STABBはLEXにおけるバイアスの変換システムで，バージョン空間が概念を含まないことからバイアスの不適切性を検出し，新しい概念を挿入することによって，バイアスをより弱いものに変換する．ただし，初期のバイアスとしては十分に強いものを設定する必要があり，その生成には経験則が用いられる．

STABBを操作化問題としてとらえると，バイアスに関して，「バージョン空間が概念を含むだけ弱い」ことと，「その問題領域にとって“正確”であるだけ強い」ことの相反する２つの操作性規範があると考えられる．これらとバイアスの変換手法の関係を調べることによって，より効果的なバイアスの変換機構，あるいは手法の限界を示すことが期待される．

操作化問題としてのＥＢＬでは，バージョン空間の概念の導入によって操作性の保証の問題が効果的に解決されることが期待される．従来のＥＢＬの枠組みは操作化問題と新和性が高いが，操作性の保証の問題の考察のためにはいくつかの点で拡張が必要であると思われる．

第１に，学習が単一の例題に対して１セッションずつ完結する形式を，複数の例題にわたって継続的に適切な一般化を探索する形式に改める方が望ましいと考えられる．現在の形式では，いったん獲得された記述を後の経験に基づいて変更する可能性が排除されるからである．

第２に，探索される対象は例から生成される一般化の記述ではなく，その一般化を領域理論に付加して知識を変換することまで含めることが望ましいと考えられる．操作性の判定は，おなじ一般化に対しても，それが実行システムにどのように使われるかによって変化するからである．

このように拡張されたＥＢＬはある意味でＳＢＬであると考えられる．すなわち，練習例は個々の知識変換であり，獲得される概念は操作性規範であるとみなすことができる．一般化関係が操作性規範を保存するならば，ＳＢＬと同様にバージョン空間法による効果的な探索が期待される．

次章では，このようなＥＢＬの拡張形を論理プログラム上に形式化し，理論的考察を試みる．

３．論理プログラムにおけるＥＢＬ

本章では，操作化問題のひとつの研究として前章で述べたＥＢＬの拡張形を論理プログラム上に形式化し，理論的考察を行なう．

3.1. 論理プログラムにおけるＥＢＬの形式化

論理プログラムにおけるＥＢＬの拡張形を操作化問題の枠組みにしたがって形式化し議論の準備をする．以下，ＥＢＬの拡張形を単にＥＢＬという．論理プログラムについては既知とする．

まず，問題解決システム関する仮定を行なう．

【仮定 3.1】　記述システムに関する仮定

1)　領域理論は確定節の有限集合からなる．

2)　概念は領域理論の単一のアトムで表現される．

3)　具体例は単位節の有限集合からなる．領域理論Ｄと概念Ｇに対して具体例Ｅが次の条件を満たすとき，ＤにおいてＥはＧの例であるという．

　　Ｄ∪Ｅ⊢ Ｇ．

4)　領域理論Ｄにおいて，具体例Ｅが概念Ｇの例であるとする．このとき，目標節←Ｇからはじまり，Ｄ∪Ｅを入力節とする線形入力反駁Ｒを，ＤにおいてＥがＧの例であることの説明（証明）という．

5)　線形入力導出において入力節のなす木構造を説明構造という．　■

1～4)は従来のＥＢＬの枠組みに直接対応する．5)は新たに導入したデータ構造であり，3.2節で考察する．

【仮定 3.2】　実行システムに関する仮定
1)　プログラムはＳＬＤ導出器によって実行される．
2)　推論は節の格納順序によって制御される．■

実行システムは純Prologインタープリタを想定する．

次に，知識変換手法に関する仮定を行なう．

【仮定 3.3】　知識変換手法に関する仮定
1)　ある説明構造から，裁断，マクロ化，変数化を経て得られるＤのマクロをＥＢＧという．
2)　過去の例題をすべて考慮し，それらのＥＢＧからなるマクロの集合によって領域理論を更新する．その際，初期の領域理論と付加されるマクロの集合を区別して，以前に付加されたマクロの集合を削除し，新しく得られたマクロの集合を付加して領域理論を更新する．さらに，マクロの集合は適当に順序付けられて，領域理論よりも前に挿入される．■

1)は説明構造に対して導入された一般化であり，3.3節で考察する．2)について，従来のＥＢＬに対する拡張の要点は，探索空間が複数の説明構造に対して定義され，探索が継続的に行なわれることである．複数の説明構造に関する最小ＥＢＧについて3.4節で考察する．

最後に操作性規範に関する仮定を行なう．

【仮定 3.4】　操作性規範に関する仮定
後戻りを生じないことが望ましい．これは，過去のすべての例について後戻りなく証明が再現されることと，マクロの個数が少ないことと解釈する．■

後戻りの除去はＳＬＤ導出器における典型的な操作性規範であると思われる．3.5節では，後戻り除去という操作性規範と最小ＥＢＧの関係について考察する．

3.2. 説明構造

説明構造を形式化し，その性質を考察する．

論理プログラムでは，入力節がすべて確定節であるために，初期目標節が単一リテラルからなるとき，線形入力導出は木構造をなす．これを導出の構造を表わすデータ構造とする．

【定義 3.1】　線形入力導出
論理プログラムＰにおける線形入力導出Ｓは次のような導出の列で表わされる．

$$S = [d_1, \cdots, d_n] \quad (n \geq 1)$$

ここで，個々の導出d_iは次のような内容からなる．
1)　$n(d_i)$：　目標節．
2)　$l(d_i)$：　導出リテラル(literal resolved upon)．
3)　$i(d_i)$：　入力節．Ｐの確定節の変数の書換え．
4)　$s(d_i)$：　導出リテラルと入力節との最汎単一化．
5)　$r(d_i)$：　導出形(resolvent)．次の導出の目標節．

また，Ｓに関して次の用語を定義する．
1)　d_1の目標節$n(d_1)$を，Ｓの初期目標節という．
2)　d_nの導出形$r(d_n)$を，Ｓの終端導出形という．■

以下，初期目標節が単一リテラルからなる線形入力導出のみを扱う．

【定義 3.2】　リテラル木

論理プログラムPにおいて，次のように再帰的に定義される木構造をPのリテラル木という．

1)　RとLがPの単一化可能な正リテラルのとき，式R:LはPのリテラル木である．

2)　RとAがPの単一化可能な正リテラル，$\langle B_1 \rangle$がB_1を根とするPのリテラル木のとき，式$R:A \leftarrow \langle B_1 \rangle \wedge \cdots \wedge \langle B_n \rangle (n \geq 0)$はPのリテラル木である．■

1)においてRを根リテラル，Lを葉リテラルという．2)においてRを根リテラル，$\langle B_1 \rangle$の葉リテラルを葉リテラルという．X:Yをノード，確定節 $A \leftarrow B_1 \wedge \cdots \wedge B_n$ をリンクという．リテラル木tに対して，根リテラルをR(t)で，リンクの頭部をH(t)で表わす．

【定義 3.3】　説明構造

Pのリテラル木tにおいて，すべてのリンクがPの確定節からなるとき，tをPの説明構造という．■

【定義 3.4】　リテラル木の訪問(traverse)

Pにおけるリテラル木tに対して，図3-1に示すリテラル木の訪問アルゴリズムによってtから得られる導出の列をtの訪問という．■

リテラル木の訪問はは非決定的で，一般にひとつの説明構造に対して複数の訪問が生成される．

【命題 3.1】　説明構造の性質

Pの説明構造の任意の訪問はPにおける線形入力導出である．また，訪問の解答代入，具体化，および終端導出形は訪問に対して不変である．■

特に，説明構造の深さ優先の訪問はＳＬＤ導出である．

【例題 3.1】

次の領域理論Dと例題e_1において，目標g_1に対する説明構造を図3-2に示す[Dejong 86]．

```
D = { kill(X,Y)←hate(X,Y)∧possess(X,Z)∧weapon(Z),
      hate(W,W)←depressed(W),
      possess(U,V)←buy(U,V),
      weapon(gun),
      weapon(knife)} ,
e1 = { depressed(john),
      buy(john,gun)} ,
g1 =    ←kill(john,john).
```

図中，“←”および“∧”はルールを表わし，“‥”は単一化を表わす．実線①と②は，次節で述べる裁断の例である．■

3.3. 説明構造の一般化

説明構造の一般化を形式化し，その性質について考察する．[Mitchell 86]の目標回帰アルゴリズムは，次の２つの操作によってマクロを生成する．

1) 目標を変数化して，マクロの結論とする．
2) 説明を目標から回帰してたどり，事実との照合の部分で“切って”，マクロの前提条件とする．

本研究ではこれらの操作を一般化し，裁断，マクロ化変数化の３段階にわけて形式化する．

【定義 3.5】 説明構造の裁断

説明構造ｔの部分木ｎ＝$R:A\leftarrow\langle B_1\rangle\wedge\cdots\wedge\langle B_n\rangle$ に対して，ｎの訪問の解答代入をθ，訪問の具体化をσとする．このとき，ｔにおいてｎを $R:(A\theta\circ\sigma)$に置き換えたリテラル木ｔ'を説明構造の裁断という． ■

Ｐの説明構造の任意の裁断とその余分は，やはりＰの説明構造である．

【定義 3.6】 マクロ

Ｐの線形入力導出Ｓにおいて，初期目標節を$\leftarrow A$，終端導出形を$\leftarrow B_1\wedge\cdots\wedge B_m$，各導出における最汎単一化を$\sigma_1,\cdots,\sigma_n$とするとき，確定節
$(A\leftarrow B_1\wedge\cdots\wedge B_m)\ \sigma_1\circ\cdots\circ\sigma_n$をＳに基づくＰのマクロという． ■

【定義 3.7】 説明構造のマクロ化

説明構造ｔの訪問をＳ，訪問の具体化をσ，Ｓに基づくマクロをＭとするとき，$M\sigma$をｔのマクロ化という． ■

説明構造のマクロ化は訪問に対して不変である．ｔのマクロ化をＭ(ｔ)で表わす．

【定義 3.8】 変数化

同じリンクからなり，葉のみが異なる説明構造t_1とt_2において，
$M(t_1)\sigma=M(t_2)$となるような代入σが存在するとき，t_1はt_2の変数化であるという． ■

変数化の条件$M(t_1)\sigma=M(t_2)$は[Plotkin 70]における語の汎化にである．これらの操作によって説明構造の一般化が定義される．

【定義 3.9】 説明構造上の一般化関係

２つの説明構造t_1とt_2において，t_1がt_2の０回以上の裁断の変数化であるとき，t_1はｔより一般的であるといい，$t_1\lesssim t_2$と書く． ■

【定義 3.10】 ＥＢＧ

領域理論Ｄと具体例Ｅにおける説明構造ｔに対して，$t'\lesssim t$なるｔ'のマクロ化Ｍ(ｔ')が，$D\vdash M(t')$を満たすとき，ｔ'をｔのＥＢＧという． ■

【定義 3.11】 基本裁断

領域理論Ｄと具体例Ｅにおける説明構造ｔに対して，Ｅの事実とのすべての導出 $L:F\leftarrow$ を $L:F$ に置き換える裁断をＤに対する基本裁断という．基本裁断によって得られる説明構造をＴ／Ｄで表わす． ■

【命題 3.2】　ＥＢＧの性質

領域理論Ｄと具体例Ｅにおける説明構造ｔに対して，ｔ'≦ｔなるＤ∪Ｅの説明構造ｔ'が，Ｄの説明構造であるための必要条件は，ｔ'≦ｔ／Ｄである．■

[Mithcell 86]のゴール回帰アルゴリズムはすべての事実を裁断し，最も変数化したＥＢＧを生成する．

【例題 3.2】

例題3.1の説明構造の一般化の例を示す．

1)　図3-2の実線①は基本裁断である．基本裁断①によるＥＢＧは，変数化によって次のa)を上限とし，b)を下限とする有限の束構造を形成する．

a)　kill(W,W)←depressed(W)∧buy(W,gun).

b)　kill(john,john)←depressed(john)∧buy(john,gun).

a)とb)の間にあるＥＢＧは，すべて領域理論Ｄのマクロである．a)に現われる定数gunを変数化したc)のようなマクロ，および①より特殊な裁断によってえられるd)のようなマクロは，Ｄから演繹されない結果を導くのでＤのＥＢＧではない．

c)　kill(W,W)←depressed(W)∧buy(W,Z).

d)　kill(W,W)←depressed(W).

2)　図3-2の実線②は基本裁断①に加えてweaponにおける裁断を行う．裁断②によるＥＢＧは，変数化によって次のa)を上限とし，b)を下限とする有限の束構造を形成する．

a)　kill(W,W)←depressed(W)∧buy(W,Z)∧weapon(Z).

b)　kill(john,john)←depressed(john)∧buy(john,gun)∧weapon(gun).

a)とb)の間にあるＥＢＧは，基本裁断①と比べて“銃”が“武器”に一般化されている．■

このように，単一の説明構造から裁断と変数化によって生成されるＥＢＧは，≦のもとで基本裁断を下限とする束構造をなす．

3.4.　複数の説明構造の最小ＥＢＧ

複数の例題の説明構造に対して，裁断による一般化がどのような性質を持つかを考察する．

【命題 3.3】　２つの説明構造の最小ＥＢＧの存在

領域理論Ｄにおいて，t_1を具体例E_1における説明構造，t_2を具体例E_2における説明構造とする．このとき，$t' \preceq |t_1|/D$，かつ$t' \preceq |t_2|/D$なる任意の説明構造ｔ'に対して，ｔ'≦ｔが成り立つようなＤの説明構造ｔが一意に存在する．■

［証明大略］　２つの説明構造に対して，同じリンクからなり，葉のみが異なる部分的な説明構造は一意である．また，変数化についても語の共通汎化は一意であることが知られている[Plotkin 70]．従って，２つの説明構造の最小な一般化が存在する．■

命題3.3は容易に一般化される．

【定理 3.1】　n個の説明構造の最小ＥＢＧの存在

領域理論Ｄにおいて，t_1，…，t_nをそれぞれ具体例E_1，…，E_nにおける説明構造とする．このとき，　$t' \lesssim |t_1|/D$，かつ…，$t' \lesssim |t_n|/D$なる任意の説明構造t'に対して，$t' \lesssim t$が成り立つようなＤの説明構造ｔが一意に存在する．■

以下，定理3.1におけるｔを最小ＥＢＧという．

【例題 3.3】

例題3.1の説明構造と，同じ領域理論Ｄと例題c_2における目標g_2に対する説明構造の最小ＥＢＧ，$LEBG_{12}$は次の通りである．

$e_2 = \{ depressed(tom),\ buy(tom, knife) \}$，

$g_2 = \leftarrow kill(tom, tom)$

$LEBG_{12} = kill(W, W) \leftarrow depressed(W) \wedge buy(W, Z) \wedge weapon(Z)$.

ここで，$LEBG_{12}$は図3.2における裁断②と，johnとtom，gunとknifeの変数化からなる．■

3.5.　後戻りの除去と最小ＥＢＧ

ＳＬＤ導出器における後戻りの除去という操作性規範と，最小ＥＢＧの関係について考察する．

後戻りは，問題解決における探索コストを反映する．後戻りの除去には次の２つの側面がある．

1)　過去のすべての例題について，後戻りを生じない．

2)　将来の例題について，なるべく後戻りを生じない．

1)は例題を解決し直すことによって厳密に検証できる．2)の検証は困難で，1)とトレードオフの関係にある．1)では特殊性が選好されるが，2)では一般性が選好されるからである．そこで，1)と2)を考慮した次のような操作性規範を定義する．

【定義 3.11】　操作性規範

t_1，…，t_nをそれぞれ具体例E_1，…，E_nにおける説明構造とする．このとき，t_1，…，t_nのＥＢＧからなる集合Ｔが，操作性規範を満足するとは，次の２つの条件が満足されることである．

1)　適当な順序付けが存在して，Ｔのマクロを領域理論の先頭にその順序で挿入したときに，具体例E_1，…，E_nに対して，t_1，…，t_nが後戻りなしで再現される．

2)　t_1，…，t_nのＥＢＧの集合Ｔ'が1)の条件を満足するならば，Ｔ'のマクロの個数はＴ以上である．■

前章で述べたように，一般に操作性規範と一般化関係は無関係である．操作性規範を満足するＥＢＧが一般化の空間において範囲を形成するようであれば，ＥＢＧの探索にバージョン空間の概念が利用できる．

【命題 3.4】　$\lesssim$による操作性の保存

一般化関係$\lesssim$は操作性規範を保存する．■

命題3.4により，t_1，…，t_nのＥＢＧの集合は，バージョン空間を形成する．上限は後戻りがないこと，下限はマクロの数によって制限される．

【定理 3.2】　最小ＥＢＧとバージョン空間の関係

最小ＥＢＧが操作的ならば，それはバージョン空間の下限である．　■

最小ＥＢＧが操作的でなくなった場合，例題の集合を適当に分割すれば，それらの最小ＥＢＧからなるマクロの集合が操作的であれば，やはりそれらはバージョン空間の下限となっていることが予想される．この予想の概念図を図3-3に示す．図は３つの説明構造のＥＢＧ空間の重なり工合を表わし，斜線は操作的な範囲を表わす．左斜線は過去の問題について後戻りしない範囲を，右斜線はその中でマクロの個数が少ない範囲を表わす．図3-3-aは最小ＥＢＧが操作的な場合，図3-3-bは最小ＥＢＧが操作的でない場合に，問題を分割して２つの操作的な最小ＥＢＧを得たところである．この予想の確認は将来の課題である．ここでは，最小ＥＢＧが非操作的となる場合の原因について考察する．

【定義 3.12】　問題領域，手続き化可能性，実行概念

領域理論Ｄ，概念Ｇ，具体例のクラスＥの組を問題領域という．問題領域が手続き化可能であるとは，有限個のＤのマクロと節選択関数が存在して，任意のＥの具体例がＧの例であることの説明を，後戻りなしに得ることをいう．このときのマクロの集合を実行概念という．　■

【定義 3.13】　説明構造の集合の一貫性

問題領域が手続き化可能であるとき，過去のすべての例題の説明構造が，同一の実行概念によって構成されているならば，その説明構造の集合は一貫性を持つという．　■

【定理 3.3】

最小ＥＢＧが操作的でないとき，つぎの３つのいずれかが成り立つ．

1)　問題領域が手続き化可能でない．
2)　例題の説明構造の集合に一貫性がない．
3)　実行概念が記述概念（述語）と一致しない．　■

3.6. 適用例

前節で述べた最小ＥＢＧの概念をハノイの塔における再帰的手続きの獲得に適用する．ハノイの塔はよく知られた問題解決の例題である．

【例題 3.4】　ハノイの塔

３本の塔A,B,Cがあり，そのうちAには何枚かの大きさの異なる円盤がある．塔から塔へ１度に１枚ずつ円盤を移動して，Aのすべての円盤をCに移動する手順を求めよ．ただし，移動先の塔には円盤がないか，または移動される円盤は，移動先の塔の頂上にある円盤より小さくなければならない．　■

ハノイの塔の問題は，通常状態計算による探索問題として形式化され，図3-4に示すような手続によって決定的に解決されることが知られている．ハノイの塔を探索問題として形式化した領域理論のもとで，異なる枚数の円盤を持つ例題を与え，最小ＥＢＧによって再帰的手続きを獲得することを試みる．

領域理論を図3-5に示す．これらは必要最小限の状態遷移規則と，再帰的手続きを得るために付加された知識とからなる．必要最小限の状態遷移規則からでは，再帰的手続きを得られないことに注意されたい．再帰的手続きは「塔Fromの一番下の円盤以外の円盤をWithに移動し，Fromの一番下の円盤をToに移動した後，Withの円盤をToに移動する」ことからなる．生成される副問題が主問題と同一の手続きによって再帰的に解かれるためには，少なくとも塔の役割を識別することと，一番下の円盤の固定を識別することが必要である．

付加された知識は次の内容からなる．

1)　塔の交換規則

「２つの塔を交換しても問題は変らない．」

2)　円盤の固定規則

「状態遷移の前後である塔の一番下の円盤が移動しないならば，その円盤を固定して解いても問題は変らない．」

円盤の固定規則は，一番下の円盤の識別のために特殊化して実装されている．状態は述語hanoi/3によって表現される．第１引数は塔の交換規則で参照される塔の役割を表わす．第２，３引数はそれぞれ状態遷移の前後の状態を表わす．個々の塔の円盤の固定状況は重リストの形式で表わす．例えば，[1,2,3]-[3]は円盤1,2,3がこの順でその塔にあり，3が固定されていることを表わす．

領域理論において，call/1は制約の計算を示すメタ述語で，理論上各例に固有の事実とみなし，必ず裁断されるものとする．単一化はoccur checkをともなう．

最小ＥＢＧの生成器を，PC98XA，MS-DOS上のArity Prolog Compiler (Ver.4.0)に実装して，２枚の円盤と３枚の円盤からなる２つの問題について，それぞれ解答を教示して最小ＥＢＧを生成させた．獲得された最小ＥＢＧを図3-6に，その説明構造を図3-7に示す．これは前述の再帰的手続きに対応する．

獲得された再帰的手続きについて考察する．獲得された手続きは，塔Aに２枚以上ある円盤を塔Cにそっくり移動するタイプの問題に適用可能である．このタイプの問題のみが与えられる場合にはマクロの個数が増えることはない．このことは，例題の増加に従ってマクロの数が増加し続けるシステムと比較して重要な性質である．

この領域理論では，塔Aの円盤を塔Cにそっくり移動するタイプの問題に限らず，任意の状態から任意の状態へ遷移する問題が記述可能である．このタイプから外れたとき，最小ＥＢＧは，過去の例に対しても後戻りを生じる過剰汎化を引起こす．このとき，例題の集合を分割して，マクロを集合として維持することは今後の課題である．ハノイの塔は，任意の状態遷移に対しても手続化可能であることが知られている[Korf 85]．

獲得されたマクロの第２，３連言項は，広い意味で冗長な制約の計算である．第２連言項の単一化失敗の検証は，現在の領域理論においても部分計算によって除去可能である[Prieditis 87]が，第３連言項のsafe_to_stackについては不可能である．

4．おわりに

本研究では，問題解決システムにおける学習研究のための新しい枠組みとして，操作性規範に基づいてＳＢＬとＥＢＬを統合化した操作化問題を提案した．操作化問題の研究例として論理プログラムにおけるＥＢＬの拡張形について理論的に考察した．その結果，複数の説明構造の最小ＥＢＧは後戻りの除去という操作性のバージョン空間の下限をなすこと，最小ＥＢＧとしてハノイの塔の再帰的手続が獲得されることを示した．

操作化問題と従来の機械学習の枠組みとの根本的な違いは，単なる“概念”ではなく，学習された概念を利用するシステムにとって有用である“操作的な概念”を獲得することが目的となる点にある．従って，正例・負例の区別は，操作性規範との関連から見直さなければならないことに注意されたい．ＳＢＬでは，例と仮説の整合性が操作性の必要条件であるために，概念に対する正例・負例の区別が重要である．これに対して，本研究で示したＥＢＬの拡張形では，後戻りの除去のような実行にかかわる操作性が求められるために，概念に対する正例・負例の区別は重要ではない．むしろ，「現在のマクロでは過剰一般化のために後戻りを生じてしまう」ような例（概念に対しては正例）が負例に相当する情報を持っている．

今後の課題としては，単一の最小ＥＢＧだけではなく，マクロを集合として維持し，手続化可能な問題領域全般に有用なＥＢＬ機構の開発，制約的知識を含む領域への拡張などが考えられる．

操作化問題においてはＥＢＬの操作性規範とＳＢＬのバージョン空間法の利点が結合される．従来の接近法では困難とされていた領域における学習問題の解決に役立つものと期待される．

参考文献

[Mitchell 82] Mitchell,M.T."Generalization as Search," Artificial Intelligence,Vol.18,pp203-226(1982).
[Mitchell 86] Mitchell,M.T. Keller R.M. and Kedar-Cabelli,S.T. "Explanation-Based Generalization," Machine Learning, Vol.1, pp.47-80 (1986).
[Keller 87] Keller,R.M. "Defining Operationality for Explanation-Based Learning," AAAI 87,482-487(1987).
[DeJong 86] DeJong G. and Mooney,R. "Explanation-Based Learning: An Alternative View," Machine Learning, Vol.1, pp.145-176 (1986).
[Mostow 84] Mostow,D.J. "Machine Transformation of Advice into a Heuristic Search Procedure," in Michalski R.S. et.al. ed., Machine Learning, Springer-Verlag (1984).
[Utgoff 84] Utgoff,P.E. "Shift of Bias for Inductive Concept learning," Dr.thesis, Ritgers Univ., 1984.
[Plotkin 70] Plotkin,G.D. "A Note on Inductive Generalization," Machine Intelligence 5, 1970.
[Laird 86] Laird,J.E. Rosenbloom P.S. and Newell,A. "Chunking in Soar: The Anatomy of a General Learning Mechanism," Machine Learning, Vol.1, No.1, pp.11-46 (1986).
[Korf 85] Korf,R.E "Macro-Operators: A Weak Method for Learning," Artificial Intelligence, Vol.26, pp.35-77 (1985).
[Prieditis 87] Prieditis A.E. and Mostow,J. "PROLEARN: Towards A Prolog Interpreter that Learns," AAAI 87, pp.494-498 (1987).
[山村 88 a] 山村雅幸,李相能,小林重信 "ＳＢＬとＥＢＬの融合による知識獲得," 第７回知識工学シンポジウム資料 (1988).
[山村 88 b] 山村雅幸,小林重信 "操作性規範に基づくＳＢＬとＥＢＬの統合化," 学習と知識獲得に関する特別講演会資料 (1988).

(Given)
1) 具体例記述言語(I):具体例を記述する言語.
2) 一般化記述言語(G):一般化仮説を記述する言語.
3) 照合述語(∈)　　:具体例と仮説の照合を調べる述語.
4) 練習例(T)　　:具体例と正例・負例の判定の対の集合.
(Determine)
次の整合性条件を満たす一般化仮説 g ∈ G ;
∀i∈T　(positive(i)→i∈g) ∧ (negative(i)→i∉g).

図 1-1 SBLの枠組み

(Given)
1) 目標概念　:操作性規範を満たさない初期記述.
2) 練習例　　:目標概念の単一の正例.
3) 領域理論　:ルールと事実の集合.説明を構成するのに用いられる.
4) 操作性規範:学習される概念記述を記述すべき形態を特定する述語.
(Determine)
操作性規範を満たす練習例の説明に基づく一般化.

図 1-2 EBLの枠組み

〈Given〉
1) 概念記述　　：（ＥＢＬで生成される一般化．）
2) 実行システム：性能向上のために1)の概念記述を用いるシステム．
3) 実行目的　　：要請されるシステムの性能向上の型と範囲を特定．
〈Then〉
次の２つの条件を満たすならば，概念記述は操作的である．
1) 利用可能性　：実行システムで利用可能でなければならない．
2) 効用　　：実行システムで利用したときに，実行目的に沿った性能向上がなければならない．

図 1-3 Kellerの操作性規範

〈Given〉
1) 問題解決器　：記述言語，インタープリタからなる機械．
2) 知識変換手法：SBG，EBG等，現在の知識から新しい知識を得る手法．
3) 操作性規範　：能力向上の観点から，知識の適切さを判定する規範．
4) 現在の知識　：領域理論，制御知識，過去の経験．
〈Determine〉
操作性規範を満たす現在の知識の変換．

図 2-1 操作化問題の枠組み

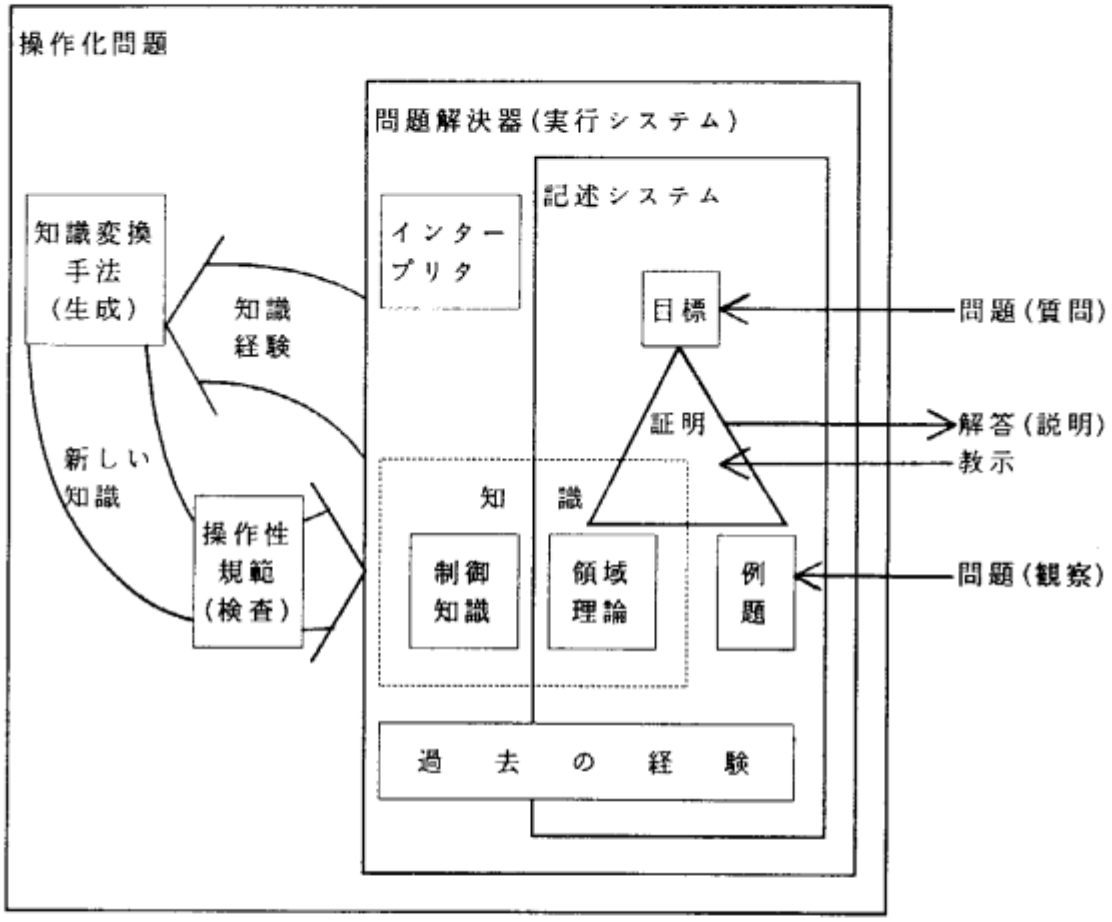

操作化問題
問題解決器(実行システム)
記述システム
知識変換
手法
(生成)
インター
プリタ
知識
経験
目標
問題(質問)
証明
解答(説明)
教示
新しい
知識
知識
操作性
規範
(検査)
制御
知識
領域
理論
例題
問題(観察)
過去の経験

図 2-2 操作化問題

```
リテラル木の訪問(t, S, θ, σ)
  入力:  リテラル木t.
  出力:  導出の列S = [d1,…,dn], 解答代入θ, 具体化σ.
begin
  T = {t} ; θ = φ ; σ = φ ; n(d1) = ←R(t) ; i=1 ;
  while T ≠ φ do
    l(di) = リテラル選択(n(di)) ;
    t     = リテラル木選択(T) ;
    s(di) = 最汎単一化(l(di), H(t)) ;
    リテラル木の分解(t, i(di), T', λ) ;
    r(di) = 導出(n(di), i(di), l(di)) ;
    T = (T − t) ∪ T' ; θ = θ ∘ s(di) ; σ = σ ∘ λ ; n(di+1) = r(di) ;
    i = i+1 ;
end

リテラル木の分解(t, ρ, T', λ)
  入力:  リテラル木 t = R:A←<B1>∧…∧<Bn>.
  出力:  リンク ρ = A←B1∧…∧Bn, リテラル木集合T', 具体化λ.
begin
  T' = φ ; λ = φ ;
  for i from 1 to n do
    Bi = R(<Bi>) ;
    if <Bi> = R:L then λ = λ ∘ 最汎単一化(R, L)
                  else T' = T' ∪ {<Bi>}
end
```

図 3-1 リテラル木の訪問アルゴリズム

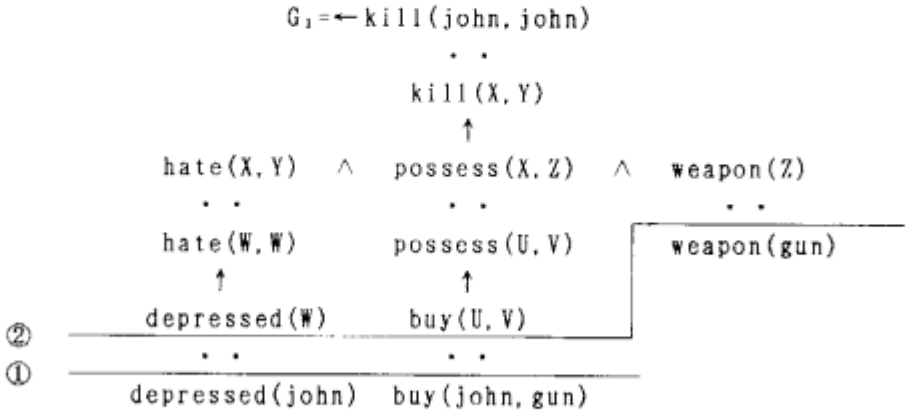
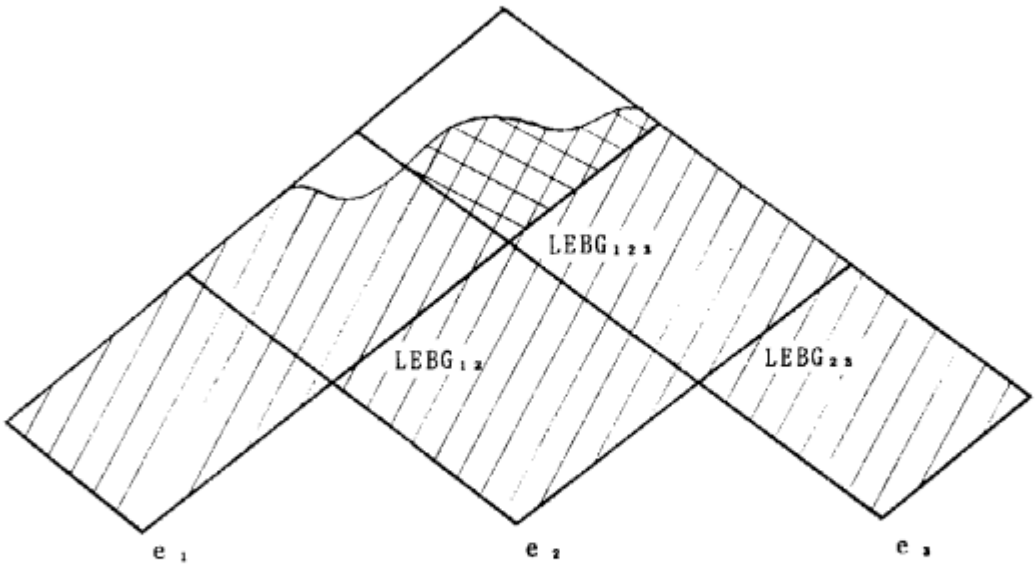
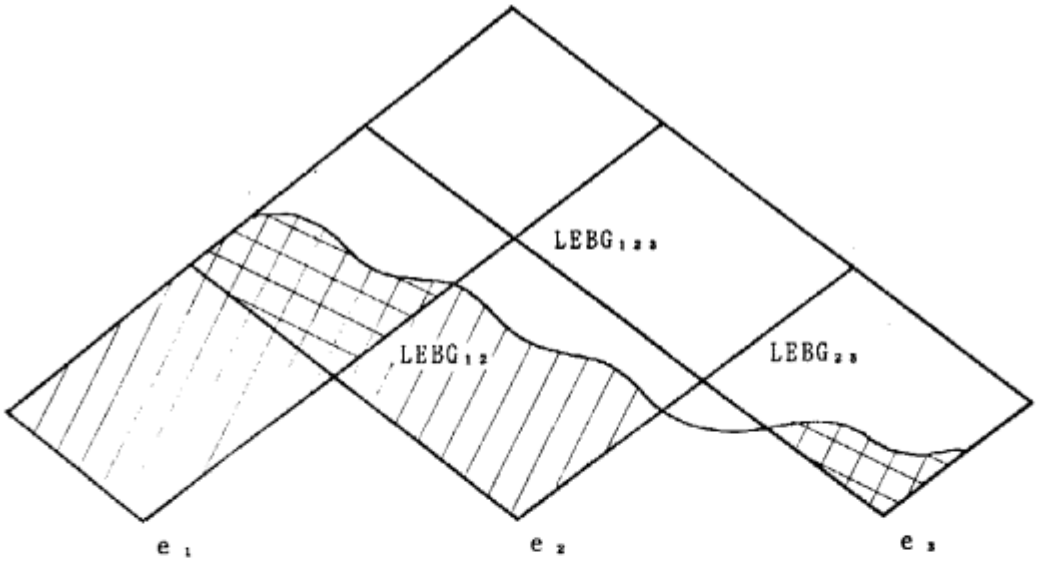


図 3-2 例題3.1の説明構造



a)　最小ＥＢＧが操作的な場合



b)　最小ＥＢＧが操作的でない場合

図 3-3 最小ＥＢＧと後戻り除去の関係

```
hanoi(1,From,With,To)←move(1,From,To).
hanoi(N,From,With,To)←
        M is N - 1∧
        hanoi(M,From,To,With)∧
        move(N,From,To)∧
        hanoi(M,With,From,To).
```

図 3-4 ハノイの塔の手続き

```
/* 円盤移動 */
hanoi(_,s([P|A]-AT,B-BT,C-CT),s(A-AT,B-BT,[P|C]-CT))←
      free(P,[P|A]-AT)∧
      safe to stack(P,C-CT).
           ・・・(6通り)
/* 円盤の固定チェック */
free(P,T-TT)←call(T≠TT).
/* 円盤の大きさチェック */
safe_to_stack(P,[]-[]).
safe to stack(P,[Q| ]- )←call(P<Q).
/* 行動の合成 */
hanoi(T,S1,S3)← hanoi(T,S1,S2)∧hanoi(T,S2,S3).
/* 塔の交換規則 */
hanoi(t(From,To,With),s(A-AT,B-BT,C-CT),s(NA-AT,NB-BT,NC-CT))←
    hanoi(t(From,With,To),s(A-AT,C-CT,B-BT),s(NA-AT,NC-CT,NB-BT)).
           ・・・(6通り)
/* 円盤固定規則 */
hanoi(T,s(A-AT,BT-BT,CT-CT),s([P|AT]-AT,B-BT,C-CT))←
    hanoi(T,s(A-[P|AT],BT-BT,CT-CT),s([P|AT]-[P|AT],B-BT,C-CT)).
           ・・・(3通り)
hanoi(T,s([P|AT]-AT,B-BT,C-CT),s(A-AT,BT-BT,CT-CT))←
    hanoi(T,s([P|AT]-[P|AT],B-BT,C-CT),s(A-[P|AT],BT-BT,CT-CT)).
           ・・・(3通り)
```

図 3-5 ハノイの塔の領域理論

```
hanoi(t(F,W,T),s([1,2|AH]-A,B-B,C-C),s(A-A,B-B,[1,2|CH]-C)←
        hanoi(t(F,T,W),
                s([1,2|AH]-[P|A],C-C,B-B),
                s([P|A]-[P|A],C-C,[1,2|BH]-B))∧          ・・・①
        call([P|A]≠A)∧                                  ・・・②
        safe_to_stack(P,B-B)∧                           ・・・③
        hanoi(t(W,F,T),
                s([1,2|BH]-B,A-A,[P|C]-[P|C]),
                s(B-B,A-A,[1,2|CH]-C)),                 ・・・④
```

図 3-6 ハノイの塔の最小ＥＢＧ

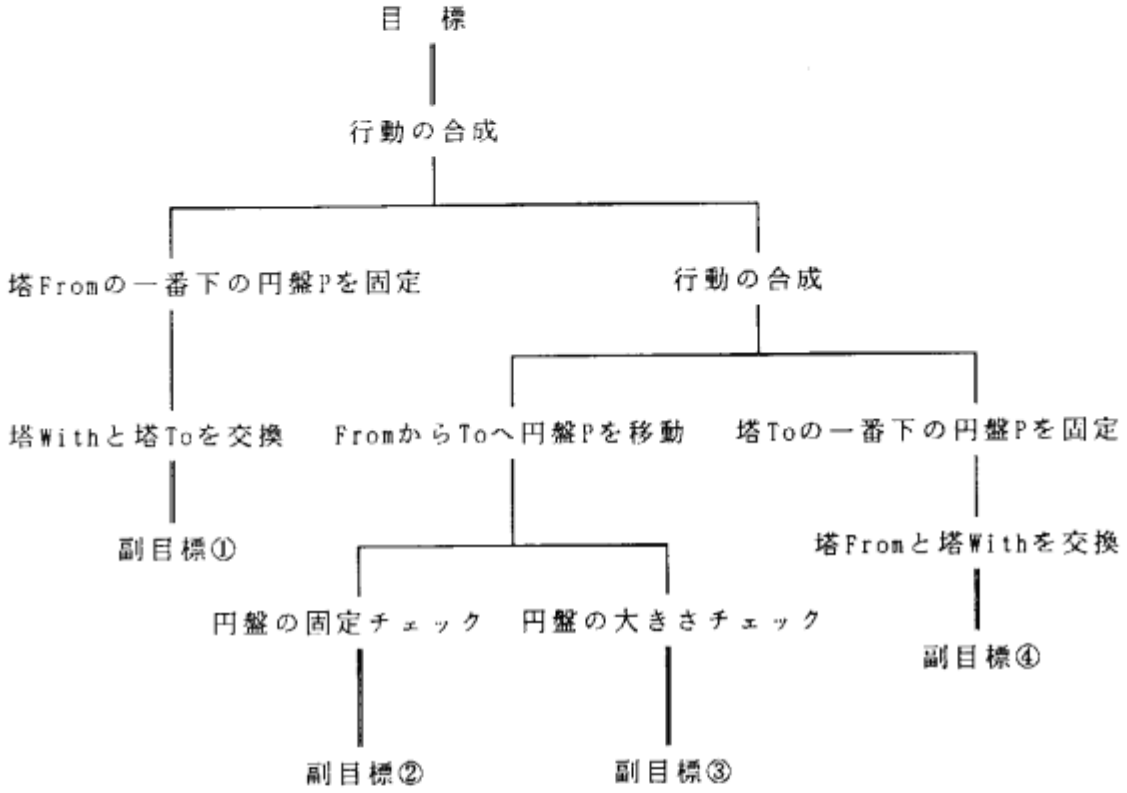


図 3-7 最小ＥＢＧの説明構造

３．４　設計事例からの問題解決知識の獲得

京都大学 工学部 精密工学教室
片　井　　修

設計過程における計算機支援を考えるとき、設計過程の明確なモデル化を与えることは、極めて重要な問題となる。そのような意味から、ここでは Mostow[1] による設計過程のモデル化に関する survey paper を紹介する。彼によれば、このモデル化のためには、設計の状態、目標構造あるいは設計における決定とその根拠、さらには設計過程の制御および設計における学習の役割を調べること等が重要な項目とされる。以上のことを考慮するとき、設計された（されている）事物そのものに対するモデルを導入することは極めて重要なものと考えられる。

ここでは、設計物に対するモデルとして、その目的（N）、機能（F）、構造・属性（S）、実体（R）の４つのレベルから事物を表現する方法を導入し[2]、設計事例からの一般化された設計に関する知識の獲得法について検討する。

このモデルにおいては、一つの設計物は、その最終目的（n）を達成するための機能群（f）およびそれを支える構造・属性群（s）、さらにはそれを実現する実体群（r）から構成されるものとして説明・理解される。

この説明には、生物学に見られるような目的論的なものと、物理学に見られるような因果機械論的なものが考えられる[3]。何れにしても、この説明のためには、それぞれのレベル N, F, S, R の間に、これらを関連付け、その根拠を与える一般的法則群（被覆法則 covering law[4]）の設定が必要となる。例えば、F と S の間については、いわゆる物理法則群を設定することによって、ある機能 f の達成に対してある構造・属性群 s が必要とされる理由を示すことができる。

ここでは、このような説明に基づいて、設計の際に必要となる一般的な知識の獲得について、EBL に類似した方法の導入を行う。また、これと EBL の関係についても検討する。さらに、S と R の間のように、一般的な法則群が前もって設定できず、経験的知識の活用が重視される部分については、weak な 一般的法則群（domain theory）の下での一般的知識の獲得を与える、Russell による DBR（determination-based reasoning）の活用が考えられる。この意味から、Russell らの論文[5,6]に準拠しながら、DBR の紹介を行うとともに、これの設計知識獲得への活用法について検討を加える。

３．４．１　設計支援と設計過程のモデル

知識工学的手法に基づいた設計支援システムを考えるとき、設計過程のモデル化を行うことは不可欠のことと考えられる。Mostow[1] によれば、このモデル化のためには以下の視点が重要とされる。

1)　設計の（中間）状態を陽に表現すること。
2)　設計の目標（goal）を陽に表現すること。

3) 設計における決定を陽に表現すること。
4) 設計における決定の根拠を明らかにすること。
5) 設計過程の制御を理解できるようにすること。
6) 設計における学習の役割を調べること。

それぞれについては以下のようなことが述べられている。

3．4．1．1 設計状態の表現

設計の状態とは、設計されつつある物の経時的な表現として表すことができる。設計過程の抽象化された表現を行うことが重要である。この一つの方法としては、無駄に終わった選択枝を除去することである。理想化された設計履歴を作ることは、ドキュメンテーション、理解可能性、デバッギング、検証、解析、説明、自動化等に有用であることが知られている。

設計物の表現として、抽象的精密化モデル（abstract refinement model）では、設計物は仕様からそれを実現する要素への分解を通して表現される。一方、変換モデルでは、設計物は、仕様から（正当性の保証された）一連の変換によって実現されている物として表現される。これら変換は一度に複数の要素に作用することがあり、したがってこのモデルの方が一般的といえる。

3．4．1．2 設計の目標の表現

一般に目標構造は木として表すことができ、設計物に対する目標木モデル（goal tree model）を考えることができる。仕様からその実現に至る変換は、目標を実現するものとして表されている。このような目標構造の表示は改良を含む設計の再試行を容易にする。また、完全な目標構造は設計者が扱うには煩雑過ぎており、人間支援の計算機による設計（human-aided design ≠ machine-aided design）が必要となる。

目標構造の表現では、互いに干渉する目標を適切に表現する必要がある。この干渉関係としては、以下のものがある。

目標の競合： 両者を共には達成することができない。
目標の共有： 一方の副目標の達成が他方の達成にも貢献する。
目標の先行関係： 一方が他方より先に達成される必要がある。

さらに、設計における付加的な目標も表現される必要がある。これらには以下のものが含まれる。

機能的目標： 機能的仕様の実現。
性能的目標： 効率、コスト、信頼性等。
知識目標： 設計に必要な情報の収集。
設計過程の目標： 設計過程に要する時間、コスト等。

3．4．1．3 設計における決定の表現

目標木モデルにおいては、目標がどのような方法によって副目標に分解されるかは説明されていない。設計における決定は、選択集合としてモデル化できる。この場合、目標木には、各目標を達成するための複数の競合した方法が付随することになる。また、設計の仮定および拘束を陽にする必要がある。このとき、設計における選択に際して、特定の仮定や拘束を置く必要があるかについて推論するメカニズムが必要となる。

さらに、意思決定は陽な目標として表される必要がある。この意思決定の包括的なモデルには次の過程が含まれていなければならない。

決定の枠組みを作る。
代替案を生成する。
代替案評価の基準を作る。
基準にしたがって代替案を評価する。
受理可能な代替案を選択する。
これが不満足であれば、これを取り消す。

3．4．1．4　設計における決定の根拠

設計に関する根拠づけとして、次のような種類が考えられる。

正当性：　目標達成のプランが正しく働くこと。
適切性：　他の代替プランでなくこれが選ばれた理由。

これらの根拠は、新しい設計問題を解く上で既存の設計過程を再実行する際に有用である。

また、要素が仕様を満足することを形式的に証明することは、類似の要素を自動的に実現するための規則として一般化することができる。換言すれば、”説明にとって良いものは設計にとっても良い”と言うことができる。

3．4．1．5　設計過程の制御

互いに干渉する目標を取り扱うための戦略を陽に表現する必要がある。目標間の関係としては、以下のものが考えられる。

a）独立性：　互いに影響を及ぼし合わない。
b）協　働：　一方の達成が他方の達成を容易にする。
c）競　合：　一方の達成が他方の犠牲によって可能となる。
d）干　渉：　一方の達成には他方の目標を考慮する必要がある。

関連した制御戦略としては、a）については、特に戦略を必要としない。b）の場合には、他方の前提条件あるいは他方よりもより一般的な目標である場合には、それを先に達成する方法が考えられる。c）の場合、より重要でない目標を犠牲にするか、あるいはこれを緩和して他方の目標と両立可能にする、さらには、目標が相対的な選好として与えられている場合には、トレードオフとして扱うことが

考えられる。d）の場合には、解に対する拘束がより少ない目標から順に達成して行く、クリティカルな決定を最初に行う、目標を併合する、あるいは目標を選択基準として用いる等が考えられる。また、ボトムアップ型設計の場合、要素を個別に設計し、それらの間のインターフェースを設計するのも一つの方法である。

大量の探索を伴う設計タスクを実行する為には、ヒューリスティックな方法とアルゴリズミックな方法を統合して用いる必要がある。記号処理とテストケースに対するシミュレーションは、設計過程の重要な部分である。前者は、欠落あるいは矛盾した要素に対する注意を促し、後者は、ボトムアップなアイデアの活用を促す。さらには、コンパイルされた目標と選択に対するモデル化を考える必要がある。例えば、ルーチン的な設計においては、熟練者は多くの目標に注意を払っていない。したがって設計システムは、全ての目標を陽に表現する必要はないが、システム（あるいはそれが作る設計）の完全な説明では、どのようにして全ての目標が達成されているかを記述する必要がある。

設計作業におけるプロトコールの研究は、トップダウンの目標駆動的処理以外の推論制御メカニズム、例えば、欠落していた情報が利用可能になったときにある目標を復帰させるデモン等、の可能性を示しつつあるが、今後の研究課題である。このような認知的アプローチはまた、人間と機械の協働の型を開発するためにも必要となる。

3．4．1．6　設計過程における学習の役割

人々が設計問題を解く（あるいは解くのに失敗する）とき、同時に、特定の設計に関する知識と問題領域に関する一般的な知識を獲得していることに注目する必要がある。特定の設計問題に対する学習は重要であるり、今後の研究課題である。これによって、設計空間における探索を大幅に低減することができる。また、探索それ自身も一種の学習と見なすことができる。すなわち、探索者は、失敗した候補を二度と試みることはない。より巧妙には、失敗した候補よりもさらに一般的な候補に対しても試行しない方法が考えられる。

専門家の作業に基づいた学習は、設計知識の獲得に有用なパラダイムである。例えば、人手によって設計された一つの回路が正しく動作することを証明し、これを一般化することによって同様の回路の設計知識を獲得することができる。このような形式的な証明が活用できない場合、非形式的（経験的）な設計知識が必要とされるが、これの獲得の方法は今後の研究課題である。

また、経験からの一般的な設計知識の獲得法を開発する必要がある。このためには、探索を制御するための知識の自動獲得、設計過程のガイドの仕方の学習、次に取り掛かるべき目標を選択することに関する学習が必要となる。

3．4．1．7　設計過程の構成要素と学習の分類

以上をまとめると、設計は多重の問題空間で作動する、と言うことができる。その構成要素としては、以下のものが考えられる。

設計物（あるいはその部分）の記述：　機能的および構造的な記述がしかもいくつかの詳細さのレベルで必要とされる。

目標：　設計物に対して何を為すべきかを規定する。設計物が満足すべき条件から構成要素に対する制約が課せられる。これはしばしば要素に対する制約伝播によって達成される。

信念：　設計物記述に対する解析、シミュレーション、記号計算、ヒューリスティックな仮定等から導かれるものであり、設計当初の設計物記述では陽ではないが、推論過程において生成され、設計過程をガイドするために使われる。

例題：　設計の性質を発見する上で有用である。テストデータに対する設計解のシミュレーションは、バグを見いだしたり、性能推定に有効である。

正当化：　信念が導かれる道筋を示すものであり、形式的証明や、ヒューリスティックな推論則がある。これらは、設計の導出、再実行、説明、一般化に有用である。

決定：　代替案間の競合を表すものであり、陽に表されれば、目標、信念によって参照することができ、決定すること自身も一つの目標として考えることができる。

設計に関する最近の人工知能研究の中心的課題は、これらの要素を含む設計過程のモデルを計算機に乗るように構成することである。この場合、設計過程における状態の良い経時的表現が必要となる。これによって、状態を変化させるためのオペレータ、目標、決定の同定、解析、実行、決定の根拠、決定の基礎となる仮定、設計過程をガイドする制御戦略を導くことができる。

設計における学習は以下の３×３×２の分類が可能である。

獲得される設計知識の種類：
- 設計目標を解く新しい知識
- 与えられた目標に対して適用すべき方法を判断するための基準
- 各時点で取り組むべき目標を選択するための戦略

獲得される知識の一般性：
- 個別の設計問題に固有の知識
- ある領域に適用可能な知識

学習のためのデータの源：
- 自身の経験
- 専門家の挙動の観察

これらの組合せの中で機械学習システムとして実現されているものは殆どない。このためには上述の表現問題の解決が必要とされる。

ここに述べてきた種々の側面をよりよく理解することによって、設計過程モデ

ルを共有しながら人間と協働するような設計システムの構築が期待される。

3．4．2　DBR － 決定関係に基づいた推論による一般化

Russell によれば、類推と一事例からの一般化（SIG: single-instance generalization）は、もし正当化できるのであれば共通の基盤を持つものとされる[5]。これら正当化（justification）の様式としては、説明に基づく推論（EBR: explanation-based reasoning あるいは EBL: explanation-based learning）と 決定関係に基づく推論（DBR: determination-based reasoning）がある。これらはともに健全（sound）な推論を保証するが、後者の方がより広範囲に適用可能である。すなわち、事例（instance）に含まれる情報を活用することによって、強力な領域固有の知識（strong domanin theory）を必要としない。人工知能の分野では、類推と一事例からの一般化は異なった分野としてとらえられてきたが、哲学の分野では、両者は妥当性（validity）に関して同一の条件を持っていることが知られていた。以下では、類推および一事例からの一般化の正当化のために必要とされる領域固有の一般法則の形を明らかにすると共にこの結果と上述の DBR および EBR の関係について検討する。

3．4．2．1　類推の正当化

source と target の間の類推は、以下のように表される。

$$P(S,A) \wedge P(T,A) \wedge Q(S,B) \xrightarrow{anal.} Q(T,B)$$

ここに、P は既知の共通性、Q は推定された共通性、A，B はそれらの source と target に共有される事例（instantiation）である。

ここで注意すべきことは、類推においては Q(T,B) が、P(T,A) の成立以外の T の性質を用いることなく導かれていることから、もし、これが正当化できるのであれば、以下の SIG が成立するものと考えることができる。

$$P(S,A) \wedge Q(S,B) \xrightarrow{SIG} (\forall x)(P(x,A) \rightarrow Q(x,B))$$

ここで A と B は単なる事例に過ぎないことに注目すると、次のような一般則も導かれていると考えることができる。

$$(\forall A)(\forall B)((\exists x)(P(x,A) \wedge Q(x,B))) \rightarrow ((\forall x)(P(x,A) \rightarrow Q(x,B))) \quad (*)$$

この法則は、通常の類推が S と T の類似性を根拠にしているのに対して、P と Q のある種の関連性あるいは依存性を示すものである（図１参照）。

3．4．2．2　正当化の背後にある知識

上述の法則を導く背景となる知識としては、例えば、結論である

$$(\forall x)(P(x,A) \rightarrow Q(x,B))$$

そのものが領域固有の知識として与えられている場合がある。このとき、Q(T,B)は、P(T,A) から source S に参照することなく導かれている。EBR（EBL）による一般化は、実はこのような強力な背景知識を用いていることになる。

すなわち、EBL の過程は以下のように表すことができる。

P, P', P": arbitrary predicates (or predicate schemata)
Q: predicate (corresponding to goal concept)
1) $P''(x) \leftrightarrow Q(x)$: definition of goal concept
   P": not operational definition for Q
2) $P(A) \wedge Q(A)$: a training example
3) a domain theory which is strong enough to show that
   $P(x) \rightarrow {}^{\exists} P'(x)$ (#)
   and
   $P'(x) \rightarrow P''(x)$ ($)
   where
   P': operational
4) EBL returns P' as a sufficient operational definition for Q.

先にも述べたように、training example は本当は必要とされない。この意味から事例に基づいた推論としての以下の要件を満たすものではない。

非冗長性（non-redundancy）[6]：　背景となる知識が不十分であり、souce S の持つ情報が類推あるいは SIG において本当に活用される必要がある。

より弱い知識によっても上述の法則的知識が導かれる場合がある。(*) に示したような P と Q の間に関連性がある場合を考える。最も単純な場合として、A と B の値域がそれぞれ {1} と {0,1} であり、P(x,1), Q(x,0), Q(x,1) をそれぞれ P(x), Q(x), Q'(x) と表す。もし、Q と Q' の間に排他性と包括性

$$(\forall x)(Q'(x) \rightarrow \neg Q(x)) \wedge (\forall x)(\neg Q(x) \rightarrow Q'(x))$$

が成立するものとすると、(*) は以下のような関係を表していることがわかる。

$$(\forall x)(P(x) \rightarrow Q(x)) \vee (\forall x)(P(x) \rightarrow \neg Q(x))$$

すなわち、どのような x に対しても性質 P が "Q であるか否かを決定" していることになる。この関係があれば、source S ならびに target T について、P(S), Q(S), P(T) から Q(T) の成立することが導かれ、かつこれは S の性質である "Q(S) の成立していること" なしには導かれない。（Q(S) と P(S) の存在によって、$(\forall x)(P(x) \rightarrow \neg Q(x))$ の場合が排除される。）　また逆に、A の値域が {1,0} であり、Bの値域が {1} であるときにも、P に関する排他性と包括性

$$(\forall x)(P(x) \rightarrow \neg P'(x)) \wedge (\forall x)(\neg P(x) \rightarrow P'(x))$$

が成立するならば、やはり上述の P と Q の間の決定関係の成立が示される。

このような排他性を持つ二値の変数を Russell は、"極性変数（polar variable）"と呼んでいる[5]。

3．4．2．3　決定関係に基づいた正当化

Russell によって導入された"決定関係"は、(*) を拡張した形で以下のように与えれれる[6]。

$$P(x,y) \succ Q(x,z) \quad (\text{"P determines Q"})$$
$$\Longleftrightarrow \quad (\forall \underline{y})(\forall \underline{z})((\exists \underline{x})(P(\underline{x},\underline{y}) \wedge Q(\underline{x},\underline{z})) \rightarrow ((\forall \underline{x})(P(\underline{x},\underline{y}) \rightarrow Q(\underline{x},\underline{z})))$$

ここに、変数ベクトル $\underline{y}$ と $\underline{z}$ には極性変数が含まれることを許す。例えば、P, Q として、それぞれ

$$P(x,y) \equiv (\text{Athlete}(x) \wedge \text{Student}(x) \wedge \text{School}(x,s) \wedge \text{Year}(x,v) \wedge \text{Sport}(x,w) \wedge \text{Female}(x,i)),$$
$$(\underline{y}=\{s,v,w,i\},\ i:\text{polar variable})$$
$$Q(x,z) \equiv (\text{Coach}(x,c) \wedge \text{Sit-ups}(x,j)),$$
$$(\underline{z}=\{z,c,j\},\ j:\text{polar variable})$$

をとると、i = 0 と i = 1 はそれぞれ女性であるか否かを、また j = 0 と j = 1 はそれぞれ腕立て伏せをするか否かを表しており、したがって、"P ≻ Q" は、"学校、学年、性別がその選手（x）のコーチと腕立て伏せを行うか否かを決定する"ことを意味している。このように、P, Q は、単一の述語でなくともよく、一般には述語スキーマ（predicate schema）と呼ばれる。

さて、上述の EBL の過程において用いられた領域固有の知識（$）の替わりに

$$P'(x) \succ P''(x),\ \text{i.e.},\ (\forall x)(P'(x) \rightarrow P''(x)) \vee (\forall x)(P'(x) \rightarrow \neg P''(x))$$

を仮定しても EBL と同様の結果を得る。すなわち、2）および（#）より、P'(A) の成立が、また 1）および 2）より、P"(A) の成立が示され、上記決定関係は、($) を導くことがわかる。ここで本質的なことは、training example の持つ情報 2）が活用され、domain theory の欠落した部分を補っていることである。

3．4．2．4　知識獲得とＤＢＲ

一般に知識獲得の場面においては、強力な一般性を持った含意則（strong implicational rule）の直接的な入手は困難とされている。しかし、しばしば注目する事柄を決定している要因の集合を特定することが可能な場合がある。DBR を用いれば、このような決定関係に関する知識から一般的な含意則を獲得することができる。

この種の知識が想定される場面として、

物理現象過程：　境界条件 ≻ 定常状態

法的決定過程:　判例記述 ≻ 法的判断
アルゴリズムによる処理過程:　入力 ≻ 出力
問題解決過程:　仕様 ≻ 解

等が考えられる。

DBR に基づく類推は、従来のものと較べて効率の面で優れている。これは、target と source の照合問題が解消されているためである。このことはまた、ルールベース・システムにおける照合の問題の解消にも活用可能であることを意味している。このとき、多数の含意則は一つの決定則と事例群として記述される。照合は関連した事例記述のみに留められる。

3．4．3　設計物に対する説明の生成と一般知識の獲得

設計物の働きを説明するのに二つのアプローチが考えられる。一つは、プラトン、ガリレイに起源を発するとされる”因果機械論的説明”であり、他方は、アリストテレスに始まるとされる”目的論的説明”である[3]。以下では、渡辺[2]により導入された設計物の説明モデルを拡張することによって、上記の両説明を統合した設計物モデルを考え、この上で生成される説明を活用することによって、設計に関する一般的な知識の獲得法を導入し、流速測定装置を例にとって説明を行う。また、本法と EBL との関連についても検討する。さらに、設計知識の獲得に対して、 DBR の活用について検討を加える。

3．4．3．1　設計物の階層的モデル

ここでは、設計物に対するモデルとして、図２に示すように、その目的（N）、機能（F）、構造・属性（S）、実体（R）の４つのレベルから、事物を表現する方法を導入する。このモデルにおいては、一つの設計物は、その最終目的（n）を達成するための機能群（f）およびそれを支える構造・属性群（s）、さらにはそれを実現する実体群（r）から構成されるものとして説明・理解される。

この説明のためには、それぞれのレベル N, F, S, R の間に、これらを関連付け、その根拠を与える一般的法則群（被覆法則 covering law[4]）の設定が必要となる（図２参照）。例えば、機能集合空間 F と構造・属性集合空間 S の間については、いわゆる物理法則群を設定することによって、ある機能 f の達成に対してある構造・属性群 s が必要とされる理由を示すことができる。また、目的集合空間 N と F の関連については、一つの目的 n に対してどのような機能群 f がその達成を可能にするのかに関する、いわゆる目的達成に関するプラン形知識が想定されることになる。さらに、S と実体集合空間 R の間には、S 上に設定されたものを実現化することに関する種々の知識が介在することになる。

これらを説明のタイプから見ると、N と F の間を継ぐ説明は、設計物の目的合理性に注目した説明（目的論的説明）であり、F と S を継ぐ説明は、因果合理性に注目した説明（因果機械論的説明）であり、また S と R を継ぐものは、設計

物の経済合理性に注目した説明と言うことができる。

３．４．３．２　流速測定装置に対する一般的説明の生成

図３は、高温流体である溶融鉛 M の流速測定装置を示すものである[7]。図中、(b) が流速測定に直接関与する部分である。本装置は校正機能も持っており、これは、(a) に示す垂直回転軸の回転によって与えられた既知の周速と比較することによって行われる。

図４は、(b) における流速測定のプロセスを、物理法則集合空間（M）を媒介として、機能集合空間（F）と構造・属性集合空間（S）の関連の上で説明したものである。達成されるべき機能 measure(objectM, velocity) が、種々の物理法則を介して他の機能に置き換えられ、最終的に measure(object5, change_in_resistance) として実行される。これら物理法則の成立を支えているのが、空間 S 上の種々の関係あるいは属性群である。

これら物理法則は測定プロセスの因果機械論的な説明の生成に用いられるものであり、図５に示す基本的なタイプが考えられる。図中、⇒ が因果関係を表している。法則の構成要素すなわち、対象物、物理量、物理性質は、因果関係の成立する範囲で出来るだけ一般的な形で与えるものとする。このとき、結果側の物理量表現を除いて、他の全ての構成要素について、それらを特殊化しても法則の成立することがわかる。この特性を利用することによって、上記の測定プロセスは、図６に示すような法則群の連結した因果連鎖として説明される。すなわち、力の伝達法則 M1 における、液体を溶融鉛に特殊化し、固体を弾性体に特殊化する。また、ひずみ伝達効果 M3 において、固体１を弾性体に、固体２をひずみゲージに特殊化する。これによって、M1 から M4 までの法則が連結される。

(b) に示された構造中でのこのような法則群と上記特殊化の成立は、関連した対象物の汎化階層関係に基づいて示すことができる。図７に示すように、M2 の成立の背後には、石英ガラス棒である object4 が弾性体であることが、さらに、M3 と M4 の連結には、ひずみゲージが固体であることが前提となっている。

測定プロセスの説明である因果連鎖の生成の際の特殊化は、必要最小限に留められており、したがって、この説明は、流速測定の一つの一般的な方法（道筋）を与えていることになる。

３．４．３．３　ＥＢＬとの関係およびその他の被覆法則

前節での一般的な測定方法知識の導出法は、EBL 法に基づくものではない。すなわち、物理法則の成立に注目し、さらにこれら法則を因果的なものに限定し、これらをできるだけ一般的な形で与え、その特殊化を最小限に留めていることによって、EBL における goal regression を用いることなく一般的な知識が、しかも測定プロセスという、目標達成の方法に関する知識の形で、導かれている。

EBL との関連で述べると、domain theory としては、空間 M および空間 S 上の知識、すなわち、図６および図７に示された知識が対応するものと思われる。また operationality criterion としては、物理量の測定が、（空間 S 上の）構

造・属性的記述のみによって（十分条件として）与えられることであると考えられる。

以上が、流速測定プロセスの説明であるが、これ以外に”高温”流体に関する測定でああることが、測定装置の構造・属性に反映されている。すなわち、弾性体としては特殊なものである石英ガラスが object4 として採用されているのは、その耐熱性によるものと考えられる。また耐熱性および局所流速の測定を可能にするために、object3（石英円筒ガラス）による object4 の被覆が行われている。このような説明の生成に必要となる一般法則（被覆法則）は、常識あるいは経験的知識の範囲に属するものが多く、一般に domain theory として設定することが困難であると考えられる。常識の定式化あるいは類推に関する研究の進展が望まれる。

設計プロセス全体をガイドする一般的な法則として、Suh[8] らは、公理論的な体系を提唱している。これは「機能独立化」と「情報量最小化」の二つの公理から構成されており、前者は設計物を複雑化する方向へ、後者は単純化する方向へと導く（図８参照）。このような法則も説明のための domain theory の一部を構成するものと考えられる。

３．４．３．４　ＤＢＲと設計知識獲得

さらに、構造・属性空間 S と実体空間 R の間のように極めて複雑かつ大量の知識によって媒介されており、一般的な法則群の設定が困難かつ経験的知識の活用が重視される部分については、先に述べた weak な domain theory の下での一般的知識の獲得を与える、DBR（determination-based reasoning）[5,6]の活用が考えられる。

これは、3.4.2.4 節に述べたように、設計問題における問題解決過程の入力である仕様と出力である解（設計物あるいはその一部の構造）の間には、決定関係が想定され、したがって DBR に基づいた一般的設計知識の獲得が期待されることによる。しかし、問題解決過程内の決定（選択）が必然的なものでなければ、獲得される知識は一般性を欠くものとなる。したがって、設計過程の各フェーズにおいてどのような選択が行われたかを明示した上で、仕様と解（設計された実体）の間に必然的な関係の成立が保証された場合について DBR を用いれば、設計に有効な一般性を持った知識の獲得が可能となる。

このような、設計における選択（機能設定、構造選定）を明示するものとして今まで述べてきた設計物モデルを採用するとき、DBR による知識獲得の対象として、空間 S と R の間の関係知識すなわち、構造・属性が与えられた下での対応する実体の選定に関する知識が考えられる。勿論、この選定過程において、多くの選択が考えられ、それが必然的なものでない場合には、これを明示したモデルの上で DBR を適用する必要がある。

このように考えると、DBR による知識獲得の際にも、設計過程の詳細な分析が必要であり、したがって、DBR と EBL は関連して活用される必要があるように思われる。

3．4．4　むすび

設計問題に関して、一事例に対する説明から、問題解決知識（設計知識）の獲得について考えるとき、3.4.3 節に示したアプローチ以外にも種々のものが考えられる。このためには、3.4.1 節に紹介した設計過程（問題解決過程）そのもののモデル化が必要となる。3.4.3 節に導入したモデルは、この設計過程モデルの一部を構成するに過ぎない。

とくに、最終的な設計解の記述だけでは、設計過程中に考慮された多くの代替案との比較が不可能であり、3.4.1.4 節に述べた設計解の根拠づけの内で「正当性」のみについての説明が得られるに過ぎない。「適切性」、すなわち何故に、（他の代替案との比較の上で）本解が選択されたかの説明を与えることができない。

例えば、例示の高温流体流速測定装置の場合、測定に関する一般的なプラン、すなわち、測定すべき物理量の「検出」（変換）、検出された物理量の「伝達」、伝達された物理量の（電流などの物理量への）変換による「計量」、の３段階のプロセスを想定するとき、図３の装置の特色は、伝達される物理量が”ひずみ”という、高温な環境に対してもロバストなものが採用されているところにある。このように設計における本質的な部分が、他の代替案との比較、あるいはより一般的なプラン形知識を活用することによって、初めて明らかになる場合がある。

本 3.4 節では、合成型問題を考える上で極めて示唆に富むと思われる Mostow の paper、知識獲得の見地から今後重要と思われる Russell の DBR を紹介するとともに、これらに関連しながら、一つの設計解からの設計知識の獲得法について（中間的な）検討結果を報告した。

参考文献

1. Mostow, J.: Toward Better Models of Design Process, The AI MAGAZINE, pp.44-57, Spring, 1985
2. 渡辺 大助：ＶＥ理論の研究－設計学的アプローチ，第１７回ＶＥ研究論文集，pp.117-124
3. von Wright, G. H.: Explanation and Understanding, Cornell University Press, Ithaca, New York, 1971
4. Hempel, G. H.: Aspects of Scientific Explanation and other Essays in the Philosophy of Science, The Free Press, New York, 1965
5. Russell, S. J.: Analogy and Single-Instance Generalization, International Workshop on Machine Learning, pp.390-397, 1987
6. Davies, T. R. and Russell, S. J.: A Logical Approach to Resoning by

Analogy, IJCAI-87, pp.264-270, 1987
7. 岩崎 幸雄: 高温流体の流速測定装置, 特許公報 昭 60-10580, 1985
8. Suh, N. P., Bell, A. C. and Gossard, D. C.: On an Axiomatic Approach to Manufacturing and Manufacturing System, Trans. ASME, J. Engg. Ind., Vol.100, No.2, pp.127-130, 1978

Figure Captions

図１　述語 P と Q の間の関連性

図２　設計物に対する階層的説明モデル

図３　高温流体の流速測定装置「可とう棒」

図４　可とう棒における測定機能達成の説明

図５　物理因果法則の一般形

図６　可とう棒における測定の因果機械論的説明

図７　可とう棒構成要素に関する汎化階層（一部）

図８　Suh の設計に関する公理体系（一部）

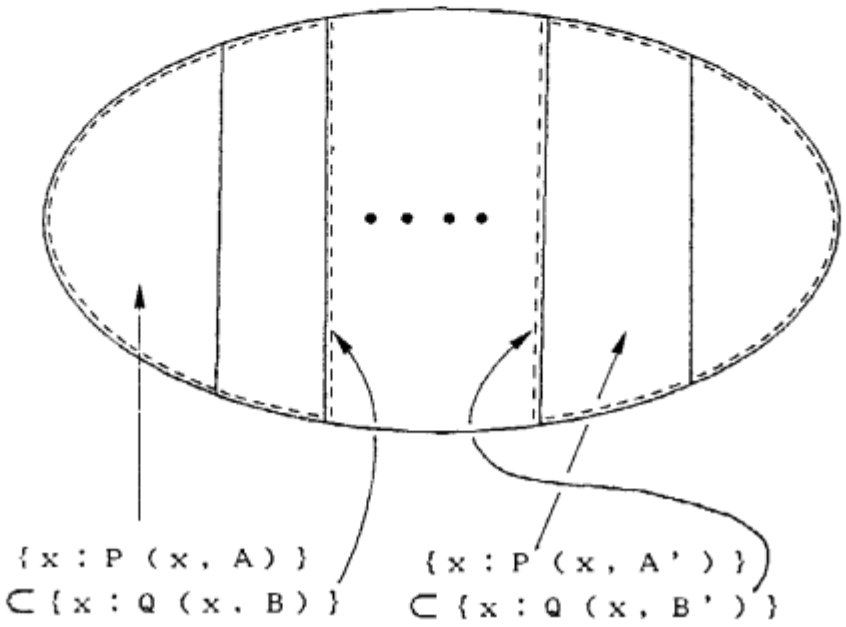


図1　述語 P と Q の間の関連性

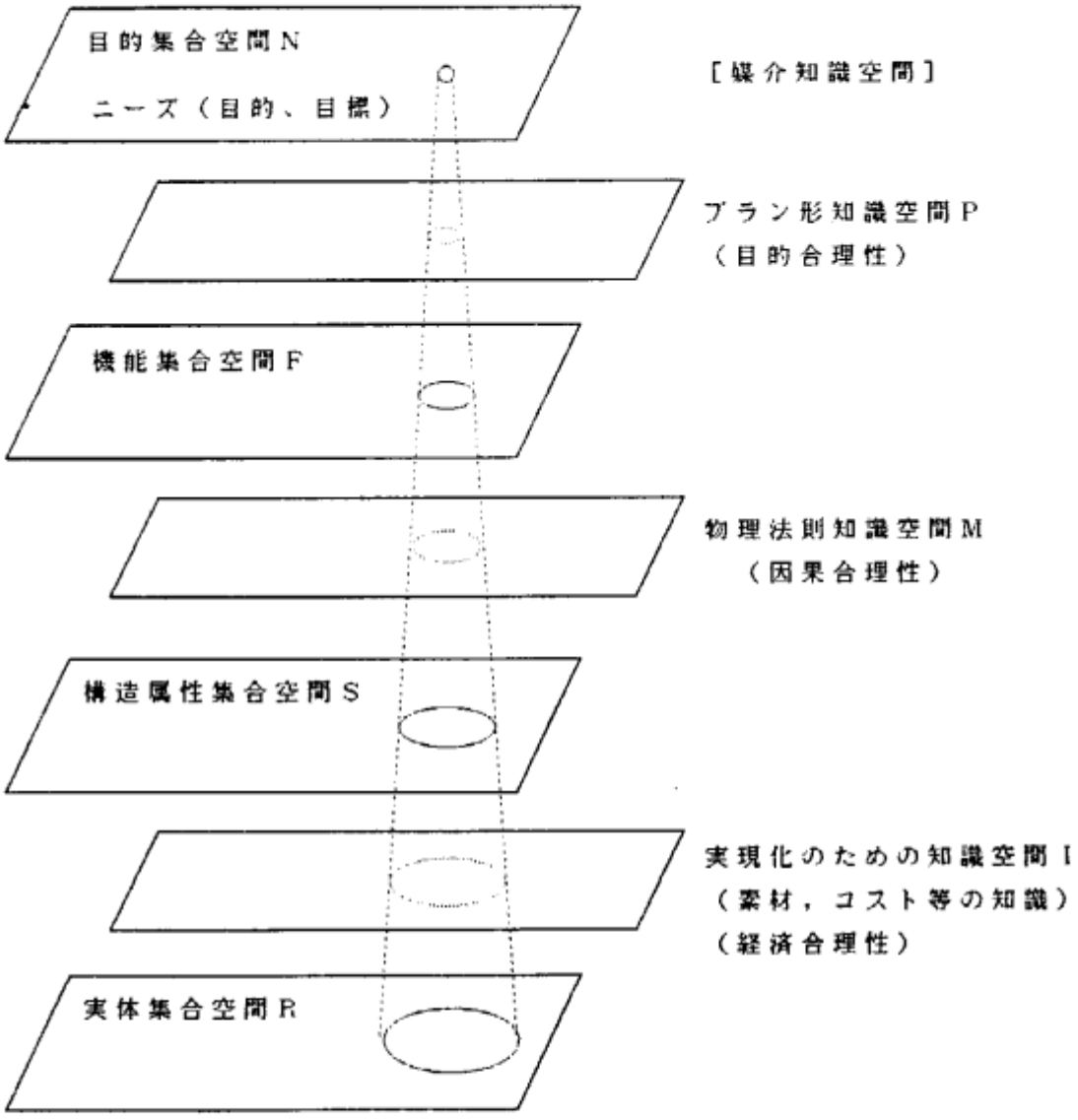


図２　設計物に対する階層的説明モデル

１：左右方向ガイド
２：支持腕
３：筒管（石英円筒ガラス）
４：可とう棒（石英ガラス棒）
５：ひずみゲージ
６：出力信号ケーブル
１０：測定部（１－６）
１１：槽
１２：本体
１３：垂直回転軸
１４：動力
１５：ベベルギア
１６：電気的出力接続手段
Ｍ：溶融鉛

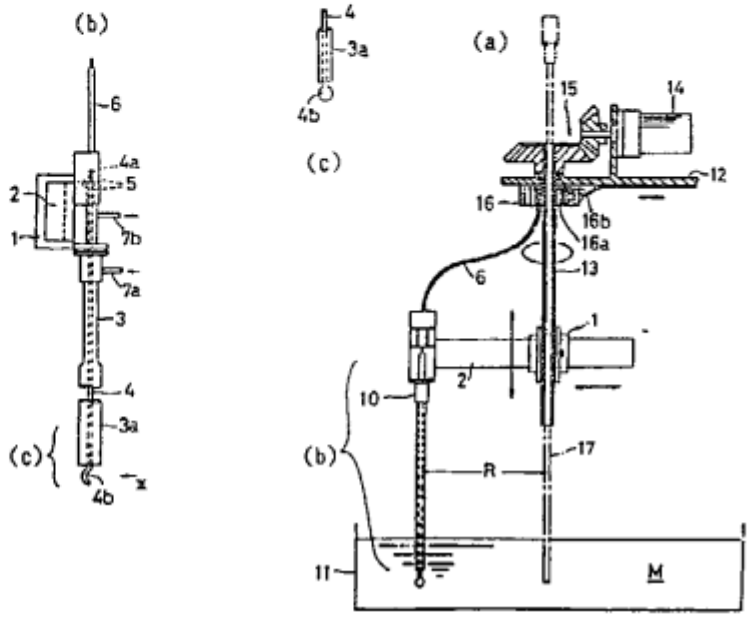
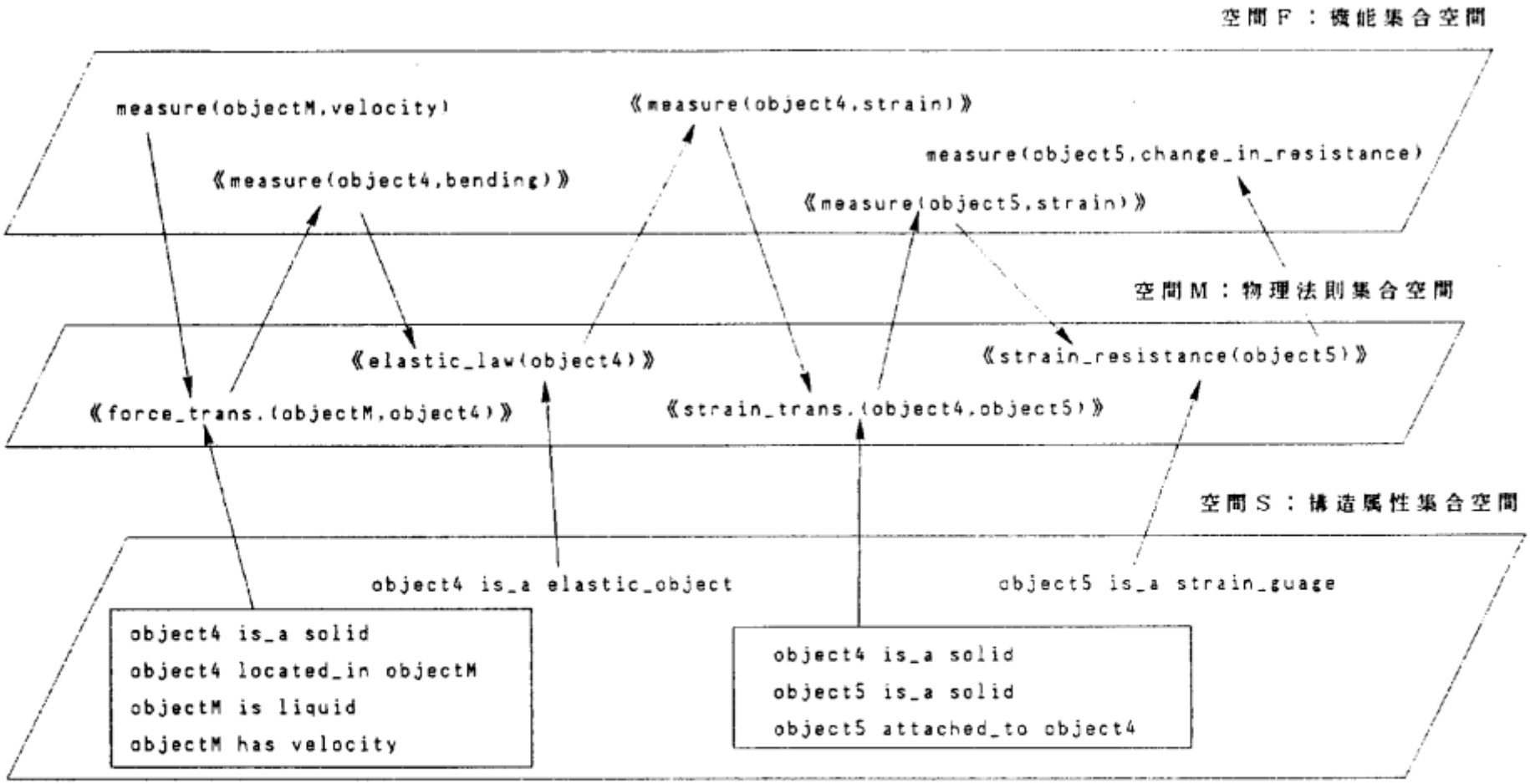


図３　高温流体の流速測定装置「可とう棒」



《》推論により挿入された節

図4　可とう棒における測定機能達成の説明

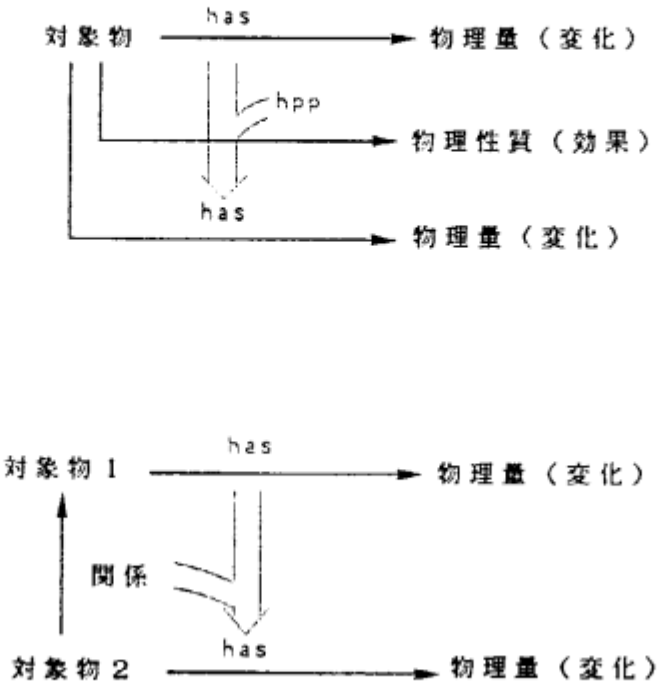


図5　物理因果法則の一般形

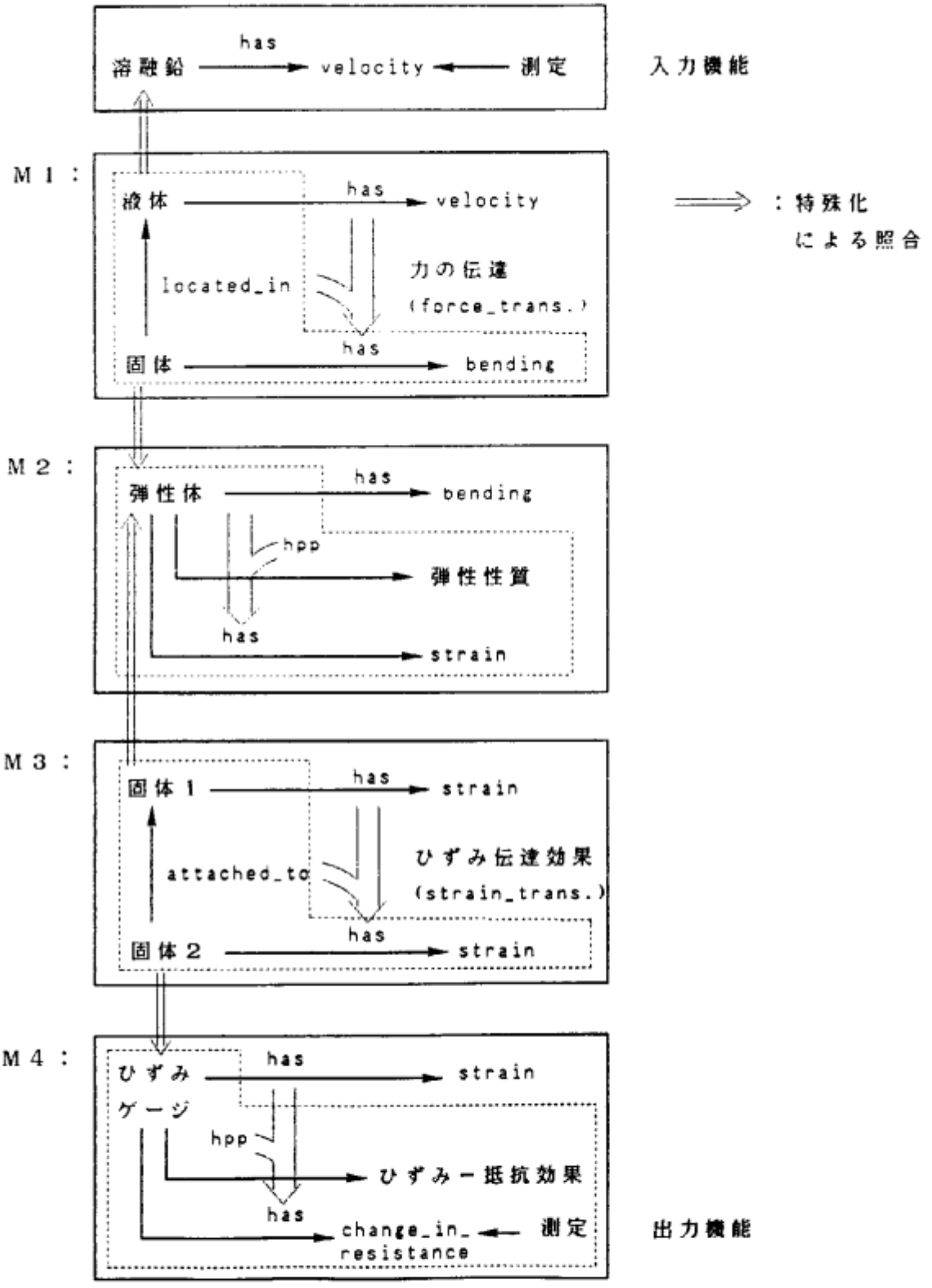
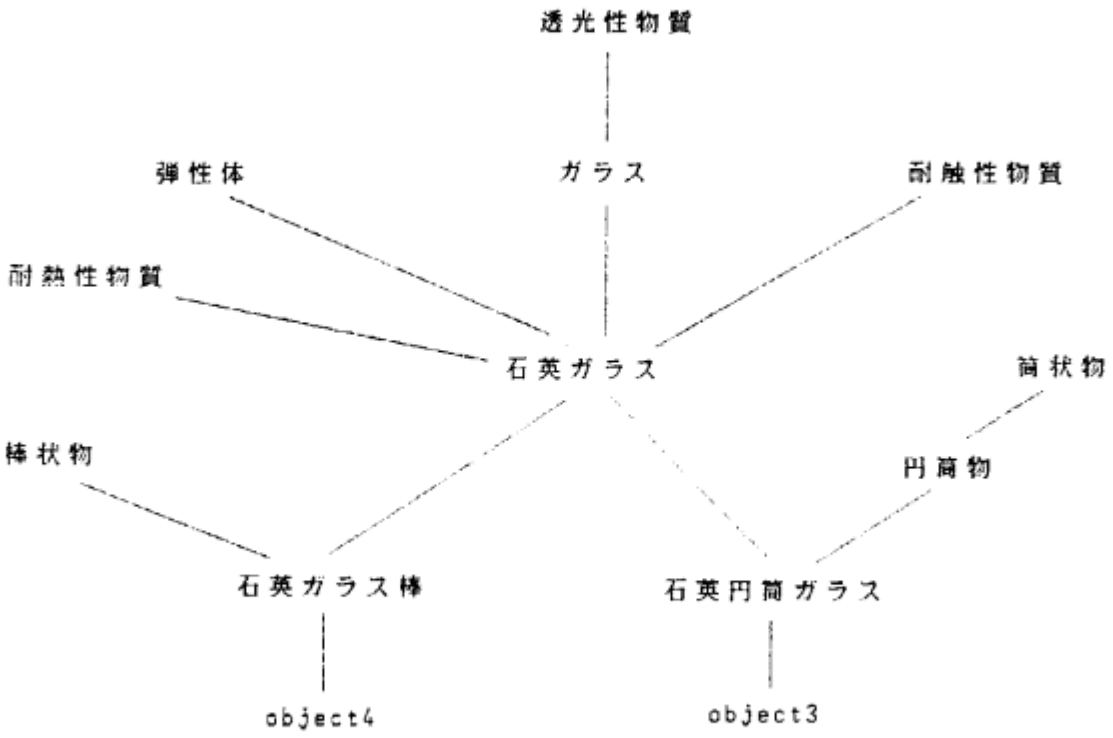


図６　可とう棒における測定の因果機械論的説明



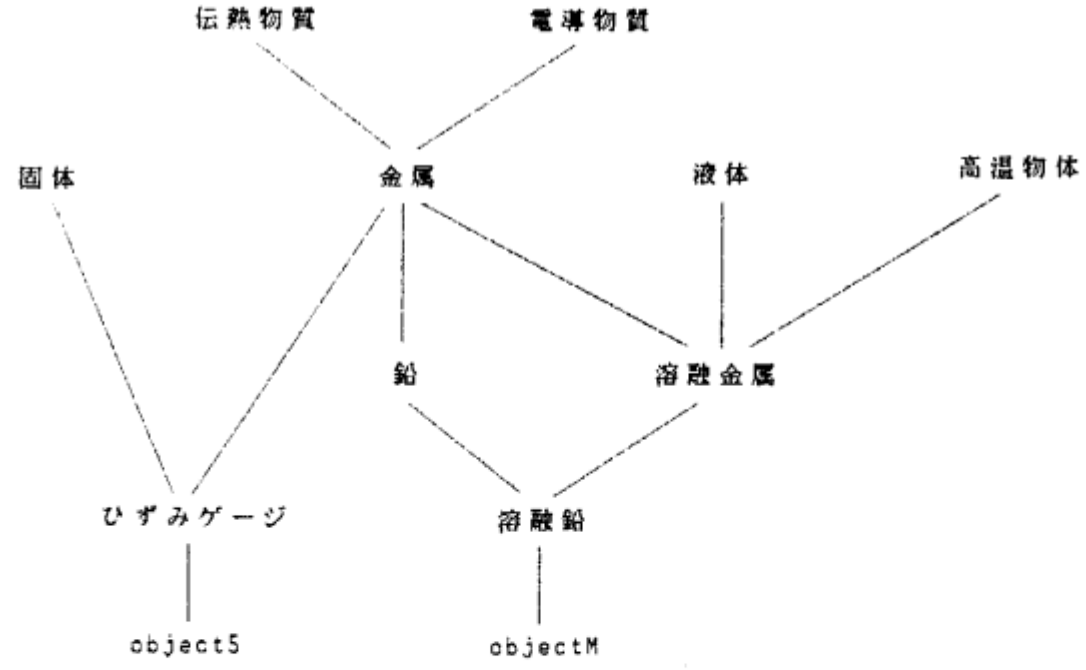


図7　可とう棒構成要素に関する汎化階層（一部）

Axiom 1: In good designs the independence of functional requirements is maintained.

Axiom 2: Designs are optimized by minimizing information content.

Corollary 1: Decouple or separate parts of aspects of a solution if functional requirements are coupled or become interdependent in the designs or processes proposed.

Corollary 2: Conserve materials and energy.

Corollary 3: Minimize the number of functional requirements and constraints.

Corollary 4: Integrate functional requirements in a single part or solution if they can be independently satisfied in the proposed solution.

Corollary 5: Use standardized or interchangeable parts whenever possible.

Corollary 6: A part should be a continuum if energy conduction is important.

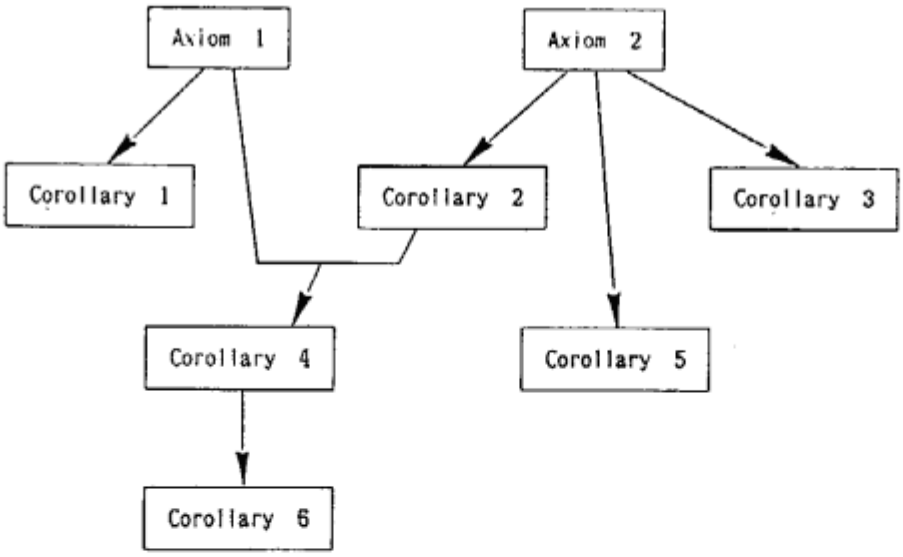


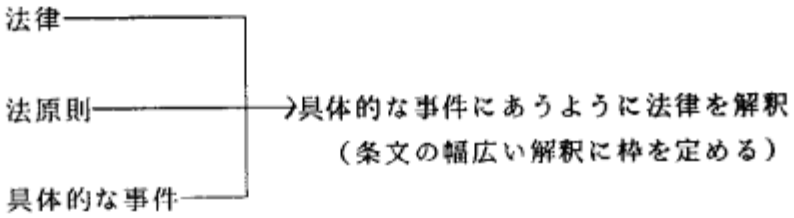
図８　Suh の設計に関する公理体系（一部）

３．５　判例とＥＢＬ

ＥＢＬ(Explanation Based Learning)の考え方が判例についての考え方と類似する点があるので、判例のコンピュータ処理にＥＢＬが使えないか考察する。

判例とは

「判例」とは１つの個別の事件を解決するためになされた法律判断である。判例には法律の解釈が含まれ、他の事件にも適用される可能性がある。裁判によって事件を解決する場合、①事実認定、②法律適用の２段階でなされる。「事実認定」は証人、文書、鑑定などの証拠を提出しあうことによって、事実の有無を裁判官に判断させることである。裁判官に採用された証拠のみが法律的に意味を持つ。「法律適用」とは、認識された事実に基づいて実際に法を適用することである。この場合、法律概念の解釈に幅があるため、法目的や社会通念や政治的判断や他の判例などを考慮しながら（これらを法原則と呼ぶことは2.1を参照のこと）具体的な事例にあうように法を解釈し、適用していくのである。法原則は具体的な解釈ルールや抽象的な法解釈の指針などからなっており、しかも矛盾する法原則があるときに個々の法原則の重要性を考慮して解釈を行わなければならないので、法解釈のプロセスはかなり複雑な仕組になっていることが予想される。



例えば、民法709条［故意又ハ過失ニ因リテ他人ノ権利ヲ侵害シタル者ハ之ニ因リテ生シタル損害ヲ賠償スル責ニ任ス］を考える。この規定の損害賠償ができるための要件は、過去の判例から

１．違法行為（権利侵害）があったこと
２．損害が発生したこと
３．行為と損害に因果関係があること
４．加害者に故意、過失があったこと
５．加害者に責任能力があったこと

と整理される。事件が生じたときはこの要件の１つ１つを立証しなければならず、１つでも証明できないときは請求できないのである。要件部分はこのように整理されても、どういう条件があれば故意があったといえるのか、損害の発生をどのように証明するのかなどは法律には書かれていないので、個々の判断には別の解釈／判断のためのルールが必要である。

さて、裁判は判決・決定・命令などの告知により成立・発効する。判決は主文と理由付けからなり、「理由付け」には、採用した証拠、法律解釈について詳細に述べられている。民事事件で90万円以上のの事件の場合、判決に不服がある場合には高裁に控訴し、それでも不服のときは最高裁に上告できる。ただし、事実認定についての不服は高裁までであり、最高裁には法律適用の不服のみ申立てできる。

裁判における法律解釈は、個々の裁判官にまかされており、１つの判決の中でなされた法律解釈は、他の裁判を拘束しないのが建前である。しかし、上級審での法律解釈は下級審の裁判官は心理的に拘束する。上級審と異なる法律解釈をすると上級審で覆る可能性が高いからである。また、最高裁においては過去の判例を覆す時には小法廷(3名の裁判官による審理）ではなく大法廷(15名の裁判官全員の審理）で審理しなければならない。これは最高裁の過去の判例が他の事件を事実上拘束していることの表われといえる。従って、実際には、過去の法律解釈は１つの規範として他の事件にも適用されると考えられる。（判例の拘束力が心理的なものにとどまるのか否かについては法律上の問題点であるが、ここでは立ち入らない。）

例として昨年になされた水害訴訟の判決を考える。主文としては住民敗訴であり、その理由として［行政の責任を問うには非常に厳格な条件が必要である］ことを述べている。この理由付けにより、今後の水害訴訟で住民が勝訴する見込はまずないといわれている。これは、行政に過失を認めるための条件が判例として形成され、以後、この基準が１つのルールとしての効力を持ちえるからである。逆の例として、持分の２分の１以下の共有者の共有林の分割請求を禁じる森林法186条が違憲であるとする昨年の最高裁の判決は、財産権の保証よりも公共の福祉を優先する合理的理由がないとする判断であり、最高裁が財産権保証の重要性を確認した例であるとされる。この判決により財産権よりも公共の福祉を優先する法律については厳しい合理性が要求され、それを持たない法律については違憲判決がなされる可能性が少し高くなったと言える。他の例として、議員定数の訴訟でも［定数格差が3.0倍以下なら合憲 ］というのは判例から生まれたルールであって、必ずしも法律の裏付けがあるものではない。（このような判例によるルールの生成は一種の立法行為であって、三権分立に反するという意見もあるが、ここでは立ち入らない。）

判例と学習

上記のように判例によって法律の解釈ルールが作られ、条文と一体となって以後の事件に適用されていく。従って、判例によって法律体系がより厳密なものになっていくのであり、判例の獲得は法律知識の学習と考えることができる。ただし、判例は特定の事件を解決するための判断であり、そのままで他の事件に適用できる形になっているとは限らない。なんらかの一般化が必要なことがある。

英国や米国のように判例が法を構成する国においては、判例の意味が特に大きい。判例

の中からレイシオ・デシデンダイ(ratio decidendi)（判決の基礎となった原理）とオビタ・ディクタム(obiiter dictum)（判決に不可欠ではない裁判官の意見）を区別する作業が法律家の重要な役割となっている。レイシオ・デシデンダイは拘束力を持つが、オビタ・ディクタムは拘束力を持たない。米国などでは判例が膨大なため、レイシオ・デシデンダイの抽出と統合が困難な状況となっているので、コンピュータによる判例処理（学習）ができれば非常に利用価値がある。

類推(analogy)による判例からの推論技法はすでに英米などで研究がなされている。その１つは、判例から一般ルールを抽出し、類似の事件に適用するものであって次の３段階で推論を行うものである。

判例の中で事件Ｙに似ているケースＸを発見する。

判例Ｘのためのルールを作成する。

このルールをＹに適用する。

この方法ではＸでのルールがそのままＹに適用できる（一般的なルールが作れる）ことを前提としている点で無理があるが、現実で法律分野で行われている判例の利用に近いものである。この技法を後のＥＢＬのところで再び考察する。

これとは別の方法として、判例から一般ルールを生成することなく、直接、事件の類似性だけから結論を推定する試みもある。

ＡとＢとＣを満たす事件Ｘでは原告が勝訴した。

ＡとＢとＤを満たす事件Ｙでは原告が敗訴した。

今の事件ＺではＡとＢとＣとＤが満たされているとすると、結論はＣとＤの類似度の重要性による。

というようなものである。この方法は事件のパターン化によって問題解決をしていくものである。事例が集れば簡単なエキスパートシステムが作成できる可能性があるが、すべてのケースについて網羅的に判例が集ることはなく、また、特徴の重み付けが自動的にできるか疑問である。

判例とＥＢＬ

さて、判例の問題とＥＢＬの類似点を考えてみる。ＥＢＬでは、①特定の問題をdomain theoryを用いて証明する、②証明を広いクラスに利用できるように一般化する、③一般化された証明から新しいtheoryを引出す、の３段階からなる。その際、operationalでない初期記述からoperationalな記述が得られることになる。

判例では、①特定の事件を法律と法原則を用いて解決する（法律を法原則を用いて動的に解釈し、事件に適用する）、②判決の中から判断の基礎となった法律の解釈を一般化し、新しい解釈ルールまたは法原則を抽出する、③新しい法律解釈のルールの追加、または、法原則による従来の法律解釈の見直しを行う、の３段階がＥＢＬの各段階に相当する。ＥＢＬの「domain theory」は法律分野では「条文そのものと条文の解釈と法原則」に相当

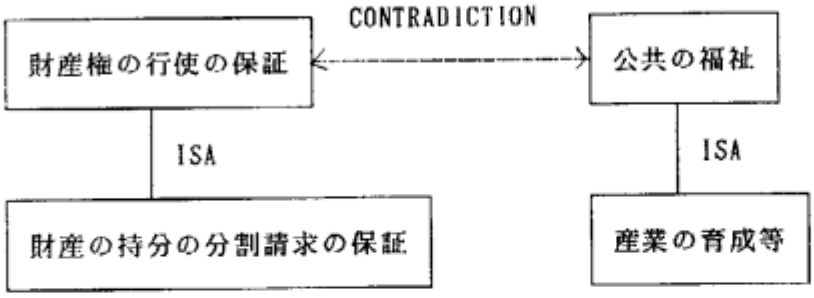
し、ＥＢＬの第２段階の「証明」は判例では「具体的な問題を解決するための法律の解釈プロセス」に相当し、「一般化された証明」とは「解釈プロセスの抽象化」に相当し、「新しいtheory」とは「抽出された新しい法原則（解釈ルール）」に相当するものと考えられる。法律概念は抽象的で一般の世界からはわかりにくい表現になっている。しかし、得られた法律解釈によって法律概念が厳格にかつ具体的に定義されることになる。法原則のうち具体的なレベルのものは、われわれの日常の生活で生じる事実（operationalな記述がなされている）と法律用語（operationalでない記述）の橋渡しをしていることになる。

森林法186条の例をもとにＥＢＬを考える。186条では、財産権の行使は自由であるべきだとする法原則と、森林の処分を自由にすると森林が細分化されて公共の福祉に反するとする法原則の対立がある。最高裁は財産権の制限を①保安や風紀の弊害防止のため、と②産業の育成や弱者の保護のために限って認めているが、そのうち、前者を厳しく判断し、後者をゆるやかに判断してきた。森林法186条は後者に属する問題であり、従来の経緯からは合憲とされる可能性が高いとされていた。しかし、森林法186条には不合理な例外規定（EX.過半数が認めれば分割できる、単独所有者は自分の森林を細分化して譲渡できる、等）があり、最高裁はこの点を指摘して違憲判決をした。この結果、不合理な規定を含むような財産権の制限立法は違憲とする立場が確立されたことになり、これが新しい法原則となって、他の法律の違憲判断に影響を及ぼすことになる。

この例をＥＢＬとして形式化できるか否かを検討する。

まず、この事件に関する法原則について考える。財産の持行使の保証と産業の育成等の目的は下図のように矛盾する場合がある。この場合、産業の育成等を優先させるというのがdomain theoryである。

財産権の保証　∧　保安等　→　違憲
財産権の保証　∧　産業の育成等　→　合憲



森林の持分の分割請求は、財産の持分の分割請求という点からは当然認められるべきであるが、森林が細分化されると産業の育成に反するとして禁じられていたのである。しかし、最高裁は森林法186条を違憲としたことにより

森林の分割請求の禁止　∧　産業の育成等　∧　不合理な例外→　違憲

という判断を示した。この判断の理由付けのため、法原則についての詳しい検討がなされているがこれが証明に相当する。この証明はそのまま一般的な事件にも適用できるので結論を一般化すると、

財産権行使の制限　∧　公共の福祉　∧　制限の不合理性　→　違憲

すなわち「財産権の行使の制限には十分な合理性がなければならない」とする新しいルールが生成され、domain theoryに追加されたことになる。

まとめ

ＥＢＬと判例についての類似点について考察した。裁判においては条文や法原則から事件を演繹的に解決されることも多いが、初めに判決をある程度決めて後からその正当性を証明するための根拠を決定することも少なくないようである。後者の場合にＥＢＬとの強い類似性があるように思われる。ここでは森林法の例をあげたが、重要な証明プロセスの考察は不十分であり、動的な法律解釈のためのdeep model（社会通念や立法目的や他の法律とのバランスなどを考察した解釈モデル）がなければこれ以上の検討は困難である。また、森林法の考察は１つの例にすぎず、実際には判例の利用は様々なレベルがある。

本稿を書くにあたり加賀山茂先生（大阪大学法学部）と小松弘先生（弁護士）に有益なコメントをいたいたことを感謝いたします。法律上の大きな誤りは訂正したものの、追加した記事についてはまだ誤りがある可能性がある。