08-R 008

大規模知識ベースに関する調査研究

― オントロジー工学に関する調査研究 ―

報　告　書

平成 9 年 3 月

JIPDEC

財団法人　日本情報処理開発協会

KEIRIN 00 この事業は、競輪の補助金を受けて実施したものです。

ごあいさつ

近年、情報処理システムは、知識処理技術などの発展を背景として、ますます高度化の方向に進んでおります。これとともに大量の知識情報を体系的に蓄積し、共有する大規模知識ベース（VLKB：Very Large-Scale Knowledge Bases）は、より高度な知識処理を実現する情報基盤として期待されております。

しかしながら、大規模知識ベースは、様々な分野・領域（学術、技術、産業など）の知識情報が多様なメディア（文字、言語、画像・映像、音声・音響など）により表現されることから、各分野・領域、各メディアで蓄積・表現された知識は共通の概念体系のもとに関連づけられることが求められます。この課題に対し、「オントロジー（概念の体系）」が貢献するものとして期待されています。

このような状況のもと、当協会では、平成８年度「大規模知識ベースに関する調査研究」として、オントロジーを取り上げ、その考え方、定義等の整理および工学的な構築、活用方法を探るため、本年度より２年計画で「オントロジー工学に関する調査研究」を実施することといたしました。

実施にあたりましては、大学、企業などの研究者、専門家からなる「オントロジー工学調査委員会（委員長　溝口理一郎　大阪大学産業科学研究所教授）」を設置して、委員各位による調査報告ならびに審議・検討を行うとともに、米国に調査員を派遣し、オントロジー構築における主要テーマの１つである言語情報からの概念抽出などに関する研究開発状況を調査しました。

本報告書は、平成８年度に実施した調査研究（中間）成果を取りまとめたものです。構成は３部から成り、第１部では、オントロジー工学の確立に向け、様々な視点から現状、課題について、第２部では、関連する技術等に関し国内の研究開発状況（論文）について、第３部では、海外動向として、米国調査の概要のほか、標準化の動向について報告しています。

本書が、今後の大規模知識ベース関連のプロジェクトの推進ひいては高度情報化の発展に寄与することを念願する次第です。

最後に、本調査研究の実施にあたり、ご指導ご協力いただきました委員各位ならびに関係各位に深甚なる謝意を申し上げます。

平成９年３月

財団法人　日本情報処理開発協会

委員会名簿

（平成９年１月現在）

オントロジー工学調査委員会（敬称略、50音順）

委　員　長：	溝口理一郎	大阪大学 産業科学研究所 知能システム科学研究部門　教授
委　　員：	伊藤　公俊	埼玉大学 工学部 機械工学科　助教授
	小山　照夫	文部省 学術情報センター 研究開発部　教授
	鶴丸　弘昭	長崎大学 工学部 電気情報工学科　助教授
	寺野　隆雄	筑波大学 大学院 経営・政策科学研究科 経営システム科学専攻　教授
	外池　俊幸	名古屋大学 言語文化部　助教授
	冨山　哲男	東京大学 大学院 工学系研究科 精密機械工学専攻　助教授
	西田　豊明	奈良先端科学技術大学院大学 情報科学研究科　教授
	堀　雅洋	日本アイ・ビー・エム (株) 東京基礎研究所　副主任研究員
	増永　良文	図書館情報大学 図書館情報学部 図書館情報学科　教授
	山口　高平	静岡大学 情報学部 知能情報工学科　助教授
オブザーバ：	武田　英明	奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻　助教授
	藤本　康二	通商産業省 機械情報産業局 電子政策課　課長補佐
	大澤　活司	通商産業省 機械情報産業局 電子政策課　国際係長
	市川　隆	(財)日本情報処理開発協会　常務理事
事　務　局：	片岡　幸一	(財)日本情報処理開発協会 技術企画部 技術課長

以　上

目　次

第2部　研究開発動向

第３部　海外動向

第１部　オントロジー工学確立に向けて

１．　オントロジー工学の必要性

第１部　オントロジー工学確立に向けて

１．オントロジー工学の必要性

人工知能研究のなかには、論理や知識表現などを扱う「形式指向」の研究と知識の内容を研究の対象とする「内容指向」の研究がある。人工知能研究はこれまで「形式指向」の研究を中心に行なわれてきたが、近年、内容指向の研究の重要性が高まっている。それは、現実に存在する知識処理の課題の多く、例えば、知識の再利用、複数のエージェントが協調するために必要な通信、理解に基づくメディア統合、大規模知識ベース／常識ベースの開発、知識の標準化による知識共有などの問題の解決には、推論などの形式的な操作の高度化だけではなく、様々な形態で存在する知識の「内容」を扱う基礎研究と高度技術が不可欠であることが明らかになってきたからである。

形式化の一つである論理は形式的に健全な議論を規定するための強力な道具を提供する。そこでは、形式的に正しい思考を規定することができ、人間の思考の抽象的な限界を議論することも可能である。しかし、健全な推論によって、与えられた問題を解決するためにはどのような知識を用意すればよいか、そもそも知識とは何か、我々は認識をどのように構成し、知識の具体的内容はどういう構造と性質を持っているのかといった疑問に「形式」は答えてくれない。

その意味で、「知能」を形式的に捉える、あるいはその根源を問題解決機能（推論、論理など）に求めることの代わりに「知識」の重要性を力説したFeigenbaumの主張は正しかった。このことはエキスパートシステムの成功が証明している。しかし、それは更に深化しなければならない。即ち、Feigenbaumの時代の知識に対する理解は、「専門家の専門知識をルールで表現する」ということが本質であった。現在はそこから遥かに進歩している。知識は様々な形態を取って存在することから分かるように、知識源は実に多様である。専門家、データ集合（データベース）、ドキュメント、画像の中にも存在する。そして、知識を扱う主体としてコンピュータ（ソフトウェアエージェント）と人間との２種類のエージェントが存在する。即ち、知識はこれら両方の

(注) 筆者等が使う「エージェント」という単語は人間とコンピュータとを一般化した主体的な動作主という意味である。

エージェントのために存在するべきである。知識は複数のエージェントの間で共有され、様々な目的のために再利用されなければならない。そしてコンピュータによる知識処理は、このような多様な知識源に潜在する知識を両エージェントにとって操作可能な形に、抽出、変形、組織化する技術、即ち、知識メディア技術に進化しなければならない。その基礎を与える学問がオントロジー工学である。

「内容指向」の研究の重要性はある程度認識されつつある。それにも関わらず、これまでの人工知能研究の多くは形式指向の研究が重要視され、内容指向の研究は軽んじられてきた。これにはいくつかの理由があるが、最も重要なことは内容指向の研究が以下のような問題を抱えていることであろう。

1）個別の状況に依存したアドホックな議論になりがちであること。従って、

2）形式理論の様に積み重ねが効くような方法論がないこと。

3）しっかりした基盤研究、あるいは技術がないこと。

これらの問題点を解決することなしには、内容指向研究の発展は望めない。多くの場合、「開発」にとどまり、「研究」へと昇華することは難しい。この様な問題を解決するのがオントロジー工学である。オントロジー工学は、知識ベースの設計意図、核となる概念化、基本概念の意味の厳密な定義などを与えるだけでなく、実世界の情報モデル構築のために不可欠な、知識を「積み上げる」技術と理論を提供する。

オントロジーに関する研究はこの数年間で急速な広がりをもって行なわれている。しかし、未だにオントロジー自身に関する定義が定まっていないのが現状であり、まして外部にいる研究者には不透明な部分が多い。更に、我が国においてはオントロジー研究の重要性が必ずしも正しく理解されているとは言えない。オントロジーは哲学的な理論から現実の問題解決に直接関係する具体的な課題まで幅広い議論が必要であり、それだけに奥が深く、理解も混乱しがちである。しかし、工学としての知識処理の将来を考えたとき、「内容指向」研究を支える基礎としてオントロジー研究は今後ますます重要となる。

さて、「全てのソフトウェアにはオントロジーは存在しており、全ての経営者の頭の中には「企業オントロジー」はある。従って、オントロジーを議論する必要性が理解できない」という意見がある。一見、その意見は正しいようにも思われるが、それらは「潜在」しているのであって、決して明示的には記述されてはいない。オントロジーを明示的に記述する事によって得られるメリット、即ちオントロジー工学の意義を明確にする。

まず、オントロジー工学の究極の目的は、「実世界、厳密には情報科学が対象とし

得る全ての対象のモデル構築の基盤を与えること」にあることを宣言しておこう。そしてそれは共有／再利用が可能で、人間とコンピュータの双方に理解可能でなければならない。

オントロジーの役割を設計問題を例にとって考えてみる。設計における人工知能の研究で最近注目を集めている研究に、Design rationale（設計意図）の問題がある。設計の世界では、設計後しばらくして改良設計を行なうことや、他人が行なった設計を元にして新たな設計を行なう場合がしばしばある。そのような場合、ある部品が付加されている理由や特殊な構造に出くわしたとき、なぜそのようになっているか理由が分からない場合には、その既存の設計結果を再利用することができない。その「理由」はDesign rationale（以降、DRと呼ぶ）と呼ばれるが、通常の設計ではそのような情報が残されることは少ない。設計行為は、部品の選択、部品のパラメータの値の選択（決定）、部品間の位相的接続関係の設定など、実に様々な、そして数多くの「決定」の集積であり、その全てには設計者の意図と合理的な理由が存在する。その情報はある意味では設計図と同じくらい重要な意味を持っている。設計物とそれが持つ有用な情報の再利用には、設計意図の理解が不可欠だからである。

知識ベースのオントロジーは正しくDRに対応する働きをする。他人が構築した知識ベースを理解するには、前提とされている条件や環境、解くべき問題が要求する仮定などの暗黙的な情報、そしてそれらを反映した対象の世界の概念化に関わる根本的な情報を知ることが必要である。しかしながら、設計と同様、これまで実に多くのシステム（知識ベース）が構築されてきたにもかかわらず、その様な情報が明示的に示されることはほとんどなかったと言っても過言ではない。オントロジーはそのようなDR情報を提供し、知識ベースの構築を支えるバックボーンとしての役割を持っている。今後の知識処理は明示的に示されたオントロジーに基づいて行なわれなければならない。

このように、オントロジーはこれまで暗黙の世界に放置されてきた、前提、世界の成り立ち、設計の意図、などを明示的に扱うことによって、新しい知識処理の基盤を提供する。しかし、現状ではオントロジーを扱う技術は全く未成熟であると行っても過言ではない。オントロジー工学の確立は、今強く望まれている。

2．　オントロジー工学確立の視点

2. オントロジー工学確立の視点

2.1 知識表現とオントロジー工学

2.1.1 知識ベースシステムと知識表現

エキスパートシステムをはじめとする知識ベースシステムでは、人間の持つ知識を機械処理可能な形式で記述し、これを機械に利用させることにより、様々な問題解決を行うシステムを構築することが目的とされる。このようなことが可能となるためには、人間の知識を機械が処理可能な形で記述することが必要であり、人間の知識を機械処理可能な形で記述するための枠組みを、一般に知識表現と呼んでいる。この意味で知識ベースシステムにおいては、知識を記述するための枠組みとしての知識表現は基本的な問題であり、どのような知識表現の枠組みを採用するかで、機械に実現させることのできる能力が決まってしまうと言うこともできるのである。

エキスパートシステム実用化の期待が高まった1980年代前半には、知識表現の枠組みに関する広範な検討が行われ、様々な知識表現形式が提案された。それらの中にはかなりの独自性のあるものもあったが、1980年代中頃になると、一部の論理型の枠組みを除いては、フレーム（オブジェクト）とルールの組み合わせが主要な知識表現の枠組みとして広く認められるようになってきた。

この、フレームとルールの組み合わせは、様々な工夫をするならば、幅広い内容の知識を記述することが可能であることが実証されており、かつ、記述された知識を効率的に機械処理するという面できわめて優れた特徴を持つことが明らかとなっている。実際に、様々なシステム実装を通じて、確かにこの枠組みの下できわめて強力な問題解決システムが構築できることも証明されてきた。

しかしながら一方では、このようなレベルでの知識記述の枠組みが決まったとして、それだけで様々なシステムの実装が可能となるわけではないことも指摘されるようになってきている。オブジェクトやルールというレベルでの記述の枠組みが決まったとして、それではそのような枠組みを用いていかにして実際の人間の知識を記述できるかという問題は実は未解決のまま残されてきたと言うこともできるであろう。フレーム（オブジェクト）とルールという表現形式は、記述された知識を効率よく適用するための枠組みではあっても、人間、特に様々な分野の専門家の知識を直接に記述する

ための枠組みとしては不十分なものであり、実際に専門分野の知識をこの形式に従って記述するためには、人間が知識を与えられた枠組みに合うように変換してやる必要があるのである。しかし実際にシステムを実装する立場からすると、問題はむしろ、このような枠組みを用いていかにして実際のシステムが構築でき、その維持が可能となるかである。この点に関する限り、1980年代の知識表現の枠組みは、システム開発者に具体的なシステム実装に関する示唆を与える点がきわめて少なかったと言うことができる。

このことは、一つには特定のシステムに関して定義された知識が他のシステムを構築する上でほとんど利用できないという、知識再利用性の困難さとなって現れており、また、一つには既存のシステムを状況の変化に応じて適切にメンテナンスしていくことの困難さとなって現れている。知識表現の単なる形式としてのオブジェクトとルールという枠組みだけでは、記述された知識を効率よく処理する枠組みとはなり得ても、問題領域での知識をどのように記述すればよいかなど、システムを構築する上での指針を明らかにすることはできないことが明らかとなってきたのである。

それでは対象領域の知識を記述する上での指針とはどのようなものであり、知識をどのような形で機械処理可能な形で記述できるのであろうか。この問題に対する回答を与えるものとして、様々なオントロジーが検討されているといえる。

2.1.2 問題解決の枠組みとしてのタスクオントロジー

問題領域の知識記述の指針を与えるものとして重要なものに、タスクという考え方がある。特に適用分野を限定しない、問題の性質に依存するジェネリックタスクと呼ばれるものが存在するという考え方は、タスクオントロジーの提唱につながるものとして重要である。タスクオントロジーという形で、問題解決のための一般的な枠組みが整理されてきた結果、システムの持つ機能的な側面について言えば、システム実装のための指針がかなりの程度明らかにされてきたということができる。例えば医療診断では、観測された症候に照らし合わせて、症候の原因になる疾患、あるいは疾患の組み合わせを仮説として生成し、ついでそれらの仮説の中で明らかに矛盾を含むものを削除するとともに、残りの仮説の間に重要度に応じた順序づけをする。また、疾患仮説の重要度に応じた治療方針を決定し、治療を行った場合の結果を予測する。という一連のタスクが繰り返し行われていると考えることにより、システムの達成すべき機能的な側面ははるかに理解容易なものとなるとともに、分野を異にする様々な診断

問題との間の関係もうまく整理することが可能となる。

このようなタスクオントロジーの提唱は、確かに知識ベースシステム構築の方法論に大きなインパクトを与えるものであったと言える。しかしそれでは分野ごと、あるいは分野をまたがってのタスクオントロジーを定めることができればそれで全ての問題が解決できるかと言えば、実はそうではない。

タスクそのものは、実際に行われるべき処理の概要を示すものであるが、そのような処理が具体的にどのような形で実現されるかまでを決定するものではない。これはルールが、実際に何をするためのものかを明示的に示すものではない代わりに、どのような形で実行されるかはかえって明らかであることと比較して、明確な対照をなしていると言うことができる。タスクの実行は何らかの形で機械処理可能な手続きとして計算機上に実装されなければならないが、このためには、実はシステムにおいて現実の世界の情報がどのような形で記述されているかが問題となる。つまり、タスクそのものが現実世界に対応する何らかの情報に対する操作を規定するものである以上、タスクの実行のためには処理の対象となるデータの特性が知られていなければならないのである。このことは、基本的に同一のタスクであっても、現実の世界をどのようなデータ形式の下に捉えるかによって、タスク実行の方法も変わってくることを示している。例えば、医療診断においても、機械の故障診断においても、観測される症候から推定できる原因のリストを構築することは、一つの重要なタスクであるが、それでは症候として、その存在だけを問題にするのか、具体的な属性までを考慮に入れるのか、属性値の間の両立性や排他性をどのように解釈するのかなどによって、採用されるデータ形式は様々に異なってくる可能性があるし、それに伴って実際にタスクを実行するための処理も異なってくることが考えられる。実際のシステム構築にあたっては、問題に対するタスクの構造を決定するだけでは不十分で、現実の状況をどのようなデータ構造の下に記述するかを決定する必要があるのである。

2.1.3 ドメインオントロジーと領域独立オントロジー

実際に問題となる状況をどのようなデータ構造の下に管理するかの問題は、一般にはドメインオントロジーの問題とされているが、これまでの知識ベースシステムでいうならば、データ構造としてのフレームないしはオブジェクトの構造を決定することに相当する。これまでの所、このようなデータ構造の決定は、特殊な場合を除いては個別のシステムの要求に合わせたいわば場当たり的な定義がされてきたにすぎない。

このことはタスクオントロジーが整理されたとしても、依然として知識ベースシステムの実装に個別的、場当たり的なものが残ることを示している。この問題に関する限り、依然としてシステム実装の具体的な指針が明らかにされているとは言えない。

ドメインオントロジーが、少なくとも個別のシステムに強く依存するのではなく、一定の広がりを持つアプリケーションに適用できる一般性の高い形で定義できるならば、そのようなドメインオントロジーは、システム実装の指針を与えるものとして有用なものとなることが期待される。また、この考え方を拡張するならば、特定の応用領域に限定されない、概念の体系としてのオントロジーを確定できるのではないかという考え方にも通じることとなる。実際にこのような汎用オントロジーの記述を目指した研究もなされている。しかしながら問題は、このような分野独立の汎用的な形でのオントロジーは、どのようにして確定することができるか、また、そのようなオントロジーは、実際のシステム実装にどのように役立つかである。

例えば Cyc や電子化辞書、Wordnet などでは、概念体系の整理が試みられており、「トップレベルオントロジー」という形でその概念体系が公開されている。これらのトップレベルオントロジーはある種の概念の体系を表しており、また、Cyc などではマイクロセオリーという形で、いくつかの基本的な質問に答えることのできる知識を付加してあるなど、それ自体興味深い枠組みを提案しているということができる。しかしそれではこのような形で整理されたトップレベルオントロジーは、知識表現という観点から、システム構築にあたっての何らかの指針を与えることができるのであろうか。

2.1.4 知識表現とオントロジー

先にも述べたように、情報の操作という観点からは、タスクオントロジーがシステム実装にあたって有用な指針を与えてくれる。一方、領域独立なオントロジーが、ドメインオントロジーの定義にどのような効果を持つかが知識表現の観点からは興味のある問題となる。

ところで、一般に知識ベースシステムにおけるドメインオントロジーは、問題解決の観点から対象分野の物事をどのような情報として取り扱うかを規定するものである。フレームないしはオブジェクトの枠組みでいえば、どのような概念クラスとしてのオブジェクトを取り扱う必要があるか、また、それらのオブジェクトの状態を示すための変数としてどのようなものを想定するか、変数の値としてどのようなものを考える

かなどを規定する必要がある。また、多くの場合、オブジェクト間の関係や、変数の値に関する制約、変数値の間の相互制約などについても管理する必要がある。知識表現の立場からは、例えば領域独立なオントロジーが、特定の知識ベースシステムに必要なドメインオントロジーの構築にどのように役立つかが興味の対象となる。逆に言えば、領域独立なオントロジーとして、どのような情報が提供されればドメインオントロジーの構築に有効かという見方もできるであろう。

例えば Cyc などに見られるオントロジーでは、対象領域における概念体系（Taxionomy）や、概念間の様々な制約関係が記述されている。このような体系を利用することにより、例えばどのような概念クラスがシステムの取り扱う問題に関係してくるか、また、そのような概念クラスをどのように名前づけるかということに関しては、貴重な情報が得られると期待できるであろう。とりわけ概念クラスに対する名前付けが標準化されることは、異なるシステムで取り扱われる情報にどの程度の共通性があるかを評価する上で有用であると考えられる。また、概念クラス間の制約関係は、システム実装にあたってのいくつかの問題を事前に回避できる可能性を提供するという意味から、やはり有用な情報となりうると考えられる。

しかし一方で、Cyc などに見られる汎用的なオントロジーがそのままの形で特定の問題解決に関わるドメインオントロジーとして利用できるかといえば、これはこのままの形では難しいことが予想される。まず第一に、Cyc などに見られるオントロジーは、基本的には概念クラスの間の関係を記述するものであって、そのインスタンスの振る舞いに関する記述を行うものではない。一方で、多くの知識ベースシステムでは、個々の具体的存在としてのインスタンスの振る舞いが問題となる場合が多いのである。

しかも、これらのインスタンスをどのようなものとして見るかは、個々の知識ベースシステムによって異なる場合が多い。例えば医療診断の場合など、症状の存在そのものだけである程度の推論が行える場合もあれば、細かな属性値までを問題としなければならない場合も考えられる。また、時間的前後関係や、空間的関係なども考慮に入れる必要があるかどうかによっても、インスタンスを構成する情報の構造は多様に変化することが考えられる。要するに、様々な対象分野の具体的な問題を解くための知識ベースシステムでは、その利用するデータ構造に関しても、個々のシステムに固有なものが要求されることが多いわけである。この背景には、概念というものを細かく定義していこうとすると、その記述の持つ共通性というものを保証することが徐々に困難な問題となるという事実があると考えることができる。

2.1.5 知識表現から見たオントロジーへの期待

様々な分野における様々な実体に関する概念を整理したものとしての汎用オントロジーについて、記述の詳細さと結果としての情報の共通性の間に上記のような問題があるとして、それでは知識表現という観点から、オントロジーをどのように考えればよいかが問題となる。特定の応用領域におけるドメインオントロジーを構築するにあたっては、およそ次の項目について定義を行う必要があろう。

1. 対象領域で取り扱う様々な物事をどのような概念構造の下に記述するか
2. それぞれの概念クラスの間にどのような関係ないし制約が存在するか
3. それぞれの概念クラスをどのような属性の集合として記述するか
4. それぞれの属性に対応する属性値はどのようなものか。特に時間的空間的関係まで含む記述が必要かどうか
5. 属性値そのもの、あるいは属性値の間にどのような制約が存在するか

これらの内で、特に1から3までについては、汎用のオントロジーで定義しておくことに大きな意味があると考えられるであろう。特に概念クラスをどのような名前（記号）で呼ぶか、また、概念クラスに関連する様々な属性をどのような名前で呼ぶかについて一定の指針が与えられることは、異なる知識ベースシステムの間でどのような共通の情報が取り扱われているのかに関して、有用な情報を提供するものであり、ひいては知識の共有に関する有力な手がかりを与えるものであると期待できるであろう。これらの情報は、複数の知識ベースシステムの間で共有されることに大きな意味のあるものであると考えることができる。

一方、個々の属性値の取り扱いにまで関する情報は、個々のシステムごとにある程度の個別性を要求されるものである可能性が大きいと考えられる。これらの情報は、システムの間で全く独立のものとは言えないにしても、全てのシステムの間でただ一つの体系を共有できれば良いとすることにはかなりの無理があるのではないかと考えられる。これらの情報については、システムの設計者がその都度参照できる形で、いわばライブラリに近い形で整理しておくことが望ましいのではないだろうか。個々のシステムの設計者は、これまでに設計されたシステムの中で属性値がどのように取り扱われているのかを検索し、現在の目的に最も近いものをいわば流用し、目的に合わせて変更することによって新しいオブジェクトの設計を効率的に行うことができるよ

うになる。一方で、このようにして定義された属性値情報を、その元となった情報への参照も含めて新しいライブラリとして追加していくならば、システムごとのデータの取り扱いについて有用な情報を蓄積することができると考えられるのであり、個々のシステムごとの知識の取り扱いの相違を明らかにし、知識の再利用性を高めることにもつながるのではないかと期待される。

知識ベースシステムの構築においては、知識表現の枠組みをどのようなものとするかが重要な課題となる。特にドメインオントロジーの構築に際して、広い概念体系に関する様々な情報を、情報の共通性のレベルに応じて適切に管理できる汎用のオントロジー管理の体系を構築することにより、システム構築の効率を向上させるとともに、知識そのものの共有性や再利用性を高めていくことが今後期待される。

2.2　語彙知識の獲得とオントロジー工学

2.2.1　はじめに

言語理解、機械翻訳など自然言語の機械処理で意味処理を本格的に行うためには、実用規模の単語についての意味情報（語彙知識）が不可欠となる。このような情報を、何から、どのように収集し、構造化すればよいかが重要な問題となっている。我々は、この問題に対して、電子化された言語データ、特に辞書データから(半)自動的に語彙知識を獲得するためのアルゴリズムを明らかにし，語彙知識ベースの構築を目指した研究を進めている。この研究では，具体的には、単語の意味を記述した語義文から、単語の表す概念間の上位－下位関係のみならず部分－全体関係など他の種々の意味情報を抽出する手法や、獲得された意味情報を広い意味でのシソーラス（語彙知識ベース）として統合化する方法などを解明する必要がある。基本的にはボトムアップによる手法が重要であろう。ここでの語彙知識ベースとは、単語の表す概念間の上位－下位関係を基盤に個々の語（概念）の意味の定義を明確にしたものを想定している［Yoshida 82］。さらに、このような語彙知識ベース上での、単語間の意味の類似性の問題や概念階層における上位概念の持つ属性の下位概念への伝播に関した問題などについても検討する必要がある。

ところで、自然言語処理の分野では、シソーラス、意味辞書、語彙知識ベースなどの用語が用いられてきており、オントロジーという用語（概念）が使用されることは少なかった。しかし、我々が考えている語彙知識ベースは、基本的には文献［溝口95］で定義されているオントロジー（ontology）の一つと捉らえることができると思われる。ただし、自然言語で対象とする語彙は膨大であり、しかも一般に単語は多義であるため、機械処理向きにその意味の記述を行うことは、非常に困難な問題を含んでいる。従って、本調査研究で提案されているオントロジー工学は、語彙知識ベース構築における基盤技術をなすものでもあるわけで、オントロジー工学の確立を切に期待すると共に、我々の研究がその確立へささやかでも貢献できれば幸いである。

本報告では、オントロジーの構築技法に関する観点から、国語辞典情報を利用した語彙知識ベースの構築に関して、主として語彙知識の獲得に関する調査研究の一端について紹介したい。まず、国語辞典にから上位－下位関係（階層関係）の（半）自動的な獲得法の提案とその実験システムの概要について報告する。この詳細な内容については、本報告書の第2部第2章で紹介する予定である。次に、その成果に基づいた、

上位ー下位関係シソーラス作成支援システムと試作シソーラスについて、その概要を報告する。さらに、目標としている語彙知識ベースの構築に向けて、その基本方針について述べる。

2.2.2 国語辞典からの上位ー下位関係（階層関係）の獲得

(1) 語義文の特徴と定義語

国語辞典には、単語の意味・用法に関する膨大な情報が含まれている。これらは言語（語彙）に関する貴重な知識であり、特に、語義文は見出し語の意味を言語で記述したもので、単語の意味に関する重要な情報源と考えられる。語義文の記述的な特徴の一つに、語義の中心となる語（以下、定義語と呼ぶ）がいろいろな側面（見方・観点）から限定され、特殊化されている、ということが挙げられる。また、定義語は、一般に見出し語の上位概念を表す語（上位語）であり、語義文の末尾に現れる場合が多い。このような特徴を利用して、語義文から定義語を抽出し、見出し語との上位下位関係を求めようという試みがある。

例えば、［中野 81］は、新明解国語辞典を用いて分類語彙表の増補へ応用している。［富浦 87］は、動詞の語義文から階層関係の抽出を試みている。［荻野 87］は、明解国語辞典を用いて人手により解析を行い、名詞シソーラスを作成している。また、LOMGMAN英英辞典を利用したものとしては、［Michiels 82］、［Boguraev 98］、［中村 87］などの研究が報告されている。しかし、語義文の中には、定義語がその末尾に現れなかったり、定義語が上位語ではなかったりする場合があるため、見出し語と定義語との間の階層関係付けまでも自動的に行っている研究は少ない。

(2) 上位ー下位関係抽出（階層関係付け）システムの開発

語義文から定義語を抽出し、見出し語との階層関係を求めるためには、定義語が語義文中でどこに現れるのか、また、関係付けのための情報は何を手掛かりにしたらよいか、などを明らかにしておかなければならない。このため、語義文の調査が必要となる。我々は、関係付けに有用な情報を担っている文末表現（以下、機能表現と呼ぶ）とその機能を整理し、語義文構造のモデル化を提案している［鶴丸 92, 95b］。調査に使用した国語辞典は、主として新明解国語辞典であるが、随時、岩波国語辞典、旺文社国語辞典など、同様な市販の辞典も参照した。

語義文の構造的特徴と機能表現の意味関係情報とに基づいて、定義語抽出の手順と

関係付けためのアルゴリズムの開発を行った。これらを評価するために、関係付けシステムを作成し、新明解国語辞典の名詞見出し語 約3,000語の語義文で実験を行った。その結果、約94%の精度が期待できることが確認された。［本報告書、第２部第２章を参照］

2.2.3　国語辞典を利用した上位ー下位関係シソーラスの構築

ここで、上位ー下位関係シソーラス［田中 87］とは、概念間の上位ー下位関係や部分ー全体関係に基づくシソーラスのことを指す。即ち、文献［溝口 95］で定義されているTaxonomyと類似の概念である。

(1)　シソーラス作成支援システムの開発とシソーラスの試作

シソーラス作成支援システムは、データ作成システム、エントリー作成システム、および、検索システムの３つのサブシステムからなる。データ作成システムは、関係付けシステムで抽出された定義語に対して、表記の統一、同形異音（例えば、飲料［いんりょう・のみりょう］など）、多義などの処理に用いられる。これとエントリー作成システムにより、‘見出し語概念’として約 80,000語から成る名詞シソーラスを試作している［鶴丸 93］。

検索システムは概念間の階層関係を検索するためのもので、シソーラスの利用に不可欠のツールである。また、これにより、階層関係の循環（サーキュレーション）やバイパスなどのチェックが可能となる。

(2)　試作シソーラスでの問題点［鶴丸 95a］

国語辞典は人の利用を前提に作成されたものであり、その内容を理解するには、高度の知識と判断力が前提とされている。そのため、国語辞典に基づくシソーラスでは、極大語の存在、同位語間の不均質などの問題点が生じる。ここで、極大語とは、階層構造上で上位語（上位概念を表す語）をもたない語のことをいう。従って、極大語があると階層関係が途切れることになる。同位語とは、同じ上位語に直接関係付けられている複数の下位語のことである。従って、その上位語の下で同じレベルにあるのことになる。また、語義文中で使用されている単語（定義語）が、語義番号で陽に区別されていない意味で使われている場合があり、語義の曖昧性が問題となる。

2.2.4　国語辞典に基づく語彙知識ベースの構築に向けて

(1)　語彙知識ベースにおける単語（概念）間の意味的関係と見方［鶴丸 76］

単語（概念）の意味を記述するためには、それらの間の意味的関係（見方）として、どのようなものを用意すればよいであろうか？また、これは実際に可能であろうか？この問題は意味素の設定と関係しており、非常に困難かつ問題を含んでいると思われる。哲学的な意味でのオントロジーとも関連している作業であろう。我々は、工学的な観点から、ある程度、概念の範囲を限って、その範中で設定することを提案してきた。

これまでの研究で得られた代表的な意味関係を列挙してみる。独断的な関係もあると思われるが今後検討して行きたい。

① 階層関係（上位(AKO：<)、下位(OKA：>)、同義(EGU：≡)、など）

② 射影関係（‘具体的もの’概念とその概念の側面（属性）との関係、次項を参照）

③ 格関係（主体(AGT)、作用の対象(OBJ)、動作の対象、起点的対象、終点的対象など）

④ 描写的定義関係（分解関係とも呼ぶ。‘こと’概念での全体部分関係とも見做せる）

⑤ 様相(様態)関係（ある概念とその概念の様相(様態)との関係）

⑥ 因果関係（手段、方法、目的、原因、理由、結果、など）

⑦ 論理関係、⑧ 時間的前後関係、⑨ 条件関係、⑩ 対義関係

また、‘具体的もの’概念の中で‘生産物’概念に関しては、表2.2-1のような見方が考えられよう。括弧の中は略記号である。

(2)　語義文中に現れる記述的特徴と意味的関係（見方）［鶴丸 95b］

語義文中に現れる記述的特徴と意味的関係（見方）として、2つのタイプを考えている。

① 語義文の末尾に現れる機能表現

② 定義語の修飾部に現れる機能表現

前者に関しては既に考察している。後者に関しては、(a) 助詞など概念間の関係を規定するもの、と (b) 見方と密接に関係している概念語で表される表現とが考えられる。(b) の表現に関して、例えば、語義文中に‘作る’が現れていれば、“製造に関する見方”から制限されていると推定されよう。また、“製造に関する見方”の一つである

表2.2-1　‘生産物’に対する代表的な見方の例

	基本的見方
使用に関する見方	使用目的(POU)、使用方法(HTU)、…
製造に関する見方	製造方法(HTM)、材料(MTR)、…
構成に関する見方	部分(PRT)、要素(EMT)、成分(CON)、…
性質に関する見方	物理的性質、性能、…
量・単位に関する見方	個数、厚さ、重さ、…
様相に関する見方	色、形、価値、程度、作用相(AFF)、…
時間に関する見方	存在時間
場所に関する見方	存在場所

“材料”に対しては、次のように、さまざまな表記方法が取られている。

‘～から作る(った)［清酒、日本酒］’、

‘～を原料として作った［葡萄酒］’、

‘～から造った［諸白］’、‘～で造った［新酒］’

しかし、見方の表記が陽になされてない場合もある。例えば、‘～に供える酒［御神酒］’のように“連体形＋名詞”の形式をしている場合などである。このような場合、“ための”や“ために用いる”などの語を補足して矛盾しなければ、“使用目的”などの関係が想定される。しかし、このような解析を行うためには、省略語の補間など人のかなりの知識と高度な判断に依らざるをえない。また、格関係、様相の分類などが必要となる。

(3)　語義文からの語彙知識の獲得に向けて［鶴丸 95b］

語義文から求められる語彙知識として、次のような意味的関係（見方）が考えられる。

① 見出し語と定義語との間の意味的関係

② 定義語とその修飾部（を構成する語）と間の意味的関係

③ 修飾部を構成する語の間の意味的関係

ところで、①に関しては、前項で獲得のためのアルゴリズムを明らかにしているので、ここでは、②と③を求める方法に関して、例を用いて、その基本方針について説明したい。

処理の大枠は次の通りである。

(a) 定義語を直接修飾している修飾部を求める。

(b) 修飾部を解析し、見方と関係付けられる語句を求める

(c) 処理し易い（例えば、Frame-likeな）形式に変換する。

例として、次の‘焼酎’の語義文について考えてみよう。

【焼酎】(0010)：酒かす・サツマイモなどを蒸留して作った、アルコール分の強い酒。

(a) 定義語‘酒’を直接修飾している修飾部を求める。修飾部が２つあるので、分ける。

イ. 酒かす・サツマイモなどを蒸留して作った酒

ロ. アルコール分の強い酒

(b) 修飾部を解析し、見方と関係付けられる語句を求める。

イの解析結果から、‘焼酎’の“製造方法”が‘蒸留すること’であり、‘蒸留すること’の“対象”が‘酒かす・サツマイモなど’であることが求められよう。さらに、‘焼酎’の“材料”が‘酒かす・サツマイモなど’も推測されよう。

ロの解析結果から、‘焼酎’の“成分”が‘アルコール’であり、‘アルコール’の“作用”が‘強い’であることが求められよう。

(c) Frame-like な形式で表すと次のように書ける（図２.２-１）。

```
{焼酎[しょうちゅう](0010)
  {AKO {酒[さけ](0010)                    }}
  {OKA { {粕取り[かすとり](0010)            }
         {泡盛[あわもり](0010)            }}}
  {EQV {酎[ちゅう](0010)
       {AKO {酒[さけ](0010)}            }}}
  {HTM { '蒸留する'
         {OBJ { ?
                {MTR {{ '酒かす' }
                      { 'サツマイモ' }}}}}}}
  {CON { 'アルコール'
         {AFF { '強い' }                 }}} }
```

図２.２-１ ‘焼酎’の語彙知識(意味記述)の一部

2.2.5 おわりに

オントロジーの構築技法に関する調査研究の観点から国語辞典情報を利用した語彙知識ベースの構築に関する調査研究の一端について報告した。現在、既に開発が進んでいる部分もあるが、本質的で難しい問題が山積している。語義文は、自然言語で記述してあり、省略も多く、人手に頼らざるを得ない部分が存在する。しかし、大量のデータを処理する必要があるため、支援システムの開発が不可欠となる。自然言語文の解析ツール（形態素解析、構文解析など）の開発、計算機上での語彙知識の表現法（データ構造）などについての研究も大切である。なお、国語辞典は使用者のある程度の知識を前提に編集されているわけであるから、それを機械処理に利用する場合には、暗黙の前提とされている知識としてどのようなものがあるかの解明が望まれる。

以下に本文で触れなかった主ないくつかの課題を紹介して今年度の報告としたい。

(1)オントロジー構築においては、概念間の（意味的）関係についての厳密な検討が必要である。例えば、部分－全体関係についての考察は興味のある課題の一つである［Winston 87］。我々は、部分－全体関係を3つのタイプに分けて論理的な定義を与え、これらが自然言語（語義文）では特に区別されずに使用されている場合があることを指摘した。また、上位－下位関係との融合により部分－全体関係を拡張するためのアルゴリズムを提案しているが［鶴丸 93］、これらについてもさらに調査・検討したい。

(2)シソーラスの試作では、現在のところ、一つの辞書（新明解国語辞典）から得られた情報しか使用していない。このため、一つの閉じた系として種々の考察ができる利点はあるが、情報の欠落などによる不備な点も存在する。例えば、階層構造上での単語間の跳躍や、極大語の存在などである。また、その辞書特有な意味しか記述されてなかったりする場合もあると思われる。従って、複数辞書からの語彙情報の獲得と統合化が重要な課題となる。そのためには、語義文で記述された語義の類似度の評価が重要となろう。

(3)シソーラスの応用に関して、シソーラスでは一般に、複数上位語を許している。そのため、複数パスが生じた場合に上位概念からの属性の継承に対して冗長性や曖昧性が問題となる。どのようにして最適なパスを優先的に選択したらよいかなどは、推論と関連した重要な課題である。

(4)動詞、形容詞、抽象語などの語彙知識の獲得と統合化。

(5) 一般分野の概念体系と専門分野の概念体系との有機的な統合システムの開発。

参考文献

[国研 73] 国立国語研究所: 分類語彙表, 秀英出版, 1973.

[鶴丸 76] 鶴丸弘昭, 藤田毅, 首藤公昭, 吉田将: 日本語の機械処理, 電子通信学会電子計算機研究会資料 EC76-49, 1976.

[中野 81] 中野 洋: 分類番号付け支援システム, 情報処理学会計算言語学研究会資料 25-5, 1981.

[Yoshida 82] S.Yoshida, H.Tsurumaru, T.Hitaka: MAN-ASSISTED MACHINE CONSTRUCTION OF A SEMANTIC DICTIONARY FOR NATURAL LANGUAGE PROCESSING, Proc. of COLING', 82, pp.419-424, 1982.

[Michiels 82] Michiels, A., J.Noel: APPROCHES TO THESAURUS PRODUCTION, Proc. of COLING'82, pp.227-232, 1982.

[田中 87] 田中穂積, 仁科喜久子: 上位／下位関係シソーラスISAMAP1の作成 [Ⅰ], [Ⅱ], 情処研報 (NL), Vol.87, No.84, pp.25-44, 1987.

[荻野 87] 荻野綱男: シソーラス作成の問題点, 日本語学, Vol.6, No.5, PP.64-71, 1987.

[中村 87] 中村順一, 酒井桂一, 長尾真: 英英辞典を用いた名詞の意味関係の分析, 電子情報通信学会, 言語処理とコミュニケーション技術研究報告, NLC86-23, 1987.

[富浦 87] 富浦洋一, 日高達, 吉田将: 語義文からの動詞の階層関係の抽出, 情処研報, Vol.87, No.84, pp.17-24, 1987.

[Winston 87] Winston, M.E., Herrmann, D.: A Taxonomy of Part-Whole Relation, Cognitive Science, Vol.11, pp.417-444, 1987.

[Boguraev 89] Boguraev,B., Briscoe,T. edited: Computational Lexicography for Natural Language Processing, pp.310, LONGMAM, 1989.

[鶴丸 92] 鶴丸弘昭, 井上淳, 日高達, 吉田将: 語義文からの階層関係の自動抽出, 信学技報, NLC92-2, No.24, pp.9-16, 1992.

[鶴丸 93] 鶴丸弘昭, 前田英幸, 山本和博, 日高達, 吉田将: 国語辞典に基づくシソーラスの構築に関する一考察, 信学技報, NLC93-58, Vol.93, No.367, pp.29-36, 1993.

[EDR 94] EDR電子化辞書研究所: 評価版EDR電子化辞書評価版, 1994.

[鶴丸 95a] 鶴丸弘昭, 城戸武雄, 肥後真一, 吉田将: 国語辞典に基づくシソーラスと単語の意味分類に関する一考察, 信学技報 (NLC95-44), Vol.95, No.321, pp.31-38, 1995.

[鶴丸 95b] 鶴丸弘昭, 城戸武雄, 日高達, 吉田将: 語義文の機能表現について, 信学技報 (NLC95-65), Vol.95, No.429, pp.47-54, 1995.
[溝口 95] 溝口理一郎: オントロジー工学序説.

2.3　言語学から見たオントロジー工学

2.3.1　はじめに

ここでは一人の言語学者としての筆者から見たオントロジー工学の現状と将来について重要だと考えられる点を簡単にまとめる。要点をまとめようとしているので、一つ一つが、また相互関係が大きな意味を持っているが、まずは切り分けて論じる。具体的には、後の第2部 3章「言語学者／外国語教師が貢献できること」で、どういうことがその例と考えられるかを論じるのでそちらを参照していただきたい。

2.3.2　重要だと考えられる相互に関係する3つの観点

次の3つの点が重要だと筆者は考えている。

(1)　普遍性を目指すか、個別性を目指すか？

言語学者にも個別言語によらない言語の本質を探ろうとする方向を目指す研究者と、言語の本質の具体的な表れである個別言語・個別の現象に興味を持っていると考えられる人とがいると見ることができる。もちろん、筆者もそうであるが、個別言語・個別の現象に比喩的な言い方をすれば横断的に見られる（と考えられる）普遍性に興味を持つ人もいるわけで、この二分法は単純に可能な2つのどちらかを選択すればいいというだけではすまないという問題をはらんでいる。オントロジーに興味を持つ人にも、この二分法が有効であるとすれば、言語学者の側からすれば、2つのどちらかを目指す人と両方を視野に入れた人、少なくとも3つのタイプの人がいてほしい。これは当然比喩であるので、程度の問題、視野という比喩で考えると角度は連続的ではないのかということが問題になり、実際には、まわりからどれかのタイプだと思われがちな人も、動いている主体なのであろう。その行動は、まわりの人間からするとタイプ分けしたくなるものであるとしても。

(2)　分野・領域に依存する知識と横断的な知識の区別

ここでも二分法が問題になる。言語学者としては、言語の本質だと考えられる文脈依存性の問題が、分野・領域の区別、さらには言語・文化・状況の個別性の区別と相互に関係していることの本質を捉えたいが、やはり問題は多くのオントロジーの研究

者あるいは知識に興味を持つ人達が共有していると思われる、切り分ければ済むわけではない大きな問題であり、そこで立ち止まらざるをえない。筆者は、電子的なネットワークが爆発的に拡大する中で、むしろ個別的な知識の公開と共有を進めていくのが当面の課題なのではないかと考えている。その知識から普遍性を引き出すことはできないかもしれない。また効率が悪いかもしれない。しかし、電子的、また人的ネットワークを通して知識を外在化し、それを共有することが、知識の普遍性を明らかにできないかもしれないが、多くの人にとって役にたつ情報になることは確かだと思える。もちろん、そういうふうにして公開された知識から、新たな知識の普遍性を探ることはできるはずで、そこから普遍性の探求に向かう研究者には、新たな地平を開いてくれるだろうという意味でがんばってほしいと思う。

(3) 人間の関係の変化

やはり日本社会の問題点を指摘したい。研究者の集団も研究者ではない人々の集団も縦割りになってしまっているところが問題で、それをいやがおうでも壊してしまう動きが進みつつあるように思える。研究者も研究機関も、蛸壺の中には安住していられなくなっていると考えられる。ここでも個人の意志・集団の暗黙の合意が崩れつつある。ここで言いたいことは、学部・学科・講座という小さな組織から離れて、さらに専門の名前を忘れて協力しあえることを探し、それを蓄積し、有効利用し、さらに引き継いでいくことだと思える。さらに言えば、専門家と素人の区別も無くなっていくしかないだろう。そういう方向の中で、「オントロジー工学者」だと考える、あるいはまわりから考えられている人と「言語学者」だと考える、あるいはまわりから考えられていて、そうだと思っている人との新しい関係をどう作っていくかがこれからの重要な課題だと考えられる。課題は、真剣に考えないといけない立場にいる人ではなくて、そういう立場にいない人によって克服されることが多いように思える。そうすると、専門家の役割は何なのかを問題にしないといけなくなるだろう。こういう意味では、本項を読んで下さる方々が、ここで取り上げている問題を解決する糸口を切り開いていけるかどうかには大きな疑問ある。しかし、我々がはっきりした問題意識を持つことで、多くの芽を摘まないで、育てていくことができる可能性こそ大事にすべきだろう。

2.4 オントロジー工学とデータベース

2.4.1 データモデリングとオントロジー

データベースとは実世界の写し絵である。ということは、実世界を記述するための記号系が必要で、データベースではそれをデータモデルという。データベースが初めて世に出たのは1963年の米国GE社によるIDSという製品であるといわれているが、そのデータベースを記述するために使われたデータモデルはネットワークデータモデルと言われている。いわゆる第一世代のデータベースシステムである。よく知られているように、データベースはその後転機を迎え、1970年になると米国IBM社のSan Jose研究所のCodd博士によりリレーショナルデータモデルが提案されることになり、現在のリレーショナルデータベース時代を迎えることになる。現在、世の中の主たるデータベースシステムはOracle、 DB2、Sybase、Informixに代表されるリレーショナルデータベースシステムが主流であるが、1990年代になるとCADやマルチメディア応用に効果的なオブジェクト指向データベースシステムが出現した。これがデータベースの第三世代である。

しかしながら、どの世代のデータベースにも共通して言えることは、データベースは、スキーマとインスタンスの２つの側面から成り立っているということである。例えば、リレーショナルデータベースでは、リレーション社員（社員番号、氏名、所属、給与）は、リレーション名が社員で、それは社員番号、氏名、所属、給与という属性を持ち、さらに属性社員番号のドメインは整数であり、社員番号がこのリレーションの主キーである、という時間的に不変なリレーションの構造を決めている側面があり、それをリレーションスキーマと呼んでいる。一方、このリレーションに登録される社員がいて、例えば社員が千人いる会社のリレーション社員は千本のタップルからなる。ここで、注意しないといけないことは、明日には新しい社員が入ってくるかもしれないし、あるいはある社員が会社を辞めてしまうかもしれない。リレーションが社員の最新情報を保持しているのならば、当然新入社員に対応して新たなタップルがリレーション社員に挿入され、一方社員が辞めれば当該タップルはそのリレーションから削除されねばならない。このように、リレーションという言葉は二義的である。

さて、データベースを設計するということは、スキーマを設計するということである。インスタンスはデータの入力作業の結果得られる（にしか過ぎない）。データベースがオントロジーと関わるのはまず、このデータベース設計のプロセスである。そこ

では、ドメイン、属性、リレーション名、主キー、外部キー、関数従属性などが認識されねばならない。

この議論は、オブジェクト指向データベースでさらにはっきりする。なぜなら、オブジェクト指向データベーススキーマを構築するということは、使われる属性やそのドメインに加えて、クラスライブラリとしてデータベース化の対象となった実世界の概念階層が構築されねばならないからである。これはオブジェクト指向データベースでは、いわゆるIS-Aの関係で概念の階層化が明示的に行われなければならないからである。

さて、リレーショナルデータベースにしろ、オブジェクト指向データベースにしろ、これまでデータベースとオントロジーが積極的にどう関わってきているのか、この問題に考察が加えられてきたことはなかった。データベースでは、例えば対象物の属性として「色（color）」を考えた場合、その属性がとりうる値（value）を属性 色のドメインからとる。例えば、虹の色を我々日本人が表す場合、赤から紫までの７色を用意しておけばよい。しかし、「赤」とはどういう色なのかは、データベースでは議論していない。つまり、データベースのメタデータ管理の立場は、ユーザからの質問（query）や更新（update）に効率よく答えるために必要であることは言を待たないが、もう一つ大事な点は、データベースの一貫性を保つためにぜひ必要なのである。つまり、もしあるユーザが虹の色を７色以外の色に変更しようとした場合には、データベース管理システムはそれを察知してそのような色の変更を禁止して、データベースの内容がおかしくならないように、つまりデータベースが実世界の実情と齟齬をきたさないように制御するために使われる。

オントロジーに共通する議論はオブジェクト指向データベース管理システム、あるいはオブジェクト指向プログラミング言語の処理系の構築時に見られる。とくに、オブジェクト指向プログラミング言語Smalltalk-80では"Everything is an object."であるので、実世界の一事象を表すためのオブジェクトはもちろん（そのオブジェクトを規定するクラスのインスタンスとしての）オブジェクトであるが、それを規定しているクラスもオブジェクトでなければならないので、そのクラスを表現するためにクラスのクラス、すなわちメタクラスが導入されることになる。この議論は、明らかにさらに高次のメタクラスを導入しないとけりがつかない所となるが、データベースにしろプログラミング言語にしろ、ユーザが関わる第一義的なところの意味がそのレベルではっきりすればよいという立場をとるから、一段階上がったメタデータ、メタクラスのレベルで実世界の意味記述を打ち止めにしている。したがって、赤は他の６色との対立に

おいて赤なのであり、それが赤の定義なのであるという、まことにソシュール（Ferdinand de Saussure）的な意味解釈の世界と言うことができる。

2.4.2　データモデルへの時間的概念の導入

実世界をモデリングしようとしたときに、時間の概念をどう取り込むかが問われるときがある。これには二つの立場があり、一つは、例えば社員の給与歴を記録しておいてそれらに関するユーザからの質問に答えようとするもので、これは履歴データベース（historical database）といわれてきた。Snodgrassらが中心となってまとめているTSQL（Temporal SQL）データベース言語はこのためのものである。一方、時間の概念が問われたもう一つの側面にマルチメディアデータベースがある。そこでは、ビデオとか音とかを記述しようとしたときには、それらのある瞬間の値だけを切り出したのでは意味を持たず、それらはある一定時間の連続したつながりをもって初めて意味を持ってくる。このような例は外に対象物の挙動を記述してデータベース化したいと考えるときにもある。このようなデータを時間的オブジェクト（temporal object）とよび、そのようなデータベースを時間的データベース（temporal database）という。

さて、時間的オブジェクトをどのように記述するかは幾つかの研究があるが、それを時区間（time interval）として捉える方法は広く受け入れられている。例えばビデオはある時間に収録が始まり、ある時間にそれが終わる時区間である。音も同様である。時区間に対する論理的な考察はAllenにより行われた。そこでは、任意の二つの時区間の間には、equal, before, after, during, contains, overlaps, overlapped_by, meets, met_by, starts, started_by, finishes, finished_byなる１３種の時間的関連のうちの一つが成立することを明らかにしている。

時区間によるマルチメディアデータの表現を考えたときに、避けて通れないのが時間的オブジェクト間の時間的関連をどのように表現するかである。しかしながら、この問題はこれまでデータモデルの問題として真剣に議論されてきたことがなかった。ここでは、筆者らが考案してきた枠組みを一つ紹介する。これは、Cattellらがオブジェクト指向データベースの標準データモデルを提言しようとして1993年に発表したODMG-93オブジェクトモデルを拡張したものである。ODMG-93データモデルでは、オブジェクトとオブジェクトの参照制約（referencial constraint）を記述する目的で、クラス定義の中に関連を表現するためのrelationship句が導入された。これを使うと、例えば学生と科目の間に、クラス学生には属性「履修する」がありそのドメインがクラ

ス科目で、一方クラス科目には属性「履修される」がありそのドメインはクラス学生である場合、ある学生が都合で退学したとき、その学生を表しているオブジェクトがクラス学生のインスタンス集合から削除されるのは勿論のこと、その学生が履修していた科目オブジェクトの属性 履修される、の値（＝この科目を履修しているすべての学生へのポインタの集合である）から、当該学生へのポインタを取り除くことにより、在籍もしていない人がその科目を履修しているという非現実がデータベース内で発生しないようにして、データベースの一貫性を保証できる。

この関連記述を拡張して、マルチメディアオブジェクト間にAllenの１３種の時間的関連をクラス定義に明示的に表現することが可能となる。例えば、ある音楽（演奏）を一つはビデオで映像を、もう一つはマイクロフォンで音を収録する状況を考える。この時、収録された映像データと音データの同期をとるために両者は同時に始まり同時に終了されねばならない。つまり、両者の間にはAllenのequals関連が定義されねばならない、ということである。この間連はODMG-93クラス定義のrelationship句にオブジェクト間のアクセスパスをドット記法をも許すように拡張することで次のように定義できる。

【例　題】

ある音楽オブジェクトmはその音成分オブジェクトsとビデオ成分オブジェクトvから成り立ち、sとvは同時に収録を開始され、また同時に収録を終了されるものとする。このとき、sとvのequals関連はクラスMusic、クラスSound、クラスVideoで次のように記述される（ただし、クラスVideoはクラスSoundに準ずるので記述を省略する）。

```
class Music: TimeInterval{
// attributs
char* title;
Sound s_track;
Video v_track;
...
//operations
void playback();
...
};

class Sound: TimeInterval{
// attributes
Music s_track_of;
...
// relationships
relationship Video s_track_of.v_track equal Video::v_track_of.s_track;
...
// operations
...
};
```

なお、クラスSoundのrelationship句での指定により、クラスSoundの一つのオブジェクトをsとしたとき、それを音成分としている音楽オブジェクトをmとすれば、m.s_track=s, s.s_track_of=m, m.v_track=v, v.v_track_of=mなる関係があり（ここにvはmのビデオ成分）、その結果、音オブジェクトsについては関係equal (s, s.s_track_o.v_track) が、ビデオオブジェクトvについては関係equal (v, v.v_track_of.s_track) が成り立たなければならないことを宣言していることになっている点である。

2.4.3 時空間データベースに向けて

従来型のデータモデルに共通しているのは、実世界では姿・形のあるものがデータベース化されるとレコードとか（ネットワークデータモデル）、タップルとか（リレーショナルデータモデル）、あるいはオブジェクト（オブジェクト指向データモデル）としか表現されないと言うことである。これはどう言うことかというと、実世界の一台の自動車は、例えばリレーショナルデータモデルでは、一本のタップルとして表現されるにしか過ぎないと言うことである。具体的に、登録番号「東京55す1234」、「コクサン」社製、オーナーが「ジェームスボンド」氏の車両はリレーション車両（登録番号、メーカー、オーナー）の１タップル（東京55す1234、コクサン、ジェームスボンド）として表される。オブジェクト指向データモデルで表現した場合、リレーショナルデータモデルと違ってオブジェクトの属性値に加えて振る舞いを表現できる点が進化している。例えば、走り出す、停るなどの車が持っている動作をメソッドとしてオブジェクト自身がそれらを持てる。しかし、タップルにしろオブジェクトにしろ、それらは概念空間の一つの”点”でしか表現されていないと言うことである。本当は、自動車は３次元の”形”を持っているのに。

さて、実世界の対象物が持っている”形”そのものをモデル化するという概念は従来のデータモデルにはない。従って、もしそれを表現しようとすれば、それらを表す属性を導入することになる。例えば、直方体を表すにはその頂点の座標を与えることが考えられる。そうすれば、それらの物体を存在させている３次元の空間座標系を前提に、例えばリレーショナルデータベースの場合、埋込み型SQL言語を使ってプログラミングして、他の物体との関連やそのものの挙動を計算できる。しかし、この枠組みはあくまで計算できるということであり、必要な属性とその値を与え、計算完備なプログラミング言語を使ってアプリケーションプログラマが目的に合致したプログラムを書いたということである。計算完備であるから、およそ計算できることは必ずプ

ログラミングできるのであり、それはそれで結構なことであるが、データモデルの観点から見るとデータモデルの実世界表現能力が不十分な部分を計算完備なプログラミング言語が補っていると捉えられる。

では、実世界の対象物をあるがままに表現できるデータモデルがあったとしたら一体どういうことになるのであろうか？まず、第一にモデル化されたものを我々はこの目で見ることができる。自動車は丁度そのミニチュアがデータベース内に出来上がり、我々はそれを見ることができる。第二に、もちろんその自動車は動くことができる。これは、オブジェクト指向データモデルではオブジェクトは振る舞いを持っていることに対応する。第三に、従ってデータベース内に実世界が時間的・空間的な性質を保存したまま再現される。そこでは、属性、振る舞い、基本的な時間的・空間的関連が表現されており、データベース内で相当のシミュレーションを行うことができる。表現しきれなかった制約や振る舞いについては、なお計算完備なプログラミング言語で援用することが必要となろうが。

このような新しいデータモデルを4次元時空間データモデルと呼ぶことにする。より具体的には、現在筆者が構築しているそれは、実世界の対象物を丁度（子供が遊ぶおもちゃの）LEGOのブロック（を組み合わせて出来上がる複合ブロック）で表すので、それを「ブロックデータモデル」と名付けることにする。もちろんLEGOのブロックにはエンジン付きのそれがあるように、ブロックデータモデルの構成要素であるブロックはそれ自身空間内を動ける。ブロックデータモデルでは実世界の対象物をMBB（Minimum Bounding Block, 極小外接ブロック）で表す。これは基本的には実世界の対象物をブロックで表すというモデルである。いくつものブロックを組み立てて複雑な形を作り上げるように、ブロックデータモデルでは基本ブロックを組み合わせて複合ブロックを作り上げてよい。しかし、実世界のオブジェクトをすべてブロックとして表現すると本来ブロックではない形状のオブジェクトもブロックと表現されるために表現誤差が生じる。この誤差をできるだけ小さくするために、オブジェクトが複合オブジェクトである場合、オブジェクト間の厳密な干渉チェックを行うことにより、それを基本ブロックレベルの誤差に引き下げることができる。ブロックデータモデルは仮想現実感システムとオブジェクト指向データベースシステムを高速のネットワークで結合して実現される。そこでは両システム間のオブジェクトの挙動の時間的・空間的同期のとりかたが問題の一つとなる。外に重要な問題として、4次元時空間データベース質問言語の設計と実装があるが、ここではそれを言及するにとどめる。

2.5　オントロジー工学への生成的アプローチ

2.5.1　アプローチ

オントロジー生成の支援について述べる。異種の知識の交わり、結びつけ、編集を支援する枠組みが重要であることを指摘し、そのためのアプローチを提案する。

このアプローチの長期目標は、これからの知識中心型の世界にむけてそのための知識基盤を作り出すことである。知識基盤を実際に作ろうという試みは、これまでCycなどでも行われた。Cycでは、CycListという少人数の人たちが百科事典や自分が本来持っているような知識に基づいて情報を集め、一定の表現言語で知識を延々と記述した。これはあまりうまくいかなかった。

オントロジー生成は本質的には知識の体系化であると考えることができる。知識の体系化のプロセスはトップダウン・観念的に力任せにやるのではなく、ボトムアップ・実証的に漸進的に進めていくのが望ましいと考えられる。換言すると、異種の知識の交わり、結びつけ、編集のプロセス（図2.5-1）を詳細に分析し、不均質情報を結びつけたり、それを編集したりすることによって知識基盤を構築するプロセスを支援するところから着手すべきであると考えている。

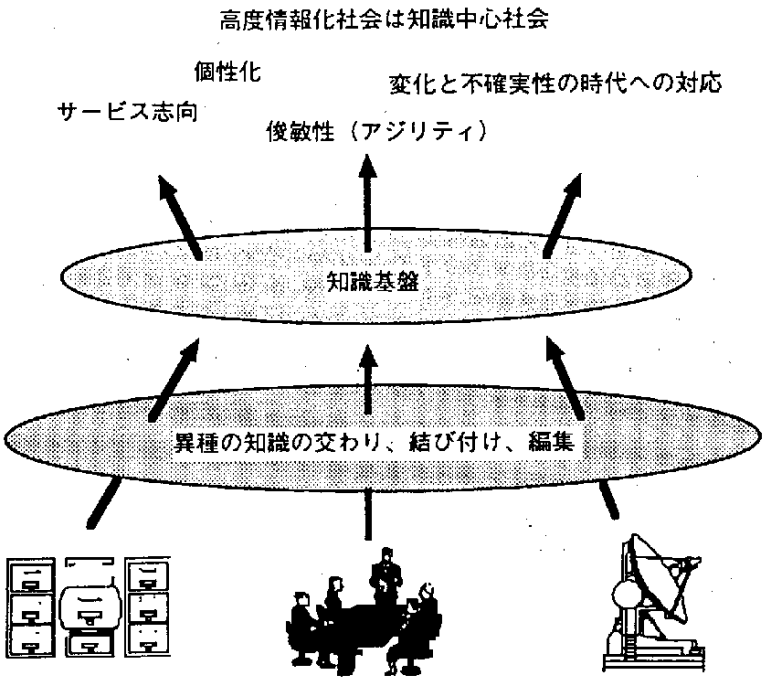


図2.5-1　知識コミュニティプロジェクトのねらい

基本的に、知識体系の構築は人間が主体となって進めるものであると考えられる。ここで知識をもった多くの人たちが交わり、それを知的プロセスが支援するようにしたい。

オントロジーは、本来生活に密着した具体的な事物から発生するものであろう。そうした素朴な世界観を体系化するプロセスを確立することが重要であろう。

2.5.2　人間の知識共有支援とCoMeMoアーキテクチャ

人々の間のインタラクションの支援を目標としたCoMeMoプロジェクトでは、人々に共有される暗黙知や文脈を最大限に利用することをねらっている。日常記憶の構築・共有支援システムを設計する上で、実生活環境における暗黙知や文脈を情報科学の視点から解明することは非常に重要である。CoMeMoプロジェクトでは、人間のインタラクション支援システムを研究開発し、評価をしてもらうことによって、暗黙知と形式知の相互作用、文脈と状況の認知と生成のメカニズムを実証的に解明することを目指している。

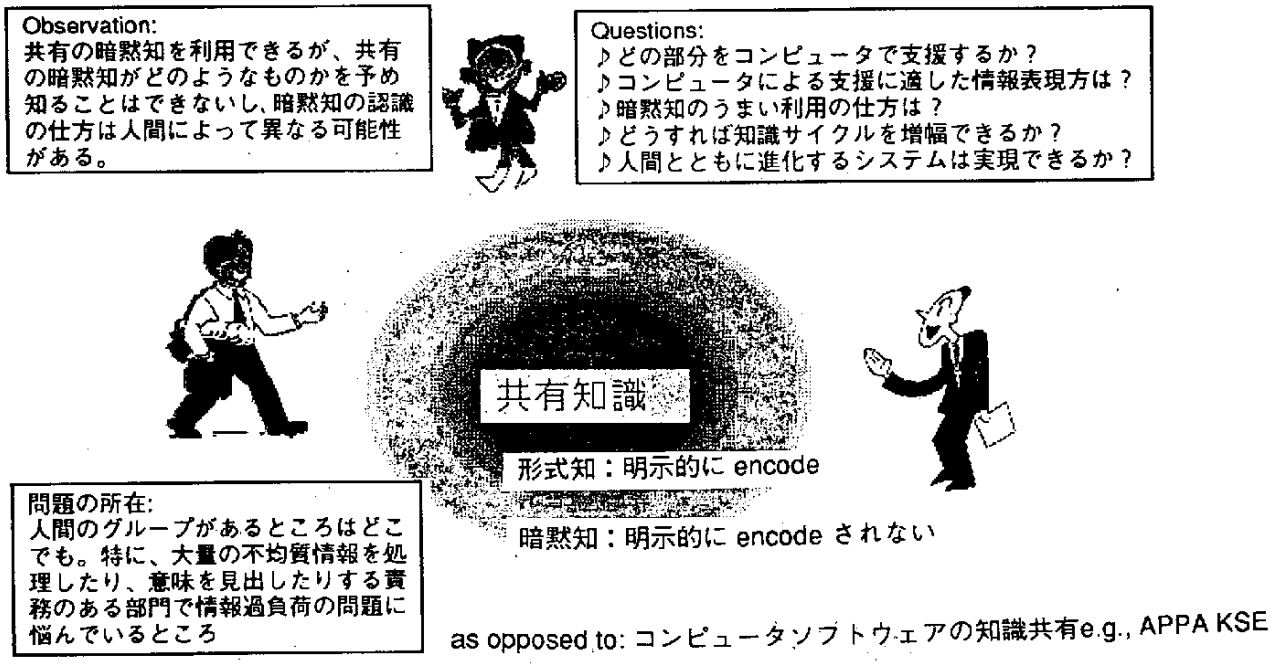


図2.5-2　人間の知識共有

人間の知識共有の問題は我々のまわりには遍在する（図２.５-２）。コンピュータ向きの知識共有と比較すると、人間の背景知識を利用できるので、人間が解釈できるところは記述する必要はなく、受け手が分かる程度の情報を明示的に記述すればよい反面、背景知識はまだ詳細に理解されていないところがあるので、誤解や意図伝達不能などの問題が生じる。コンピュータにあらゆる情報を与え、活用できるようにすることが不可能である以上、豊富な知識を扱おうとすると知識を解釈する人間の背景知識に委ねざるを得ない。

CoMeMoプロジェクトでは、大量の不均質情報を扱うためには弱い情報構造を用いたゆるやかな情報の関連付けが有効であるという作業仮説をたてている。従来の人工知能では、よく定義された一種の情報表現言語というのばかり考えて、何とかして自然言語テキストやイメージを計算機向きメディアに記述しようとしてきた。そのような目標の達成は非常に困難であるばかりでなく、変換の段階で重要な情報が落ちてしまう。

たとえば、「場」の情報というのが落ちたまま音声データをテキストに変換してしまうと、その場の雰囲気や発言に関わる状況的な情報は落ちてしまう。自然言語のような人間向き情報メディアか人工言語のようなコンピュータ向き情報メディアか、というような問題設定をしないで、「結びつけること」という視点から取り組む方が有益ではないだろうか。

情報の厳密な記述、完全な記述をすることを目指すのをやめて、例えば「奈良と桜と吉野山という間に何らかの関連がある」という縮退した情報だけを使うアプローチを提案する（図２.５-３）。これによって、情報構造をまだ的確に与えられない生データ、すなわち未解釈情報、未分節情報がさまざまに活用できることが期待される。

このようなアプローチは今のコンピュータサイエンスの主流（よく定義されたものだけを使う）とは逆行するものであるが、敢えてその有用性を再評価する価値があると考えられる。

工学には、機械（＝コンピュータ）をツールとして利用する場合、機械は人間に言われたとおりのことをすればいいのであって、あまり知的なことはしない方がよい、という意見が根強く存在する。ここで目指すところは、機械と人間をあわせた系の知的創造性を高めることである。機械があまりきっちりしたことをしなくても、人間の知的活動の大きな刺激になるのかもしれない。

従来の知識表現を厳密で形式的に定義しようという方向に対して、ここでは表現の意味を漠然としか定義せず、表現の意味解釈を書き手と受け手にまかせようというア

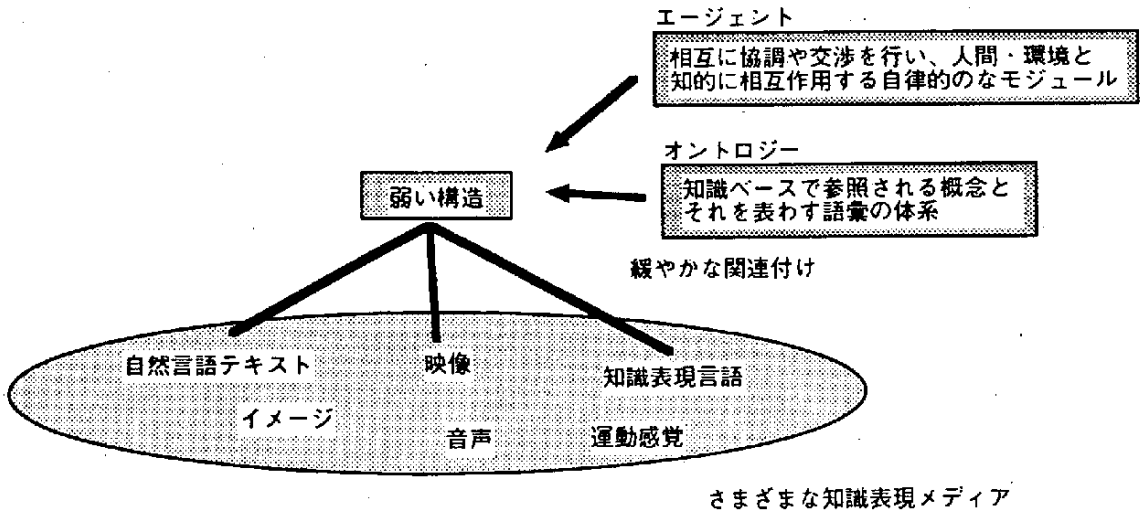


図2.5-3　弱い情報構造を用いたゆるやかな情報の関連付け

プローチをとっている。こうすることによって、マルチメディアデータ、ちょっとした思いつきやメモ、精度の十分でない自然言語処理システムの抽出データなど、広範な情報を手早く入力しやすくなる。CoMeMoアーキテクチャは収集された断片的な情報の共有と活用を促進するためのものである。

現在注目しているのは、図2.5-4のように「何が与えられると何が想起されるか」という連想関係を用いた情報表現である。結合されるものはユニットと呼ばれ、概念シンボル、テキスト、イメージデータなどに対応する。

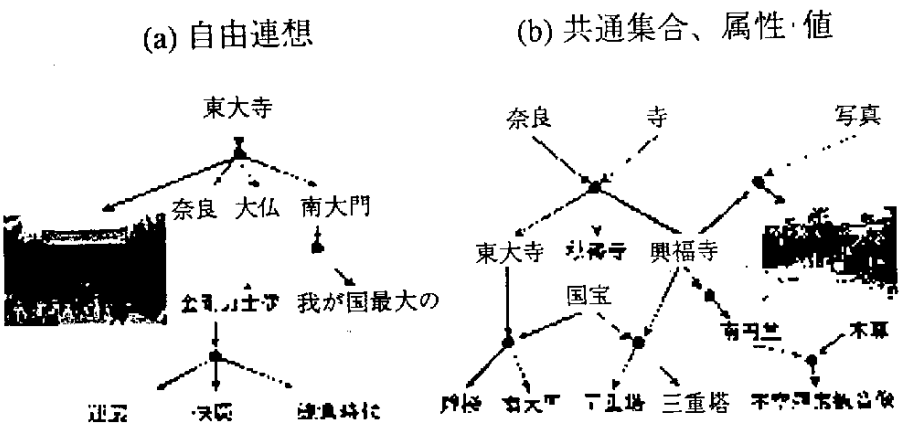


図2.5-4　連想関係による典型的な情報表現の例

このような連想構造を一定の視点から集積したものをワークスペースと呼ぶ。CoMeMo情報ベースはこうしたワークスペースを集めたものである（図2.5-5）。

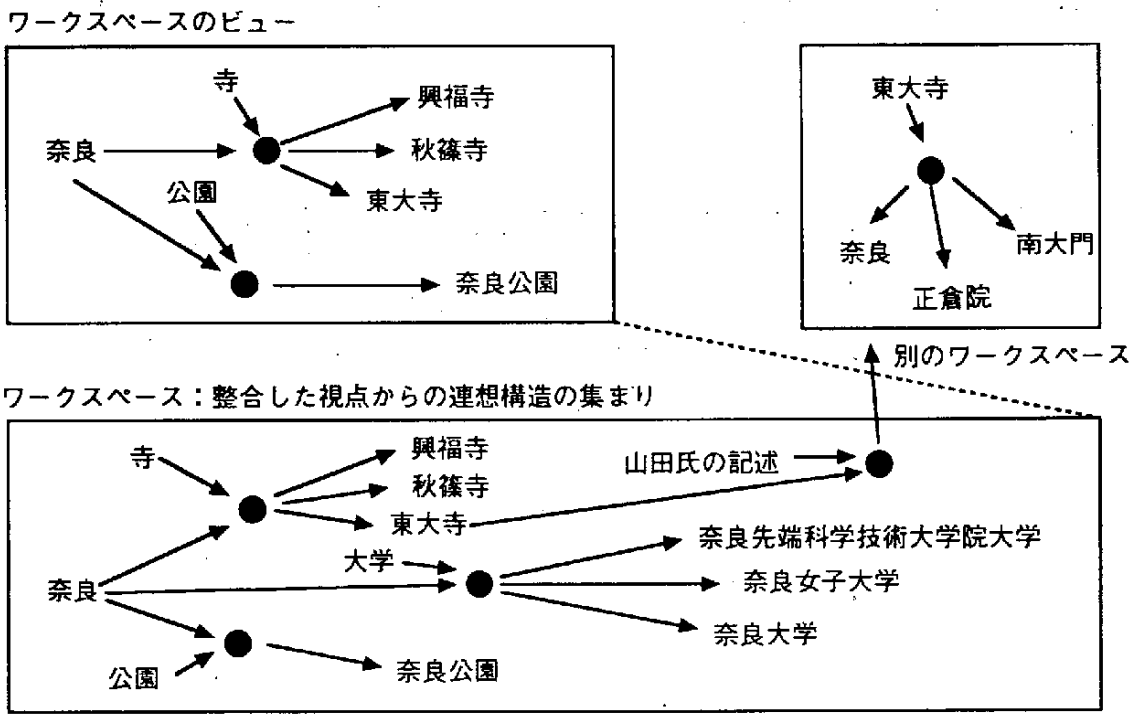


図2.5-5　情報ベースはワークスペースの集まり

このような枠組みの中で人間の連想がどのような形で現れるかは興味ある研究テーマである。

CoMeMoアーキテクチャは、図2.5-6に示すように、

1. CoMeMoプロトコル：CoMeMoアーキテクチャの構成要素の間の情報交換のためのプロトコル。
2. CoMeMoワークベンチ：弱い情報構造による情報ベースの探訪、ブラウジング、編集を支援する。
3. CoMeMoモバイル端末：CoMeMoワークベンチをモバイル端末上に実装し、携帯可能にしたもの。問題の発生する現場での情報入力と利用を可能にする。
4. CoMeMoインコーポレータ：さまざまな情報源からの情報をCoMeMoアーキテクチャが取り扱い可能な形式に変換する。

5. CoMeMoサーバ：弱い情報構造による情報表現を蓄積し、必要に応じて提供する。
また、オントロジーを用いて情報提供者と利用者の間の仲介を行う。
6. CoMeMo エージェント：(a) ユーザの委託を受けて、処理を非同期に行う代理機能、(b) エージェント同士で共通の問題を解決する交渉機能、(c) ユーザに対して CoMeMo情報ベースの内容を提示するエージェント指向インタフェース機能をもつ。

から構成される。

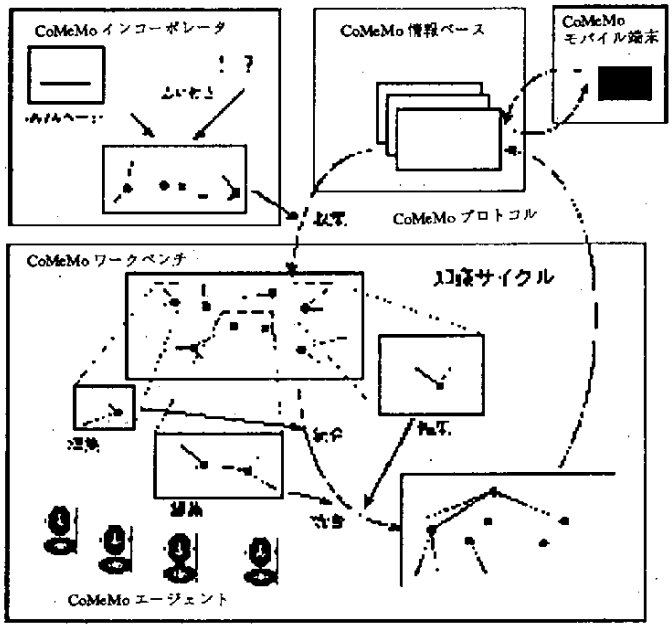


図2.5-6：CoMeMoアーキテクチャ

2.5.3 CoMeMoによるボトムアップな知識・オントロジー構築の支援

CoMeMoを用いた、インターネットなどの多様な情報源からのボトムアップな知識・オントロジー構築の支援システム（図2.5-7）のプロトタイプのかなりの部分を構築した。

CoMeMoは種々の情報源からの関連する情報をまとめて、構造化する過程を支援する。

CoMeMoによる支援は

1. 情報キャプチャ：HTML文書や自然言語テキストなどからの連想構造生成

2．視点変換：連想の視点を変換し、キーと値を適当に入れ替える。
3．異なる情報ベースからの連想構造の統合。
4．連想構造の整形：観点の一定しないと考えられる連想構造を修正する。
5．強構造の示唆。

から成る。CoMeMoは情報の意味は理解しないが、構造化を支援するプログラムは与えられた連想構造における情報の意味をヒューリスティックを用いて抽出することを試みる。

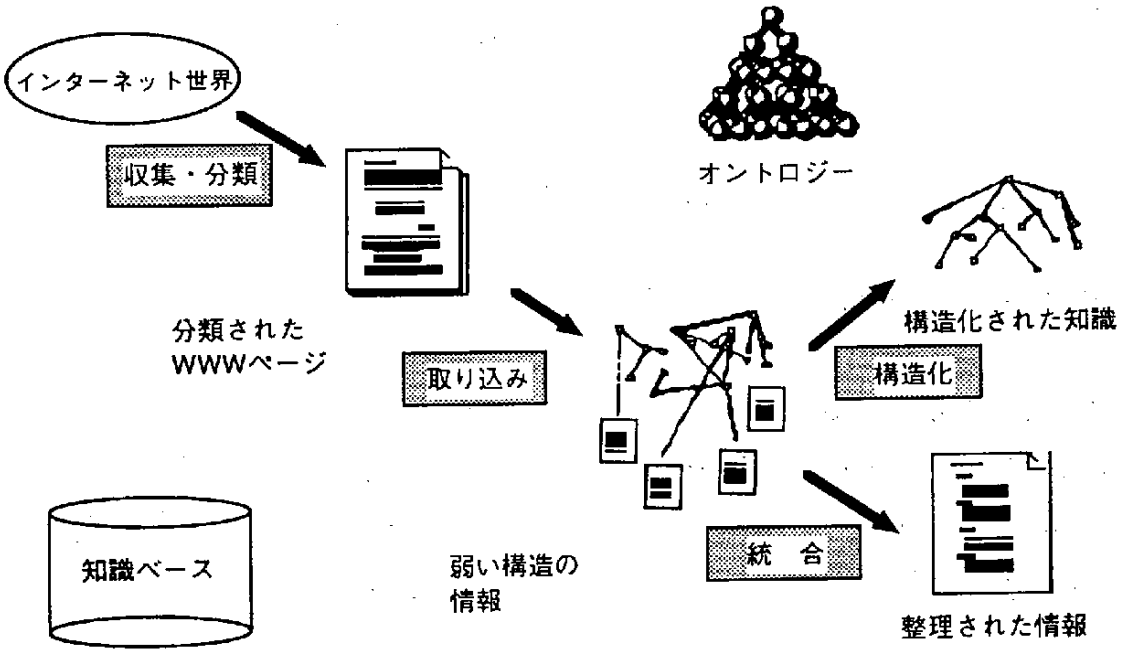


図2.5-7　ボトムアップのオントロジー構築の支援

2.5.3.1　情報キャプチャ

完全な自然言語処理は期待できないので、形態素解析によってソースからキーワードを切り出し、HTML文書の構造などを手がかりに連想構造を生成し、ヒューリスティックなどを用いて整形する（図2.5-8）。

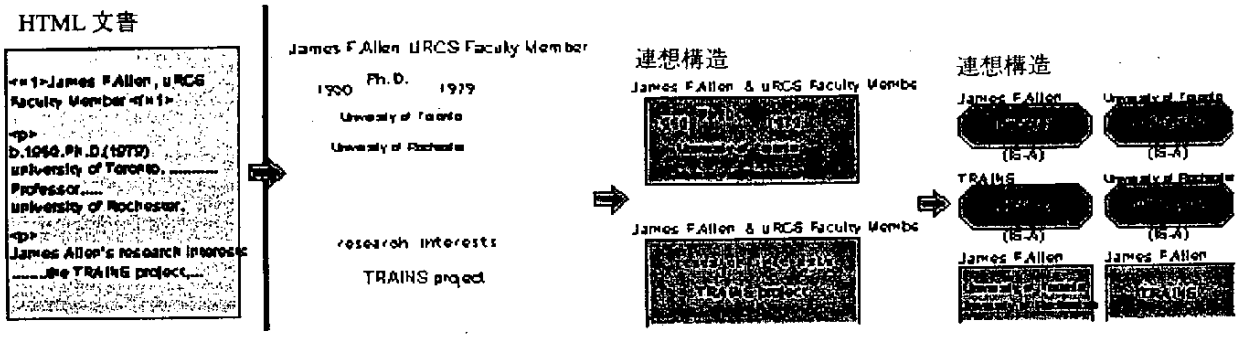


図２.５-８　情報キャプチャ

２.５.３.２　視点変換

与えられた視点に基づいて連想構造を再構造化する。連想検索を用いて実現する。例えば、図２.５-９中央の連想構造は「James F. Allen」と「プロジェクト」の直接的な連想関係を明示したものではないが、この二つのキーワードによる連想検索を行うことによって、構造の中に暗黙的に含まれている連想関係を抽出し、明示している。

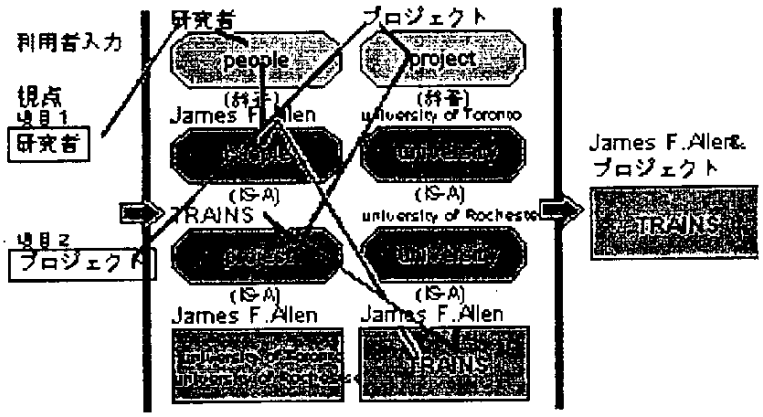


図２.５-９　視点変換

2.5.3.3　異なる情報ベースからの連想構造の統合

複数の情報ベースに含まれる連想構造を統合することによって、一様な視点からの包括的な連想構造の集まりを生成する。例えば、図2.5-10上の3つの連想構造を統合すると図2.5-11下のような連想構造を生成する。視点変換をしてから、連想構造のマージを行っている。

連想構造の間で語彙の使い方に一貫性がない場合はあらかじめ語彙の整合を行う必要があるが、本研究ではこの問題に対する本格的な取り組みはまだ行っていない。

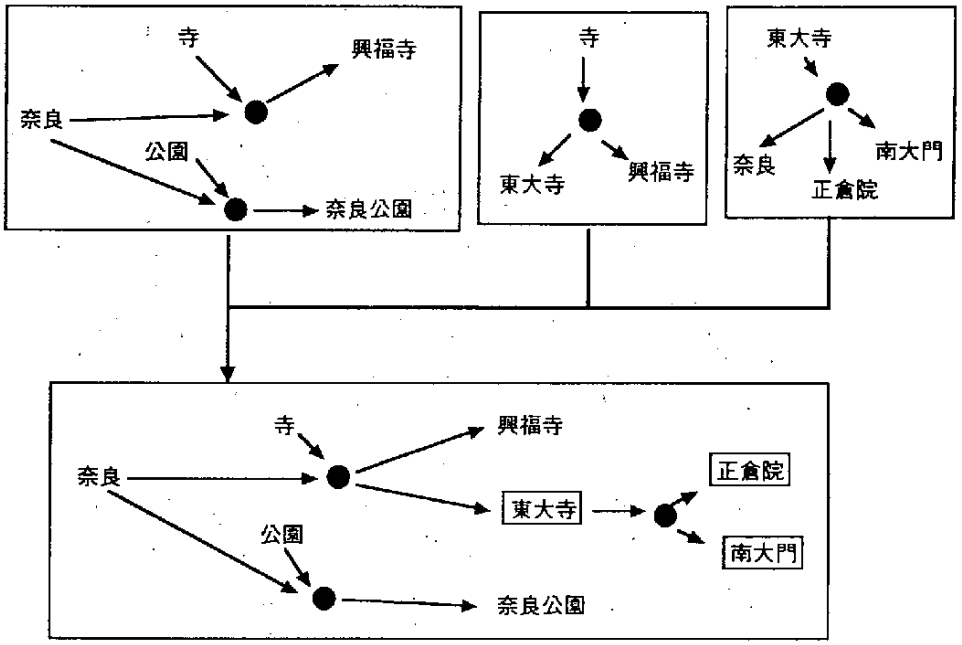


図2.5-10　連想構造の統合

2.5.3.4　連想構造の整形

このフェーズでは、与えられた連想構造の集まりの観点のばらつきをヒューリスティックを用いて補正する。詳細度が異なる情報は詳細な方にあわせる方向に補正する。例えば、図2.5-11は直交分解と類推的詳細化とよぶヒューリスティックの適用例を示す。第1ステップで「生駒山上遊園地」と「花」の両方から連想されている「桜」を認識し、それが「生駒山上遊園地」と「桜」の2つをキーとする連想構造として明示している。第2ステップで、与えられた情報ベースにおける「あやめ池遊園地」と

「生駒山上遊園地」の挙動の類似性と、「ハナショウブ」と「桜」の類似性に基づいて、「ハナショウブ」を「あやめ池遊園地」と「花」をキーとする連想構造の値として推定している。

はじめは、「あやめ池遊園地」と「ハナショウブ」の間に単なる連想関係があることしか示されていなかったが、連想構造の整形によって両者の間の関係に関する情報が詳細化されている。

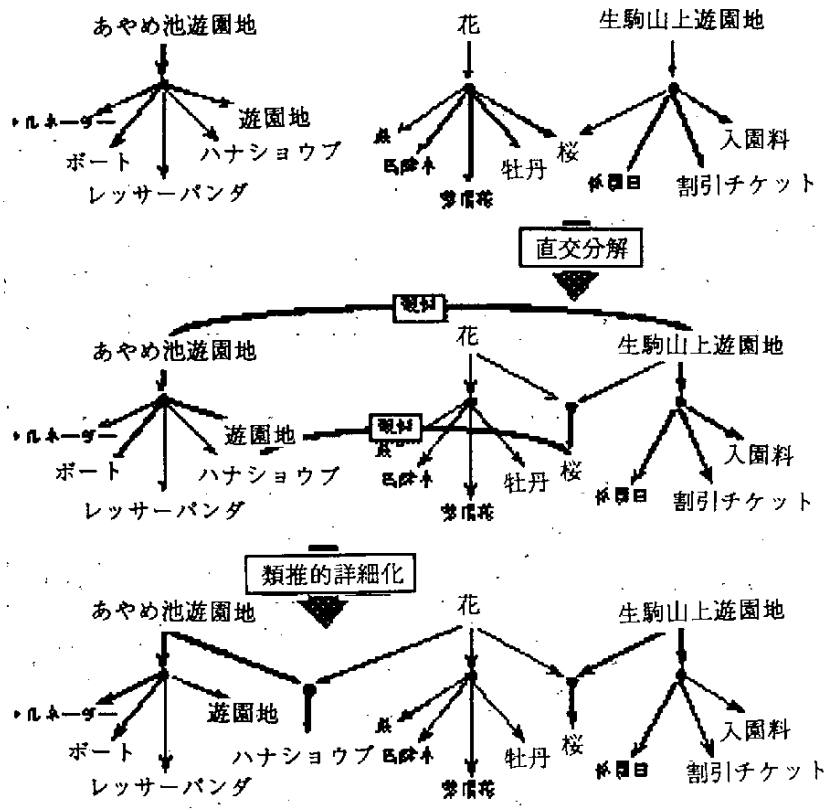


図2.5-11　ヒューリスティックを用いた連想構造の整形

2.5.3.5　強構造の生成

以上のように収集・整理された連想構造に意味付けを行って強構造を生成する。強構造の生成は、強構造の抽出と一般化から成る。図2.5-12では、「奈良」を主要概念、「寺」や「公園」を属性としている。今のところどれを主要概念とし、どれを属性概念とするかの決定や、暗黙的な属性（例えば、「部分」）の挿入は人間が行うものとしている。

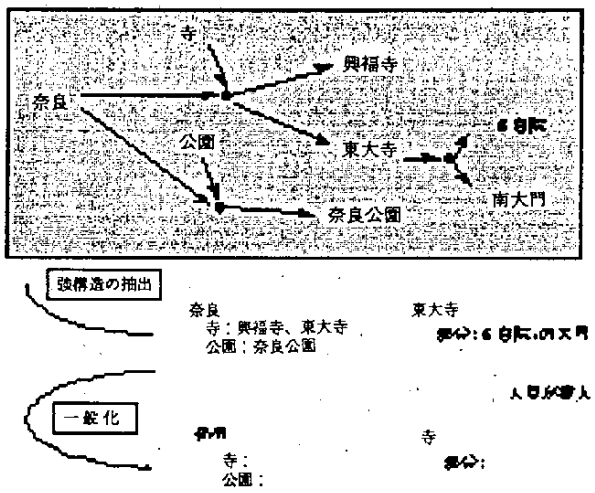


図２．５-１２　強構造の生成

2.6 知識ベースシステム開発におけるオントロジーの役割

2.6.1 はじめに

知識ベースシステムは、従来、知識がゼロの状態から再開発されてきた。似た分野のシステム開発にあたっても、以前に開発したシステムの知識ベースを再利用することはされていなかった。今後、知識の共有と再利用を促進するためには、知識表現、推論手法、さらには開発方法論を含めた知識ベース技術の標準化が重要な問題となる。しかしながら、知識ベース技術に関しては、人工知能技術の進歩が非常に速い、圧倒的なシェアをもつ企業が存在しないなどの理由から標準化の動きは比較的少なかった。ところが、最近、米国やヨーロッパでは、知識ベース技術の標準化に関する活動がさまざまなかたちで行なわれている。この1つの柱が、オントロジーに関する研究である。本稿では、このような活動の背景について分析を加えるとともに、代表的な知識ベース技術標準化の動向と現状とそれらにおけるオントロジーの役割について考察する。

2.6.2 標準化活動の背景とオントロジー

コンピュータ関連の産業においては、製品そのものよりも技術について標準化がなされる。そのため技術の実際の普及に先立ち、市場が主導的な役割をはたし、開かれた標準が、さまざまな会合でコンセンサスを得て、事実上作られていくことが多い[Hall91a]。そしてある程度のコンセンサスができた時点で、それを公的・法律的にオーソライズする手続きがなされる。これには一般に非常に時間がかかる［ANSI/IEEE 88］，［情報処理 93］。

知識ベース技術に関する標準化活動は1990年代にはいってから開始されたばかりである。それに参画するメンバーの立場も次のようにさまざまなものがある。

2.6.2.1 知識ベースシステム利用者の立場

システム利用者の立場からは、AIシステムは問題解決システムとしての性能が高いこと、さらには、システムの移植性や保守性、あるいは、他システムとの統合性が重要である。人工知能理論や知識ベース技術を表に出す必要はない［寺野 93b］。この

ような立場からは、利用者が意識しないですむような標準化はきわめて望ましい。このような意識は、さまざまなレベルの応用システムを利用する大規模ユーザにとって強いものとなる。共通に使える概念については機械部品の規格のようにオーソライズされた体系がドメインオントロジーとして整備されることが望まれる。

このような要求がとくに強いのは米国の国防省（DOD）や航空宇宙局（NASA）である。このような点から、システム利用者にとっては、AIシステム評価、とくに、システムの検証・妥当性検査［寺野92］，［寺野93a］を重視すると同時に、他システムとの統合を意識した標準化活動が望まれる。この意味では、従来型ソフトウェア開発においても、ドメインの用語が整備されていない以上、知識ベースシステムを単独に取り上げて（ドメイン）オントロジーの問題を議論することは、現状ではあまり意味がない。

2.6.2.2 知識ベースシステム開発者の立場

システム開発者の立場からは、知識ベースシステムに対する開発要求の増加に対するシステム開発コストの低減が標準化への動機である。とくに、知識ベースシステム開発ツール・言語、ならびに利用者やシステム間のインタフェースに対する標準化の要望は強い。この点からは、タスクオントロジーの概念は重要であり、タスクを限れば充分に実用的なオントロジーが整備でき、その利用を前提とした、オントロジーエンジニアリングに基づくシステム開発方法論を構築していける可能性はある。

しかし、知識ベースシステムを真に普及させようとするか、また、それを利用して市場の支配をねらうかの立場の違いは表面的な活動をみているだけではなかなかわからない。前者の面からは、軍やNASA、エネルギー産業などに大規模システムを納入しているソフトウェア開発会社において、知識ベースシステム開発の必要性が高まっているにもかかわらず、その開発・評価基準が定まっていない現状に対し、不満・不安が高まっている。また、後者の面からは、標準化活動にあわせて新たなツールを開発し、それをビジネスに直結させたい、ツールベンダー・ユーザのグループが存在することが上げられる。たとえば、後述するスタンフォード大学KSLが提供するオントロジーサーバには、彼等が開発に直接携わったシステムの視点からの情報は豊富だが、知識の共有・再利用の点からはまだまだ不十分なものである。

2.6.2.3 AI研究者の立場

知識ベース開発に必要な知識表現・推論機構などの要素技術は、最近１０年あまりのAIブームを経て成熟の域に達している。このような状況のもとで、AI研究者の間で今後の知識ベース関連分野の研究の主導権をにぎろうとする動きが活発になるのは当然のことといえよう。そのような活動の１つとして知識ベースの共有・再利用は大きなテーマとなってきている［ASTEM 92a］，［ASTEM92b］。今年度の研究委員会の活動も当然この一環である。現在のところは、各分野ごとの参加者の間の概念の共有化が、主な成果になっており今後の方向性を明確にすることが重要であろう。

2．6．3 標準化活動の動向とオントロジー

知識ベース技術に関連する標準化活動は、大きく、1) 知識の変換・再利用を重視した標準化活動、2) 対象領域を特化した標準化活動、3) 標準化を支援する活動、4) 知識ベース技術そのものではないが関連の深い従来型ソフトウェアに対する標準化活動の４つに分類することができる。最初の知識の変換・再利用の面からの標準化活動は、さらに、1-1) 問題解決・オントロジーレベルなどの知識レベルの標準化、1-2) 強力かつ標準的な言語を設定しようという知識表現言語レベルの標準化、1-3) 知識ベース間の変換を支援するための適当な中間言語を設定しようとする知識変換レベルの標準化、1-4) 知識ベースシステム間の情報交換を支援するための質問言語・通信規約の標準化活動にわけることができる。

2.6.3.1 KSE（Knowledge Sharing Effort）［Neches 1991］

米国国防省が主なスポンサーとなり、USCならびにStanford大学の研究者が中心となって知識の共有化（Knowledge Sharing）をめざして実施している研究活動である。知識（ベース）の交換を促進する形式（KIF; Knowledge Interchange Format）、中間言語（Interlingua）の設計・開発、知識システムの共通仕様の定義、外部インタフェースの定義、共有・再利用可能なオントロジーの記述（Ontolingua）［Gruber 93］と、それを用いる分散知識ベースシステムの開発（PACT）［Cutkosky 93］の研究を行なっている。知識の共有・再利用を目的とするプロジェクトには、この４つの問題をとりあげているものが多い。

Ontolinguiaについては、オントロジーの意味論が不十分である、操作性の概念に欠けるなどの理由で問題があるとされている。しかし、ここで提供されているオントロジーサーバでは、Ontolinguaで記述したオントロジーの追加、編集、修正などの機能をWWW上で公開している。これを利用したプロジェクトはWEB上のスマートカタログを対象とするCommerceNet、企業モデルの開発を支援するEnterprise Project、医療情報処理を対象とするInterMed、医療試験データの知識共有をねらいとするTrial Bank、インターネット上の製品、サービスに関する情報を対象とするNetwork-based Information Broker、会計情報システムなどさまざまな分野に適用されており、普及が進んでいるオントロジーの１つである。

2.6.3.2 Common KADS Library [Breuker 93]

Common KADSは、ヨーロッパESPRITにおいて進行中の知識ベース開発の方法論である。Common KADS Libraryは、Common KADSで使われる問題解決方法（Problem Solving Methods; PSMs）を問題型、制御構造、推論構造などの概念とともに、And-ORネットワークとして構造化したものである。これによって再利用可能な知識ベースシステムの構成要素が領域オントロジーとネットワーク上の演算に基づいて定義される。PSMはさまざまな立場から作られ、また、利用される。このような枠組みにより、知識ベースの共有・再利用を促進しようという考えである。Common KADSに基づく実現モデルはこの他にもいくつか提案されている［Voss 93］。

2.6.3.3 EuroKnowledge [Georges 93]

EuroKnowledgeでは、知識ベース技術の発展を加速するために、知識表現標準を提案し、その効果を評価し普及を助けるソフトウェアの開発を目指している。これは、３年間のプロジェクトとして本年度から開始されたプロジェクトで、現在の技術レベルにおける知識技術のビジョンを明確化し、それに対する標準化の要求を同定するとともに、供給者、ユーザ、研究者間に必要とされる協調関係を志向している。

KSEと同様に、知識再利用を重要なテーマとしており、既存知識の変換、知識ベースシステムと既存システムの統合化、知識レベル表現・ソースレベル知識表現の形式化、ファイル変換フォーマット、質問言語・通信規約の標準化を項目にかかげている。

2.6.3.4 IEEE P1252 (IMKA; Initiative for Managing Knowledge Asset) [IMKA 90]

これは、Carnegie Groupが、DEC、TI、Ford、US West各社とともに始めた活動である。現在のAIツールの欠点を克服し、特に、統合型知識システムで、高速かつ柔軟に利用できるようなツールの仕様をまとめ、その実現を目指している。第1フェーズで、ツール仕様を決定し、第2フェーズでは、知識検索操作のための標準言語の仕様を定めるとともに、データベース検索言語SQLとのインタフェースを定義するための活動を行っている。IMKAは、プロジェクト開始当初は、マルチクライアント・プロジェクトであったが、これが、現在IEEEの標準化活動の1つになってきている。ここでは、知識表現言語そのものが標準化の対象となっており、オントロジーを重視するという思想はない。

2.6.4 標準化活動の問題点に関する考察

標準化へむけての活動が盛んになることは好ましいが、その一方で、標準化活動の数ばかりが増加するのはいたずらに知識ベース技術に混乱を引き起こすのみである。その意味で、これらの標準化活動に対してかなり強行な反対意見も見られる。知識共有・再利用をねらいとする標準化活動の問題点については以下のような問題が存在する。

知識レベルの標準化については、対象領域知識と問題解決知識をともに、明確なオントロジーとして記述しなければならない［ASTEM 92a］。これは、対象の存在とその認識とに関する哲学的な議論を要する問題である。したがって、タスクや領域を限っても明確な定義にいたるには膨大な時間がかかるし、仮に明確な定義を得たとしても、現在の知識表現手段の未熟さから、その実現には膨大かつ複雑なレベルの作業が不可欠である。この典型例として、Cyc［Lenat 90］において時間を複数の方法で表現し実現したことが上げられる。

知識表現言語レベルの標準化活動については、理論的な立場からは一階述語論理とフレームをベースにもつKL-oneタイプの言語が注目を集めている［SIGART 91］。しかし、このような言語族には事実上の標準は存在しない。ここで、ツール提供者の間での確執が生ずる可能性が高い。さらに、最近の知識表現研究［Brachman 92］で明らかになったように、これらの言語には本質的に知識の表現力と推論の計算量の間のトレードオフ問題が存在する。たとえば、フレーム表現に関連するConcept Languageある

いはTerminological Logicの研究では、効率良く推論できる言語のクラスが非常に狭いことが明らかになっている。

知識変換レベルの標準化についても、知識表現分野の共通の問題が存在する。研究のレベルでは、変換のための知識表現に対する構文的・形式的な議論が重視され、その意味の問題が無視されがちである。しかし知識のもつ意味を保存しながら知識を変換する方法について我々は十分な知見を持ち合せていない。

質問言語や通信規約の標準化についても、知識表現言語で述べたのと同様な、表現力と計算量のトレードオフ問題が生ずる。一見すると、Interlinguaのような質問言語は記述力が高いほうが望ましいように思えるが、ある知識ベースの語彙を質問言語に変換するには高速な知識処理が不可欠であり、それには表現言語の単純さが必要となる。また、各種の表現メディアや従来型システムとの統合のためのインタフェース規約も重要である［Gaines 91］。

一方、応用分野に特化した標準化については、今後ともさまざまな活動が行われるようになると考えられる。この際の課題としては適切な分野を選択し、重複なく、開かれた標準を設定することが上げられる。

従来型システムを含む標準化活動全般については、オブジェクト指向データベースや質問言語などに見られるデータモデルやSQLの標準化、ドキュメント処理におけるSGML［Herwijnen 90］の活動、通信規約におけるTCP-IPやOSIモデル、また、PCTE［Thomas 89］などのソフトウェア開発環境の標準化活動と協調しながら、開かれた標準を目指していく必要があろう。

参考文献

[AAAI-STD 91] Wright, R.G., (ed.): Working Notes on the AAAI Workshop on Standards in Expert Systems. 1991.

[ANSI/IEEE 88] 東基衛: ANSI/IEEEソフトウェア規格集, 日本規格協会 (1988).

[ASTEM 92a] ASTEM: エキスパートシステム構築支援のための大規模知識ベース構築を目指して. ASTEM技術報告 TR-P-048-92, (1992).

[ASTEM 92b] ASTEM: 大規模知識ベース方法論. ASTEM技術報告 TR-P-049-92 (1992).

[Breuker 93] Breuker,J.: Reusable Components for Artificial Problem Solvers: the Common KADS Library Experience. in [IJCAI 93], pp.36-60, 1993.

[Brachman 92] Bracjman, R.J., Levesque, H.J., Reiter, R. (eds.): Knowledge Representation.

MIT/Elsevier (1992).

[Cutkosky 93] Cutkosky,M.R., et al.: PACT: An Experiment in Integrating Concurrent Software Engineering. IEEE Computer, Vol.26, No.1, pp. 28-37 (1993).

[Gaines 91] Gains, B.R., Boose, J.H.: Standards Requirements, Sources and Feasibility in Knowledge Acquisition. in [AAAI-Std 91] pp. 5-10 (1991).

[Georges 93] Georges,M., Goodall,A.: Euroknowlege.in [IJCAI 93], pp.36-60, 1993.

[Gruber 93] Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Stanford University Knowledge Systems Laboratory Technical Report KSL 93-04. (1993).

[Hall 91a] Hall, C.F.: The AIAA Committee on Standards for Artificial Intelligence. in [AAAI-Std 91] pp. 47-49 (1991).

[Hall 91b] Hall, C.F.: Why Standards?: The CAIA'91 Workshop on Standards for Artificial Intelligence. IEEE Expert, Vol.6, No.6, pp. 49-63 (1991).

[Herwijnen 90] Herwijnen, E.V.: Practical SGML.Kluwer, 1990. (SGML懇談会実用化WG (監訳)：実践SGML. 日本規格協会, 1992.)

[Hori 92] Hori, M., Nakamura, Y., Hama, T.: Methodology for Configuring Scheduling Engines with Task-Specific Components. Proc. JKAW'92, pp. 215-230 (1992).

[IJCAI 93] Workshop Notes of IJCAI-93 Workshop on Knowledge Sharing and Information Interchange. (1993).

[IMKA 90] Carnegie Group, et al.: IMKA Software Specification Phase 1, Knowledge Representation. 1990.

[情報処理 93] 情報規格調査会: 情報技術の国際標準化と日本の対応－1992年度の情報規格調査会の活動－. 情報処理, Vol.34, No.9, pp. 1204-1212 (1993).

[機械振興協会 92] 機械振興協会: 知識アーカイブ研究開発計画－大規模知識獲得支援ツール (VLKB)に関する調査研究報告書. 1993.

[小林 90] 小林重信, 寺野隆雄（編）：知識システムハンドブック, オーム社, (1990).

[Lenat 90] Lenat, D.B., Guha, R.V.:Building Large Knowledge Based Systems. Addison-Wesley, 1990.

[Mizoguchi 92] Mizoguchi, R., Tijerino, Y., Ikeda, M.: Task Ontology and Its Use in a Task Analysis Interview System - Two-Level Mediating Representation in MULTIS -. Proc. JKAW'92, pp. 185-198 (1992).

[Neches 91] Neches, R. et al.: Enabling Technology for Knowledge Sharing. AI Magazine, Vol.

12, No.3, pp.36-56, 1991.

[西田 92] 西田豊明: 大規模知識ベースへの試案. 電子情報通信学会AI研究会／人工知能学会KBS研究会資料, 1992.

[Scheibenzuber 91] Scheibenzuber, H.: Expert System Standards for Test and Diagnosis. in [AAAI-Std 91] pp. 45-46 (1991).

[SIGART 91] SIGART BULLETIN: Special Issue on Implemented Knowledge Representation and Reasoning Systems.Vol.2, No.3, 1991.

[寺野 92] 寺野隆雄（編）：エキスパートシステム評価マニュアル, オーム社, (1992).

[寺野 93a] 寺野隆雄: 大規模エキスパートシステムの評価について, 人工知能学会誌, Vol.8, No.1, pp. 8-16 (1993).

[寺野 93b] 寺野隆雄: 知識システム開発方法論, 朝倉書店, (1993).

[Thomas 89] Thomas,M.I.: PCTE Interfaces: Supporting Tools in Software Engineering Environments. IEEE Software, Vol.6, No.6, 1989.

[White 90] White, M., Goldsmith, J. (eds.): Standards and Review Manual for Certification in Knowledge Engineering. Systemsware-Corp., 1990.

[Wright 91] Wright, R.G.: An Overview of Expert System Standards Activities. Proc. the World Congress on Expert Systems. pp.656-659, 1991.

[Voss 93] Voss, A., Karbach, W.: Implementing KADS Expertise Models with Model-K.IEEE Expert, Vol.8, No.4, pp.74-81 (1993).

2.7 知識システム構築とオントロジー工学

これまでの知識工学研究において問題解決知識の断片的共通性については多くの議論がなされ、部品合成による知識システム構築の可能性は示されたといえる。しかしながら、実用レベルでの再利用を考えた場合、当初想定されていなかった適用領域の境界が顕在化したり［Winograd 86］、周辺環境との相互作用を通じて境界領域自身が変化する［Brown 94］といった状況に直面することが予想される。このような知識の文脈依存性は知識システム単体で考えれば再利用性や保守性の阻害要因となる。ところが、システムの利用者／管理者／開発者として関与する人間や相互運用されるシステムを包含する範囲でとらえれば、当該システムの開発と運用を通して提供された知識がその適用領域を明確化あるいは拡大することによって進化する過程として捉えることができる［堀 97］。

知識システムの開発／運用過程において見られる相互作用は包括的に解消されるものではなく、現実として受け入れた上で様々なトレードオフの中で均衡点を見い出す必要がある［Hori 97a］。知識の文脈依存性を考えれば、そのようなトレードオフや均衡点も適用環境において恒常的であり得ない。したがって、知識システム構築方法を体系化し前提知識として提供されるオントロジーを整備していくには、適用環境との相互作用を踏まえて知識システムのライフサイクルを見直すとともに実用レベルでの知識再利用を通して実証的研究を積み重ねていくことが不可欠である。

知識システムの構築にあたって推論機構と領域知識の役割を区分することの意義は周知の通りであり、タスクとドメインの観点はオントロジー設計への指針として有用である。しかしながら、知識の表現はその利用手順と独立ではなく［Bylander 87］、タスクとドメインに関わるオントロジー相互の依存関係についても議論がなされている［Gennari 94, 高岡 95, 堀 95, van Heijst 97］。

本報告では再利用性と相互運用性の２つの軸に着目することによって、知識システム構築におけるオントロジーの設計と利用に関わるトレードオフについて論じる。

問題解決知識を特定の対象領域に限らず広範囲で再利用することを考えた場合、タスクの観点から汎化された抽象レベルで問題解決過程をとらえることは有用である。一方、特定の対象領域において相互に連携しながら協調する知識システム／情報システム／現場担当者間の相互作用まで考えた場合、複数のタスクがどのように関連し合うかを明示的にするためにドメインの切口から対象領域をモデル化することができる。言い換えれば、前者は問題解決知識の再利用性を重視しタスクの切口から様々なドメ

インをカバーすることを目指すアプローチであるのに対して、後者は相互運用性に重きを置くことによって特定ドメインにおける複数タスクが連携する文脈を規定するものであると言うことができる。もちろん、これら２つのアプローチは独立ではなく、再利用性と相互運用性についてどちらを重視するかによって様々なアプローチが可能である。

80年代の知識システムは「孤立した一枚岩」(isolated monolith)であったと言われる。従来の知識システムはモジュール性に欠け機能の断片的な再利用が困難なだけでなく、関連システムとの連携も考慮されていなかったために、再利用と相互運用のいずれもが容易でなかった。一枚岩であることを克服するために問題解決知識のモジュール化／部品化が試みられてきた。領域固有の性質をタスクの観点から汎化し細粒度の構成要素を提供することによって広範囲での再利用性は期待できるが、それによって特定の対象領域において連携する複数タスクの関連について領域固有の前提を明示的にすることは困難となる。個々のタスクを中心としたアプローチには、対象領域からの独立性を重視したもの［Tijerino 93, Benjamins 93, Hori 94］、対象領域との依存関係を考慮したもの［Gennari 94, 高岡 95, Heijst 97］、さらにリモートサイト間での問題解決手続きと領域知識の相互運用を目指したもの［Gennari 96］などがある。特に、分散環境での相互運用では相互にコミュニケーションを図るエージェント間で構文レベルでの合意が必要であり、KQMLやCORBA IDLはそのためのコミュニケーション・オントロジー［溝口 97］としての役割を果たす。そこで、コミュニケーションに介在するエージェントが前提とする知識内容については別途規定する必要がある。

それに対して、対象領域のモデルを前提としてその領域内での複数タスク間の関連を規定することによって、データやサービスの授受を相互に図る複数のエージェントに対して共通の利用文脈を与えることができる。これは再利用範囲を狭めることと引き換えに、データやサービスの授受に介在する要求者と提供者間での相互作用が「意味をなす」(make sense)こと、すなわち意味レベルでの相互運用性を保証するための前提を明示的にするアプローチであると言える。特定の対象領域を設定しその上で複数タスク間の連携を扱ったものには［Laresgoiti 96, Hori 97b］などがある。

筆者等はこれまでスケジューリング問題をジョブ割付問題として定式化し、部品合成によるプロトタイプ開発環境の構築［Hori 94］、ならびに問題解決手続きの部品化に取り組んできた[Hori 95]。ジョブ割付問題は、旅客機の搭乗ゲートへの割付、製鋼工程の生産計画、レストランでの人員配置など様々な分野におけるスケジューリングを対象とするもので、これらの分野に共通する断片的な機能単位が問題解決部品とし

て抽出された。当初はスケジューリング問題全般に対して対象領域固有の性質を汎化することによって、製鋼工程の生産計画やサービス業での人員配置など様々な対象領域（ドメイン）における割付問題（タスク）を統一的に扱うことを目指していた。しかしながら、それらの部品では問題解決過程において前提となる対象分野に特徴的な構造が明示的ではなかった。そのため、プリミティブな推論操作が前提とするデータ構造と概念レベルの対象モデルを明確に対応付けることが容易でなかった。

そこで、類似事例の分析、商用ソフトウェア・パッケージの開発／営業担当者へのヒアリング等を経て、現段階では製造業における生産スケジューリングに特化した部品ライブラリを構築し、実用システムの開発を通してその評価を行なっている。ドメインを特定したことによって、製造、出荷、受注といった関連業務との連携を考える上でより具体的な対象領域のモデルを得ることが可能となった。ジョブ割付問題による定式化から生産スケジューリング・システム構築のための部品ライブラリの実現に至る一連の過程は、現在の部品群に対する設計理由（design rationale）を与えるもので、実運用の環境との相互作用を通して到達した設計上の一つの均衡点となっている。

再利用知識の蓄積と維持には多大な労力を要する。その労力に見合う効果を得るには、知識管理において空間的ならびに時間的な２つのスケーラビリティを考えなければならない。前者は知識システムとそれ以外の情報システムや利用者との連携を通して広範囲での間接的な知識の利用を促すことに関わるもの である。それに対して、後者は知識システムの継続的な運用を通して顕在化する暗黙的な前提を踏まえて再利用知識を洗練し、知識を陳腐化させることなく資産として継承していくことを意味する。したがって、時間的なスケーラビリティには単に現在利用可能な知識を維持するだけにとどまらず、実社会との相互作用を通してその知識に新たな価値を付与していくという積極的な意義がある。

再利用可能な知識を用いたシステム構築では、初期開発の効率化だけでなく継続的な保守／拡張作業への効果も期待される。さらに、予め用意された仕様を前提とする場合と、実問題を対象とする場合によって評価の枠組も異なる。予め用意された仕様を前提とする場合、いくつかの手法を同一の問題に適用することによって異なる手法や知識を相互に比較することが可能である［Schreiber 96］。この場合、評価のための条件を整えた上で検討を進めることができるが、実問題への適用可能性について有益な知見を得ることは期待できない。それに対して、実用レベルの知識再利用に対して有益な知見を得るために、実問題を対象として特定の手法や知識を類似の状況での初期開発に適用したり［高岡 95］、初期開発から機能拡張の段階に至るまで継続的に適

用することが考えられる［堀 96］。このような様々な状況設定における実証的研究の成果を積み上げていくこと［Simon 95, 溝口 96］は研究者コミュニティ全体によって取り組まれるべき課題であると言える。

実証的研究においては、実験結果から得られた知見が一般的であっても実験環境と不可分な条件に依存する場合があり［Hanks 93］、その意味で特定の状況設定での結果をオントロジー設計や知識再利用に対する普遍的な原則や指針として一般化することは必ずしも容易でない。むしろ、知識を安易に一般化することなく個々の状況設定も含めて経験的に得られた知見や知識内容を共有していくことが重要である。

参考文献

[Benjamins 93] R. Benjamins : Problem Solving Methods for Diagnosis. Ph.D. thesis, University of Amsterdam, 1993.

[Brown 94] J. S. Brown and P. Duguid : Borderline issues: Social and material aspects of design. Human-Computer Interaction, Vol. 9, pp.3-36, 1994.

[Bylander 87] T. Bylander and B. Chandrasekaran : Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. International Journal of Man-Machine Studies, Vol.26, pp.231-243, 1987.

[Gennari 94] J. H. Gennari, S. W. Tu, T. E. Rothenfluh, and M. A. Musen : Mapping domains to methods in support of reuse. International Journal of Human-Computer Studies, Vol.41, pp.399-424, 1994.

[Gennari 96] J. H. Gennari, A. R. Stein, and M. A. Musen : Reuse for knowledge-based systems and CORBA components. Proceedings of the Tenth Banff Knowledge Acquisition for Knowledge-Based Systems, pp. 46.1-46.16, Banff, CA, 1996.

[Hanks 93] S. Hanks, M. Pollack, and P. Cohen : Benchmarks, test beds, controlled experimentation and the design of agent architectures. AI Magazine, Vol.14, No.4, pp.17-42, 1993.

[Hori 94] M. Hori, Y. Nakamura and T. Hama : Configuring problem-solving methods: a CAKE perspective. Knowledge Acquisition, Vol.6, No.4, pp.461-488, 1994.

[Hori 95] M. Hori, Y. Nakamura, H. Satoh, K. Maruyama, T. Hama, S. Honda, T. Takenaka and F. Sekine : Knowledge-level analysis for eliciting composable scheduling knowledge. Artificial Intelligence in Engineering, Vol.9, No.4, pp.253-264, 1995.

[Hori 97a] M. Hori : Trading cost of reuse in KBS development using explicit ontologies.

International Journal of Human-Computer Studies, Vol.46, No.2/3, pp.319-325, 1997.

[Hori 97b] M. Hori and T. Yoshida : Domain-oriented library of scheduling methods: Design principle and real-life application. Research Report 209635-9702, Tokyo Research Laboratory, IBM Japan Ltd., 1997.

[堀 95] 堀 雅洋, 吉田武稔, 吉川博文 : 問題解決手続きの構成とドメイン・オントロジーの関連について. 第9回人工知能学会全国大会論文集, pp.383-386, 1995.

[堀 96] 堀 雅洋 : 知識システムにおけるドメイン・オントロジーの安定性 ---実システムの開発・運用に基づく考察---. 第10回人工知能学会全国大会論文集, pp.199-202, 1996.

[堀 97] 堀 雅洋 : 知識システムのライフサイクルについて ---組織的知識創造の枠組に基づく考察---. 人工知能学会研究会資料, SIG-KBS-9603, pp.25-32, 1997.

[Laresgoiti 96] I. Laresgoiti, A. Anjewierden, A. Bernaras, J. Corera, A. Th, Schreiber, and B. J. Wielinga : Ontologies as vehicles for reuse: a mini-experiment. Proceedings of the Tenth Banff Knowledge Acquisition for Knowledge-Based Systems, pp.30.1-30.21, Banff, CA, 1996.

[溝口 96] 溝口理一郎 : 形式と内容 ---内容指向人工知能研究の勧め---. 人工知能学会誌, Vol.11, No.1, pp.50-59, 1996.

[溝口 97] 溝口理一郎, 池田 満 : オントロジー工学序説 (投稿中).

[Schreiber 96] A. Th. Schreiber and W. P. Birmingham : The Sisyphus-VT initiative. International Journal of Human-Computer Studies, Vol.44, pp.275-280, 1996.

[Simon 95] H. Simon : Artificial intelligence: an empirical science. Artifical Intelligence, Vol.77, pp.95-127, 1995.

[高岡 95] 高岡良行, 広部健治, 溝口理一郎 : 再利用可能知識ベースの構築 ---変電所事故復旧問題を例にして---. 人工知能学会誌, Vol.10, No.5, pp.786-797, 1995.

[Tijerino 93] Y. Tijerino and R. Mizoguchi : MULTIS II: enabling end-users to design problem-solving engines via two-level task ontologies. In N. Aussenac, G. Boy, B. Gaines, M. Linster, J.-G. Ganascia, and Y. Kondratoff, Eds., "Knowledge Acquisition for Knowledge-Based Systems", pp.340-359, Lecture Notes in Artificial Intelligence 723, Berlin: Springer-Verlag, 1993.

[van Heijst 97] G. van Heijst, A. Th. Schreiber, and B. J. Wielinga : Using explicit ontologies in KBS development. International Journal of Human-Computer Studies, Vol.46, No.2/3, pp.183-292, 1997.

[Winograd 86] T. Winograd and F. Flores : "Understanding Computers and Cognition: A New Foundation for Design". Albex, Norwood, NJ, 1986.

2.8 機械設計とオントロジー工学（1）

2.8.1 機械設計向きCADとオントロジー

機械設計の研究においてオントロジー研究は、単なる一アプリケーションとしての興味という以上に、現在、機械（に限らず）設計が一般に抱えている問題を解決するために設計を統合的に支援するシステムの研究開発が精力的に進められていることと関連を深く持っている。すなわち、設計あるいは多くのエンジニアリグ活動は本質的に多くの技術者による共同作業であり、その計算機による支援（いわゆるComputer Supported Collaborative Work）の実現は、人工知能・知識処理の研究としてもまた実用的にも多くの話題を含んでいる。計算機による設計や生産の統合は、CIM（Computer Integrated Manufacturing）として多くのシステムが開発され、また実用に供されている。最近ではこの考え方をさらに推し進め、製品のライフサイクルすべてに関わる活動を計算機上でモデル化し管理しようという Virtual Manufacturing、Virtual Engineering という考え方が提唱されている。

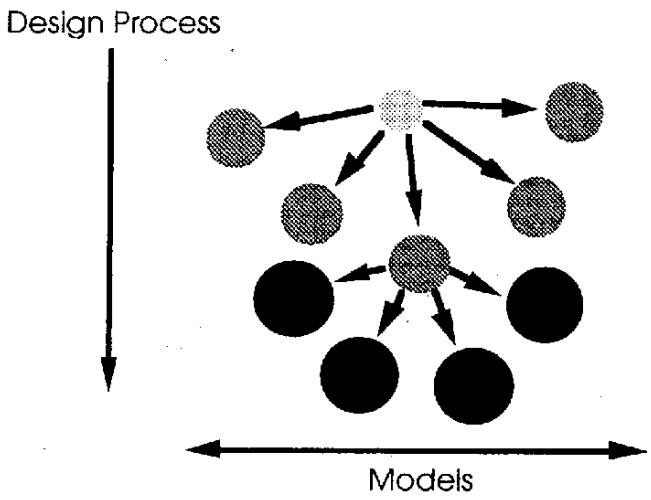


図2.8-1　設計過程と設計モデル

例えば図2.8-1は、設計対象物に関するモデルを表現し操作するCADを用いて、さまざまの設計解候補を並列的に検討していくのが設計過程であるということを示す。このように計算機による統合的なライフサイクル活動の支援を考えると、まずライフ

サイクル活動を表現するモデルが必要であることと、これらのモデルを利用してライフサイクルに関わるエンジニアリング活動を支援するシステムの全部または一部がコンピュータ上にあるため、エージェントアーキテクチャが有利であることが分かる。ここで全体として単一の大規模知識ベースとして構築することの実現はまず困難であり、図2.8-2に示すように何らかの形で分割することが必要となる。困難であることの最大の理由は、知識ベース全体に共通するオントロジーの定義や全体の整合性を確保することが極端に難しいことである。

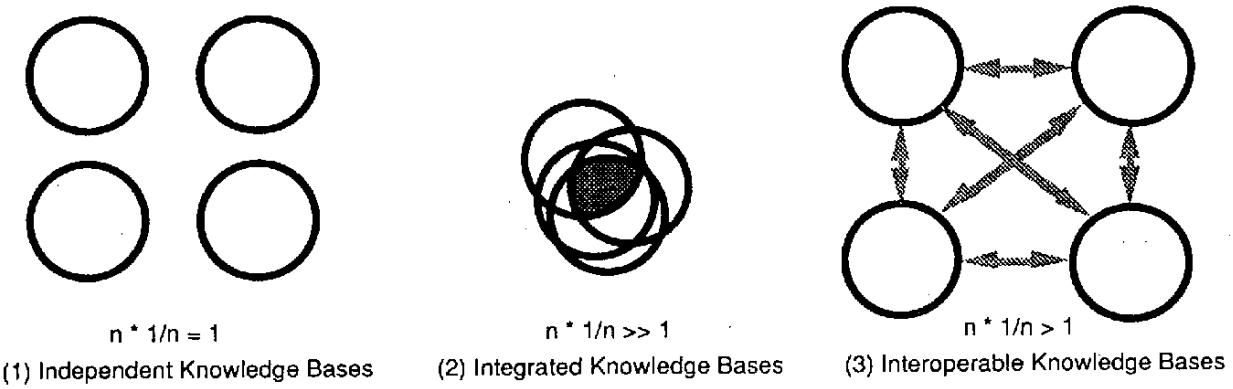


図2.8-2　エージェントの知識構造

図2.8-2(1)に示すようにn個の独立した知識ベースを集めたとき、全体としては1の力が出ていると考える（一つ一つの知識ベースを1/nの大きさとする）が、しかし互いに独立しているので個々を個別に使用するのと変わらない。図2.8-2(2)は、その知識ベース同士が重なりを持ち、互いに知識を補強し合うケースであり、全体として発揮される力は1より大きくなる理想的なケースである。しかし、このような知識ベースを構築することは巨大な知識ベースを構築することと実質的に等価であり、困難であることには変わりない。そこで、図2.8-2(3)に示すようなエージェントアーキテクチャによって相互に通信可能な構造が考えられる。この時には、全体の力は恐らく1より少し大きい程度であろうが、互いに内部構造を知らなくてもよいなど情報隠蔽の効果があり、構築は比較的容易になると考えられる。図2.8-2(3)のような疎結合されたエージェントアーキテクチャを想定すると、今度はそのエージェント間の通信や言語が問題となり、結局オントロジーの問題を避けることは出来ない。しかし、

個別の知識ベースが何を知っているかが記述してあればよいのであり、個別の知識ベースの細部について知る必要はなく、情報隠蔽の効果はそれほど失われない。

図2.8-2(3)のようなエージェント構造においてオントロジーがどのような役割を果たすかを示すのが図2.8-3である。図2.8-3では、まずシンタックスレベルでのhowに関する知識表現が統一されており、オントロジーはその上でどんな知識、つまりwhatを定義するものと考えており、これを知識モデリングの問題と呼ぶことにする。各エージェントは、そのオントロジーの上で互いに通信をする。したがって、オントロジーはまず知識記述のための最小単位（語彙）を規定し（これをターミノロジーと呼ぶことにする）、語彙に含まれる言葉の間の関係のうち最小限のもの（階層やラダーを使って分類整理するいわゆるタクソノミー）などが含まれる。しかし、このように収集された語彙がオントロジーとして機能するためには、それだけでは不十分であり、これに加えて言葉間の関係を規定する概念が必要となる。例えば部分・全体関係に関する概念であり、また特に機械設計などの分野ではこれに加えて空間や時間に関する概念をオントロジーに含むと都合がよい。

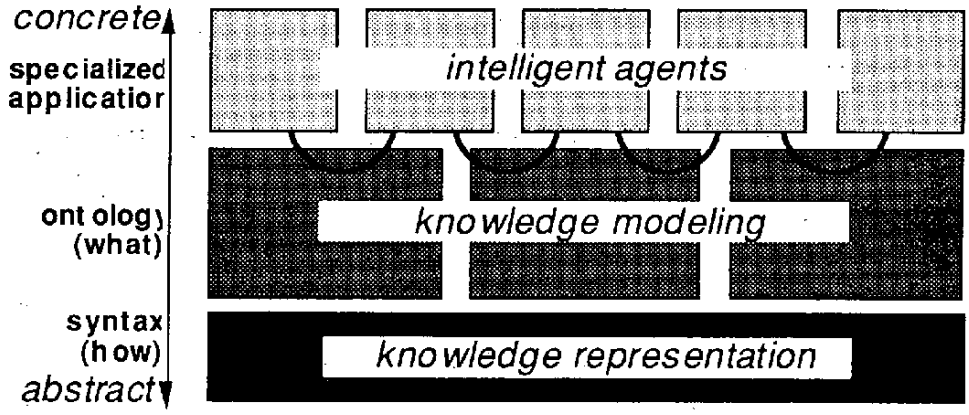


図2.8-3　オントロジーの役割

すなわち、機械設計向きCADのような応用を考えるとき、基本的な語彙（概念）には、次のような概念種類を記述しておくとよいと考えるが、これらの頭文字をとりCATSオントロジーと呼んでいる。これらの詳細については、次項以降で述べる。

物理的因果関係（Causality）：我々の研究ではForbusの定性プロセス理論［Forbus 84］を利用している。

行為（Action）
時間（Space）
空間（Time）

2.8.2 知識集約型工学のためのフレームワーク：KIEF

我々は製品ライフサイクル全体に亘るさまざまのエンジニアリグ活動に関する知識を集積・集約して用いることで付加価値を向上するという新しい工学のスタイルを知識集約型工学（Knowledge Intensive Engineering）と呼び、それを統合的に支援するフレームワークとしてKIEF（Knowledge Intensive Engineering Framework）の開発を進めている［Tomiyama 96］。KIEFは前項で述べた疎結合されたエージェントアーキテクチャに基づいており、エージェント間の通信、すなわちさまざまのエンジニアリシグ活動に関する知識をエージェント間で共有するために、KIEFが持つべき知識ベースはオントロジーのレベルから統合されている必要がある[Ishii 95]。

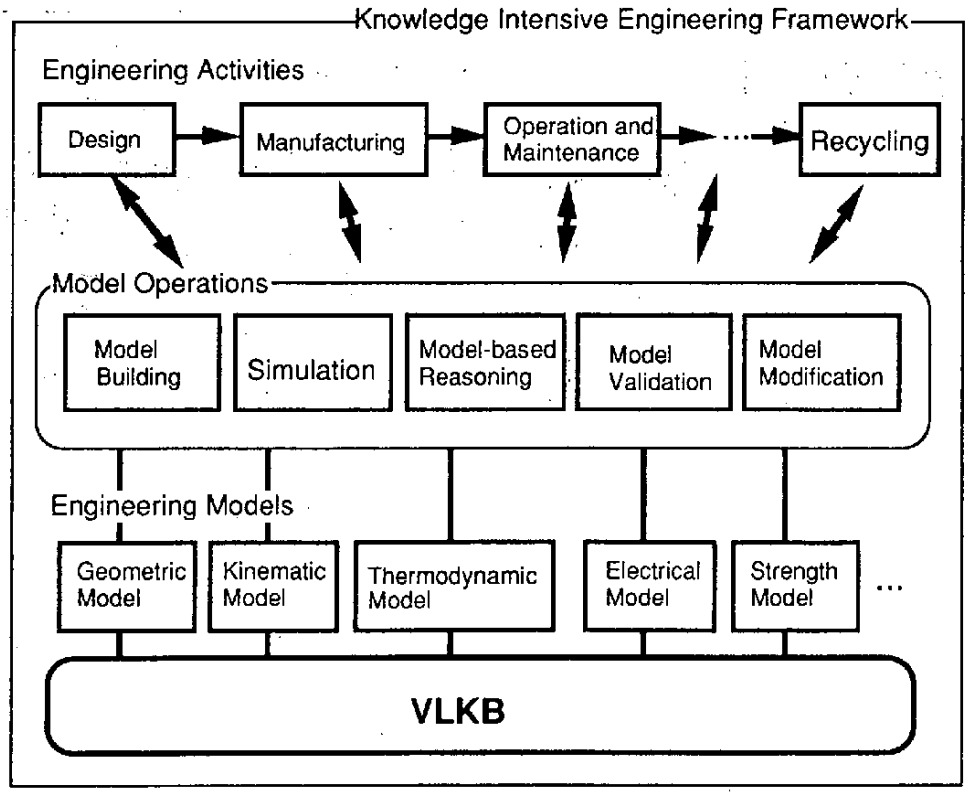


図 2.8-4　製品ライフサイクルとKIEF

ここで、図 2.8-4は、製品ライフサイクルにおいてKIEFがモデルベースの支援を行うことを概念的に表示したものである。例えば設計者は、設計対象物である製品に関するさまざまの情報をKIEFの上で生成し、さまざまの局面におけるその性能や挙動についてモデルを用いて評価を行う。同様に、設計時に生成された情報は生産や製品の運用、保全、リサイクルにも利用されるべきであり、それぞれの段階で技術者が設計情報にアクセスし、逆に設計者が各ライフサイクルで得られた情報にアクセスして設計にフィードバックする。この一連の行為でKIEFの重要な役割は、さまざまのモデル（形状モデル、有限要素モデル、ブロック線図等の各々の設計対象モデル）を統合することである。したがって、図 2.8-5に示すKIEFのアーキテクチャの中心はモデルを統合する枠組みにあり、我々はモデルのモデルという意味でメタモデル機構と呼んでいる。

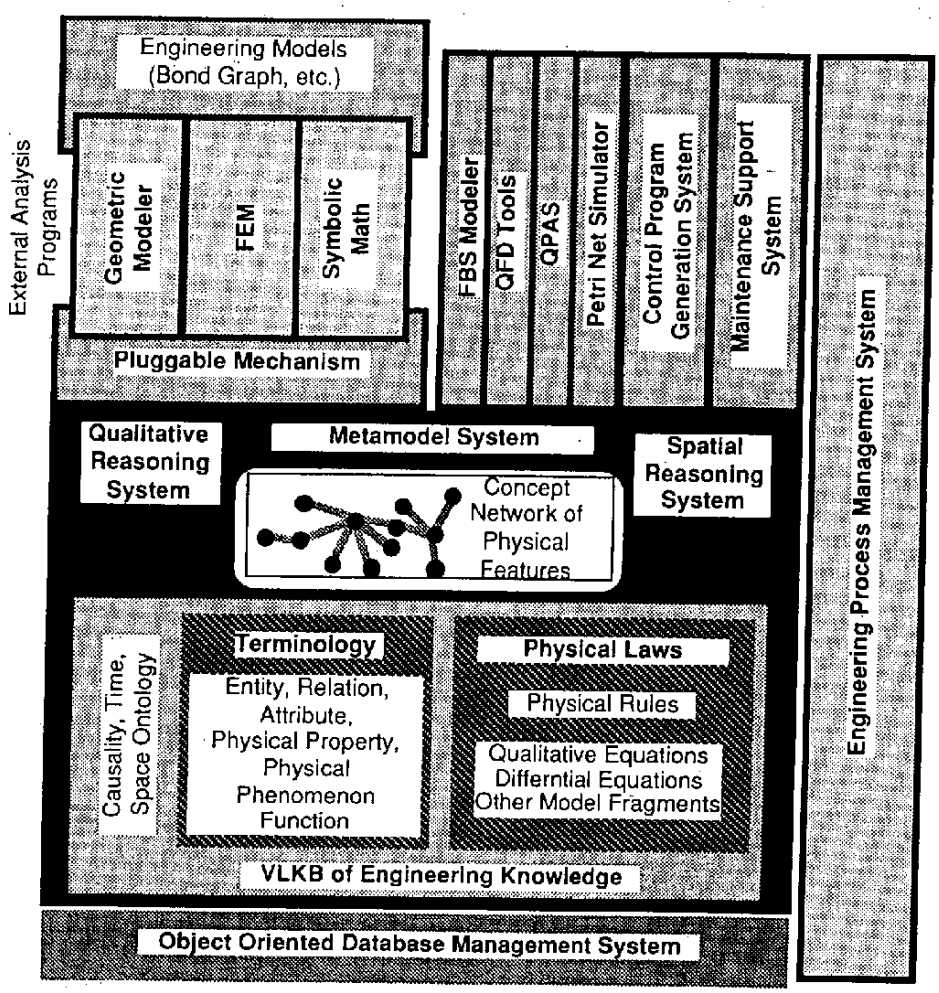


図 2.8-5　KIEFのアーキテクチャ

メタモデルについては、次項で詳しく述べるが、KIEFシステムは、メタモデルを用いたモデルの一貫性の管理や作成支援を行い、各種の設計ツールやモデラーが接続するためのプラガブル・メタモデル機構と、メタモデルの表現に用いる物理概念や、設計対象モデルの表現と物理概念との対応等、設計対象モデル作成に必要な知識を提供する知識ベースから成る。オントロジーはこの知識ベースの構成に関わる問題である。

2.8.3 メタモデル

製品のライフサイクルの初期である設計過程の、さらに最初の段階である概念設計段階では、要求される仕様を満足させるにはどのような挙動を発現させれば良いのかに重点がおかれる。設計の過程が進行するに伴い、設計対象を表すモデルが構築と解析され、挙動実現のための基本的機構や機構同士を結び付ける構造、そして機構を形作る部品の形状や材質といった情報が付加され詳細化される。また、設計対象の評価とが反復的に行われて、最終的な設計解が得られる。メタモデルはこれらの複数の設計対象モデルを統合的に取り扱うと共に、そこで表される設計対象の情報の一貫性の管理に用いる中心的モデルとしての役割を持つ。

また、概念設計段階において、設計対象の細部の情報は考慮されず、対象の大まかな構造の決定が行われるので、メタモデルの構築には定性的な知識が用いられる。例えばある機械でシャフトを回転させる必要がある時に、「2秒で3000回転に達することが出来る。」といった具体的数値は概念設計段階では現れない。「素早く高回転に達する」といった定性的に表現された挙動が重要となる。メタモデルは対象を定性的に表現した概念モデルとしての役割を持つ。

図2.8-6はメタモデルが、各モデルに出現するデータレベルではなく概念レベルでモデルを統合することを示すものであり、図2.8-7は、KIEFにおいてメタモデルがどのように構築されて、またどのようにして設計対象モデル間の整合性を管理するかを示している。メタモデルは、各種のモデルが取り扱うさまざま物理概念をノードとし、このノードをリンクで結んだ概念ネットワークとして表現する。ここで、ある設計対象物に関するメタモデルは、フィジカルフィーチャーを組み合わせて作成する。

一方、設計対象モデルはそれぞれのモデルに固有の理論体系に基づいて作成・利用されるが、しかし、その背景にある物理概念は共通であると考えられる。各モデルを構築する際には、大規模知識ベース内のモデルライブラリに存在するモデルフラグメントを呼び出して、メタモデル機構の方から対応するデータを供給してモデルを構築

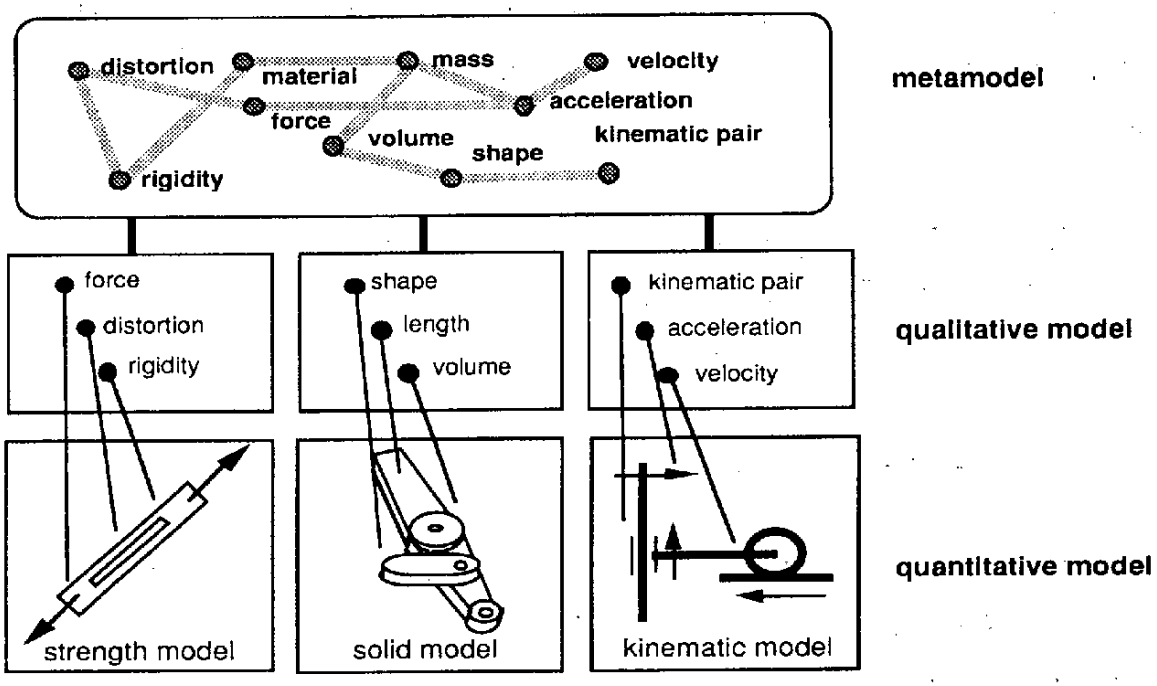


図 2.8-6　メタモデルによるモデルの統合

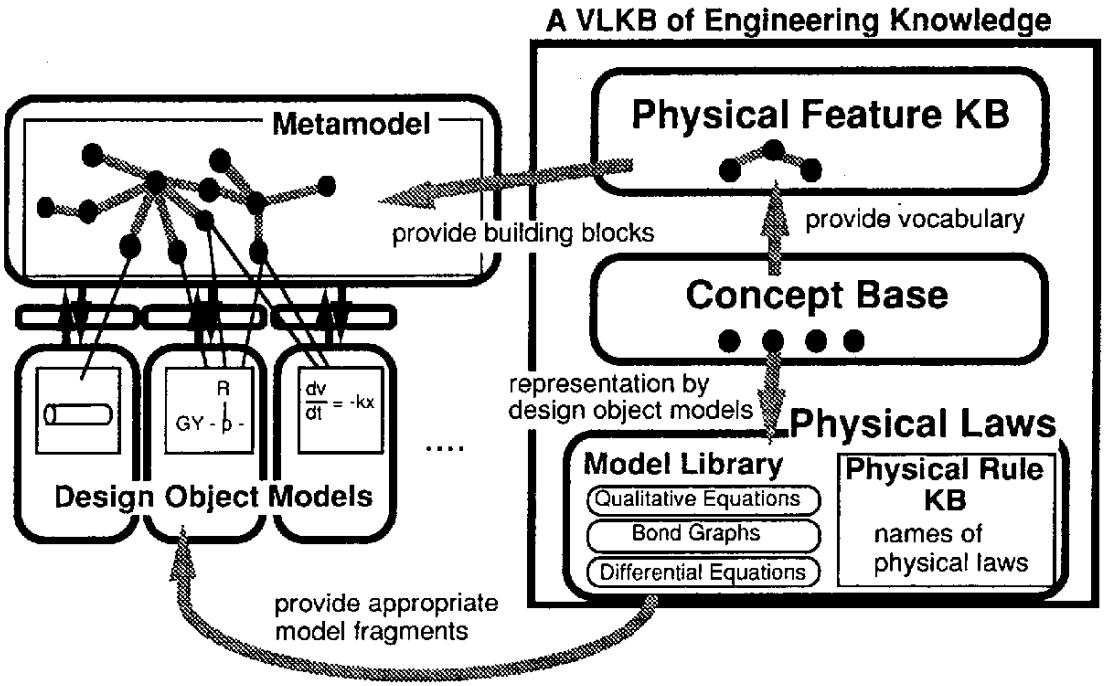


図 2.8-7　KIEFにおけるオントロジーの役割

する方式をとっている。

このようにして、物理概念を用いてメタモデルを表現すると共に、物理概念と各々の設計対象モデルの表現との対応関係を知識と保持することで、次のことが可能になる。

・設計対象の情報の一貫性の管理
・メタモデルを元にした設計対象モデルの作成
・メタモデルを通じた設計対象モデル間での情報の交換

2.8.4 オントロジーの分類

KIEFにおけるオントロジーとして、既に述べたようにCATSオントロジーを採用している。現時点では、このうち物理的な因果関係に関するオントロジーを中心にして概念辞書が作られている。具体的には、以下の6種類の概念に関して知識収集を行っている。

実体 (Entity)
属性 (Attribute)
関係 (Relation)
物理特性 (Physical Prop-erty)
物理法則 (Physical Rule)
物理現象 (Physical Phenomenon)

各々の物理概念はフレーム形式で定義する。実体・属性・関係・物理現象の各々の概念では、概念間に抽象具体の階層関係を設定して概念を定義している。下位の概念は階層関係の上位の概念で定義された概念の性質を継承する。これらの概念の定義には、Supersというスロットがあり、上位の概念の名称が記述される。本研究では上位の概念を複数定義することを許しているので、概念の性質は多重継承される。階層関係の最上位には各々の概念の種類を表す名称を持つ概念が定義されている。例えば、実体概念の最上位の概念は、Entityという概念であり、属性概念の最上位の概念は、Attributeである。

2.8.4.1　実体概念

物理世界の事物を表現する概念である。ばねや歯車、ねじ、モータ等の様に具体的な実体から、「機械部品」や「回路素子」のように抽象的な実体も実体概念に含まれる。抽象・具体の階層関係により、上位の実体概念が持つ性質は、下位の概念に継承される。例えば、Metalという実体概念に対して、「導電性」という性質が定義する場合、Metalの下位に定義される概念も「導電性」の性質を持つ。実体概念は次のスロットの記述によって定義される。

Attributes

実体に関わる属性を記述する。

Physical Properties

実体の性質、形状等を規定する物理特性を記述する。

実体の例としてSpring の定義を表2.8-1に示す。

表2.8-1　実体の例

スロット名	知識例
Name	Spring
Supers	MechanicalPart
Attributes	Length
Physical　Properties	Elastic

2.8.4.2　属性概念

速度、質量、力などの物理量のことである。属性概念は次のスロットの記述によって定義される。

Data Type

データタイプは扱う物理量が、スカラー量かベクトル量であるかを記述する。

Differential

物理量同士の微分関係を記述する。例えば、加速度は速度を時間に関して微分した量であることが記述される。

Dimension

物理量の次元とSI 単位系で定義された、その属性量を表す単位が記述される。

属性概念の階層構造では、基本的に上位概念のDimension を継承する。属性の例を表2.8-2に示す。

表2.8-2　属性の例スロット名知識例

スロット名	知識例
Name	AngularAcceleration
Supers	Acceleration
DataType	Vector
Differential	differentiate AngularVelocity with respect to Time
Dimension	$L^0M^0T^{-2}$ I^0temperature^0; *rad/s*^{2}

2.8.4.3　関係概念

複数の実体概念間の様々な関係を表す概念である。フィジカル・フィーチャの記述や物理現象の定義で用いられる。関係概念は、次の3つに分類される。

空間的な位置関係　　：In、On、Nearなど
機構上の拘束・接続関係　：二つのギアが噛み合うことを表すMeshed、
　　ねじ対偶を表すScrewPair など

特定の物理量に関する関係：実体間で電流が流れることを表すElectrical Connection
など

関係は次のスロットの記述で定義される。

Prerequisites

Relations

物理現象の定義に関係概念を用いる際の記述形式を定める。例えば、ある実体の上に別の実体が置かれた状態を表す関係Onでは、表2.8-3のように上にある実体を表すupperを先に、下にある実体を表すbaseを後に置いている。base や upper がどのような実体であるかは Prerequisites で記述している。On という関係は特定の実体間だけではなく、全ての実体に置いて成り立つと考えているため、upper、base 共にEntity と定義されている。これはEntity の下位の概念、つまり全ての実体概念においてOnという関係概念が成り立ち得ることを意味する。

表2.8-3　関係の例

スロット名	知識例
Name	On
Supers	Relation
Prerequisites	upper _Entity, base _Entity
Relations	(on upper base)

2.8.4.4　物理特性概念

実体が持つ性質を表す。例えば、バネという実体の定義においては、「弾性的」や「コイル状の」という物理特性が用いられる。つまり、実体概念は物理特性概念の組合せで定義される。物理特性は次のスロットの記述で定義される。

Comments

物理特性概念を参照する者の便宜を考えて記述する。

Attributes

当該物理特性が示す内容がどのような属性に関する概念であるかを記述する。例えば、Hotという物理特性は温度に関するものであるから、このスロットにはTemperatureと記述されている。

物理特性は次のように分類できる。

属性概念の範囲を指定する特性：Hot（Temprature）、Thick（Thickness）など
形状を表す特性　　：Spiral、Curvedなど
現象に深く関わった特性　　：Combustible（可燃性の）など

物理特性の例を表2.8-4に示す。

表2.8-4　物理特性の例

スロット名	知識例
Name	Elastic
Comments	Consider the entity as elastic object, such as a rubber.
Attributes	ModulusOfElasticity

2.8.4.5　物理法則概念

一般に物理世界における法則や定理として、物理現象の背景に存在しその現象に関わる属性間の関係を定義する概念である。物理法則は次のスロットを用いて記述される（表2.8-5）。

Comments

物理法則の内容がテキストとして書かれるスロットである。

Attributes

物理法則の表現の中で用いられるパラメータを、変数として定義しているスロットである。

Field

その法則が属する学問領域を記述する。

Expression

その法則が示す内容を表現するスロットである。論理式による表現や、定性的な表現、数式による表現など様々な表現が考えられる。概念辞書とモデル・ライブラリを含めた全体的な構成に関して、物理法則の内容は各モデル・ライブラリの中で記述するとしているが、何らかの表現を与えることは知識記述者が内容を理解するための一助となる。そこで本研究では法則の一般的表現として、数式を用いている。

表2.8-5　物理法則の例

スロット名	知識例
Name	CoulombLawForElectricForce
Comments	The force of attraction or repulsion between two bodies is inversely proportional to the distance between the two bodies.
Attributes	q1 _ElectricCharge, q2 _ElectricCharge, F _ElectricForce, r _Distance
Field	Electromagnetics
Expression	$F = k \bullet q_1 \bullet q_2 / r^2$

2.8.4.6　物理現象概念

重力が働く、バネが伸び縮みする、等のように実体に働く物理的な現象を記述する概念である。物理現象概念の定義は次のスロットで行われる（表2.8-6）。

Local Entity Slots

その物理現象が起きる実体と、当該物理現象において実体に関わる属性を記述する。

Physical Rules

その物理現象の背景に存在する物理法則と、Local Entity Slots で記述した実体の属性がどの物理法則で定義されるかを記述する。

Conditions

その物理現象が起きるために必要な前提条件を、実体間の関係と実体が持つ物理特性で記述する。

表2.8-6　物理現象の例

スロット名	知識例
Name	WingEffect
Supers	PhysicalPhenomenon
Local Entity Slots	(object _(ClosedPath Lift) fluid _(Pressure Density Velocity))
PhysicalRules	(KuttaZhukovskijTheorem (Lift object) (ClosedPath object)(Density fluid))
Conditions	(PhysicalProperty (Thin object)(Flat object) (Fluid fluid)) (Relation (Soaked object fluid))

2.8.5　フィジカルフィーチャー

2.8.4項に定義した物理的因果関係に関する概念を用いて、フィジカルフィーチャーを定義する。フィジカルフィーチャーとは、ある物理的な「状況」に関するシナリオであり、物理法則、実体、関係に関する概念を用いて記述する。図2.8-8は、回転するギアとウォームギアが噛み合っており、その両者が回転することでギアによる回転伝達という物理現象が起きるというフィジカルフィーチャーを示している。このフィジカルフィーチャーは、完全にグラフィカルに編集できるため、特に知識としてフレームの形で記述する必要はない。

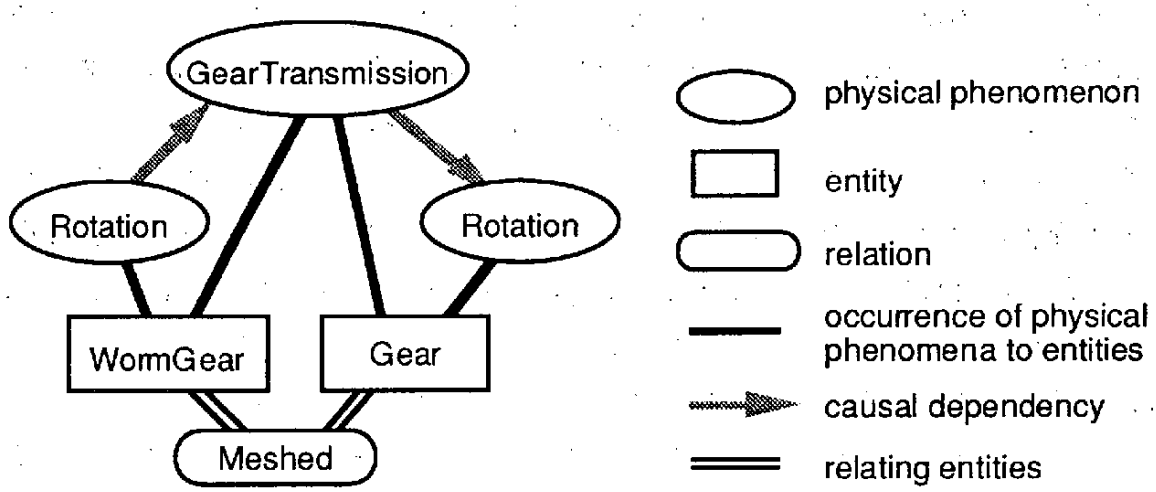


図2.8-8　ギア駆動に関するフィジカルフィーチャー

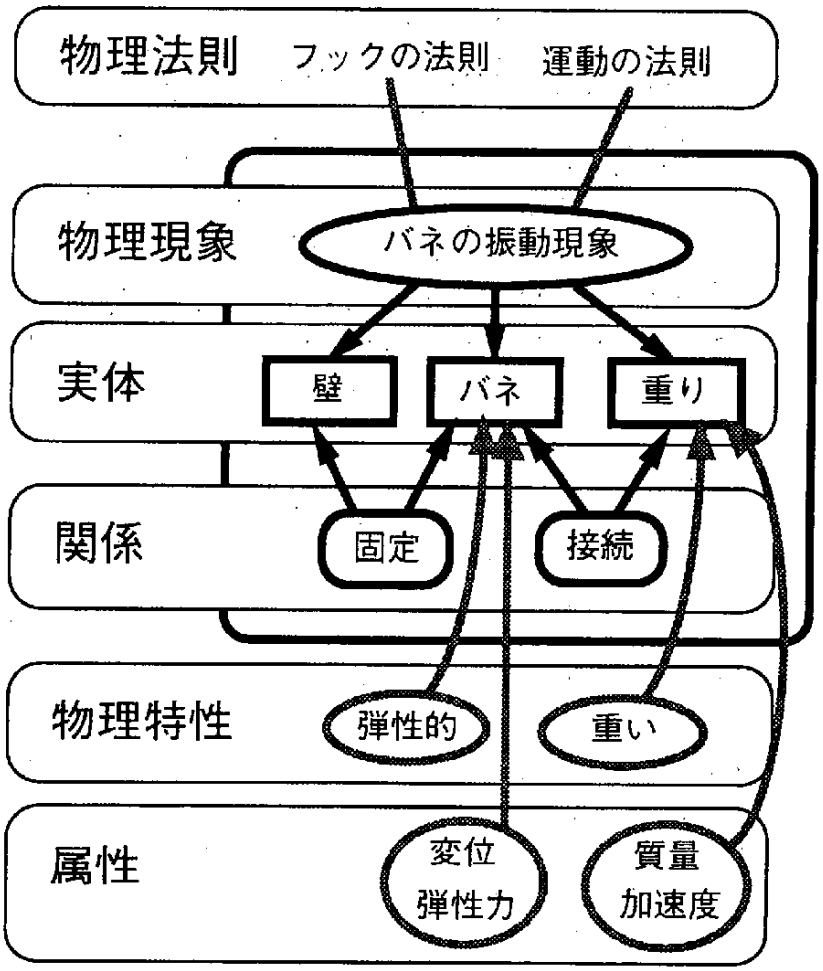


図2.8-9　プライマリモデルの例

次にこのフィジカルフィーチャーを組み合わせて、メタモデルを作るが、この段階で生成されるのはプライマリモデルと呼ばれるいわば第1近似モデルで、メタモデルシステムへの入力である。これを定性プロセス理論に基づいた定性推論システムにかけて、エンビジョニングを行い、その結果として何が起こりうるのか、知りうることをすべて導出したモデルをメタモデルとして作る。バネを介しておもりを壁に接続した簡単な機構のメタモデル（プライマリモデル）を図2.8-9に示す。実体と関係によって対象の抽象的な構造を表現し、物理現象を用いて対象の挙動を表現している。

実は現在のシステムでは、このようにして作られるプライマリモデル（メタモデル）も、プラガブル・メタモデル機構における一つのモデルとして位置づけられている。従って、他のモデル（例えば運動モデルやボンドグラフモデル）も、プライマリモデルと同様の方法で生成される。これを示したのが図2.8-10である。

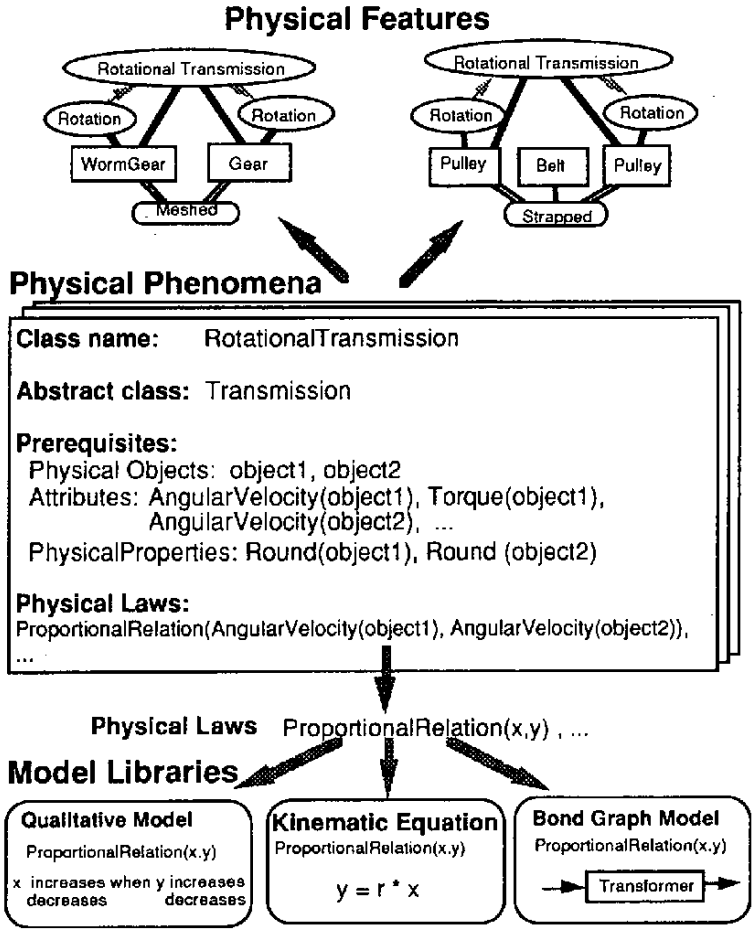


図2.8-10　フィジカルフィーチャーによるモデルの生成

2.8.6 まとめ

本稿では、我々の研究室にて研究している知識集約型工学のための計算機フレームワークKIEFの基本的な考え方についてまず説明した。KIEFはエージェントアーキテクチャを採用しており、その構築にはエージェント間で知識や情報を交換するためにオントロジーが必要である。機械設計向きCADへの応用では、物理的因果関係、行為、時間、空間に関するオントロジーが必要であり、現在のシステムではこれらのうち、物理的因果関係に重点をおいて知識を収集している。

今後は、行為や、時間、空間などに関するオントロジーに取り組む予定である。

参考文献

[Forbus 84] Forbus, K.D., Qualitative process theory, Artificial Intelligence. Vol. 24, No. 3, pp. 85-168, 1984.

[Tomiyama 96] Tomiyama, T., Umeda, Y., Ishii, M., Yoshioka, M., and Kiriyama, T., Knowledge Systematization for a Knowledge Intensive Engineering Framework. In Tomiyama, T., Mantyla, M., and Finger, S. (eds.), Knowledge Intensive CAD, Volume 1, Chapman & Hall, London, pp. 33-52, 1996.

[Ishii 95] Ishii, M., Sekiya, T., and Tomiyama, T., A very large-scale knowledge base for the Knowledge Intensive Engineering Framework. In Mars, N.J.I. (ed.), Towards Very Large Knowledge Bases, IOS Press, Amsterdam, Oxford, Tokyo, Washington, D.C., pp. 123-131, 1995.

2.9　機械設計とオントロジー工学（2）

機械設計は、古い分野で完成した領域と考える向きがあるかもしれないが、多くの未解決な問題を残しており、決して組織的に行われているわけではない。金属の摩耗、疲労などは、多数の研究が行われてきているにも拘わらず、組織的な設計が可能となるほどには解明されていない。また、ある意味で驚くべきことに機械設計を支える重要な要素である金属切削も長い研究にも拘わらず解明されていない。これらの領域は、今日でも重要性を保っており、その解決は依然として工学問題の一角を占めている。なお、ここで解明とは、電子工学に見られるとされるように科学的知見を直接応用して開発が行われるような状態を指している。これらの機械工学関連の問題は、未解明とはいえ、次のような特徴を持つ。

(1) 還元論的には解決されている。関係する現象は、個々には科学的に解明あるいは説明されているといえる。
(2) 多数の実験によって、実用的には解決されている。例えば、摩耗という現象があるために機械の製造ができないというわけではない。

以上の特徴が示すことは、機械設計の抱える問題の大部分は工学的には解決される可能性があるということである。すなわち、それらの問題は、古典力学的姿勢と図面や文書といった記述媒体の制約とによって解決が困難になっているのであって、これらの制約を情報処理技術によって緩和させられるならば解決の可能性がある。

ここで解決という意味も多様な定義が必要である。たとえば、切削加工も原子配列制御も金属の形状を創成するという点では同じ目的を持つが、後者は実現精度が理想的に高い。このことを以って、切削加工が不用になるとは言えず、逆に所要コスト（エネルギー、時間など）との関係では理想的となることもある。工学の場合は、単一の価値観では判定できず、粒度や詳細度などの観点を状況に応じて管理する必要がある。

オントロジーを対象を理解・記述する観点であるとしたとき、上記の問題意識は工学のためのオントロジーを確立するするということに繋がる。機械工学にとっては、従来の古典力学から受け継いだオントロジーに代わる（機械）工学オントロジーの確立が必要である。そのためには、機械工学内部の変革だけでは不可能であるとともに、推進力となる情報系技術の研究が溝口の提唱する「内容指向研究」に向かうことが必

要である。

以下、機械工学関連で特徴的な事項を考える。

(1) Rule-based Systemの効果

RbSは、一時大変注目され人工知能技術と同一視されたこともあったが、その結果として機械工学を中心に評価するべき影響を残している。RbSが導入される以前の機械工学は、現在以上に理学（力学）的であり、単に応用研究と考えられていた。RbSによって外界に働きかける実用性が、理学的根拠ではなく有用性によって選択されたルールによって実現できることが示されたことが大きかったと考えられる。この頃に工学的オントロジー確立に向けた素地が醸成されたと見ることが出来る。

(2) 工学的理論の領域固有性

最初のブームの後RbSの行き詰まりから「深い知識」が注目されたが、現在のところ初期の期待通りには成功しているとは言えない。その原因は、工学問題の深い知識を物理学に求めたことにあると考えられる。工学と理学は目的を異にする体系であることが軽視されていたものと予想され、その更なる原因は一般性、したがって記述の経済性への指向であったと考えられる。しかし、工学的理論は強い領域固有性を持つと考えるべきである。強い証拠のあることではないが、RbSが工学分野で歓迎されたことの理由は、そのことを直観的に技術者が理解したことに帰着されるのではないかと予想される。

2.10　オントロジー工学の研究動向と諸相

2.10.1　はじめに

最近、Yahoo サーチエンジンにおいて、人工知能の中にオントロジーというジャンルが追加され（実際は、哲学の中にオントロジーというジャンルが設けられ、人工知能からリンクが張られている。http://www.yahoo.com/Arts/Humanities/Philosophy/Metaphysics/Ontology）、オントロジー工学に関する様々な研究が紹介されている。また、人工知能に関連する国際会議やワークショップにおいても、オントロジー工学が取り上げられる機会が増え、オントロジー工学の研究への関心が高まっている。しかしながら、オントロジーについては、まだ統一された見解はなく、研究分野や研究者に依存して異なった捉え方や構築が進められているのが現状である。本稿では、G.Heijstによるオントロジーの分類を紹介する事によって、自然言語理解、データベース、知識工学の3つの研究分野とオントロジー工学の関連が深い事を示すとともに、各研究分野におけるオントロジーへの具体的な構築例を紹介し、オントロジー工学の研究動向を探る。

2.10.2　G.Heijstによるオントロジーの分類

G.Heijstは、表現された概念の構造（の型）と概念化対象という2つの観点から、オントロジーは分類できるとしている［G.Heijst 1995］。情報構造の型からは、オントロジーは下記の3種類に分類される。

- Terminological ontologies：一般辞書や専門用語辞書である。テキストレベルで用語の定義を与えている辞書も含み、情報が構造化されていなくてもよい。
- Information ontologies：データベースのスキーマ表現である。レコード構造のような情報構造を持つ。
- Knowledge modeling ontologies：知識の概念化の仕様である。Information ontologiesと比較して、階層構造のようなより豊富な情報構造を持つ。

一方、概念化対象からは、オントロジーは下記の5種類に分類される。

- Domain ontologies：問題領域における概念化の仕様である。

・Generic ontologies：様々な問題領域で利用可能な汎用的な概念の仕様である。例えば、イベント、プロセス、状態、要素などの概念が定義される。Generic ontologiesは、Domain ontologiesに対して、上位のオントロジーになる。

・Task ontologies：メソッドに関する仕様である（［G.Heijst 95］では、これをTask Modelsと呼び、オントロジーとは別個に取り扱っているが、最近は、このように呼び、オントロジーに含める事が多い）。

・Application ontologies：あるアプリケーションで要求される知識モデリングに関するすべての仕様である。具体的には、Application ontologiesは、Domain ontologiesとGeneric ontologiesであるが、Task ontologiesを含める場合もある。

・Representation ontologies：知識表現に関する仕様である。言語のシンタックスに相当する。

情報構造の観点から分類された3種類のオントロジーは、上から順に、自然言語理解、データベース、知識工学の研究分野で検討されているオントロジーに対応し、概念化対象の観点から分類された5種類のオントロジーは、知識システムを実行可能とするためのオントロジーであるKnowledge modeling ontologiesを細分化したものとして捉える事ができる。以下においては、分類された各々のオントロジーについて説明する。

2.10.3 自然言語理解におけるオントロジー

自然言語理解の研究分野では、計算機可読型辞書（MRD: Machine Readable Dictionaries）の開発が精力的に行われており、オントロジーというとMRDを指す場合が多い。よく知られているMRDとしては、WordNet（プリンストン大学）、ROGET's Thesaurus（MICRA）、LDOCE（Longman's Dictionary）、SENSUS（南カリフォルニア大学情報科学研究所）、EDR（日本電子化辞書研究所）などがある。MRDに依存して、オントロジーとしての完成度はまちまちであるが、概念階層構造が整っている事から広く使われている WordNet、階層構造としてのIS-A関係だけでなく他の概念関係子もサポートしているEDR電子化辞書、並びに、既存のMRDをマージして構築されたSENSUSについて、以下、概略を説明する。

WordNet［Miller 1990］は、図２．１０-１に示すように、名詞句辞書、動詞句辞書、形容詞句辞書、副詞句辞書、および見出し句辞書から構成されており、総計約９万個

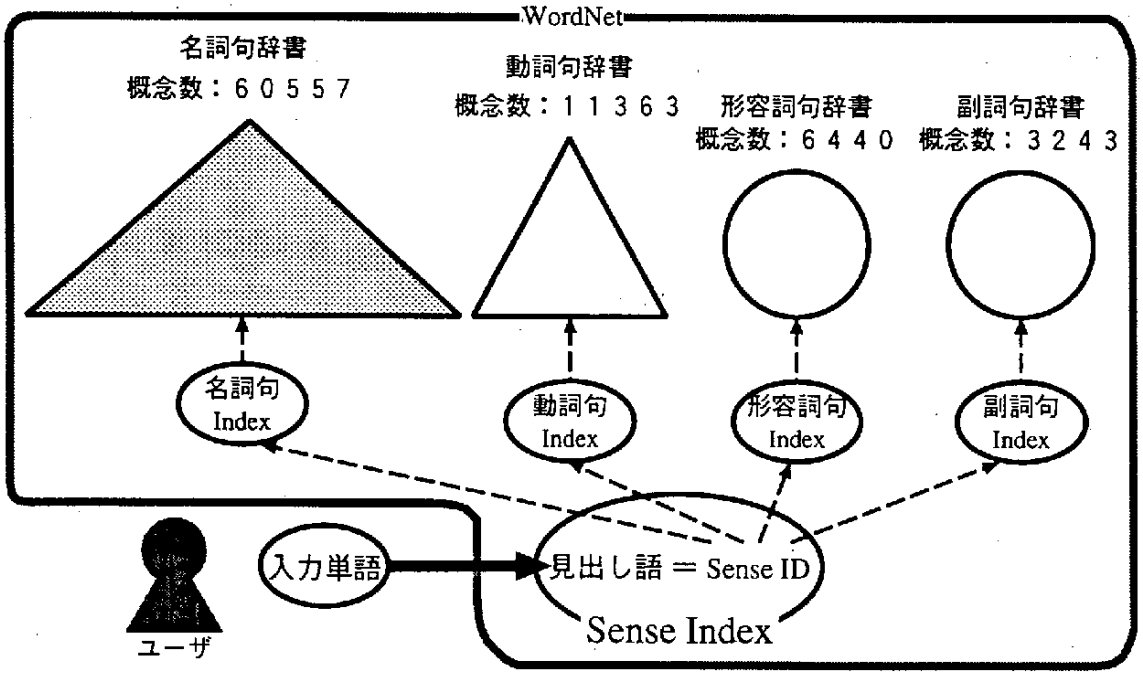


図２．１０-１　WordNet の概観

の語彙を保持している。見出し句辞書は、見出し句、意味情報としての概念ID、辞書編集情報、品詞情報などから構成されているが、概念IDが、見出し句辞書と各辞書の

間のリンクとして機能している。名詞句辞書と動詞句辞書は、概念ID、辞書編集情報、対応する見出し句リストから構成されているが、概念群は階層構造を有している。また、一部の概念IDには、反対概念の概念ID、part ofやmember ofやsubstance ofの概念IDなども与えられている。形容詞句辞書と副詞句辞書も、概念ID、辞書編集情報、対応する見出し句リストから構成されているが、階層構造は持たない。

EDR電子化辞書［EDR 1993］は、図２.１０-２に示すように、単語辞書と概念辞書から構成されている。単語辞書は、見出し情報、文法情報、意味情報、補助情報から構成されているが、意味情報には、概念辞書の各概念ノードを識別するための概念識別子が割り当てられ、単語辞書と概念辞書を結合する働きを持っている。一方、概念辞書は、概念識別子、概念見出し、概念記述から構成されており、概念記述には、多重継承を許す概念階層関係を記述する他に、agent（動作主体）、object（対象）、goal（目標）、implement（道具・手段）、cause（原因）、place（場所）、scene（場面）、a-object（属性）という８種類の概念関係子による記述が与えられる。

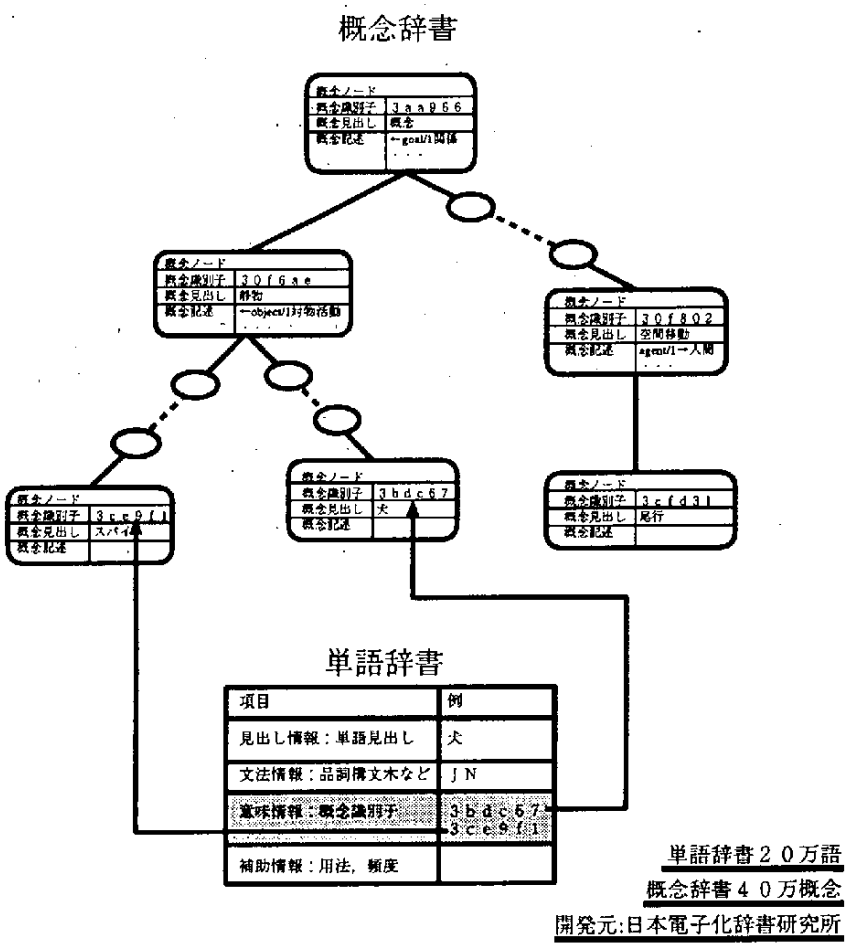


図２.１０-２　EDR の概観

SENSUS［Knight 1994］は、図２．１０-３に示すように、既に開発されているGeneric Ontologies、PENMAN Upper Model［Bateman 1990］とONTOS［Carlson & Nirenburg 1990］をマージして、上位オントロジーとしてのOntology Baseを構築した後、WordNetやLDOCEを順次その下位にマージしていく事によって構築されたMRDである。構築プロセスは、完全に自動化されていないが、２つの概念の定義文における共有単語の有無によって、２つの辞書間で対応する概念を見出したり、対応する概念の上位下位概念群に注目して、さらに対応する概念を見出すような照合アルゴリズムが利用されており、マージ作業を効率化している。

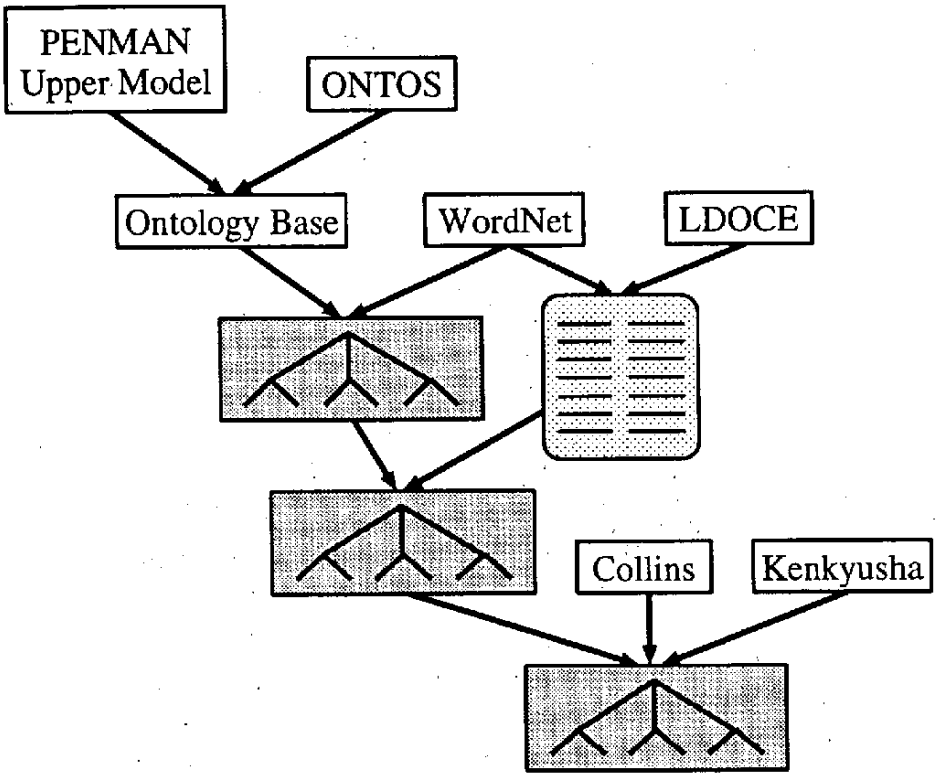


図２．１０-３　SENSUS の概観

2.10.4 データベースにおけるオントロジー

データベース（DB）の研究者は、実在するデータを重視して、オントロジーを考察する事が多いと筆者は感じている。例えば、インターネット知的サーチエンジンSIMS［Arens et al. 1996, Tejada et al.1996］では、各サイトにテーブル形式（関係DB）による生データとスキーマ（フィールド＝属性）が蓄えられている環境を想定して、テーブル名とフィールド名を概念と見なし、概念階層木を自動的に構成する研究が進められている。以下、この手法について概説する。

まず図2.10-4に示すように、3つの異なったテーブル形式のDB、Geoloc Table、Airport Table、Seaport Table が既に存在すると仮定する。各テーブルにおいて、XCodeがキーフィールドであり（キーフィールドの同定処理は自動化されている）、キーフィールド値の包含関係を調べる事により、図2.10-5のようなキーフィールドの階層木が得られる。キーフィールドは、テーブルを代表するフィールドである事から、キーフィールド名をテーブル名に置き換え、ノンキーフィールドをそれが属するテーブルに HAS-A関係により連結させる事により、IS-A関係とHAS-A関係から構成される概念階層木が得られる（図2.10-6参照）。最後に、複数個の概念（Seaport and Airport）からHAS-A関係により連結されたフィールド（RR Access）が存在するので、それらの概念群の上位にある中間概念が隠されている（その中間概念に RR Access が HAS-A関係により連結されるべき）と考え、新しい中間概念（Port）を考案し、概念階層木を完成させる（図2.10-7参照）。

概念階層は、キーフィールド値の包含関係により形成されるため、値の与えられ方によっては、無意味な概念階層を作り出す危険性があるが、MRDなどと相補的に利用する事により有用となるアプローチといえよう。

Geoloc Table

GCode	Name	Country
TYUV	Gabes	Tunisia
JHBF	Jerba	Tunisia
SDRD	Rome	Italy
GFJI	Paris	France
GBKL	Rome	Bulgaria

Airport Table

ACode	RRAccess	FuelCap
TYUV	N	250,880
SDRD	Y	375,000
GFJI	Y	567,575

Seaport Table

Scode	RRAccess	BerthQty
JHBF	Y	25
GBKL	N	38

図2.10-4 テーブルのスキーマとデータ

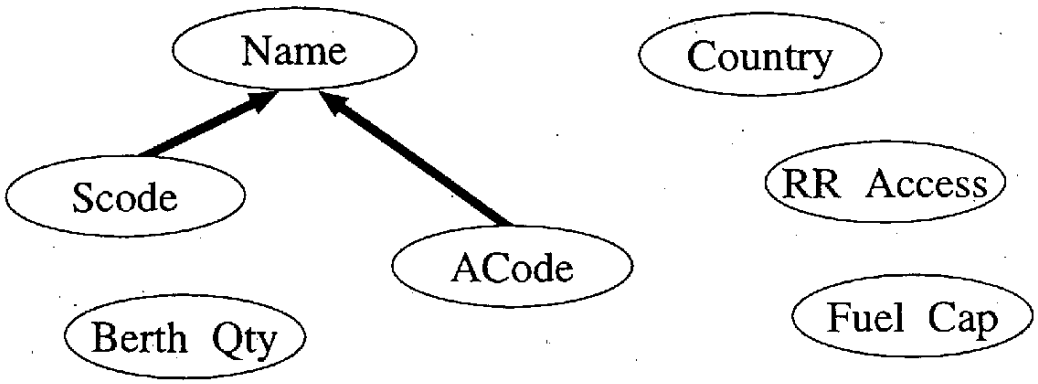


図2.10-5　キーフィールドの階層木

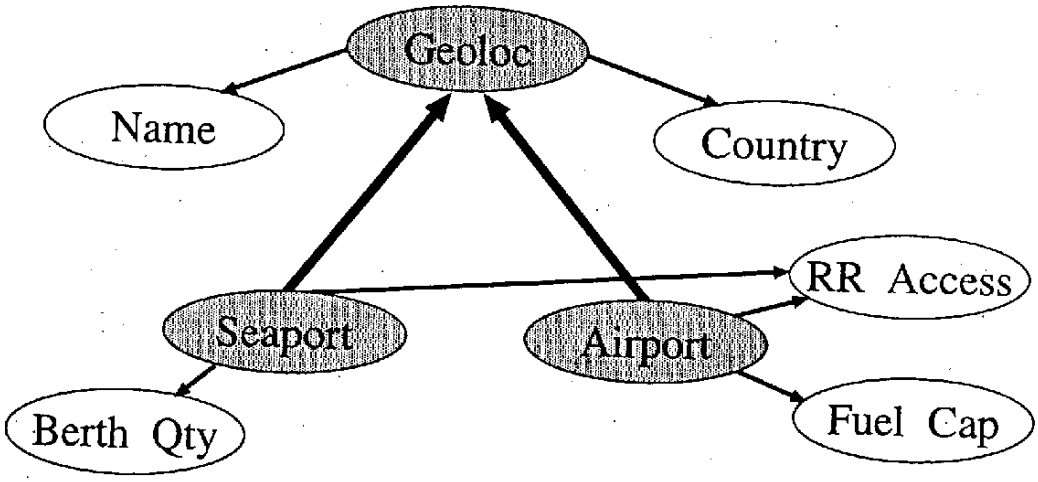


図2.10-6　テーブル名の階層木

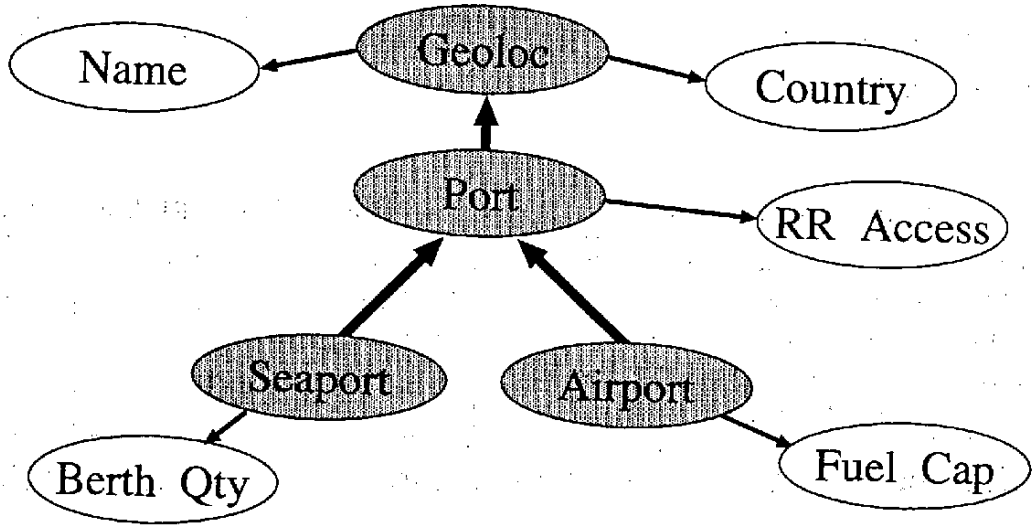


図2.10-7　最終的に得られた概念階層木

2.10.5 知識工学におけるオントロジー

2.10.2項で述べた、概念化対象の観点から分類されたオントロジーは、知識システムを実働化させるための様々なプリミティブ体系と捉える事ができる。知識システムは、知識ベースと推論エンジンが二大構成要素であるが、前者の概念プリミティブ体系として Generic ontologies と Domain ontologies、後者の概念プリミティブ体系としてTask ontologies、および、表現プリミティブの体系として Representation ontologies が対応する。以下、各オントロジーの代表的な例について概説する。

(1) Generic ontologies

自然言語理解の分野においては、2.10.2項で述べた PENMAN Upper Model と ONTOSが代表的であり、知識工学の分野においては、スタンフォード大学で開発されたOntolingua［Gruber 1992］がある。Generic ontologies は、概念階層木全体の上位部分を決定する上で核となるオントロジーであり、様々な議論が成されている（例えば、Special Issue : The role of formal ontology in the information technology,International Journal of Human-Computer Studies, Vol.43, No.5/6 等参照）。

(2) Domain ontologies

Domain ontologiesは、問題領域に含まれる概念の膨大さ、あるいは、Task ontologies との相互関連を記述する困難さにより構築が困難であることから、他のオントロジーと比較して構築例が少ないのが現状であり、構築支援環境の整備が望まれている。構築支援方法としては、MRDを活用する方法と開発済みの類似ドメインオントロジーを利用する方法が検討されている。

前者に関しては、南カリフォルニア大学情報科学研究所で開発されているオントロジーブラウザー Ontosaurus がある［Swartout 1996］。Ontosaurus では、図2.10-8に示すように、空軍作戦（Air Campaign）に関連した入力語彙と前述したSENSUS間の照合をとり、照合されたノード（ACP Seed Term）と根節点間のパス（Path to Root）を中心に分析し、そのパス上に入力語彙以外の空軍概念（Military Terms）が見出されたり、あるいは、その空軍概念の上位概念（Frequent Parent）やその下位部分木に含まれる概念（Subtree Terms）に新しい空軍概念が見出されて、空軍概念階層木が構築されていく。Ontosaurusは、人手に委ねられる部分が依然大きいが、MRDを利用した最初のドメインオントロジー構築支援環境として意義があるといえる。

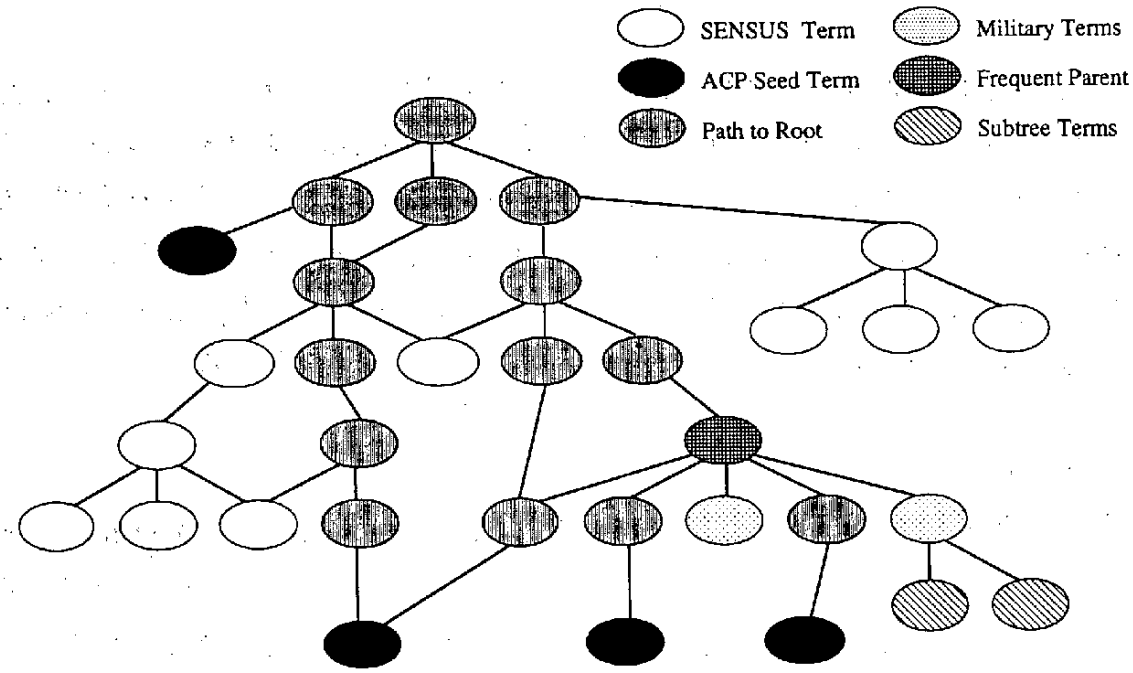


図2.10-8　MRDを利用したドメインオントロジーの構築

一方、後者については、アムステルダム大学で検討されているオントロジーライブラリーがある［G.Heijst 1995］。G.Heijstらのアプローチでは、クラス、関係、関数、インスタンスの集合から構成される定理を基本単位として、汎用性の高い定理集合としてのコアライブラリーとそれ以外の定理集合としての周辺ライブラリーを事前に構築しておく。そして、新しい問題領域におけるドメイン階層とメソッド階層を構築した時、それらをドメイン固有性とメソッド固有性の観点から、この2つのライブラリーに関連づける事によって、オントロジーライブラリーを拡張していく。

両者ともに、専門家であるユーザにかかる負担は依然大きいものがあるが、今後の研究が期待される分野である。

(3)　Task ontologies

他のオントロジーと比較して、推論プリミティブ（問題解決メソッド）の整理は取り組みやすいこともあって、タスクオントロジーの研究は盛んである。例えば、スタンフォード大学で開発されているPROTEGE-II［Puerta et al. 1992］、欧州のEspritにおけるプロジェクトCommon KADS［J.Breuker and W.Van de Velde 1994］、我が国では、溝口らのMULTIS［Mizoguchi R. et al1992］、堀らのCAKE［Hori, M. et al. 1994］にお

いて、タスクオントロジーが構築されている。PROTEGE-IIによる設計タスク構造の記述例を図２．１０-９に示すが、再利用性の高い問題解決メソッドを如何に構築するかが焦点となっている。

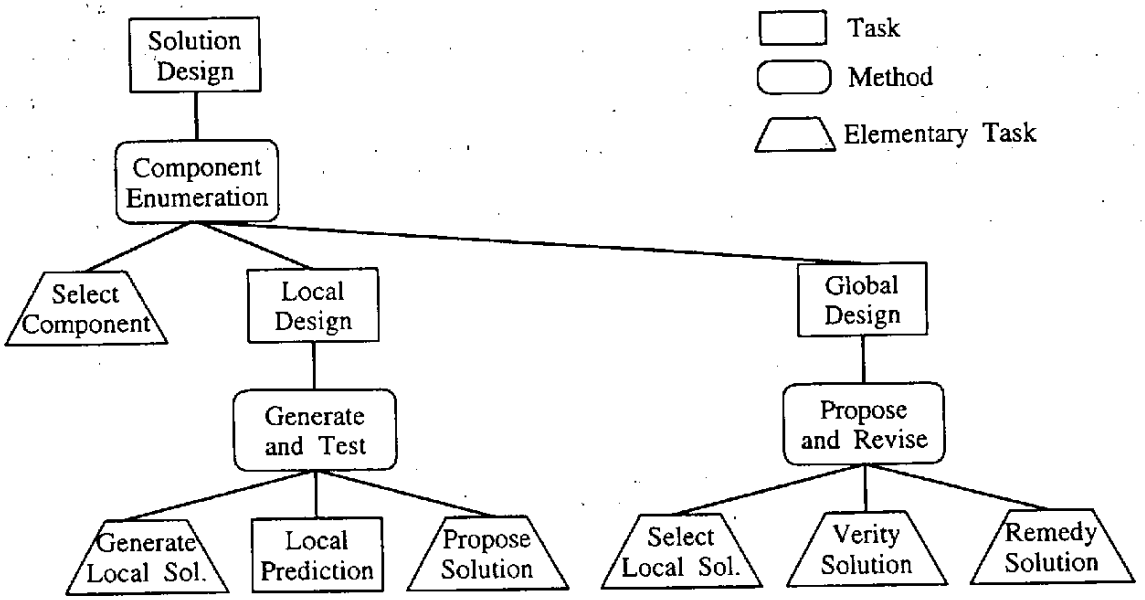


図２．１０-９　PROTEGE-II により記述された設計タスクの構造

（４）　Representation ontologies

Ontolingua［Gruber 1992］において提供されているフレームオントロジーには、Ontolingua による記述の処理方法が与えられており、ユーザが Ontolingua により記述したオントロジーを他のフレーム系の知識表現言語（例えば Loom）に変換することができる。すなわち、フレームオントロジーは表現に関する仕様を与えており、Representation ontologies の例となっている。

（５）　Application ontologies

Application ontologiesは、あるアプリケーションにおいて知識システムを動作させるために、（１）～（４）のオントロジーを合成したものと捉えられ、最近、開発されている知識獲得ワークベンチ（PROTEGE-II［Puerta et al. 1992］、CUE［G.Heijst 1995］）では、この合成プロセスを何らかの形で支援する環境を提供している。

2.10.6 おわりに

本稿では、自然言語理解、データベース、知識工学の３つの研究分野における、オントロジー捉えられ方と構築例について概説した、この分野以外でも、知識表現のコミュニティでは、論理を基礎にしたオントロジーの表現方法（Formal Ontologies）について議論が成されている。すなわち、現在、オントロジーは、図２.１０-１０に示すように４つの研究分野から、様々な観点により検討されているといえるが、これらのアプローチは、決して排斥しあうものではなく、相互に関連しながら発展が期待されるものであり、総合的なオントロジー工学が進展していくことを期待したい。

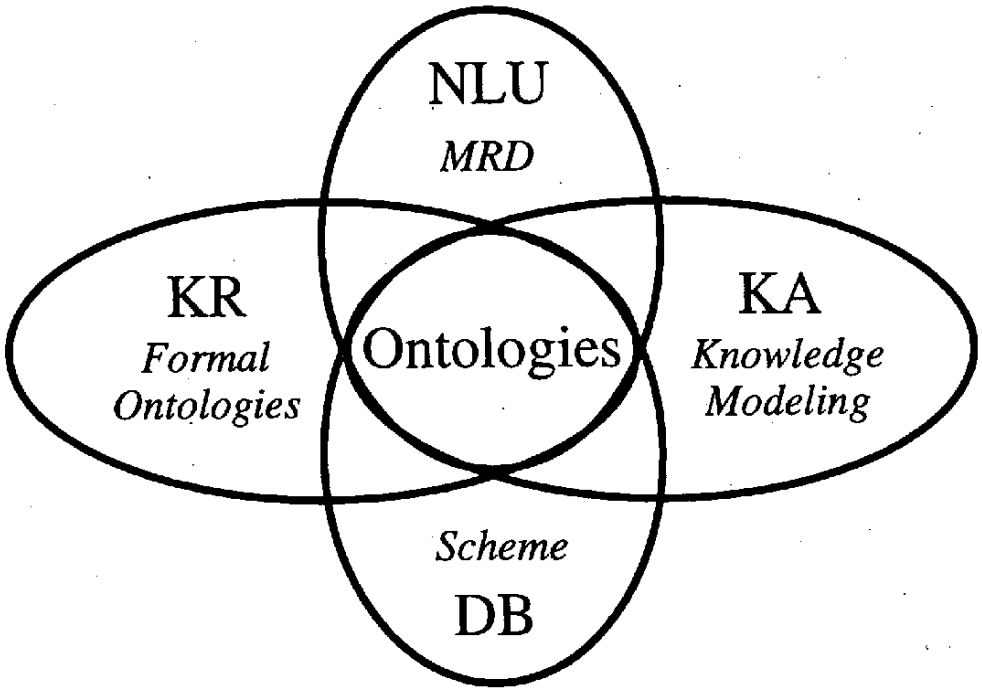


図２.１０-１０　オントロジーの諸相

参考文献

[Arens et al. 1996] Yigal Arens, Craig A. Knoblock, and Wei-Min Shen (1996). Query Reformulation for Dynamic Information Integration, Journal of Intelligent Information Systems, Special Issue on Intelligent Information Integration.

[Bateman 1990] Bateman, J.(1990). Upper Modeling: Organizing Knowledge for Natural Language Processing. Fifth International Workshop on Natural Language Generation

[J.Breuker and W.Van de Velde 1994] J.Breuker and W.Van de Velde edit (1994). Common KADS Library for Expertise Modeling. IOS Press

[Carlson & Nirenburg 1990] Carlson, L., and Nirenburg, S. (1990). World Modeling for NLP. TR CMU-CMT-90-121, Center for Machine Translation, CMU

[EDR 1993] Japan Electronic Dictionary Research Institute, LTD (1993). EDR Electronic Dictionary Technical Guide

[Gruber 1992] Gruber, T.R.(1992).Ontolingua: A Mechanism to Support Portable Ontologies, Version 3.0 TR, KSL, Stanford Univ.

[G.Heijst 1995] Gertjan van Heijst (1995). The Role of Ontologies in Knowledge Engineering. Dr. Thesis, University of Amsterdam.

[Hori, M. et al. 1994] Hori, M., Nakamura, Y., and Hama, T. Configuring Problem-Solving Methods: a CAKE Perspective, Knowledge Acquisition, Vol.6, pp.461-487

[Knight 1994] Kevin Knight and Steve K. Luk (1994). Building a Large-Scale Knowledge Base for Machine Translation. Proc. of AAAI94, 773-778

[Miller 1990] Miller, G. (1990). Wordnet: An On-line Lexical Database. International Journal of Lexicography 3(4) (Special Issue).

[Mizoguchi R. et al. 1992] Mizoguchi, R., Tijerino, Y., and Ikeda, M. (1992). Task Ontology and its Use in a Task Analysis Interview System - Two-Level Mediating Representation in MULTIS, Proc. of 2nd JKAW, pp.185-198

[Puerta et al. 1992] Puerta, A.R., Egar, J., Tu, S. W., and Munsen, M. A. (1992). A Multiple-method Shell for the Automatic Generation of Knowledge Acquisition Tools, Knowledge Acquisition, Vol.4, pp.171-196

[Swartout 1996] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ (1996). Toward Distributed Use of Large-Scale Ontologies.

[Tejada et al. 1996] Sheila Tejada, Craig A. Knoblock,and Steve Minton (1996). Learning Models for Multi-source Integration. AAAI Spring Symposium on Machine Learning in Information Access.

第2部　研究開発動向

第２部　研究開発動向

はじめに

第２部は、オントロジー関連技術の研究開発動向として、第１部 ２章（オントロジー工学確立の視点）の各視点のベースとなる諸研究について紹介している。既発表のものについては、標題ページ下にその出典を示す。

1. 自然法則の表現とオントロジー

1.1 はじめに

インテリジェントCADに関する研究テーマの一つとして、新しい人工物の発明や開発を行おうとしているエンジニアに対して、どのような種類の知識の提示が有効な支援となりうるかが興味の対象となる。このような知識にどのようなものが含まれるかを考えるに当たり、人工物の発明や開発の過程について考察することが重要となるであろう。

この問題に関して筆者らは現在次のように考えている。すなわち、エンジニアは想定されるニーズに対して、開発すべき人工物がどのような挙動を示すべきかを考えると同時に、そのような挙動の予測に役立つと思われる様々な法則を想起する。続いてこれらの法則が相互に組み合わせ可能な形に変形され、結果としてどのような人工物の挙動が実現できるかが検証される（図1.1-1）。ここで利用可能な法則としては、物理法則や効果など、いわゆる自然法則と呼ばれるものや、過去の実験結果から導かれた実験式などが挙げられる。これらの中で、自然法則は特に重要なものと考えることができる。

たとえば図1.1-2は電磁超音波による、金属の肉厚測定の原理を示すものである。あらかじめ強い磁場を設定した状況で高周波電流による変動磁場をかけると、金属表面に変動する渦電流が生じる。この渦電流と静磁場との相互作用により、超音波が発生するが、この結果生じた反対側の金属面の振動が、再び静磁場との相互作用により電流を生じ、結果として磁場の変動を引き起こす。従ってこの面での磁場変動と、反対面での電流付加の時間差を検出することにより、概略の肉厚を推定することができる。この方法は、完全な非接触で肉厚推定が可能であるという特徴を持つ。

電磁超音波の原理は、電流によって磁場が発生するというアンペールの法則、導体の回りの磁束密度の変動が起電力をもたらすというファラデーの法則、磁場の中の電流が力を受けるというローレンツ力、媒体の中の振動の伝達に関する波動方程式、磁場の中で導体を動かすことによって電流が生じる電磁誘導などの自然法則の組み合わせとして捉えることができる。ここでは、1.どのような人工物の振る舞いによって目的とする機能をを実現するか、2.そのような振る舞いに関連する自然法則にはどのようなものがあるか、3.それらの自然法則はどのような形で組み合わせることができる

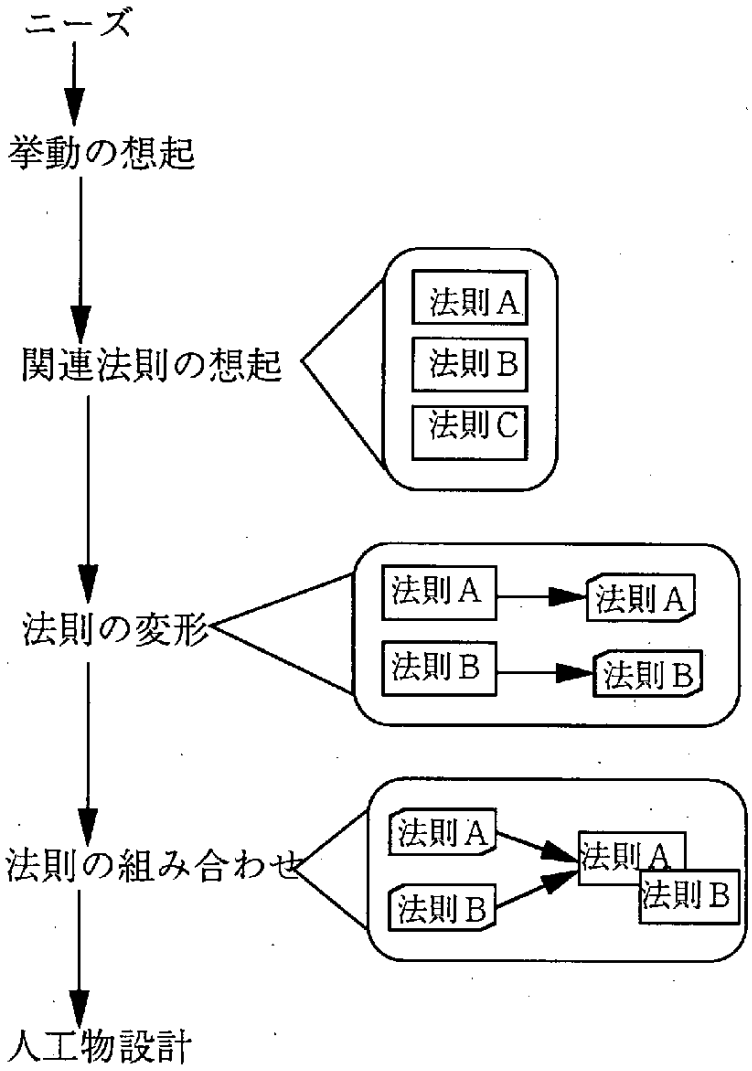
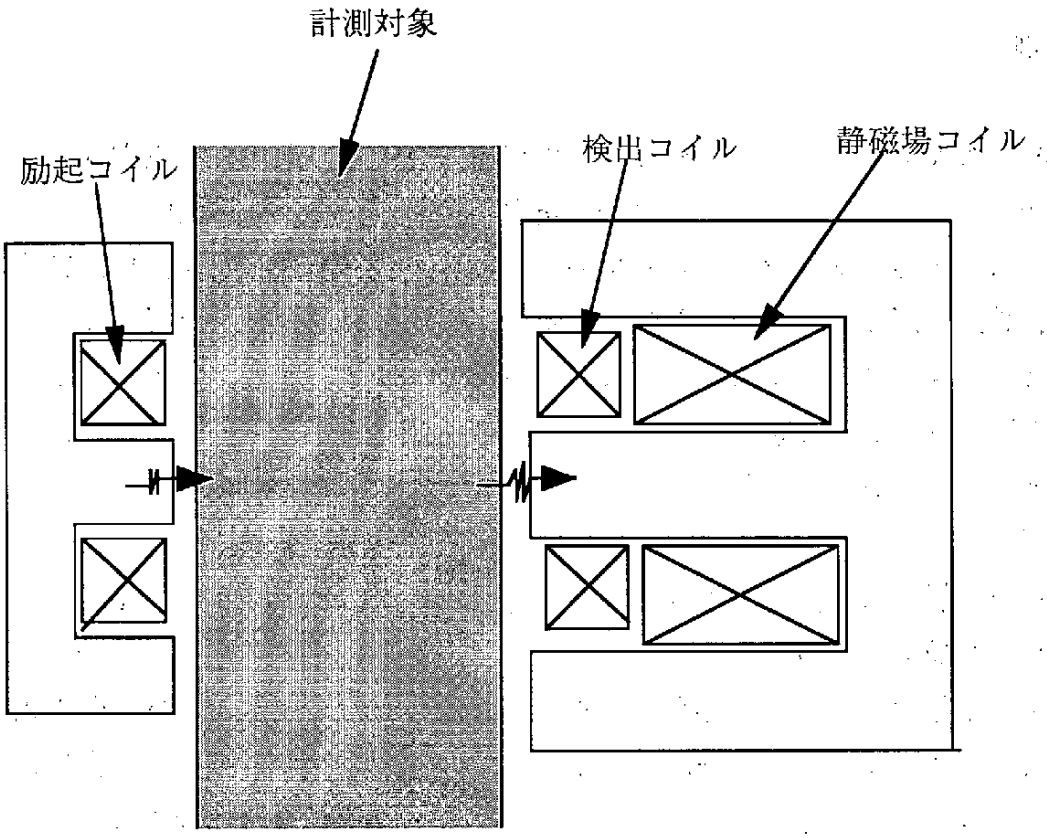


図1.1-1　人工物設計の流れ

か、を明らかにすることが主要な問題となる。

このような考え方に基づいて筆者らは、新しい人工物を設計しようとするエンジニアに対する情報支援として次のものが有効であると考えている。

1. 過去に行われた技術的発明や開発の事例集を用意し、提示すること。
2. 自然法則その他の形で得られている知識を基本知識として収集し、提示すること。
3. 複数の自然法則を適宜変形して組み合わせることにより、結果としてのものの挙動を予測して提示すること。



アンペールの法則 → ファラデーの法則 → ローレンツ力 → 波動方程式 → 電磁誘導

図1.1-2　電磁超音波厚み計測装置の原理図

1.では、様々な機能がどのような形でものの振る舞いによって実現されるかに関する知識を示すことができる。2.および3.では、ものの振る舞いをどのようにして予測できるかに関する知識が示される。

筆者らは、主として2.および3.の目的に利用するために既知の自然法則を機械処理可能な形で計算機内に蓄積する自然法則データベースに関する研究を進めている［田浦ほか 95］［小山ほか 95］。本論文ではこの研究に関連して現在までに得られた知見を紹介する。

1.2　自然法則

人類はその科学の歴史の中で、自然界の様々な現象について、再現性・規則性が存在することを明らかにしてきた。これらの内で整理の進んだものは物理法則や効果という自然法則として一定の記述を与えられている。これらの知識の多くは、自然言語による記述と、数式や図を用いて記述されてきた。また、実際このような記述は、人間が理解するという目的には有効な表現となっている。しかしながら、このような自然法則の記述を計算機によって検索、加工・操作しようとすると、その記述の不正確さ、あるいは不完全さが明らかとなる。

たとえば物理法則・効果では、それらが成立するための条件など、法則を定式化した観察者の視点を前提として定式化されているにも関わらず、一般の法則の記述からこのような視点を機械的に読みとることはしばしば困難である。人間は多くの場合このような不完全な記述から、法則の本質を読みとることにより、複数の自然法則を組み合わせることができるかどうかを判断し、法則記述のレベルをそろえた上で実際に組み合わせるという操作を行うことが可能であるが、このような操作を機械に行わせるためには、自然言語を中心にした記述の不備は大きな障害となる。

これまでに自然法則を相当程度網羅的に収集した例はあるが［Hix 58］、自然法則を成立させる前提としての視点まで含め、また、機械操作可能性を配慮した形で記述されたものはほとんどないといってよい。本研究では、自然法則記述の前提条件まで含めて、機械操作可能な形で自然法則を記述し、計算機に蓄積する方法について論じる。ここではまず、自然法則の表現としてどのようなものが必要となるかを明らかにする必要がある。すなわち、自然法則の記述に際して関連する様々な対象のオントロジーを決定するとともに、法則記述の基本的な枠組みを定める必要がある。

1.3　自然法則表現の要件

1.3.1　視点に関する制約の記述

自然界の様々な現象の多くは、複数の要因が相互に干渉しあう結果として現れる。従って、漠然とした自然観察のみに基づいてそこに存在する規則性や法則性を見出すことは、一般には困難である。幸い、自然界でいくつかの要因が集まって一つの現象を生じる場合の多くは、個々の要因による効果を加法的に重ね合わせることで十分に

説明できる。従って、自然法則を整理する一つの有力な方法は、基本的な要因となる現象を同定し、そのような現象が純粋な形で観測できる環境を設定した上で、そこに生じる現象の規則性・再現性を明らかとすることである。ここでは、法則が純粋な形で成立するための基本的な状況を設定することが要求される。これは対象とするものの見方に特定の制約を設けることに他ならない。

このような考え方に基づき、自然法則の発見者はものの見方に特定の制約を与えることによって、再現性が高く、簡潔な自然法則の記述を可能にしてきた。整理された形で表現される自然法則の背景には、常にこのような制約が存在するとみなさなければならない。自然言語と数式を用いて記述された自然法則の記述を人間が解釈する場合、このような制約の存在は多くの場合容易に読みとることができるが、計算機によってこれらの前提条件を機械的に読みとることは一般には困難である。自然法則データベースの構築に当たっては、これらの様々な制約を、機械可読な形で記述する必要がある。

このようなものの見方に関する制約には様々なものがあることに注意しよう。たとえばマックスウェルの方程式はアンペールの法則やファラデーの法則を一般化したものとなっている。ここにみられるのは、基本的に同一の現象であっても、ものの見方の制約を変えると、少なくとも表層的な記述のレベルでは一見異なった形を取ることがあるということである。

また、もう一つの場合として、たとえばアルキメデスの原理は、静水圧に関するパスカルの法則から導出できるというように、一つの法則が他の法則あるいは法則の組み合わせにより導出できる場合もある。

科学の立場から統一的な自然法則の理解をめざす場合、適用に当たっての制約がより少ない、一般的な形の法則が望ましいということが言える。この意味で科学の進歩は、より一般性の高い記述を発展させてきたと言えるであろう。

しかしながら一面では、工学的な立場から実際の状況に即して自然法則を適用し、対象の挙動を予測しようとする場合には、必ずしも一般性の高い記述が有利であるとは言えない場合がある。一般性の高い形での記述は、状況が限定されている場合には、むしろ解析を困難にする可能性もある。技術としての発明では、むしろ制約を限定することによって対象の挙動予測を容易にしてきたという側面も持っている。

従って今回目的とする自然法則データベースでは、ものの見方の制約に関する様々な一般性のレベルで記述された自然法則を並列的に収録すると同時に、制約の間の関係を取り扱うことができる枠組みが必要となる。

1.3.2 物理量の間の関係記述

自然法則の多くは、上記の制約の下での様々な対象に関わる物理量の間の関係として定義される。ここでもまた、記述のレベルにはいくつかのものが存在する。

このような関係を最もナイーブな形で表現したものとして、定性的な表現をまず考えることができる。たとえばニュートンの法則を「束縛されていないものに力を加えるとその速度が変化する」という形で定性的に記述するなどである。このような定性的な記述では、しばしばその記述から物理量の間の因果関係の存在類推させる場合もある。先のニュートンの法則では、「力が原因となって加速度を結果として生じる」という解釈もあり得るであろう。

一方で、厳密な再現性の検証が行われた自然法則では、対象とする物理量の間の関係が数式の形で整理されているものも存在する。このような物理量の間の関係式においては、先の定性記述にみられたような因果関係の類推は、直接的には導き出されない。ニュートンの法則における物理量間の関係を $F = m\alpha$ という形で記述する場合には、因果関係の類推には必ずしも結びつかないであろう。

一般性をめざす科学の立場からは、ここでも、物理量の間の関係をできる限り一般的な関係として定量的に捉えることが目標とされるであろう。一方、技術的な観点から自然法則を考える場合、定性的な表現も重要である。

新しい人工物の発明という観点からは、様々な自然法則が組み合わされた結果、どのような挙動が実現できるかが興味の対象となる。このとき、基本となる自然法則の記述レベルがあらかじめ一定のレベルにそろっているということは一般には期待できない。この場合には最も記述のレベルの低いものにあわせる必要があるが、最低限定性的関係が与えられていることが必要である。

定性的関係を因果関係として捉える見方は、一つには人間の直感的理解を助けるという意味もあるが、また、外部から操作しやすい物理量と操作しにくい物理量を区別するという意味あいも存在する。このことは、有効な法則の組み合わせの探索に当たってのヒューリスティックスとして利用できる可能性がある。

ここでも今回目的とする自然法則データベースでは、定量的な関係だけではなく、定性的な関係も併せて記述できることが要求されているということができるであろう。

1.4　自然法則データベースにおけるデータ表現

前節で述べたことから本研究で目的とする自然法則データベースでは、自然法則を記述する上で前提となる制約と、自然法則で定義される、関連する物理量の間の関係を記述するデータ表現が必要となる。

自然法則が前提とする制約を記述するためには、1.どのような実体が関係してくるか、2.それらの実体は空間的にどのように配置されているか、3.実体の周囲にどのような場を想定するか、4.実体と場の空間的関係はどのようになっているか、を明示的に示す必要がある。

また、物理量の間の関係を記述するためには、数式による関係と定性的な関係の双方について記述のできる枠組みを決定する必要がある。

自然法則データベースでは、これらに関連して記述する必要のある項目と、それらを記述する上で必要となる基本的語彙および記述シンタックスを決定する必要がある。

筆者らは、文献に収録されている自然法則の内、約150について検討を行った結果、図1.4-1に示すデータ表現を採用した。

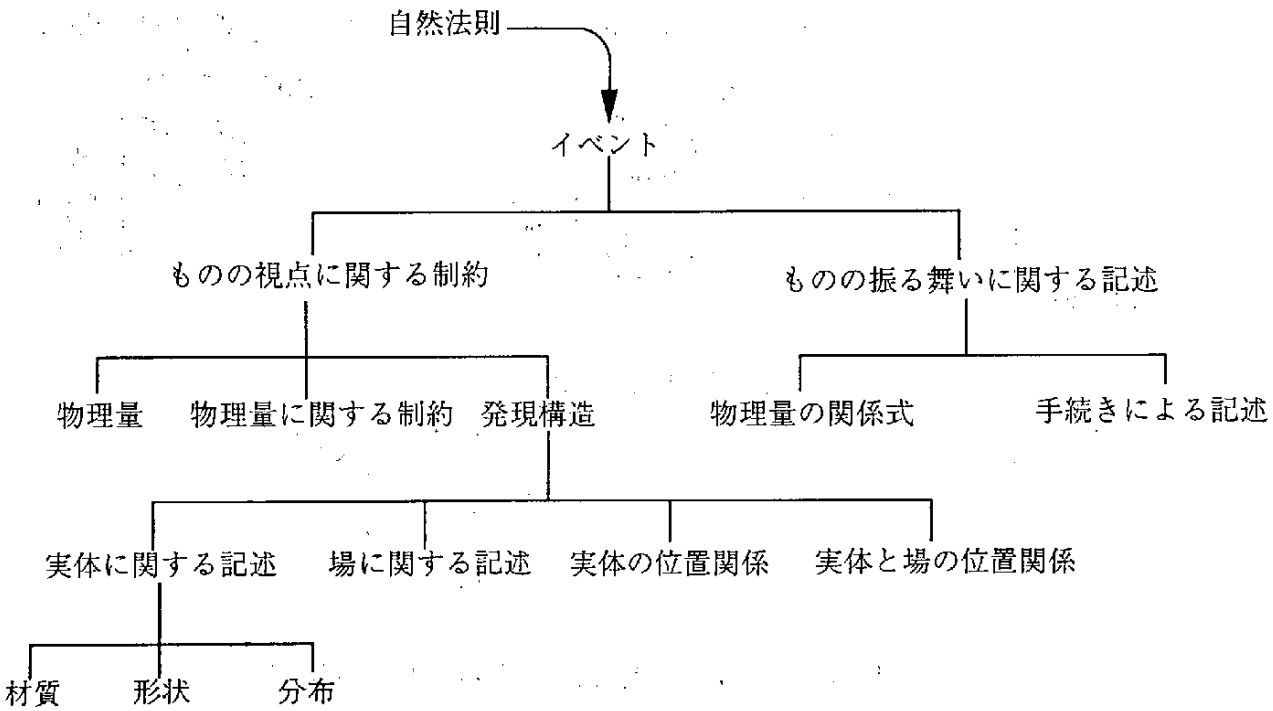


図1.4-1　自然法則のデータ構造の概念

まず、自然法則はある種の現象によって表現されるという考えから、イベントと呼ぶものを想定する。イベントはさらに、それに関連するものの視点に関する制約と、ものの振る舞いに関する記述から成り立っている。

ものの視点に関する制約には、どのような対象のどのような物理量が関与するか、物理量の範囲等に関する制限、ものに関する材質、形状、分布などの属性や、ものと場に関する配置などから成り立っている。また、ものの振る舞いに関する記述は、物理量の間の関係式または手続きによる記述を考えている。

これらの記述の内で、特にものの形状や分布、ものと場の位置関係などについては、いくつかの図式を用意することとし、これを語彙として用いることによって記述を行っている。図1.4-2にものの間の位置関係に関する図式を階層的に整理したものの一部を示す。これらの辞書は、自然法則に関係する様々な対象のオントロジーを定義したものと見ることができる。また、物理量の間の関係式については原則としてMATHEMATICAの表記によって表すこととし、手続きによる記述としては、ものや場に関する操作を表す述語を導入した上、C言語に準拠する手続きによって記述している。

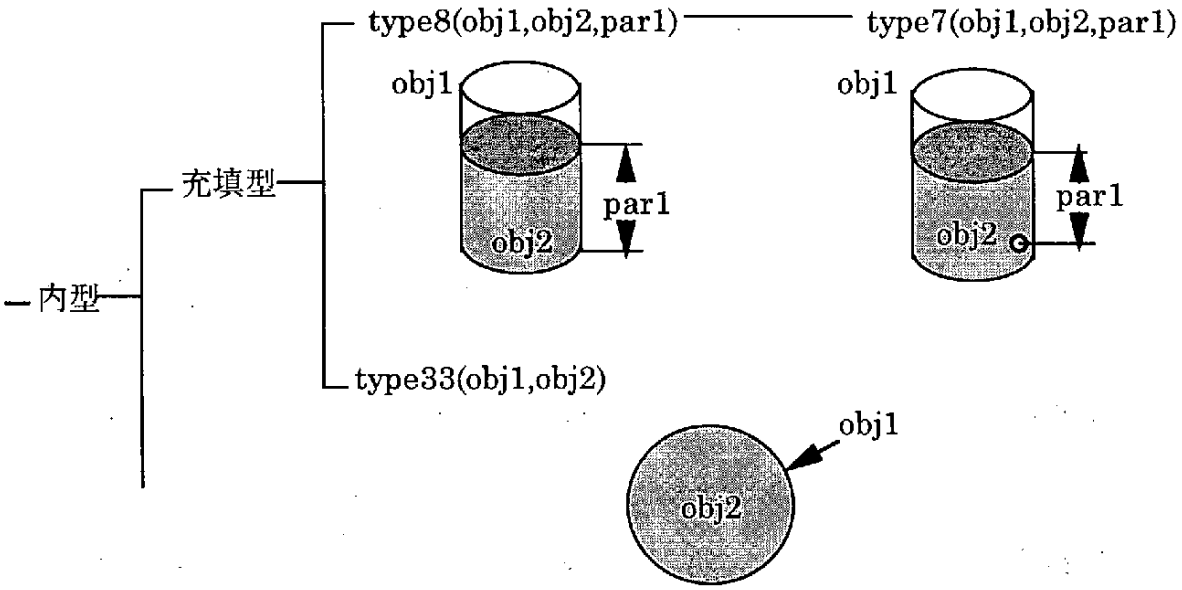


図1.4-2　ものの間の位置関係に関する図式

このような表現を考えるに当たって、記述の語彙としてどの程度のものを定義しなければならないかに興味がもたれる。約150の法則を検討した結果、図式として用意

した語彙の大きさを表1.4-1に示す。ここでは対象とする法則の数と比較して相対的に少数の語彙で記述が可能であることが示されており、ある程度一般性のある語彙の選定ができていると考えられる。

表1.4-1　種類毎に用意された図式の数

図式の種別	用意された図式の数
形状	23
分布	13
実体間の関係	39
実体と場の関係	7

1.5　オブジェクト指向データベースによる実装の試み

前節で考察した自然法則データベースのデータ表現を管理する枠組みとしてはいくつかのものが考えられるが、今回はオブジェクト指向データベース管理システム上での実装を試みた。前節のデータモデルは、様々な実体が相互に複雑に関係づけられており、オブジェクト指向データベースによる実装が有効であると考えられる。図1.5-1は今回実装したオブジェクト指向データベースにおけるオブジェクトモデル図である。このモデルに従って現在データベースの実装を進めている。図1.5-2は一部簡略表記した、このモデルに従ってクーロンの法則をデータベースに実装した例である。

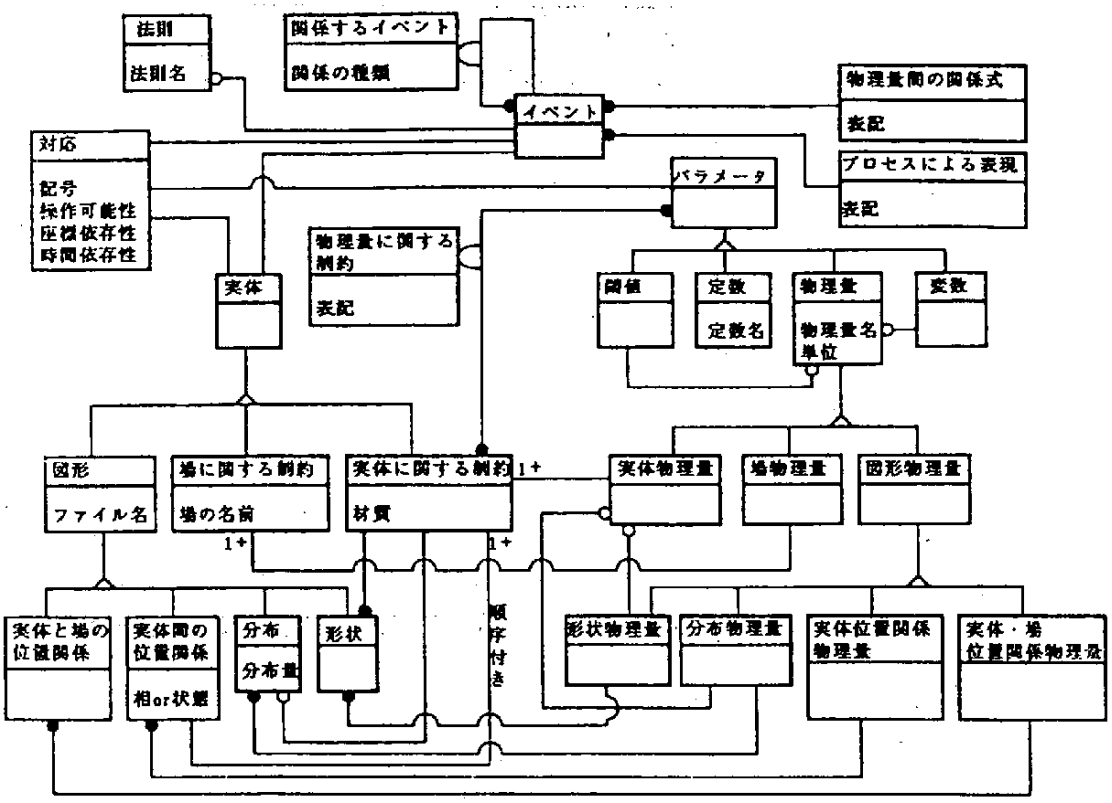


図1.5-1　自然法則のオブジェクト指向モデリング

```
class        法則
ID:                          110
name:                        Coulomb's_law
view point & action:   event10
-----------------------------------------
class        イベント

name:                        event10
related event:               none
related event constraint:    none
quantity bindings:     F1 | F::object10,unoperable,independent,_/
                             F2 | F::object11,unoperable,independent,_/
                             q1 | q::object10,operable,independent,_/
                             q2 | q::object11,operable,independent,_/
                             l | type11::l,operable,_,_/

epsilon | object12::epsilon,unoperable,independent,independent/
quantity constraint:   MATHE[F1 == -F2]/
object:                      object10/object11/object12/
object constraint:     type10
field declaration:     none
field object relation: none
relation equation:     MATHE[F1 == q1*q2/(4Pi*epsilon)(l/Abs[l]^3)
process:               none
-----------------------------------------
class        実体

name:                        object10
quantity:              F[force, N]/
                             q[electric_charge, C]/
material:              conductor
shape:                       shape10
destribution:                any
-----------------------------------------
class        実体

name:                        object11
quantity:              F[force, N]/
                             q[electric_charge, C]/
material:              conductor
shape:                       shape10
destribution:                any
-----------------------------------------
class        実体

name:                        object12
quantity:              epsilon[permittivity, F/m]/
material:              any
shape:                       any
destribution:                any
-----------------------------------------
class        実体間の位置関係

name:                        type10
filename:              Type_figure32
quantity:              none
object:                      object12 | type11/
mixing state:                none
-----------------------------------------
class        実体間の位置関係

name:                        type11
filename:              Type_figure2
quantity:              l[distance, m]/
object:                      object10 | object11/
mixing state:                none
-----------------------------------------
class        形状

name:                        shape10
filename:              Shape_figure1
quantity:              any
-----------------------------------------
```

図１.５-２　自然法則（クーロンの法則）実装例

1.6 考察と今後の課題

本研究では知的CADに関連して、自然法則を機械処理可能な形でデータベース化するためのデータモデルについて検討を行い、一部の自然法則についてはオブジェクト指向データベースの上に実装を行った。現在のところ、語彙として用意されている配置の下で特定の物質に関する特定の物理量が関連する法則の検索が可能であることを確認している。

知的CADの立場からは、データベースへのデータ蓄積と検索が可能であるというだけではなく、これらの情報を実際に組み合わせて操作可能かどうかに興味が持たれる。この問題については現在検討中であるが、実際の状況に対応して自然法則を適用するためには、自然法則自体を機械可読な形で蓄積するだけではなく、自然法則を適用する上で必要とされる知識を併せて用いる必要がある。

まず第一に、一つの法則で前提とされている状況が、別の法則で前提とされている状況と整合させることが可能かどうかの判断と、実際に整合させるための操作が可能でなければならない。このためには、まず第一には図式化された語彙の間に、どのような階層関係が存在するかが明らかにできる必要がある。次に、制約された状況の下で、物理量に関する元の関係記述がどのような形で成立するかが決定できなければならない。

また、実際問題として、データベースに蓄積されている法則が直接に仮定している状況と完全な整合性がとれる場合以外についての適用可能性を考える必要もある。このためには、いわゆる近似の成り立つ範囲に関する判断と、原状況を近似状況へ変換する方法が明らかにされる必要がある。

次に、異なる法則の間で物理量の間の関係記述のレベルをそろえる必要がある。ここでは定量的な関係式から定性的な関係式を導出する方法が確立されなければならない。

また、実際に与えられた複数の関係から実際の人工物の挙動を予測するためには、複数の関係を綜合して全体としての振る舞いを決定する仕組みが必要となる。ここではたとえば微分方程式の解法や、定性物理式の解法などに関わる知識が併せて必要とされることになる。

オブジェクト指向プログラミングの立場からは、以上述べてきた様々なデータ操作が、オブジェクトに埋め込まれたメソッドとしてどこまで実現可能であるかについても興味のもたれるところである。

今後は、データベース内容をいっそう充実させるとともに、蓄積された自然法則を実際に適用する枠組みに関する検討を進め、知的CADにおいて自然法則を利用する枠組みを確立していきたいと考えている。

参考文献

[田浦ほか 95] 田浦俊春, 小山照夫, 川口忠雄, 芝田京子, 叶迎春:「自然法則データベースの研究」, 第13回設計シンポジウム講演論文集, pp.75-82, 1995.

[小山ほか 95] 小山照夫, 田浦俊春, 川口忠雄, 叶迎春, 芝田京子:「自然法則データベースのためのオブジェクト指向モデリング」, 1995年度人工知能学会大会（第9回）論文集, pp.419-422, 1995.

[Hix 58] C.F.Hix, Jr. and R.P.Alley: Physical laws and effects, Jhon Willey & Sons, New York, Chapman & Hall, London, 1958.

2. 語義文からの階層関係の自動抽出

2. 語義文からの階層関係の自動抽出

2.1 はじめに

機械翻訳や対話理解など自然言語の機械処理で本格的に意味処理を行うためには、実用規模の意味辞書（広い意味でのシソーラス）が必要不可欠となる。このような意味辞書の開発では、単語の意味に関する情報を何から、どのような方法で収集し、どのように構造化するかということが重要な課題となる。

国語辞典には、実用規模の単語の意味・用法に関する多くの情報が含まれている。これらは、言語（語彙）に関する貴重な知識であり、特に、その語義文は見出し語（単語）の意味を記述したもので、単語の意味に関する最も有用な情報源の一つと考えることができる。

一般の市販の国語辞典では、スペースの制限で、語義記述を簡潔にするために、次のような辞書特有の方法が取られていることが多い。

(a) 適当な上位語（見出し語の表す概念の上位概念を表す語）を利用して、それの表す概念をいくつかの側面から制限する。

例①【鎌】草を刈る<u>農機具</u>。

(b) 同義語・言い換え語で示す。

例②【原爆】<u>「原子爆弾」</u>の略。

(c) いくつかの下位語（見出し語の表す概念の下位概念を表す語）で示す、

例③【乳幼児】<u>乳児</u>や<u>幼児</u>。

これら上位語、同義語、下位語などは、単語の意味を記述（定義）する場合の中心となる語と考えられるので、これらの語を定義語と呼ぶ。上記の例では下線で示してある。また、‘【…】’の中は見出し語、それに続く文は語義文（の文末部分）である。網かけは機能表現（後述）を示す。以下の例でも同様である。

ところで、単語間の階層関係（上位－下位関係、同義関係、全体－部分関係など）は、シソーラスにおいて最も基本的な意味情報の一つである。我々は、語義文のこの

ような記述的な特徴を利用して、語義文から定義語およびそれと見出し語との間の階層関係を抽出し、シソーラスへの応用を目指した研究を進めている。

このためには、次のようなことを明らかにしておくことが必要となる。

(1)定義語が語義文のどこに記述されているのか、
(2)見出し語と定義語との間にどのような階層関係が成り立つのか、
(3)その関係は何を手掛かりにどのように判定すればよいか。

本報告では、まず、(1)、(2)に関して、定義語の出現位置と機能表現の記述的な特徴に着目して、語義文の記述形式を5つの型に分類・整理している。次に、(3)に関して、定義語の抽出と見出し語との階層関係付けのための手順を提案している。さらに、これらを検証するために、実際に階層関係抽出システムを試作し、新明解国語辞典（三省堂）MRDの名詞見出し語の語義文で実験を行っている。

2.2 語義文の記述的特徴と機能表現

2.2.1 語義文の記述形式

語義文の中で定義語の出現位置や個数に着目した場合の記述的な特徴として、次のようなことが挙げられる。語義が複数の文で記述されている場合は、その第一文を対象にしている。

(1)語義文の末尾に定義語が現れる場合が多い。
(2)語義文の末尾が定義語でなく、特殊な表現が現れている場合がある。これを機能表現と呼ぶ。
(3)定義語が修飾部を持たない場合がある。
(4)複数の定義語が‘や’、‘と’、‘・’、‘または’などで結合されている場合がある。これらは並列の機能を持つので、便宜的に並列詞と呼ぶ。
(5)定義語の直後に例示助詞の‘など’が接続している場合がある。

これらの特徴から、記述形式を近似的に次の2つのタイプに分類している。

タイプⅠ：（語義文の末尾が定義語である）
語義文＝([修飾部]＋定義語)*。
例④【折尺】…たたんでしまっておけるものさし。
例⑤【スピン】旋回。
例⑥【親王】天皇の男の子または孫。
例⑦【建具】…つける戸・ふすま・障子など。
タイプⅡ：（語義文の末尾が機能表現である）
語義文＝([修飾部]＋定義語)*＋機能表現。
例⑧【瀑布】「大きな滝」の意の漢語的表現。
例⑨【青蛙】…に似た、大形のカエルの一種。
例⑩【海人草】マクリの薬品としての称。
例⑪【流体】気体と液体との総称。
例⑫【馬面】長い顔をあざけって言う語。
例⑬【御嬢様】相手の娘に対する尊敬語。

ここで、‘[…]’は任意要素、‘＋’は連接を示す。‘*’は並列詞や例示助詞が現れる場合があることを示す。‘([修飾語句]＋定義語)*’の構造をした文を基本的語義文と呼び、‘[修飾部]＋定義語’を語義文核要素と呼ぶ。

バッカス記法（BNF）を用いると、次のように書ける。

<語義文>::＝<基本的語義文>｜<基本的語義文><機能的表現>
<基本的語義文>::＝<語義文核要素>｜<基本的語義文><並立助詞><語義文核要素>
<語義文基本要素>::＝<定義語>｜<修飾部><定義語>｜<定義語><例示助詞>｜
　　<修飾部><定義語><例示助詞>
<機能的表現>::＝<句>｜<適当な記号を含んだ句>
<定義語>::＝<単語>
<修飾部>::＝<スペース以外の任意の文字列>
<並立記号>::＝ や｜と｜・｜または
<例示助詞>::＝など

機能表現については、次の２.２.２項で述べる。

２.２.２　機能表現の記述形式と機能

(1)　機能表現の記述的特徴と機能パターン

機能表現の記述形式とその機能を明らかにするために、新明解国語辞典MRDから約10,000個の語義文をサンプルデータとして抽出し、その記述形式について調査した。その結果、約1,500個の語義文が機能表現を含んでいるとみなせるものであった。機能表現は、多様な記述形態を取るが、一般に、それが表す意味関係と密接に関連する語（以下、機能語と呼ぶ。）を末尾に含んでおり、この機能語に助詞や補助用言、特殊な語句などが付属して構成されている。機能表現の記述形式を正規表現でモデル化したものを機能パターンと呼び、表２.２-１に示す。表２.２-１では、機能パターンの末尾に現れる機能語をFWと略記している。また、SSは基本的語義文を示す。機能パターンは、さらに細かく分類することも可能であるが、見出し語と定義語との階層関係を抽出すること中心に考えて、表２.２-１のように３つの型に大分類し、全体として５種の型に整理している。

3つの型の特徴は、次の通りである。

100型：「」を含み、」の後方に機能語が現れる。

200型：機能語の直前に‘の’が現れる。

300型：機能語が連体修飾されている。

語義文に現れる機能表現の取る機能パターンの型は、100型と200型がほとんどで、全体の約9割以上を占めている。100型は新明解国語辞典に特に多用されている。300型は数も少なく特殊な記述とみなせる。他の同様な国語辞典（例えば、岩波国語辞典、新潮国語辞典など）では、ほとんどが200型である。

表2.2-1　語義文の機能パターンの型と種類

型	機能パターン
100	「SS」＋σ＋FW。
200	
201	SS＋の＋FW。
202	SS＋W_{202}＋の＋FW。
300	
301	SS＋を＋σ＋W_{301}＋FW。
302	SS＋に対する＋FW。

注）σ：任意の文字列
W_{202}：として，、，の意の，、という意の，と
W_{301}：言う，いう，言った，呼ぶ，呼んだ，指す，指した

(2) 機能表現の表す意味情報と機能語

機能表現は見出し語と基本的語義文との間の意味関係を規定する機能をもっている。例えば、例④で機能表現“の一種”は、見出し語‘青蛙’と基本的語義文‘…に似た、大形のカエル’との間の下位－上位関係(<)を規定している。機能語は“一種”である。

また、例⑦で機能表現“の意の漢語的表現”は、見出し語‘瀑布’の語用に関する情報を表しているが、‘瀑布’の基本的な語義は、基本的語義文‘大きな滝’と同義であるとみなすことができる。このように、語用に関する情報を持つ機能表現には、見出し語と基本的語義文との間の同義関係(≡)を暗黙に規定しているとみなせるものが多い。例⑦の機能表現で、機能語は“漢語的表現”である。

ところで、機能表現には、次の例⑭や例⑮のような、部分ー全体関係(《)や全体ー要素関係(∋)などを表すものもある。

例⑭【水煙】搭の九輪の一部。
例⑮【天の川】…川のように見える星の群れ。

上述の上位ー下位、同義、部分ー全体、全体ー要素などの関係を階層関係と呼んでいるが、機能表現の中には、次の例⑮のように、階層関係以外の関係を表すものもあるので、便宜的に関連関係(R)を導入する。

例⑯【あひる】マガモの変種。

機能語としては、現在、約180種を求めている。表2.2-2に機能語の一部を示す。機能語は、それの表す意味情報に基づいて、次の2種類に分類している。

表2.2-2　機能語の分類

関係情報を持つ			語用情報をもつ		
FW	FP	ρ_{FE}	FW	FP	ρ_{FE}
一種	201	<	漢語的表現	100	≡
一品種	201	<		201	≡
種	201	<	略	100	≡
一つ	201	<		201	≡
一派	201	<	称	100	≡
				201	≡
類	201	>		202	≡
				301	≡
一部分	201	《	総称	201	≡
一部	201	《		202	≡
上部	201	《	語	301	≡
半分	201	《	尊敬語	100	≡
横がわ	201	《		201	≡
				301	≡
群れ	201	∋		302	≡
集団	201	∋	官庁用語	100	≡
				301	≡
変種	100	R	文字読み	100	≡
	201	R			
間	201	R	形	100	R
				201	R
部分	201	?	名	201	<R

[1] 見出し語と基本的語義文との間の階層関係を積極的に表しているもの、

[2] 見出し語の語用に関する情報を表しているもの。

表2.2-2では、それぞれの機能語に対して、それが含まれる機能パターン、およびその機能パターンによって見出し語(EW)と基本的語義文(SS)との間に仮定される意味的関係(ρ_{FE})が示されている。これらの情報は、後述の階層関係抽出システムで用いる機能語辞書の内容の一部である。

2.3　見出し語と定義語との間の階層関係

2.3.1　階層関係抽出のための仮定

定義語から見出し語と定義語との間の階層関係を求めるために、いくつかの仮定を置く。ここで、階層関係を表すのに前章で導入された記号を用いる。また、見出し語（の表す概念）をEW、語義文（の表す概念）をDS、基本的語義文（の表す概念）をSS、語義文核要素（の表す概念）をKS、定義語（の表す概念）をDW、定義語の修飾部をMDと表記する。なお、単語や句の表す概念を表示するために‘｛｝’を用いている。

（1）　見出し語と基本的語義文との関係

[1] 語義文がタイプⅠの場合、機能表現を含まない場合であり、基本的語義文と同じ構造である。近似的に見出し語の表す概念と同義な概念を表していると仮定する。

EW　≡　SS（＝　DS）

例：｛切り餅｝≡｛四角に切った餅｝

[2] 語義文がタイプⅡの場合、機能表現を含む場合であり、機能表現が見出し語と基本的語義文との間の意味関係(ρFE)を規定していると仮定する。

EW ρ_{FE} SS

例：｛青蛙｝＜｛…、大形のカエル｝

（∵FE＝‘の一種’，∴ ρ_{FE}＝‘＜’）

例：｛原爆｝≡｛原子爆弾｝

（∵FE＝‘の略’　，∴ ρ_{FE}＝‘≡’）

（2）　基本的語義文と定義語との関係

[3] 語義文核要素と定義語と間には、次のような関係を仮定する。

KS ≦ DW

(∵KS＝[MD]＋DW)

例：{草を刈る農機具} ＜ {農機具}

ただし、等号は修飾部(MD)がないときである。

[4] [3]を拡張して、基本的語義文と定義語との間に、次のような関係を仮定する。

SS ≦ (DW)*

(∵SS＝(KS)*＝([MD]＋DW)*)

ただし、等号は修飾部(MD)がないときである。

例：{…で取りつける戸・ふすま・障子など} ＜ {戸・ふすま・障子など}

[5] 並列助詞で結合された複数の定義語や例示助詞が付いている定義語（の表す概念）と、その構成要素になっている個々の定義語（の表す概念）との間には次の関係が成り立つと仮定する。

(DW) * ≧ DW

ここで、等号は、定義語が１個で、かつ例示助詞を持たない場合である。

例：{子または孫} ＞ {子}

例：{戸・ふすま・障子など} ＞ {戸}

例：{菓子など} ＞ {菓子}

2.3.2　階層関係の抽出手順

２.３.１の仮定と階層関係の推移律に基づき、次のような、見出し語(EW)と定義語(DW)との間の階層関係の抽出手順を提案する。

(Ｉ) 語義文がタイプＩの場合、

(a) DWが修飾部を持ち、

①DWが一個で，‘など’がなければ、　EW < DW
②DWが複数個か，‘など’があれば、　CD

(b) DWが修飾部を持たず、
①DWが一個で，‘など’がなければ、　EW ≡ DW
②DWが複数個か，‘など’があれば、　EW > DW

(Ⅱ) 語義文がタイプⅡの場合、
(a) DWが修飾部を持ち、
①DWが一個で，‘など’がなく、
ρ_{FE}が‘<’‘≡’であれば、　EW < DW
ρ_{FE}が‘《’‘ ’‘R’であれば、　EW ρ_{FE}DW
ρ_{FE}がそれら以外であれば、　CD
②DWが複数個か，‘など’があれば、　CD

(b) DWが修飾部を持たず、
①DWが一個で，‘など’がなければ　EW ρ_{FE}DW
②DWが複数個か，‘など’があれば、
ρ_{FE}が‘>’‘≡’であれば、　EW > DW
ρ_{FE}が‘》’‘ɜ’‘R’であれば、　EW ρ_{FE}DW
ρ_{FE}がそれら以外であれば、　CD

CD(Check Data)は、見出し語(EW)と定義語(DW)との間の関係に矛盾が生じる場合である。上記の抽出手順について、２、３の例で説明する。

例：(Ⅰ)(a)②の場合、
EW≡SS、SS<(DW)*、(DW)*>DWであるから、
EWとDWとの間の関係に矛盾が生じる。

例：(Ⅱ)(a)①の場合、
EW ρ_{FE}SS、SS<DW、となるが、ρ_{FE}が‘》’か‘ɜ’か‘R’であれば、ヒューリスティクスにより、EW ρ_{FE}DW が成り立つとしている。

例えば、例⑭では、{水煙}《{搭の九輪}、{搭の九輪}＜{九輪}の関係から、{水煙}《{九輪}の関係が成り立つと考えられるからからである。

例：(Ⅱ)(b)①の場合、

EW ρ_{FE}SS、SS＝(DW)*、(DW)*＞DW、となるので、(Ⅱ)(a)①の場合に準じて、ρ_{FE}が‘》’か‘∋’か‘R’であれば、EW ρ_{FE}DWが成り立つとしている。ただし、この場合は具体例がなかった。

2.4　見出し語と定義語との間の階層関係抽出システム

2.4.1　階層関係抽出システムの構成

システムの構成を図2.4-1に示す。各処理の概要は次の通りである。

(1)　基本的語義文と関係情報(ρ_{FE})の抽出

機能語辞書を用いて、入力された語義文のタイプ分けを行い、機能表現から得られる関係情報(ρ_{FE})の抽出と基本的語義文の抽出を行う。

ここで、語義文が、機能語を含むが機能パターンを満たしてない場合は、チェックデータ1 (CD1)として出力される。機能語と判定された語が定義語となる場合などをチェックするためである。

(2)　定義語と関係情報の抽出

単語辞書を用いて、基本的語義文から定義語を抽出する。動植物名が片仮名表記されていたり、複合語がより適切な定義語と見なされる場合があったり、定義語が複数個存在する場合などがあるので、それらを考慮して抽出を行う必要がある。また、関係情報として、定義語が複数個含まれているかどうか、定義語に修飾部があるかどうか、など情報を抽出する。

ここで、定義語が抽出されなかった語義文がチェックデータ2 (CD2)として出力される。

(3)　見出し語(EW)と定義語(DW)との関係の決定

抽出された定義語と関係情報をもとに、2.3.2の手順に従って、見出し語と定義語との関係を求める。

ここで、一意的に関係が決定できない場合、チェックデータ3 (CD3)として出力される。

以下、図2.4-1に従って、各処理の手順について述べる。

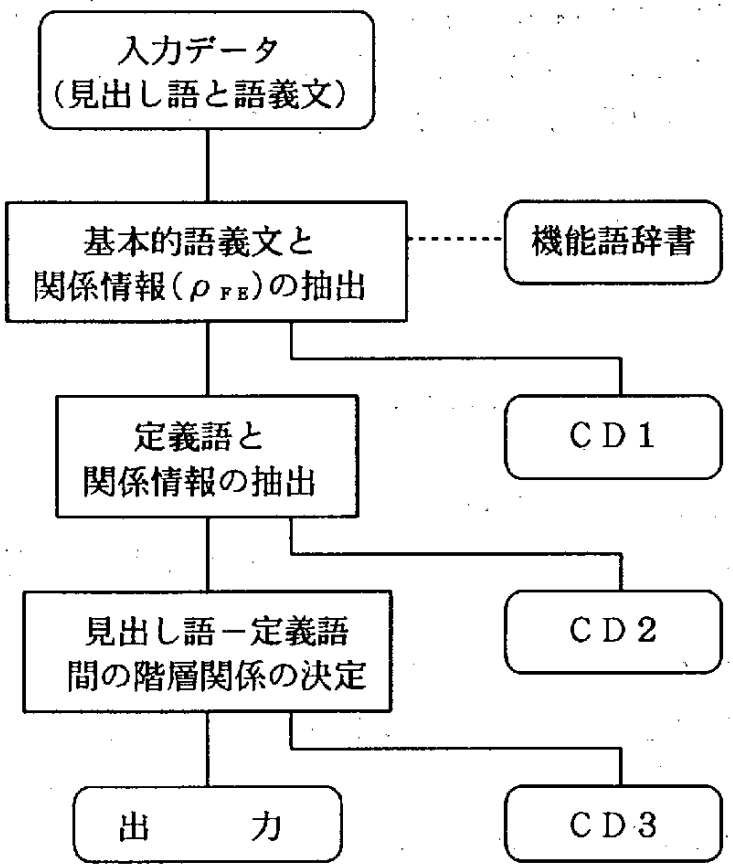


図2.4-1　階層関係抽出システムの構成図

2.4.2　入力データ

入力データは見出し語とそれに対応した語義文である。新明解国語辞典MRDから抽出されたものである。一つの見出し語が複数個の語義を持っている場合があるので、それぞれの語義に対応して見出し語と語義文の組を求めておく必要がある。

また、語義文には、括弧やドット（・）により複数の語義が一つの文となっていたり、漢字に読みが付加されていたりする場合があるので、それらを分離し、標準的な日本語文表現に変換しておくことも必要である。この処理を標準文変換と呼ぶ。

例⑰【伝令】…で、命令を伝える・こと（人）。

標準文　1.…で、命令を伝えること。

2.…で、命令を伝える人。

このため、語義番号を４つ組み(defg)で表している。それぞれ、dは新明解国語辞典での大語義番号、eは同じく小語義番号、fは語義記述で用いられている語義文の順序を示す番号、gは標準文変換の際の通し番号を示している。

２.４.３　基本的語義文と関係情報の抽出

（１）　機能語辞書

機能語辞書は次のような内容から構成されている。

（a）見出し語としての機能語

（b）機能語と組合せ可能な機能パターンの個数

（c）機能語と組合せ可能な機能パターンの型とその型に対応した関係情報（ρ_{FE}）

機能語辞書の内容の一部は、２.２.２の表２.２-２に示してある。ただし、機能語の判定は語義文の末尾の方から行うので、実際は逆引き辞書になっている。

（２）　基本的語義文と関係情報の抽出手順

Step 0 ：（語義文データのエラーチェック）

語義文入力データの末尾を検査し、‘。’でなければCD1として語義文を出力する。

Step 1 ：（機能語の判定）

機能語辞書を用いて、入力文（語義文またはStep 4からの再入力文）の末尾が機能語かどうかの判定　を最長一致法で行う。

① 入力文の末尾が機能語であれば、その機能語と組合わせ可能な機能パターンと関係情報(ρ_{FE})を機能語辞書から求めて、次のステップへ。

② 再入力文の末尾が機能語であれば、Step 6へ。

③ 末尾が機能語でなければ、Step 5へ。

Step 2 ：（機能パターンの判定）

機能パターン判定ルーチンを用いて、入力文の文末が機能パターンと一致するかどうかの判定を行う。このルーチンは２.２節の表２.２-１に基づいて作成している。

① 機能パターンであれば、次のステップへ。
② 機能パターンでなければ、Step 6 へ。

Step 3 ：（関係情報（ρ_{FE}）の抽出）

Step 2 で抽出された機能パターンとStep 1 で求められている機能パターンを比較して、一致しているものがあれば、それに対応した関係情報ρ_{FE}を抽出する。

① ρ_{FE}が一個であれば、次のステップへ。
② ρ_{FE}に曖昧さがあれば、Step 6 へ。
③ 一致するものがなければ、Step 6 へ。

Step 4 ：（再入力文の生成）

入力文から機能パターンを除去して、残りの部分を再入力文として、Step 1 へ。

Step 5 ：（基本的語義文の抽出）

Step 1 の③の条件を満足した文を基本的語義文として抽出する。入力文の末尾が機能語でない場合である。

Step 6 ：（チェックデータ 1 (CD1)の出力）

Step 2 の②、Step 3 の②と③、Step 1 の②の場合、入力文がチェックデータ 1 (CD1)として出力される。

図２.４-２に、チェックデータ 1 (CD1)の出力例を示す。図２.４-２には、読みと正書法からなる見出し語(EW)、語義の区別のための４組の語義番号(defg)、抽出された機能語(FW)、機能パターン(FP)の型、関係情報ρ_{FE}、および、チェックの理由がC1に示されている。C1は、定義語と機能語との機械的な区別が困難な場合や、機能表現の表す関係情報(ρ_{FE})に曖昧さがある場合である。後者の場合は、機能語辞書で複数の関係情報か、または、？の記号が与えられている（表２.２-２参照）。なお、入力文の区切りとして‘。’がない場合のデータエラーをチェックしている。

出力データの形式は、チェックの理由を示すC2の部分を除けば、２.４.４項の図２.４-３と同じである。

語　義　文	見出し語	(defg)	機能語	FP型	ρ_{FE}	C1
にさらしたあと、その刺激光を取り	りんこう【燐光】	(0210)				1
の相対する二つの辺が平行な四辺形。	だいけい【台形】	(0010)	形			2
「声喩・擬音語」の総称。	オノマトペ【オノマトペ】	(0010)	総称			3
の側面で、腕のつけねの下側の部分。	わき【脇】	(0110)	部分	201	?	4
生きていた時の名。	ぞくみょう【俗名】	(0210)	名	201	<R	4
人生の前の半分。	ぜんはん【前半生】	(0010)	半分	201	(	5

(注)　(defg)：語義番号情報(d:大語義番号，e:小語義番号，f:文番号，g:標準文変換の際の通し番号)，
FP型：機能表現がある場合，その機能パターンの型，
ρ_{FE}：関係情報(機能表現が表す意味的関係[EW ρ_{FE} SS])，
C1：チェックの理由
1；入力データエラー(語義文の末尾に‘。’がない)．
2；機能語はあるが機能パターンがない．
3；機能パターン判定処理ルーチンで求まった機能パターンが機能語辞書の内容にない．
4；抽出された関係情報(ρ_{FE})が曖昧である，即ち，複数個あるか，または‘?’である．
5；機能語が複数個ある．

図2.4-2　チェックデータ1(CD1)の出力例

2.4.4　定義語と関係情報の抽出

(1)　一般用語の逆引き辞書

基本的語義文から定義語を抽出するために逆引き単語辞書を用いている。この逆引き辞書は新明解国語辞典の名詞見出し語（約55,000語）から作成したものである。定義語は、基本的には、この逆引き辞書を利用して、基本的語義文の末尾語との最長一致法で求める。

(2)　定義語と関係情報の抽出手順

Step 1：（漢字列または片仮名列の抽出）

入力文（基本的語義文またはStep 6からの再入力文）の末尾が漢字列か片仮名列であれば、それを抽出する。抽出された文字列の文字数をnとする。

Step 2：（入力文の末尾語の抽出）

逆引き辞書により、入力文の末尾語を抽出する。抽出された語の文字数をmとする。

Step 3：（チェックデータ2(CD2)の出力）

n＝m＝0であれば、入力された基本的語義文がCD2として出力される。即ち、この入力文の末尾語が、片仮名列や漢字列でなく、逆引き辞書に登録されていない場合であり、定義語が抽出されないからである。

Step 4 ：（定義語の抽出）

① n ＞mであれば、Step 1 で得られた文字列を定義語とし、mと共にスタックする。

② n ≦mであれば、Step 2 で得られた文末語を定義語とし、スタックする。

Step 5 ：（関係情報の抽出と再入力文の生成）

定義語の直前に並列詞（‘・’、‘や’など）があるかどうかを判定する。

① 並列詞があれば、それから文頭側の部分を再入力する（Step 1 へ；残りの定義語の抽出のため）。関係情報として、複数定義語が含まれていると判定する。

② 並列詞がなければ、修飾部の有無を調べる。修飾部がなければ、それを関係情報として、次のステップへ。修飾部があれば、関係情報として修飾部があると判定し、さらに、適当な文字数を設定し文頭側の並列詞を調べる。並列詞があれば、チェックデータ(CD2)として入力文が出力される。並列詞がなければ、次のステップへ。

Step 6 ：（定義語の出力）

Step 4 でスタックされていた定義語は、入力された基本的語義文単位にまとめて、次の関係抽出の処理へ渡される。出力データの形式は、チェックの理由を示すC3の部分を除けば２．４．５の図２．４-４と同じである。

図２．４-３に、チェックデータ(CD2)の出力例を示す。C2にチェックの理由が示してある。C2の‘２’と‘３’は語義文の記述的な特徴を利用している。C2の‘２’は、Step 5 の②からのチェックデータで、定義語が複数個あるかどうかの関係情報の抽出ミスをチェックするためである。実験システムでは、文字数を５としている。また、C2の‘３’は、タイプⅡの入力文で、基本的語義文の末尾に「時」が形式名詞として使われている場合があったからである。

基本的語義文(SS)	見出し語(EW)	(defg)	機能語	FP型	ρ_{FE}	C2
ボートのこぎて	そうしゅ【漕手】	(0010)				1
かねじゃく	きょくしゃく【曲尺】	(0010)	文字読み	100	≡	1
計量・評価の標準	しゃくど【尺度】	(0010)		100	≡	2
あとつぎ・あととり	けいし【継嗣】	(0010)	漢語的表現	100	≡	2
自分の母親を他人に対して言う時	おふくろ【御袋】	(0010)	称	201	≡	3

(注) (defg), FP, ρ_{FE}, は, 図2を参照.
C2：チェックの理由
1；定義語が抽出されない.
2；定義語から文頭側の5文字以内に並列詞('・', 'や', 'と', 'または')がある.
3；関係情報(ρ_{FE})があり, かつ, 抽出された定義語が『時』である.

図2.4-3 チェックデータ2(CD2)の出力例

2.4.5 見出し語と定義語との意味関係の判定

2.4.3項で求められた関係情報（語義文のタイプと関係情報ρ_{FE}）と、2.4.4項で求められた関係情報（定義語が複数個あるかどうか、および、定義語に修飾部があるかどうか）を用いて、2.3.2項の手順に従って、見出し語と定義語との間の意味（階層）関係の判定を行なう。

図2.4-4にチェックデータ3(CD3)の出力例を示す。C3にチェックの理由が示されている。

図2.4-5に階層関係抽出システムの出力例を示す。図2.4-4には、抽出された定義語（KWIC形式で示されている)、決定された階層関係などが示されている。ここで、決定された関係rは、図を見やすくするために、'DW r EW'の関係を示している。従って、関係情報ρ_{FE}と逆の関係を表していることになる。ρ_{FE}は、'EW ρ_{FE} DW'の関係を表しているからである。

基本的語義文の残り	定義語	見出し語	(defg)	機能語	FP型	ρ_{FE}	β	γ	δ	C3
…人々の間に語り伝えられた	：昔話	こうひ【口碑】	(0010)				◇			1
…、雨を防ぐための	：衣類・道具	あまぐ【雨具】	(0010)				◎			1
一つの物権と見なされる、	：財産	ざいだん【財団】	(0010)	集団	201	∋				2
陰電気を帯びた、	：分子・原子	いんイオン【陰イオン】	(0012)	群れ	201	∋	○		ま	3
…べることを禁じられた	：魚肉・獣肉	なまぐさもの【生臭物】	(0010)	類	201	>	○		・	3
田畑に作る	：穀物・野菜	さくもつ【作物】	(0010)	総称	201	≡	○		や	3

(注) (defg), FP, ρ_{FE}, β, γ, δは, 図2～図5を参照. 定義語の区切り(：), 複数定義語の切れ目(・).
C3：チェックの理由
1；機能表現がなく, 定義語に修飾部があり, かつ複数定義情報がある.
2；機能表現があり, 定義語に修飾部があり, かつ定義語が1個で, ρ_{FE}が>, 》または∋である.
3；機能表現があり, 定義語に修飾部があり, かつ複数定義情報がある.
4；機能表現があり, 定義語に修飾部がなく, かつ複数定義情報があり, ρ_{FE}が<, 《または∈である.

図2.4-4 チェックデータ3(CD3)の出力例

基本的語義文の残り	定義語(DW)	r	見出し語(EW)	(defg)	機能語	FP型	ρ_{FE}	β	γ	δ
	円周	》	えんこ【円弧】	(0010)	一部分	201	《		☆	
…近い、	肩	》	かたさき【肩先】	(0010)	上部	201	《			
	手首	R	かはく【下腕】	(0010)	間	201	R		☆	と
	ひじ	R	かはく【下腕】	(0010)	間	201	R		☆	と
	土地	<	ふどうさん【不動産】	(0010)				◎	☆	や
	建物	<	ふどうさん【不動産】	(0010)				◎	☆	や
	土木建築	≡	どけん【土建】	(0010)	略	100	≡		☆	
	正味	≡	ネット【ネット】	(0210)					☆	
	写真感光板[3]	>	かんぱん【乾板】	(0010)	一つ	201	<		☆	
相手の	役人	>	きかん【貴官】	(0010)	官庁用語	301	≡			
…古来の	音楽	>	ほうがく【邦楽】	(0010)						

(注) (defg), FP, ρ_{FE}, は，図２を参照.
r：決定された意味的関係（'DW r EW'を示す．ρ_{FE}と逆の関係である[∵EW ρ_{FE} SS]．）
β：関係情報(◇;定義語の直後に'など'がある，○;抽出された定義語が複数個である，
◎：定義語の直後に'など'があり，かつ抽出された定義語が複数個である．）
γ：関係情報(☆;定義語に修飾部がない)
δ：語義文中での並列詞の表記字（'ま'は'または'を示す．）
[] で囲まれた数：単語辞書でマッチした文字数[m]．（'写真感光板[3]'の３）．

図２.４-５　階層関係抽出システムの出力例

2.5 実験結果と考察

実験用のデータとして、3,036文の語義文を用いた。これらは、新明解国語辞典の名詞見出し語の語義文の中から任意に選らばれたもので、標準文変換されている。本実験システムは長崎大学情報処理センターFACOM M760上に、PL/Iで実現されている。

(1)表2.5-1、表2.5-2、表2.5-3、表2.5-4に実験結果を示す。

＊表2.5-1は、各段階での入力データとチェックデータを調べたものである。これより、

[1] 入力文3,036文の中で、データエラー（末尾に'。'を持たないもの）が17文(0.6%)であった。

[2] 入力文3,036文の中で、基本的語義文が抽出された語義文は2,903文(95.6%)であり、その中で、

① タイプⅠの語義文が2,583文（約89.0%）

② タイプⅡの語義文が　320文（約11.0%）

[3] 基本的語義文2,903文の中で、定義語が抽出されたものが2,593文(89.3%)であった。入力文3,036文に対しては85.4%である。

[4] 入力文3,036文に対して、チェックデータの総数が482文(15.9%)であった。これより、2,554文(84.1%)に対して関係付けが行われたことになる。

表2.5-1　各段階での入力とチェックデータの数
（データ数は語義文単位である）

	入力データ	チェックデータ
(1)基本的語義文の抽出	3,036	DE : 17 FW有り : 116 計 133
(2)定義語の抽出	FE無し : 2,583 FE有り : 320 計 2,903	FE無し : 245 FE有り : 65 310
(3)階層関係の決定	FE無し : 2,395 FE有り : 274 計 2,593	FE無し : 36 FE有り : 3 39

注）DE：データエラー，　FE：機能表現

表2.5-2　複数定義語を含む語義文の調査

(a)階層関係決定されたデータ

		出力データ（語義文数）			抽出されていた定義語数
一文当たりの定義語数		1	2	3	
出力データ	FE無し	2,280	20	2	2,326
	FE有り	237	13	2	269
	計	2,517	33	4	2,595
		2,554			

(b)チェックデータ3(CD3)

		チェックデータ3（語義文数）			抽出されていた語義文数
一文当たりの語義文数		1	2	3	
CD3	FE無し	5	29	2	69
	FE有り	1	2	0	5
	計	6	31	2	74
		39			

＊表2.5-2は、階層関係付け処理の入力文(2,593文)で、定義語を複数含むもの、および、それらの文から抽出された定義語数を調査したものである。関係付けが行われた語義文(2,554文)とチェックデータ(39文)のそれぞれについて表2.5-2(a)、表2.5-2(b)に示している。

[5] 表2.5-2(a)から、2,554文の入力文の中で、複数定義語を含む語義文が37文であり、関係付けが行われた定義語が2,595個であった。

＊表2.5-3は、抽出された定義語の妥当性について調査してものである。これより、

[6] 定義語が抽出された語義文2,593文から抽出された定義語は、2,669語であった。そのうち、妥当な定義語は2,558語(95.8%)であった。

[7] 抽出された定義語2,669語に対して、階層関係付け処理で、関係付けが行われた定義語は2,595語(97.2%)であった。

[8] 関係付けされた定義語2,595語の中で、妥当な定義語は2,486語(95.8%)であった。誤りは109語(4.2%)であるが、この中には、データエラーの８個が計数されている。

表２.５-３　階層関係決定処理入出力データの定義語数

	入力データ（語義文数）		定義語数 妥当	誤り	計
出力データ	FE無し：	2,302	2,235	91	2,326
	FE有り：	252	251	18	269
	計	2,554	2,486	109	2,595
チェックデータ	FE無し：	36	67	2	69
	FE有り：	3	5	0	5
	計	39	72	2	74
計	FE無し：	2,338	2,302	93	2,359
	FE有り：	255	256	18	274
	計	2,593	2,558	111	2,669

注）FE：機能表現

表２.５-４　関係付けられた妥当な定義語データの内訳

関係	妥当		誤り		小計		計
EW：DW	機能表現 無し	有り	機能表現 無し	有り	機能表現 無し	有り	
<	1,996	76	0	1	1,996	77	2,073
>	38	30	6	0	44	30	74
≡	195	126	0	0	195	126	321
⟨		8	—	1	—	9	9
∋	—	0	0	0	0	0	0
R	0	9	0	0	0	9	9
小計	2,229	249	6	2	2,235	251	2,486
計	2,478		8		2,486		

EW：見出し語，　　DW：定義語

＊表２.５-４は妥当な定義語に対する階層関係付け出力データの内訳である。これより、

[9] 妥当な定義語2,486語に対して、関係付けが妥当とみなせるものが2,478語(99.7%)であった。誤り、または、曖昧と思われるものが、8語(0.3%)であった。

[10] 表２.５-３と表２.５-４から、関係付けの処理がなされた定義語(2,595語)に対して、妥当な関係付けがなされた定義語は、2,478語(95.5%)であった。従って、システムの解析精度は、約95.5%であるといえる。

[11] 階層関係抽出システムの出力データの誤りは、117語(4.5%)であった。誤りのほとんどが定義語抽出で起こっている。例えば、平仮名書き定義語(やきいも)や接頭語‘お’(お金)などの抽出ミスがあった。関係付けでの誤りとしては、上位－下位関係と全体－要素関係の区別の曖昧さがあった。また、国語辞典MRD化でのミスなどもあった。

（２） 定義語抽出での問題点として、次ぎのようなことが挙げられる。

[1] 定義語が仮名書きされているものがある。

平仮名見出しを登録したとしても最長一致法では、妥当な定義語の抽出には限界がある。

[2] 義語が‘こと’や‘もの’として抽出される場合が多い。今回の実験では、‘こと’が411個、‘もの’は103個抽出されていた。末尾にこのような語を持つ語義文では、より適切な定義語が別の位置にあったり(例⑱)、または、語義を記述した第二文以降にあったり(例⑲)することがある。

例⑱【刊記】…古い出版物で、…書いたもの。

例⑲【装置】…すること。また、その設備。

[3] 特殊な記述がなされている場合がある。

例⑳【所在】どこにあるか。

2.6 おわりに

本稿では、国語辞典の語義文の構造的特徴を十分に活用して、単語間の意味的関係の一つである階層関係を機械的に求めるための階層関係付けシステムの概要およびその実験結果について報告した。

同様の研究が、いくつかのグループでも進められて来ている。

シソーラスへの応用に関して、次のようなことも行っている。

[1] チェックデータからの定義語の抽出や関係付けを行うための支援システムの作成。

[2] [1]を利用した、定義語への読みや語義番号の付加。および、複合語などの処理。

今後の課題として、単語の意味記述の形式化へ向けて、語義文の文末表現のみならず文中に現れる関係表現とそれが表す意味情報を抽出・整理することが必要である。また、機能表現の表す意味的関係をさらに精密化することも大切である。

大規模機械辞書構築への国語辞典の応用などに関連して、人間用・機械処理用を兼ねた源辞書の考え方が提案されている。従って、国語辞典の情報解析に関する研究は、ますます重要になるものと考えられる。

参考文献

[金田一 81] 金田一(京), 金田一(春), 見坊, 田, 山田：新明解国語辞典, 三省堂, 第2版 (1974), 第3版 (1981)

[中野 81] 中野 洋: 分類番号付け支援システム, 情報処理学会計算言語学研究会資料 25-5 (1981)

[長尾 82] 長尾 真: 言語辞書活用のための計算機プログラムシステムの開発と言語辞書の解析, 昭和55, 56年度科研費研究成果報告書 (1982)

[Yoshida 82] S.Yoshida, H.Tsurumaru, T.Hitaka: MAN-ASSISTED MACHINE CONSTRUCTION OF A SEMANTIC DICTIONARY FOR NATURAL LANGUAGE PROCESSING, Proc.of COLING '82, pp.419-424 (1982)

[Michiels 82] A.Michiels, J .Noel: APPROCHES TO THESAURUS PRODUCTION, Proc. of COLING'82, pp.227-232 (1982)

[横山 84] 横山, 荻野: 国語辞典磁気テープのドキュメント, 電総研彙, Vol.48, No.8, pp.672-677 (1984)

[鶴丸 84a] 鶴丸, 内田, 日高, 吉田: 国語辞典からの情報抽出と構造化, 情処学自然言語処

理研資43-6 (1984.5)
[鶴丸 84b] 鶴丸, 水野, 内田, 日高, 吉田: 単語の釈義文を利用した単語間の階層関係の抽出について, 情処学自然言語処理研資45-4 (1984.9)
[辻井 84] 辻井潤一: 機械翻訳と知識処理 :「語」か「概念」か？, 情報処理学会自然言語処理技術シンポジウム論文集, pp.209-212 (1984)
[辻井 85] 辻井潤一: 辞書の構成と機械翻訳, 情報処理, Vol.26, No.10, 1985.
[吉田 86] 吉田　将: 辞書構築における諸問題, 情報処理, Vol.28, No.8, pp.933-939 (1986)
[鶴丸 86] 鶴丸, 日高, 吉田: 単語間の上位－下位関係の自動抽出, 情処学研報 (FI), Vol.86, No.3,pp.1-8 (1986)
[中村 87] 中村, 酒井, 長尾: 英々辞典を用いた名詞の意味関係の分析, 信学会NLC86-23 (1987)
[荻野 87] 荻野綱男: シソーラス作成の問題点, 日本語学, Vol.6, No.5, pp.64-71 (1987)
[鶴丸 87] 鶴丸, 兵頭, 松崎, 日高, 吉田: 語義を考慮した単語間の階層構造の抽出について, 情処研報 (NL), Vol.87, No.84, pp.9-16 (1987)
[富浦 87] 富浦, 日高, 吉田: 語義文からの動詞の階層関係の抽出, 情処研報, Vol.87, No.84, pp.17-24 (1987)
[田中 87] 田中, 仁科: 上位／下位関係シソーラスISAMAP1の作成[Ⅰ], [Ⅱ], 情処研報 (NL), Vol.87, No.84, pp.25-44 (1987)
[内田 88] 内田裕士: 電子化辞書の開発, 情報処理学会自然言語処理技術シンポジュウム論文集, pp.89-98 (1988)
[徳永 89] 徳永, 奥村, 田中: 概念階層への視点の導入, 情処学論, Vol.30、No.8、pp.970-975 (1989)
[長尾 92] 長尾　真: 辞典形式での専門分野の知識の体系的構成法, 人工知能学会誌, Vol.7, No.2, pp.136-144 (1992)
[黒橋 92] 黒橋, 長尾, 佐藤, 村上: 専門用語辞典の自動的ハイパーテキスト化の方法, 人工知能学会誌, Vol.7, No.2, pp.152-161 (1992)

3.　言語学が貢献できること

3. 言語学（言語者／外国語教師）が貢献できること

3.1 はじめに

インターネット環境の爆発的な普及が、言語学者あるいは外国語教師がオントロジー工学あるいは、様々な応用・実用化に貢献できる可能性を大きく広げたと考えられる。ここでは、考えられる具体的な可能性と、それらの可能性を追求した場合に得られる成果からさらに得られるであろうオントロジー工学での成果について検討する。

3.2 自然言語による知識の記述・蓄積とその有効利用

現在、次に示すような研究構想を持っている。既にある程度成果が上がっているものもあり、また今後出来れば進めたいと考えているものも含まれている。

3.2.1 日本語辞書（IPAL）

筆者も１０年以上にわたってプロジェクトに関係した日本語辞書。昨年度末にプロジェクトは一応終了した（http://www.opa.go.jp/STC/NIHONGO/IPAL/ipal.html）。

- 基本動詞辞書
- 基本形容詞辞書
- 基本名詞辞書

筆者がプロジェクトを離れてからも、辞書情報の統合化に関して検討が進められ、辞書情報のハイパーテキスト化された環境も開発され、また意味分類をホームページ上で参照できるようにしたIPA STaXもプロジェクトに加わった研究者により開発されている。多くの人に利用されていると考えられるが、既に開発された辞書・環境に個々の利用者が付加できる情報をどう付加し、公開するかを、著作権の問題も含めて検討して欲しい。

3.2.2 生成的辞書

Stanford大学のCSLI（Center for the Study of Language and Information）におられるAnn Copestake氏とDan Flickinger氏と日本側の研究者、筆者も含めて主に主辞駆動句構造文法（HPSG: Head-driven Phrase Structure Grammar）の枠組みで研究を進めてきた人たちで共同研究を進めようとしているプロジェクト。

日本側では、日本認知科学会第14回大会でワークショップ「辞書を中心として日本語文法の構造化」を開催の予定で、ホームページ上に既に発表者の原稿を公開している（http://lang.nagoya-u.ac.jp/tonoike/wksp97.html）。Ann Copestake氏も活発な研究を進めておられて、研究成果がホームページ上に公開されている（http://www-csli.stanford.edu/~aac/）。

3.2.3 モダリティ辞書

自然言語を通してオントロジーに関する問題を考える場合に、領域に依存した知識であるかどうかの区別だけでなく、ある表現したい内容に関して話者がどういう態度をとるかも重要である。日本語で言えば、言表事態と言表態度の区別、あるいは文のレベルという言い方で多くの研究者によって論じられてきた。発話行為を含めて文法の問題を考える必要がある。（日本語の文のレベルについては、Tonoike, T. & H. Sano (in press) 'A semantic analysis of hierarchical clause structure in Japanese,' Gunji,T & K. Hasida eds. Constraint-based Grammar of Japanese. Kluwer, を参照。また、生成的辞書的な観点で、談話も含めた検討を進めているAnn Copestake氏の研究も重要である。上記のホームページに公開されている研究成果を参照。）

3.2.4 日本語学習者作文コーパス

日本語の文法に関する問題を検討するために、日本語学習者の作文コーパスの作成とその共有化を、筆者も研究分担者として加わり進めている（大曽科研　http://cookie.lang.nagoya-u.ac.jp/）。

この研究により以下に挙げるような点で日本語に関わる研究を、今までになかった形で前進させることができるのではないかと考えている。

・　日本語辞書への否定的情報の付加
・　日本語文法に関する新しい情報源
・　新しい自然言語処理の対象（間違いを含んだ日本語文の処理）

3.2.5　英語辞書

筆者は、大学で英語を教えているので、日本語と英語との対応関係、また日本語母語話者の英語学習、さらに日本語母語話者の英語教師による学習支援にも関心がある。友人であるアメリカ人のLinda Woo氏の協力で、新たに英語基本語彙リストを作成しホームページ上に公開している（linda5000:http://lang.nagoya-u.ac.jp/~tonoike/linda5000.html）英語例文の蓄積・共有、語義分けをした辞書の構築なども、同僚を含めた英語教師で計画している。さらに、法的な問題などが解決できれば、既に言及した日本語辞書IPALとのリンクを張って公開することも可能である。

3.2.6　NULL（Nagoya University Linked Lexicon）プロジェクト構想

名古屋大学言語文化部に専任教員がいるすべての言語、日本語、英語、ドイツ語、フランス語、ロシア語、中国語、スペイン語、朝鮮・韓国語で辞書を作り、相互にリンクを張る構想がある。まだ構想段階であり、実際に使いものになるものにするためには、まだまだやるべきことが残されているが、日英の２カ国語で行えることは、相互に連結すれば、多くの言語間で可能で、しかもそれらが全体としてリンクされることの意義は大きい。

3.2.7　動画の利用

名古屋大学の言語文化部では、外国語の動画教材の作成を進めている。既に英語のプレゼンテーション教材の暫定版を公開している（http://lang.nagoya-u.ac.jp/~sugiura/igirisu/）。さらに、英語以外の外国語でも教材の作成を進めている（プロジェクトの内容に関しては、http://lang.nagoya-u.ac.jp/kyouiku95.htmlを参照）。

プレゼンテーションの実例に対して、日本語での説明を加え、対応するところにリンクを張ることで、構造を与えている。

3.2.8　領域別知識の蓄積と利用：医療支援と法廷通訳

愛知県には、製造業を中心に日系ブラジル人をはじめ多くの日本語を母語としない人が暮らしている。病気やけがで医療を受けたい場合に、言葉が通じないことで困る場合が増えているようで、先日もラジオで、ブラジル・ポルトガル語とタガログ語の通訳ができるボランティアを探している団体が紹介されていた。

上記の動画教材で利用している環境がここでも活かせるだろう。各言語の母語話者と医者と動画を作れる環境にある者が協力すれば、医者が知りたい情報に関して、それを二カ国語で対応させてホームページ上で参照できるようにすれば、役にたつものを提供できる。

さらに、名古屋大学に留学中の日系ブラジル人の大学院生が法廷通訳を支援するシステム（日本語とブラジル・ポルトガル語）を作成しようとしている。

3.3　おわりに

オントロジー工学に関する研究には、言語学が抱える広い意味での文脈依存性と同様の大きな、克服することが難しい問題があると考えられる。筆者は、その問題を直接解こうとするのではない方向を目指すことを考えている。個別の領域の知識を人間が使いやすい形で提示することで、様々な形で人間を支援することができる。従来の言語学の研究対象の限定の仕方を離れて、インタフェースとしての自然言語を活かす方向である。言語に興味を持つ人間は、この方向で広い意味でのオントロジー工学に貢献できるのではないだろうか。

4．　マルチメディアデータベースの新潮流

4. マルチメディアデータベースの新潮流
究極のマルチメディアデータベース技術を求めて

概要

本稿では、マルチメディアデータベースシステム技術の現状を論じ、将来を展望する。筆者は既にマルチメディアデータベースシステムを3世代に分けて考えると、研究・開発上分かりやすいと提案しているが、本稿ではさらに、著者自らのシステム開発事例を織混ぜながらそれらの問題点や問題解決の手法を論じる。特に、本稿ではオブジェクト指向のモデル化概念であるオブジェクト参照法に従いながらも、リレーショナルデータベース管理システム 4th Dimensionを使って構築したマルチメディアデータベースシステムをOMEGA-Ⅰと名付け、これを第1世代のマルチメディアデータベースシステムの典型例としてその概略を紹介する。次に、ビデオや音といった時間的マルチメディアオブジェクトを表現・管理・運用できる第2世代のマルチメディアデータベースシステム、これをOMEGA-Ⅱと名付けよう、オブジェクト指向データベースシステムの上に実装した例を取り上げ、何がこの際問題なのかを論じる。さて、マルチメディアデータベースの究極の狙いはデータベース化の対象となった世界を丸ごとモデル化することである。これは。3次元の空間に存在する3次元空間オブジェクトとそれら間の関連、および操作をどのように表現し管理するか、加えて時間的オブジェクトをどのように表現し管理するかを統合的に扱えるデータモデルを構築することが問題と捉えることができる。我々は区間という概念に基づいて、4次元時空間データモデルを設計中である。このモデルに基づいた第3世代のマルチメディアデータベースシステムは仮想現実感システムとオブジェクト指向データベースシステムの連携システムとして構築される。我々はこのシステムをOMEGA-Ⅲと名付ける。

著　者：Yoshifumi Masunaga, University of Library and Information Science

標　題：New Trend of Multimedia Databases -- In Pursuit or Ultimate Multimedia Database Technology --

出版物：Proceedings of the Advanced Databese System Symposium '96,

Information Processing Society of Japan, Tokyo, December 2-4, 1996.

4.1 まえがき

従来の文字・数値型のデータに加えて、テキスト　画像、動画像、音といったマルチメディアデータを表現し、管理運用できるデータベースシステムをマルチメディアデータベースシステムと言うが、現在その必要性に疑問を呈するデータベース研究者はいないであろう。実際、そのようなシステムを開発するに必要なオブジェクト指向技術、大規模な記憶デバイス技術、マルチメディアプロセッサの開発、高速通信回線の開発、あるいはデータ圧縮技術の進歩や標準化の推進などが急速に整備されている。

マルチメディアデータベースシステム技術の研究は世界的にみても、1984年頃はまだ黎明期であった。日本での取り組みもそのような世界的動向に刺激されてほぼその時期に取り組みが開始されている［Masu87］。初期の研究はマルチメディア文書のマルチメディアデータベース化といったことに焦点が当たったことが多かったが、JPEGやMPEG-1などの静止画像や動画像のデータ圧縮アルゴリズムが開発されるに連れて、画像データベースや動画像データベースに研究・開発も盛んになってきた。また、アプリケーションでもVOD（Video on Demand）などが取り沙汰されて、そのためのデータベースシステム技術も盛んに研究きれている。

さて、筆者はかって、マルチメディアデータベースシステムの研究・開発はそれを3世代に分けて考えると分かりやすいと提案した［Masu93］。それらは次のとおりである。

(1)第1世代のマルチメディアデータベースシステム

テキストと静止画像中心のマルチメディアデータベースシステム。たとえば、図版入のマニュアルや教科書などを光学式文字読み取り装置（OCR）とイメージスキャナで取り込み、それらをコンポーネントデータとしてマルチメディア文書やマルチメディア図書を構築するといった典型例がこの世代に属する。

(2)第2世代のマルチメディアデータベースシステム

第1世代のマルチメディアデータベースシステムのそれに加えて、動画像や音といった時間的マルチメディアデータを表現・管理・運用できるマルチメディアデータベースシステムをいう。システム構築のポイントはマルチメディアデータが時間依存であるということをどのようにモデル化して統一したモデル論を展開できるかであろう。それがうまくいけばシステムの実装も明らかになる。

(3)第3世代のマルチメディアデータベースシステム

実世界は3次元の空間であり、また1次元の時間軸を有する、4次元の時空間である。そこではオブジェクトは一般には3次元のオブジェクトとして存在し、様々な動きをし、オブジェクト同士が様々に関連している。そこでは、オブジェクは単に（動）画像で表わされているとかいったレベルを超えて、実世界そのものがデータベース化の対象となっている。したがって、第3世代のマルチメディアデータベースでは、3次元の空間そのものをユーザインタフエースとして再現してやる必要がある。また、モデル化された対象世界のオブジェクトは一般に3次元であり、また時間と共に動きを伴うもの、さらにそれらオブジェクトの存在している実世界の様々な制約が表現され、またそれらにオブジェクトは従わなければならないものである。換言すれば、第3世代のマルチメディアデータベースでは対象とした世界がその世界そのものの記述（たとえば、その世界は物理空間としてはニュートン力学が支配している空間なのかどうか。とか）を含む、データベース化の対象となったあらゆるものが記述きれた、そのような体系でなければならない。この意味では、世界が丸ごとデータベースの中に入っているのであり。究極のデータベースシステムと呼んでよいだろう。

本稿では、この様な観点からマルチメディアデータベースシステム技術の現状を論じ、来るべき本格的マルチメディアデータベースシステム時代への展望を示すことを目的とする。

4.2 第1世代のマルチメディアデータベースシステム技術

4.2.1 要素技術

OCRやイメージスキャナの技術進歩は著しい。特に、OCRの文字続みとりソフトウェアの進歩に目を見張るものがあり、以前はそれなりの専用ハードウェア（＝文字読みとり用コンピュータ）を用いないと読みとり精度に支障をきたすことが多かったが、最近ではパーソナルコンピュータ上で稼働する廉価な文字読みとりソフトウェアでも相当のスピードと読みとり精度を保証してくれる。イメージスキャナはカラーが当り前のものとなり、精度と絡めて、一般的な利用で不満を抱く人はいないであろう。また、カラーディスプレイはブラウン管、液晶ともにその色再現性はよい。カラープリンタは色再現性や価格に不満が残るが、第1世代のマルチメディアデータベースシステムのデータ入出力装置、現在ほぼ整っていると考えてよいであろう。

4.2.2 リレーショナルデータベースアプローチ

要素技術のもう一つの側面は第1世代のマルチメディアデータベースシステムを支えるデータベースシステムの現状であろう。大別して、マルチメディアデータベースシステムの核システムとしてリレーショナルデータベースシステムとオブジェクト指向データベースシステムの二つが考えられる。

最近のリレーショナルデータベースではリレーションの属性値にBLBO（Binary Large OBject）を許すものがほとんどである。従って、イメージを長いビット列としてモデル化してしまえば属性のデータ型をBLBOにしておけば、リレーショナルデータベースに格納できる。テキストは文字列型で格納できる。たとえば、我々が教科書の電子化に使用しているApple Macintosh上で稼働するリレーショナルデータベース管理システム 4th Dimensionのデータ型は、Iで整数、Aで文字、Tでテキスト、Pでピクチャを取り込める。従って。リレーショナルデータベースを使用して第1世代のマルチメディアデータベースシステムを構築することができる。リレーショナルデータベースシステムには、たとえば4th Dimensionでは全文検索関数が実装されており、データベース機能に加えてテキストを扱うアプリケーションに有難い機能を有している。実際我々が開発した中学校の地理の教科書の電子化版では、全文検索機能を使って教科書の細項目（＝数百字からなる記述の単位）間の意味的な関連性を見つけることを実

装している。しかし、この教科書のマルチメディアデータモデル化は次項で述べるオブジェクト参照法を用いてそれを行っており、我々はオブジェクト指向のモデル化に則りリレーショナルデータベースシステムを用いて開発したこのシステム［Masu93b］をOMEGA-Ｉと名付ける。現在、OMEOA-Ｉはインターネットに対応すべくWeb化の改良を加えられている最中である。

4.2.3　オブジェクト指向アプローチ

さて、この様なリレーショナルアプローチの最大の欠点は、文書の構造をダイレクトに表現できないことである。たとえば、一般に文書は、章、節、小節といった論理的構造を有する。この様な構造を持った文書をリレーショナルデータベースシステムで表わそうとすると、一般に多数のリレーションを定義し、また多数のタップルを生成しないといけない。これは、分かりずらいし、検索などの処理を行う場合に、リレーションの結合演算がとられることになるので、効率がよくない［Masu91］。

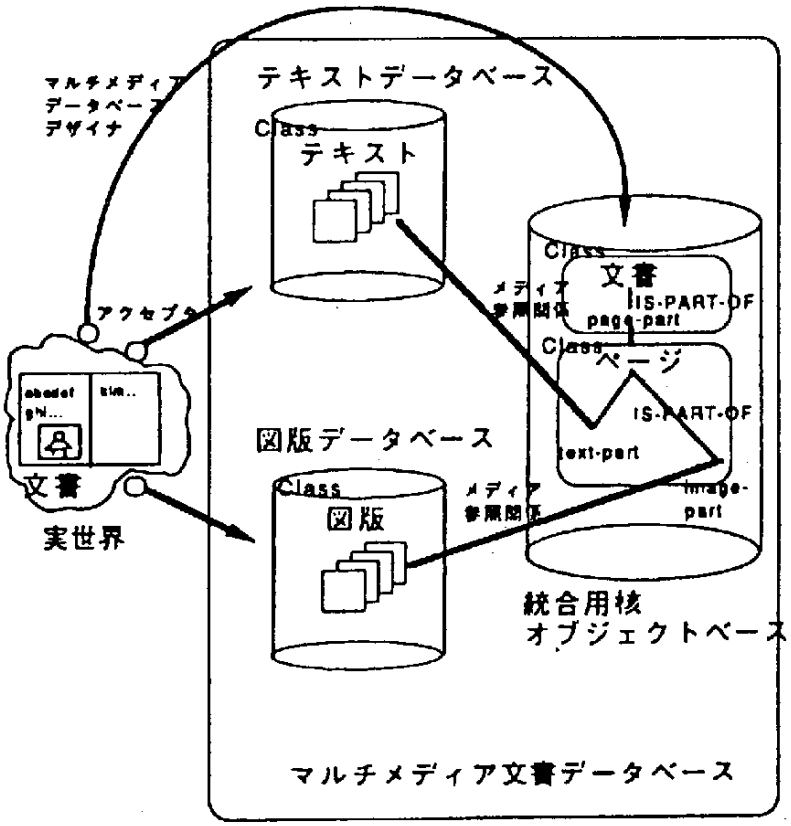


図４.２-１　オブジェクト参照法

この様な問題は、オブジェクト指向データベースシステムを使用すると解決する。リレーショナルデータベースシステムとの違いはこれまでよく論じられてきたことであるが、ここでもう一度その主要な利点をマルチメディアデータベースシステム構築の観点からまとめれば次のようであろう：(1) 複合オブジェクトの概念により、マルチメディア文書などの諸構造をそのままモデル化できる、(2) オブジェクトの異質性の同質化の機能により、テキストや静止画像といった異なるデータをオブジェクトとして統一的に表現できる。(3) ポリモルフィズムによりマルチメディアデータへのインタラクションを統一的に表現できる、など。オブジェクト指向データベースシステムを用いた初期のマルチメディア文書データベースシステムの開発は、Dr. Won Kimや筆者らのグループの研究・開発が先駆け的研究として発表きれている。我々が提案した「オブジェクト参照法」というマルチメディアデータベースシステムの構築法はまさしくこのパラダイムでの産物であり、それを図4.2-1に示す［Masu87］。

4.3 第2世代のマルチメディアデータベースシステム

4.3.1 時区間アプローチとOMEGA-II

第2世代のマルチメディアデータベースシステムで問題になるのは、その定義から、動画像や音といった時間的マルチメディアオブジェクトのモデル化である。動画像は一般に1秒当たり30枚の静止画の系列であり、一方、音は波形として捉えられる。両者はその表現形式が全く異なるが、両者を共に表現し管理・運用したいマルチメディアデータベース側の観点からすれば。同一の表現形式をとりたい。

その一つのアプローチが、両者をともに時区間（time interval）として捉える考え方である。この考え方は、マルチメディアデータの表現に時線アプローチ（time-line approuch）として、MHEGなどの標準化団体が採用したり。マルチメディアオーサリングシステムの構築にこのモデルが採用されたりして、既に知られている［Masu95］。また、時区間の概念はかって人工知能の分野でAllenがその論理的性質を示した結果がよく知られているが、時間的関連には、equals、after、during、contains、overlaps、overlapped-by、meets、met-by、starts、started-by、finishes、finished-byである［Alle83］。

我々は、動画像や音を時区間としてモデル化して、オブジェクト指向データベースシステムを用いて時区間を表わすクラスを定義・導入して第2世代のマルチメディアデータベースシステムを開発してきた。このシステムを今後 OMEGA-Ⅱと呼ぼう。このシステムに関する研究結果は幾つか既に公表されているが［Maki96, MaNK96］、そのなかでも大事な概念を幾つか列挙すると次のようになる：

(1) 時間的マルチメディアオブジェクト間の時間的関連記述

時間的マルチメディアオブジェクト間にはAllenの時間的関連が存在する。それをクラス定義時に陽に記述できる体系を明らかにしたこと。これは、ODMG-93標準オブジェクトモデルが2つのオブジェクト間の参照関連をクラスレベルで明示させるために、relationship inverseを導入したが、その構文と上位互換になるように定義されている。この結果、たとえばある音楽演奏が映像（＝ビデオ）と音によって録画・録音される場合に、集録された映像と音が同時に始まり同時に終わる時間的関連にある、というようなことが記録できるようになる。また、この情報が映像オブジェクトと音オブジェクト間に記録されるので、その映像と音を同期して再生（play back）することが可能となる。図4.3-1にクラス音楽演奏、クラスビデオ、クラス音の関係を示す。

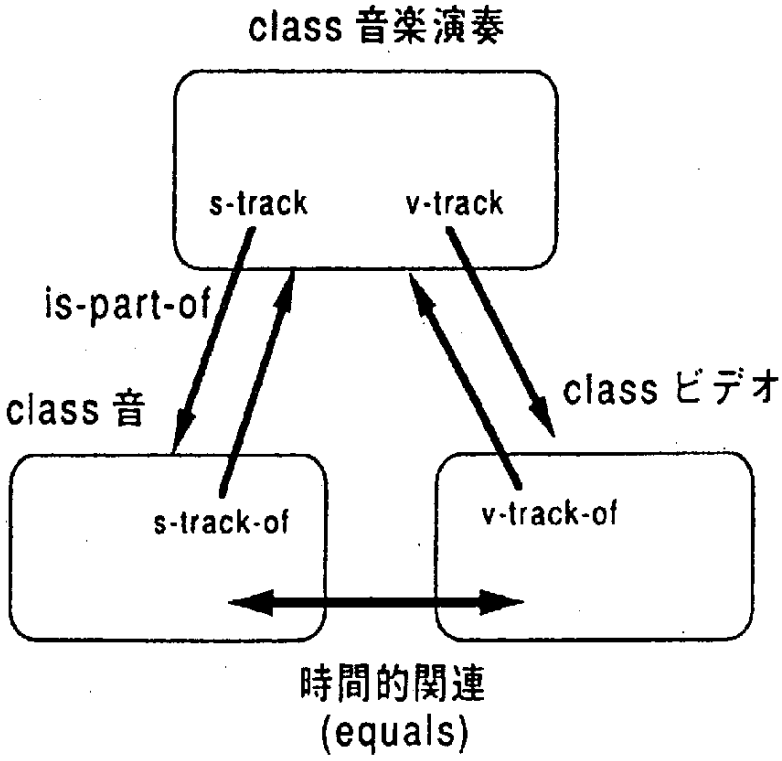


図４.３-１　クラス音楽演奏

(２)　空時区間の導入と時間的マルチメディアデータモデル化への応用

時区間的に、何もないことを陽に表わすために我々は空時区間という概念を導入した。この概念を動画像がないことを表わすために空動画像オブジェクトを使って表わし、音オブジェクトがないことを空音オブジェクトを使って表わす。実際には、空動画像オブジェクトは真っ黒なフレームの系列であり、空音オブジェクトとは振幅がゼロの音が乗ったキャリアのみの音オブジェクトとして実装している。その結果、動画像や音が複雑に混じった複合マルチメディアオブジェクトの標準形を導入することができる。その基本となるのが２つの時間的オブジェクト間の７つの時間的関連の標準形で、それを図４.３-２に示す［MaNK96］。

(３)　新しいマルチメディア同期再生スキームの提案

空時区間オブジェクトの導入により、従来知られていたマルチメディア同期法、たとえばAppleのQuick Time［Appl92］や藤川ら［FSM*93］の方法とは違う同期法を示すことができる。これは開始同期と個体同期の２つの同期により、全ての複合時間的マルチメディアオブジェトは同期再生できる、という方法である。この方法を簡単に述

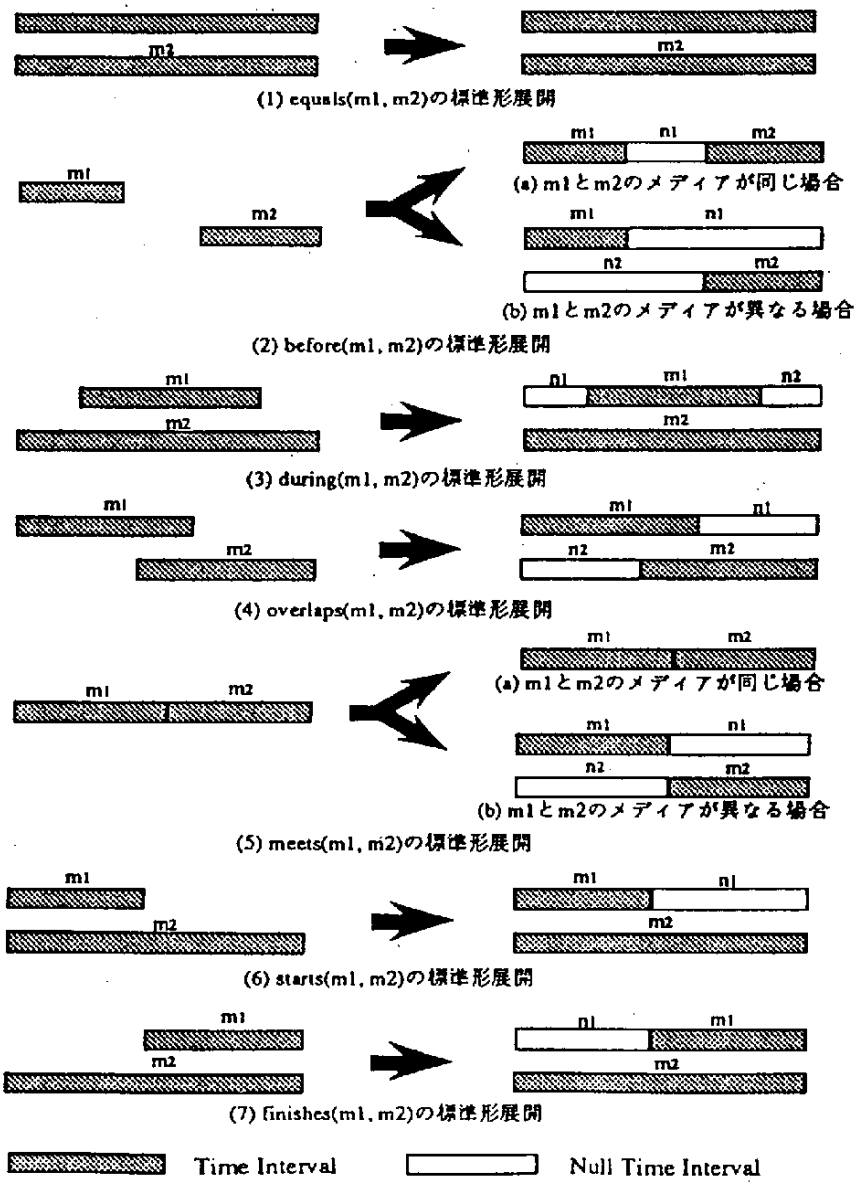


図４．３-２　７つの時間的関連の標準形

べれば、各々10秒の動画像と音オブジェクトを同期再生しないといけないない場合、それらの再生を同時に開始し（開始同期）、続いて動画像は動画像で自分の責任で10秒でその再生を終わるように、音オブジェクトはまたその自己の責任で10秒でその再生が終わるように各々制御すれば（個体同期の概念）、結果として両者の同期再生ができたとしようという概念である。このためには、たとえば動画像の再生スピードが

CPU負荷の増大などにより落ちてきた場合は、コマ落としをして自己の再生の同期をとるアルゴリズムを実装しておくことが必要である。OMEGA-Ⅱではこのスキームを実装しており、プロトタイプシステムは当初の目的どおり稼働している。なお。動画像はMPEG-Ⅰ圧縮符号化アルゴリズムで圧縮されている。一方音は20KHz。8ビットで符号化されて文字列として格納されているが時間遅れ等の問題を発生することなく再生が行われている。

4.3.2 MPEG-Ⅰ動画像データベースシステム

我々はOMEGA-Ⅱを開発していく過程で、現在の動画像データベースシステムの在り方に大きな疑問を抱いた。VODを実現すべく現在研究・開発されているビデオサーバとは一体どのようなテクノロジを採用しているのか。リレーショナルデータベースシステムを使ってVODサービスを行えるとする。たとえばOracleのシステム構成はどうなっているのか［Mika94］。調査によれば、これらのシステム構成は、ビデオデータはデータベース管理システムに格納されているのではなく、ビデオサーバと称するファイルシステムに格納されているのであり。ビデオの属性データ（タイトル、オーサ、制作年、ビデオ格納番地、等）のみがデータベースシステムに管理されている、ということで一致している。たしかに、格納されたビデオデータはVODの様な場合には更新されることはないであろうから、データベース管理システムの直接の管理下に入っていなくてもよいという側面はあるが、素材としてのビデオデータや音データを編集してまた新たなマルチメディアコンテンツを構築したいというような状況では、明らかに上記のような構成は不都合を生じる。つまり。動画像や音データを直接データベース管理システムの下に置けるデータベースシステムを構築できないのか、研究する価値がある。音データは動画像データに比べればデータ量の観点からいえば小さく問題は少ないので、研究は専ら動画像を直接データベース化できるかを研究した。MPEG-Ⅰ動画像データベースシステムの構築研究は奈良先端大学の研究グループが研究をしてきた経緯があるが考え方のみが示されており実装はされていないようである［THIU95, OgIU96, KYTU96］。

さて、動画像の最大の問題はデータ量の大きさである。圧縮をかけない状態での2時間のビデオのデータ量は数十GB（Giga Bytes）になる。これでは、現在一般に使われているワークステーション環境では1本のビデオをその2次記憶装置に格納することすら無理であろうし、ディスクアレイでテラバイト級の記憶容量が達成されても、

記憶のためのコストが掛かりすぎる。

動画像のデータ圧縮は静止画像の圧縮アルゴリズムJPEGを動画像に適用したmotion JPEGや、動画像がフレームの系列であるという特徴に着目した動画像に固有の圧縮アルゴリズムであるMPEG-Ⅰ、さらにMPEG-Ⅱが知られている。たとえば、MPEG-Ⅰを用いると動画像のデータ圧縮率は約数十分の一であり、したがって2時間のビデオは約1GB程度の大きさになる。また、この程度の大きさならば、今後の大容量記憶装置の普及や高速データ伝送路の整備などを考えると、データベース管理システムの管理下に直接置き管理・運用する形態は不自然ではない。

そこで、我々はMPEG-Ⅰ動画像をGOP（Group of Pictures）単位でオブジェクト化し、直接クラスGOPのインスタンスとして、オブジェクト指向データベース管理システムONTOSのもとで生成・管理・運用できるプロトタイプシステムを開発した。その結果は野中らにより報告されているが［NoMa96］、そのシステムでは動画像のビデオカメラおよびビデオデッキからの取り込みと、格納、および再生（順再生・逆再生・再生速度の指定などができる）が行える。また、そこでのクラス階層は時間的マルチメディアデータベースシステムOMEGA-Ⅱと両立しているので、この動画像データベースシステムはOMEGA-Ⅱの動画像データベースエンジンと位置付けてよい。

4.4 第3世代のマルチメディアデータベースシステム

4.4.1 4次元時空間データモデルと3次元空間データモデル

データモデリングの根本的概念はデータベース下の対象となった世界、つまり実世界の写絵（うつしえ）をコンピュータ内に構築することである。この写絵を記述する記号系がデータモデルと呼ばれた。これまで、ネットワークデータモデルやハイアラキカルデータモデル、あるいはリレーショナルデータモデル、そして我々がマルチメディアデータベース構築で対象にしているのがマルチメディアデータモデルである。

たしかに、マルチメディアデータモデルとは何か、世の中にコンセンサスはないが、我々がまさしく取り組んできた課題はこのマルチメディアデータモデルとは何かを明らかにしようという挑戦でもある。第1世代のマルチメディアデータベースでも、第2世代のマルチメディアデータベースでも共にそうであるが、オブジェクト指向パラダイムが有効であることは基本的に間違いがないことと世界の多くの研究者が認識していると考えられる。たとえば、オブジェクト指向データベースシステムを使わないまでも、オブジェクト指向プログラミング言語C++を使用して動画像のプレゼンテーションシステムを構築したという類の報告例がある［GiBT93］。

さて、第3世代のマルチメディアデータベースシステムを構築することに論点を移してみよう。この世代のマルチメディアデータベースシステムがそれ以前のマルチメディアデータベースシステムと本質的に異なる点は、3次元の空間的オブジェクトを表現でき、管理でき、運用に供さねばならないということである。さらに、現実の世界では時間軸が加わるので、実際は4次元の時空間に存在するオブジェクトとオブジェクト間の制約、そしてオブジェクトの操作（問い合わせや更新、あるいはアプリケーション開発）を記述できるものでなければならない。

この様な目的のためには、これまでのリレーショナルデータモデルは力不足である。また、オブジェクト指向データモデルを強化して第2世代のマルチメディアデータを記述できるように拡張したOMEGA-Ⅱデータモデルでも、マルチメディアデータの時間的側面は記述できても、空間的側面を記述することまでは考慮が払われていないので、さらに考察を加える必要がある。

この結果、4次元の時空間のオブジェクトを自然に記述できるデータモデルが得られることになる。これを単に4次元時空間データモデル、時間軸を考えないで3次元の空間のオブジェクトとそれら間の関連と操作を記述できるデータモデルを単に3次

元空間データモデルと呼ぶことにしよう。いずれの場合も、単に4次元データモデルとか3次元データモデルとかいう呼称は採用しない。なぜなら、データウェアハウスの研究分野で既に多次元データモデルという言い方か定着しており、それとの混同を避けるためである。

本来2次元や3次元の空間的オブジェクトの扱いは、たとえば地理情報システム（GIS）やCADの分野で研究・開発が先行してきた経緯がある［Egen94］。しかし、最近はデータベースの分野でもきまざまな観点から時空間データベースの研究が盛んに行われるようになってきた［KITA96, TaSh96, KYTT96, KuMa96, Masu96］。

4.4.2 OMEGA-III

4次元時空間データベースを開発するにあたり、我々はグループウェアの発想にたち、協調作業を支援できる仮想的な環境をデータベース技術に基づいて開発することを考えた［MKK*96, Masu96］。そのような作業環境は3次元の空間であり、空間そのものを物理的および時間的に支配している「場」の存在を考えることができる。OMEGA-Ⅲはそのような高度な実世界のデータモデリングを可能とする第3世代のマルチメディアデータベースシステムである。本項ではこの基本的な考え方を紹介して、今後のマルチメディアデータベースシステムを展望してみたい。

(1) OMEGA-Ⅲのシステムアーキテクチャ

OMEGA-Ⅲは仮想協調作業環境を提供するために、フロントエンドは仮想現実感システムからなりバックエンドにオブジェクト指向データベースシステムからなる。フロントエンドではユーザは従来の仮想現実感システムが提供してくれると同じ空間で協調作業が行える。しかし。仮想現実感システムにはユーザがアドホックな形で空間的な質問を発行できる機能やオブジェクトやオブジェクト間の関連に対し更新を発行するような機能は備わっていない。しかし、明らかにこの様な機能は協調作業支援のためには必要である。我々は、この様な機能をバックエンドのオブジェクト指向データベースシステムを使って実現してやることを考えた。そのために考案されたシステムの全体アーキテクチャが図4.4-1に示される［Masu96］。

さて、その図に示されるように、この協調作業空間実現のためには仮想現実感シス

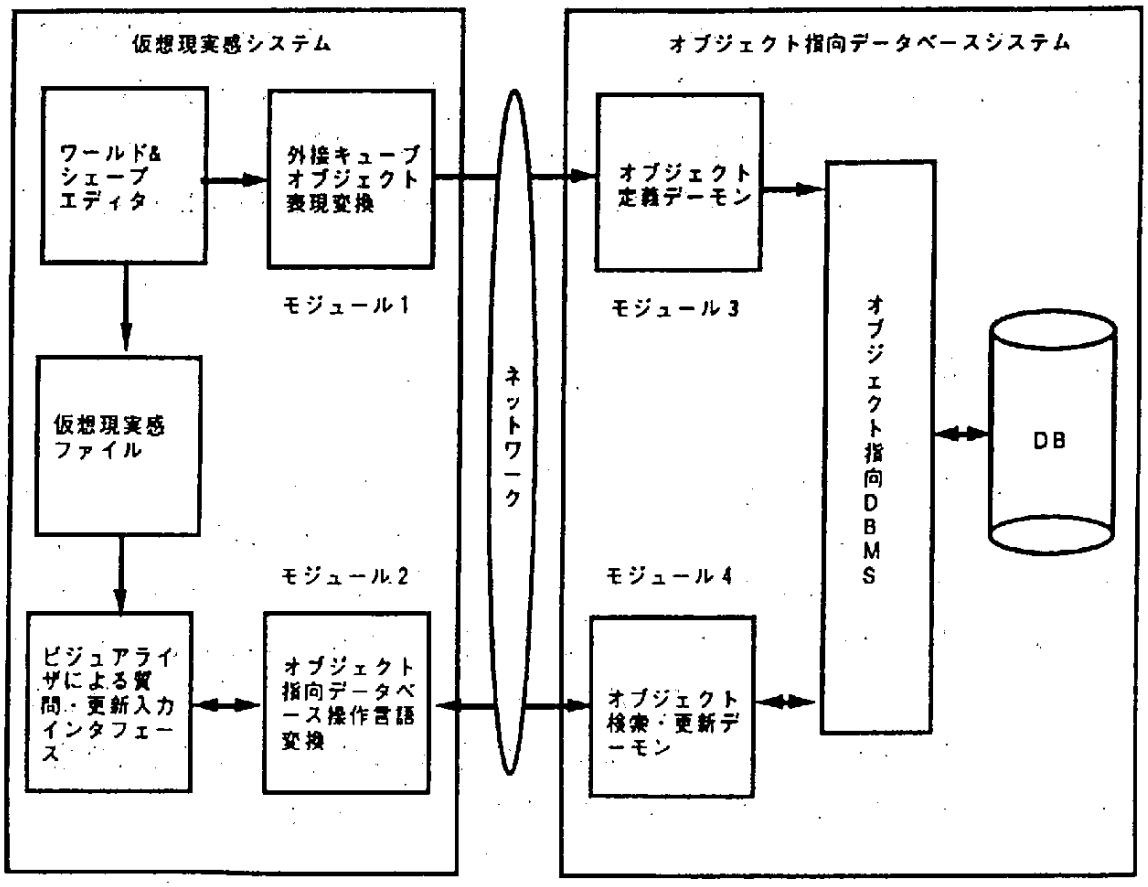


図4.4-1　協調作業支援のための4次元時空間データベースシステム
OMEGA-Ⅲのアーキテクチャ

テム上で2つのモジュールを実装する必要がある。モジュール1により仮想現実感システムで生成されたオブジェクトに対応したデータベースオブジェクトのクラス定義とオブジェクト生成定義がオブジェクト指向データベースに送られる。モジュール2は仮想現実感空間でのユーザの検索・更新要求を（空間的拡張を施された）オブジェクト指向データベースシステムが受け付けられる形に形成しそれを送出し、その結果を受け取るためのものである。一方、オブジェクト指向データベースにも対応して2つのモジュールが作成され、また3次元空間的オブジェクトデータモデルを管理できるように空間的機能拡張を施されることが必要である。また、時区間としてモデル化

される時間的オブジェクトをサポートできるように時間的機能拡張が施されている必要もある。モジュール3はモジュール1からのオブジェクト定義要求を受理するために立ち上がっているデーモンである。同様にモジュール4はモジュール2からの要求を待ち受け、必要な結果を返すために立ち上がっているデーモンである。

なお、複数の協調作業者がたとえお互いが遠隔地に離れていようとも協調作業を行えるような環境を実現するために、仮想現実感オブジェクトもデータベースオブジェクトもすべてがお互いにコミュニケーション可能となっていることが必要であり、この様な環境を実現する手段として、たとえばCORBAのようなオブジェクト指向の考え方を取り込むことが考えられる。

(2) 区間に基づく統一的4次元時空間データモデル

3次元の空間的オブジェクトをどのように表現するかには様々な方法が考えられる。たとえば、黒木ら［KuMa96］は単体複合オブジェクトを使用する提案を行っているが、数学的厳密性を有してはいるものの、モデルは実世界での意味表現能力に乏しい。我々は、3次元コンピュータグラフィックスや3次元CADでのこれまでの成果に着目し、3次元オブジェクトを「外接キューブ」で表現することを提案してきた［Masu96］。その概念を図4.4-2に示す。外接キューブは一般に本物の3次元オブジェクトの近

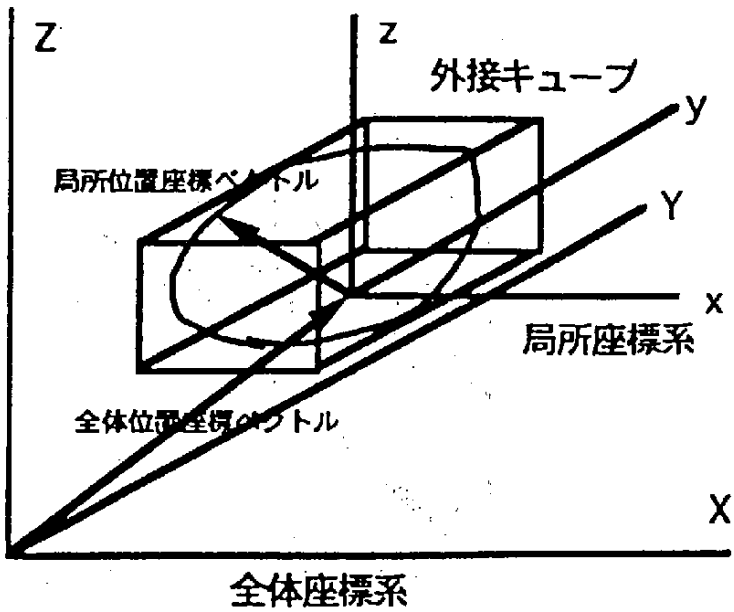


図4.4-2 外接キューブの概念

似となるので表現誤差が問題になるが、このアプローチの最大の利点が3次元空間的オブジェクト、2次元空間的オブジェクト、1次元空間的オブジェクトおよび線状性を有する時間的オブジェクトが全て区間（の直積）で統一的に表現できる点である。この様子を図4.4-3に示す［Masu96b］。

4次元の時空間データモデル、あるいは3次元の空間データモデルを考えるようになると。従来のデータモデルでは顕在化しなかった本質が突きつけられる。これは、

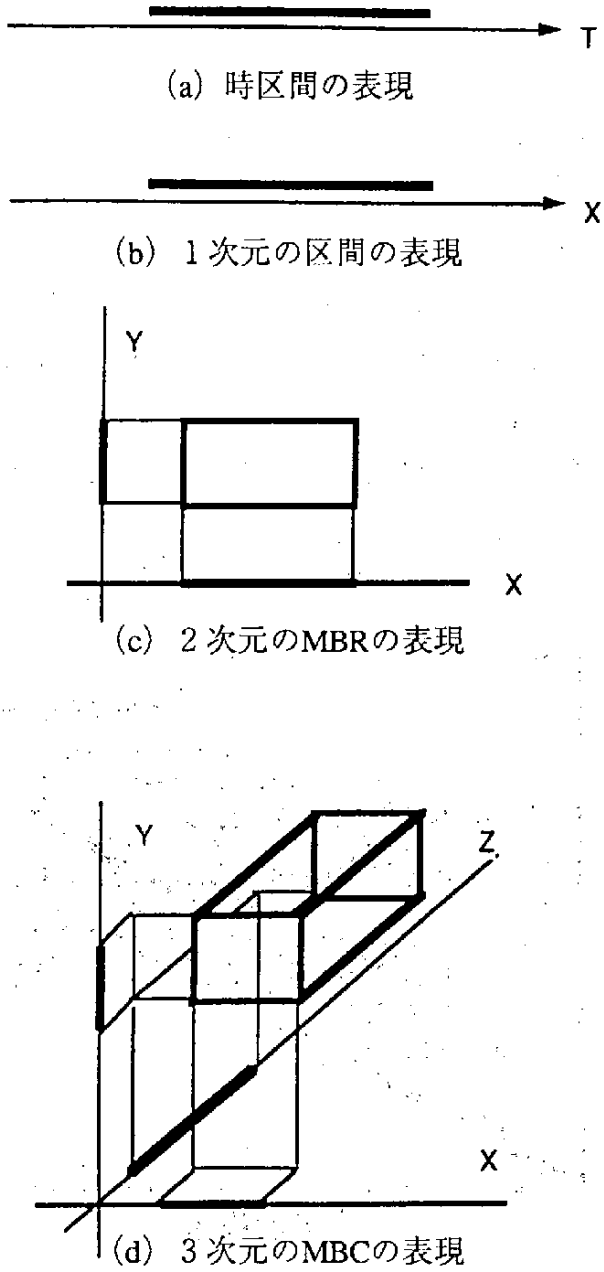


図4.4-3　区間論理による時空間統一データモデル

現在一部、仮想現実感システムで研究が行われているようであるが、対象とした実世界の「場」の記述である。たとえば、広田らは仮想現実感世界にニュートン力学場や量子力学場を考慮して、物体の表示を行えるシステム開発例を報告している［HiKi94］。３次元の空間的データモデルではこの次元の議論がこれまでのデータモデルとそれを差別化できる重要な要素として積極的に行われる必要がある。

空間的・時間的関連には、きらに細分化される意味構造があり最近それらが研究の対象となっている。我々はAllenの13種の時間的関連は、実は８種の位相的関連と３種の順序関連の混合推論の結果であることを示した［Masu96］。

(3) マルチモーダルインタフエース

さらに、３次元空間データベースシステムというようにデータベースが進化してゆくと、データベースに対するインタフエースも従来のリレーショナルデータベースに対するSQLやオブジェクト指向データベースに対するOSQLといった文字ベースのインタフェースではなく、音声やジェスチャー（＝身振り・手振り）を含むマルチモーダルなインタフェースが必要となる。また、HMD（Head Mounted Display）をかぶればキーボードは直接は触れない。もしどうしてもキーボードアクセスしたければ視野のなかにキーボードを実現してタッチするしかない。我々はこの様な状況も考慮して音声によるデータベースアクセスシステムを構築している。そこでは、センテンスパターンを用いる音声認識装置を用いてリレーショナルデータベースシステム Oracle に格納されているデータにアクセスし、答えが音声合成装置により音声で返ってくるプロトタイプシステムが構築され、問題点の同定を行っている［MrMa96］。

4.5 むすび

本稿では マルチメディアデータベースシステムの、初期から現在にいたる研究・開発の流れを概観すると共に、これからのマルチメディアデータベースシステムとして、4次元時空間データベースシステムを紹介した。そこでは、単に履歴データをデータベースにどう加えるかといったレベルの議論ではなく、本質的に3次元の空間に存在するオブジェクト、それらの間の関連、操作、と時間的次元を統合した。そういう意味で究極のデータモデリングの世界が問われているわけで、本稿では我々が研究・開発中のOMEGA-Ⅲを取り上げ、仮想現実感システムとオブジェクト指向データベースシステムの統合による実現アプローチを示した。

参考文献（引用順）

[Masu87] 増永良文：”マルチメディアデータベース概念,” 情報処理, Vol.28, No.6, pp.671-684, june 1987.

[Masu93] 増永良文：” マルチメディアデータモデルOMEGAにおける音データと動画像データのサポート,” 情報処理学会, アドバンスト・データベースシステム・シンポジウム'93 議事録, pp.163-179, December 1993.

[Masu93b] 増永良文：” マルチメディア電子教科書の開発研究,” 情報処理学会研究報告, 93-DBS-94, pp.175-184, july 1993.

[Masu91] 増永良文：” 次世代データベースシステムとしてのオブジェクト指向データベースシステム,” 情報処理, Vol.32, No.5, PP.490-499, May 1991.

[Masu95] 増永良文：“マルチメディアデータベースと時間,” 情報処理, Vol.36, No.5, pp.369-377, May 1995.

[Alle83] Allen, J.F.："Maintaining Knowledge about Temporal Intervals," Communications of ACM, Vol.26, No.11, pp.832-843, 1983.

[MaKi96] 増永良文, 清光英成, “マルチメディアオブジェクト間の時間的関連記述の一フレームワーク,” 電子情報通信学会論文誌D-Ⅱ. Vol.J79-D-Ⅱ, No.4, マルチメディア論文特集号, pp.492-501, April 1996.

[MaNK96] 増永良文, 野中和明, 清光英成, “時区間論理に基づく時間的マルチメディアオブジェクトの同期再生スキームの実装,” 電子情報通信学会論文誌D-Ⅰ, Vol.J79-D-Ⅰ, No.10, pp.820-833, オブジェクト指向技術とその応用論文特集号, October,1996.

[FSM*93] 藤川和利, 下條真司, 松浦敏雄, 西尾章治郎, 宮原秀夫：分散型ハイパメディアシステム Harmony における情報間同期機構の実現, 電子情報通信学会論文誌D-Ⅰ, Vo1.J76-D-Ⅰ, No.9, pp.473-483, Sept 1993.

[Appl92] Quick Time 1.5 Developer Kit & Quick Time Components 1.5 Developer Kit, Apple Computer, Inc., 1992.

[MiKa94] アレン・マイナー, 菅晃：" オラクル・メディア・サーバー：一般家庭向け双方向マルチメディア・システムの概要," 情報処理学会, アドバンスト・データベースシステム・シンポジウム'94 会議録, pp.153-163, December 1994.

[THIU95] 友田政明, 堀内優希, 石川佳治, 植村俊亮："圧縮映像データの格納と検索," 情報処理学会研究報告, 95-DBS-101, pp.121-128, Junuary 1995.

[OgIU96] 小川政行, 石川佳治, 植村俊亮：" 圧縮映像データベースシステムにおける映像演算と実現手法," 情報処理学会研究報告, 96-DBS-106, pp.1-8, January 1996.

[KYTU96] 清光英成, 山本憲男, 高倉弘喜, 植村俊亮：" 動画像データベースの格納構造, " 情報処理学会研究報告, 96-DBS-109, pp.81-86, July 1996.

[NoMa96] 野中和明, 増永良文："MPEG-Ⅰ動画像データベースシステムの構築実験," 情報処理学会, アドバンスト・データベースシステム・シンポジウム'96 議事録, December 1996（掲載予定）.

[GiBT93] Gibbs, S., Breiteneder, C. and Tsichritzis, D. ："Audio/Video Databeses: An Object-Oriented Approach, " Proceedings of the International Conference on Data Engineering'93, pp.381-390, 1993.

[Egen94] Egenhofer, M.J. ："Deriving the Composition of Binary Topological Relations," Journal of Visual Languages and Computing, Vol.5, pp.133-149, Academic Press, 1994.

[KITA96] 小林みな子, 今井さやか, 富井尚志, 有澤博：" リアルワールドデータベースにおける 3 次元物体のモデル化," 情報処理学会研究報告, 96-DBS-109, pp.203-208, July 1996.

[TaSh96] 谷崎正明, 嶋田茂："能動型検索機構による空間情報提供方式," 情報処理学会研究報告, 96-DBS-109, pp.209-214, July 1996.

[KYTT96] 上浦真樹, 依田和也, 田島敬史, 田中克己：" 3 次元空間データベースにおけるデータモデルとアクセス管理機構について," 情報処理学会研究報告, 96-DBS-109, pp.215-220, Julty 1996.

[KuMa96] 黒木進, 牧之内顕文：" 単体複体の概念を用いた時空間データモデル Universe の設計," 情報処理学会研究報告, 96-DBS-109, pp.221-226, July 1996.

[MKK*96] 増永良文, 上林弥彦, 金森吉成, 北川博之, 今井浩, 古川哲也, 徐海蒸, 加藤和彦：”協調能動型データベースシステム技術の研究に向けて,” 情報処理学会研究報告, 96-DBS-106, pp.173-180, Junuary 1996.

[Masu96] 増永良文：“協調作業支援のための３次元空間データモデルの考察,” 情報処理学会研究報告, 96-DBS-109, pp.227-232, July 1996.

[Masu96b] Masunaga, Y.："An Interval-Based Approach to a Spatio-Temporal Data Model for Virtual Collaborative Environments," Proceedings of the International Symposium on Cooperative Databese Systems for Advanced Applications (CODAS), Kyoto, Japan, December 1996 (to appear).

[HiKi94] 広田光一, 木島竜吾：“バーチャルリアリティ・ソフトウェア,” bit 別冊, 仮想現実学への序曲ーバーチャルリアリティドリーム, pp.132-142, August 1994.

[ArMa96] 荒川ゆう子, 増永良文：”音声によるデータベースアクセスの研究,” 情報処理学会研究報告, 96-DBS-109, pp.13-18, July 1996.

5．知識共有のためのオントロジーの構成的構築

5. 知識共有のためのオントロジーの構成的構築

本研究では異種分散知識システムのためのオントロジーのモデルについて考察する。まず、ここではオントロジーをそれぞれが一つの概念化に対応するアスペクトという単位を用意して、それから構成的に全体のオントロジーを定義する。組合せの方法としては、単に違う領域のアスペクトを合わせる組合せアスペクトと、同じ領域の違う概念化のアスペクトを合わせるカテゴリ・アスペクトがある。次にこの方法によるオントロジーに基づくマルチエージェントシステムでは、アスペクトの変換が可能であることを示す。さらに実世界エージェントにおけるオントロジーの多重性について述べる。実世界エージェンはそれぞれの物理的能力に依存してオントロジーが異なる。共有オントロジーは個々のエージェントの共通する概念を表現すると共に、異なる部分は概念の詳細度と概念の解釈の違いによって表現する。最後に、分散的にオントロジーを開発するために、異なるオントロジーの類似性や差異を統計的に調べ、支援する方法について述べる。

5.1 はじめに

知識システムが構築容易性や再利用性は、そこで使われる概念体系がどれだけ明示化がなされているかに依存する。すなわち、異なる知識システム間での知識の共有と再利用で重要になるのが、オントロジーの共有である。ここでいうオントロジーとは対象領域の概念化の結果であり、概念と概念間の関係によって構成される体系である。

共有オントロジーを研究する場合、主に単一オントロジーによって行うものが多い（例えば［Gruber 93］）。しかし、実際のシステムを構築する場合、対象は同一でも複数の概念化が行なわれていたり、逆に同一の概念に見えても、別のものを指していることも多い。すなわち、オントロジーが異なることがあり、また異なるオントロジーを同時に用いている。すなわち多重オントロジーの作成・利用を可能にすることが必要である。

著　者：武田英明, 西田豊明

標　題：知識共有のためのオントロジーの構成的構築

出版物：情報処理学会 研究報告96-FI-42, page.9-15, 1996

本稿では多重オントロジーに関して行なった我々の４つの試みを説明する。まず最初に多重オントロジーを形式的な定義を論理的な枠組みで行なう。次にontoliguaの拡張として多重オントロジーを扱う言語ASPECTOLの定義とマルチエージェント系での利用について述べる。次に、実世界エージェントにおけるオントロジーの多重性について考察する。また、人により異なるオントロジーを統計的に調べ、その統合を支援する方法についても述べる。

5.2 アスペクトによる形式化

5.2.1 アスペクト

オントロジーは対象世界を記述する体系であるが、それは対象世界の概念化（conceptualization）の結果である。しかし、単一の概念化によって、記述の詳細度の幅をある記述や広範な対象世界の記述を行なうことは困難であり、実際的ではない。実際には、我々は複数の概念化を適宜組み合わせることによって、対象世界を、広い範囲を詳細度の幅がある形で、とらえている。オントロジーの構築においても、この複数の概念化の利用を反映した方法をとる必要がある。本研究では、一つの概念化による概念の体系をアスペクトと呼び、オントロジーをアスペクトの組み合わせによって構成する。

ここではアスペクトを基本（atomic）アスペクトと複合（composite）アスペクトに分けて定義する。基本アスペクトは、他のアスペクトとは独立に定義されるもので、アスペクトの名前と用いられる概念の定義および概念間の関係からなる。これに対して、複合アスペクトはアスペクトの組み合わせによる新たなアスペクトの構成を行なうもので、他のアスペクトを利用して定義される。複合アスペクトはさらに2種類あ

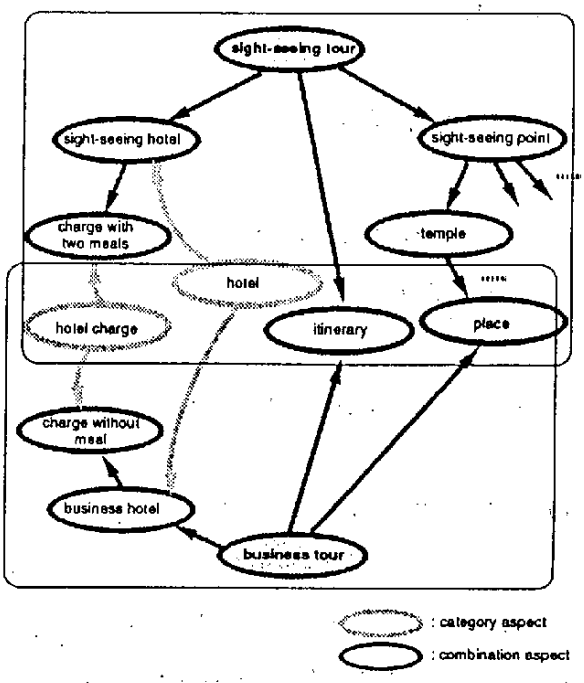


図5.2-1　共有可能な複数オントロジー

り、異なる対象領域のアスペクトを合わせることによって定義する組み合わせ（combination）アスペクト、対象領域が重なり合うアスペクト間の関係によって定義されるカテゴリー（category）アスペクトからなる。前者は基本的には二つのアスペクトの和を構成するものであるに対して、後者は異なる概念化の関係を記述している。例えば、旅行に関するオントロジー「旅行アスペクト」は「観光ホテル・アスペクト」と「行程アスペクト」などからなる組合せアスペクト、「ホテル・アスペクト」は「観光ホテル・アスペクト」、「ビジネスホテル・アスペクト」などからなるカテゴリ・アスペクトとして定義できる（図5.2-1参照）。

5.2.2 アスペクトの形式的定義

5.2.2.1 基本的定義

まず、他にアスペクトに依存しない基本アスペクトを定義する。

定義1 基本アスペクトAの一階述語論理における理論を$T(A)$とするとき、そのアスペクトの名前$aspect(A)$は次の式を満たす。

$$aspect(A) \longleftrightarrow T(A)$$

次に、組合せアスペクトを定義する。

定義2 アスペクト$A_1, \cdots, An$の組み合わせアスペクトの理論$T(A_{COM}(A_1, \cdots, An))$は次式を満たす無矛盾な理論である。

$$T(A_{COM}(A_1, \cdots, An)) = aspect(A_1) \wedge \cdots \wedge aspect(An) \wedge I(A_1, \cdots,$$

ただし、$I(A_1, \cdots, An)$はアスペクト$A_1, \cdots, An$のアスペクト間理論である。

この場合、組み合わされるアスペクトの中に同名の述語があった場合、予期せぬ結果をもたらすことになる。ここでは同名の述語がある場合は、同一の意味を持つと仮定している。そうでないときとは、以下で示すカテゴリアスペクトで表現するべき場

合だと考える[1]。

次にカテゴリアスペクトを定義する。この場合、要素となるアスペクトの理論が常に正しいのではなく、状況に依存する。このため、様相論理を導入して形式化する。以下では様相論理S4を用い、直感的には□は必然を、◇は可能を示す様相記号である。

定義3　アスペクトA_1, …, Anのカテゴリアスペクトの理論T（A_{CAT}（A_1, …, An））は次式を満たす無矛盾な理論である。

$$T(A_{CAT}(A_1, \cdots, An)) = \Diamond\, aspect(A_1) \wedge \cdots \wedge \Diamond\, aspect(A_n) \wedge I(A_1, \cdots, An)$$

ここで、I（A_1, …, An）はアスペクトA_1, …, Anのアスペクト間理論である。

組合せアスペクトやカテゴリアスペクトの要素として、また組合せアスペクトやカテゴリアスペクトを使うことができるので、アスペクトは階層的に定義することができる。いま、あるアスペクトAを$A=f$（A_1, …, An）と表現したとき、fはA_{COM}とA_{CAT}の組合せで記述できる。

5.2.2.2　アスペクト間の関係

アスペクト間の関係として、まず包含（inclusion）と真包含（strict inclusion）を定義する。

定義4　アスペクトAはアスペクトBに包含されるときは、$aspect(B) \vdash \Diamond\, aspect(A)$が満たされるときである。

定義5　アスペクトAはアスペクトBに真包含されるときは、$aspect(B) \vdash aspect(A)$が満たされるときである。

ここで包含関係とはアスペクト間の包含を意味しており、真包含関係とは論理的な

---

[1] 実際にはこの制約は厳し過ぎるので、以下の表現言語においては置換えを許している。

包含関係を意味している。同様に論理式に対しても関係を定義できる。

定義6　式fはアスペクトBに包含されるときは、$aspect(A) \vdash \Diamond f$が満たされるときである。

定義7　式fはアスペクトBに真包含されるときは、$aspect(A) \vdash f$

この定義の意味するところは、アスペクト理論を解釈するには二つの方法があるということである。すなわち、真包含関係とは通常の論理的解釈を意味し、包含関係とは、理論の代替をまで含めた解釈を意味している。

定理1　もしアスペクトAがアスペクトBに真包含されるならば，AはBに包含される。

もう一つの関係は、代替（compatible）関係で、二つのアスペクトが関係しあっていることをしめす。

定義8　アスペクトAとアスペクトBが代替関係にある時は、以下の条件のどれかが満たされる時である。

1. AとBが同一アスペクトである。
2. AとBを要素として持つアスペクトが存在する。
3. Aの要素A'とBの要素B'が代替関係にある。

定義9　式fがアスペクトAにおいて代替関係であるとは、fを含むアスペクトBが存在して、AとBが代替関係であるときである。

代替関係は無矛盾性や、後で述べる変換可能性を意味するわけではなく、その前提としてアスペクト間に共通部分があることを指している。

5.2.2.3　アスペクト間理論（1）：アスペクトレベル関係

カテゴリアスペクトの特徴はアスペクト間理論にどうようなものがはいるかに依存

する。以下ではその特徴としてコンパクトというものを定義する。

定義１０　もしカテゴリアスペクトのアスペクト間理論 I（A_1, …, An）が以下の式を満たす時、このアスペクトをコンパクトであると呼ぶ。

$$I(A_1, \cdots, An) \vdash \Box (aspect(A_1) \vee \cdots \vee aspect(An))$$

直感的には、コンパクトなカテゴリアスペクトは、その要素のアスペクトA_1, …, Anで十分であることを意味している。

定理２　コンパクトなカテゴリアスペクトが２つの要素アスペクトA_1, A_2からなり、A_1がA_2に真包含されるとき，A_1はAに真包含される。

実際にはこのAとA_1の関係はもっと強く、$aspect(A) \vdash \Box aspect(A_1)$である。これを固定的（rigid）関係と定義する。

定義１１　式fがアスペクトAにおいて固定的である時は、$aspect(A) \vdash \Box f$のときである。

もつひとつの特徴は排他的（exclusive）である。

定義１２　もしカテゴリアスペクトのアスペクト間理論 I（A_1, …, An）が次式を満たすとき、そのアスペクトは排他的であるという。

$$I(A_1, \cdots, An) \vdash \bigwedge_{k=1,\cdots,n-1} \bigwedge_{l=k+1,\cdots,n} \neg \Diamond (aspect(A_k) \wedge aspect(A_l))$$

5.2.2.4　アスペクト間理論（2）：対象レベル関係

また、個々の述語に対してアスペクト間理論を定義することができる。これを対象レベル関係と呼ぶ。例えば、アスペクトAの式pがアスペクトBのqを合意するとは

$$\Diamond\ (aspect\ (A) \wedge p) \rightarrow \Box\ (aspect\ (B) \rightarrow q)$$

と書くことができる。もっと一般的に“If アスペクトA_1の式f_1が真ならばアスペクトA_2の式f_2が真である”とは次式のように書ける。

$$\Diamond\ (aspect\ (A_1) \wedge f_1) \rightarrow \Box\ (aspect\ (A_2) \rightarrow f_2)$$

もし、アスペクトAの式pとアスペクトBのqが同値ならば、次のように書ける。

定理3　もし命題pがあるコンパクトなカテゴリアスペクトAの要素全てにおいて同値ならば、qはAにおいて固定的である。

5.3　マルチエージェント系における多重オントロジー

5.3.1　アスペクト記述言語

前節では論理的性質について述べたが、ここでは実際にアスペクトが記述可能な言語ASPECTOLを定義する。ASPECTOLはOntolingua［Gruber 92］の拡張として定義される（図5.3-1参照）。基本アスペクトはアスペクト理論に相当するOntoliguaの定義文によって定義される。組み合わせアスペクトは利用するアスペクトの宣言をさらに含む（図5.3-2（b）参照）。カテゴリー・アスペクトは利用するアスペクトの宣言とアスペクトに含まれる概念間の関係が論理式として定義される（図5.3-2（a）参照）。

```
基本アスペクト
  (define-aspect <アスペクト名>
    Ontolingua definitions are here
    ...
    )
組み合わせアスペクト
  (define-aspect <アスペクト名>
    (:include <要素アスペクト名>*)
    (:rename (<要素アスペクト名>!<述語名>. <述語名>)
*)
    Ontolingua definitions are here
    ...
    )
カテゴリ・アスペクト
  (define-category-aspect <アスペクト名>
    (:include <要素アスペクト名>*)
    (:category-type  <カテゴリ・タイプ>)
    (:rename (<要素アスペクト名>!<述語名>. <述語名>)*)
    (:default-aspect <デフォルト・アスペクト名>)
    (:translation
      (=> <変換前概念記述> <変換後概念記述>)
     ([:query-precedence <質問文変換時条件>]
      [:inform-precedence <伝達文変換時条件]
      (-> <変換前概念付帯記述> <変換後概念付帯記述>))+)*
    )
```

太字は予約語、[]は省略可能、+は一回以上の繰り返し、
*は0回以上の繰り返しを示す。

図5.3-1　ASPECTOLの文法

```
(define-category-aspect FEE
  (:use fee/a fee/b)
  (:category-type nil)
  (define-translation FEE
      (=> (fee/A!fee ?fee) (fee/B!fee ?fee))
    ((:query-precedence nil
      :inform-precedence nil
      (-> (fee-value ?fee ?value)
          (and (adult ?fee ?feel)
               (student ?fee ?fee2)
               (fee-value ?feel ?value)
               (fee-value ?fee2 ?value))))))
    ...
 )
```

(a) Category Aspect **fee**

```
(define-aspect temple/A (TEMPLE)
  (:use fee/A)
  (define-class temple (?x)
    :def (and (has-one ?x name)
              (has-one ?x fee)))
  (define-function name (?x) :-> ?n
    :def (and (temple ?x) (string ?n)))
  (define-function temple-fee (?x) :-> ?f
    :def (and (temple ?x) (fee ?f))))
```

(b) Combination Aspect **temple/A**

図5.3-2　アスペクト定義の例

5.3.2　アスペクト間の変換

知識コミュニテイ［Nishida 93］のようなエージェントに基づく分散協調システムでは、エージェントによって必ずしも同一のアスペクトに基づいているとは限らず、この場合異なるアスペクト間で知識（ここではエージェント間のメッセージ）を変換する必要がある。

競合するアスペクトにおいては、一方のアスペクトに基づく知識を他方に変換することが可能な場合がある。可能である場合とは、知識に含まれているアスペクトに関連するカテゴリアスペクトに適用可能な論理式が含まれている時である。

例えば、図5.3-3に示すような、料金の概念化が異なる二つの「寺」エージェント間のコミュニケーションを考える。今、料金区分という概念があるエージェントが料金区分のないエージェントに質問をした場合、図5.3-4に示すような手順により、料金区分のないメッセージに直される。返答はこの逆の変換が行なわれる。

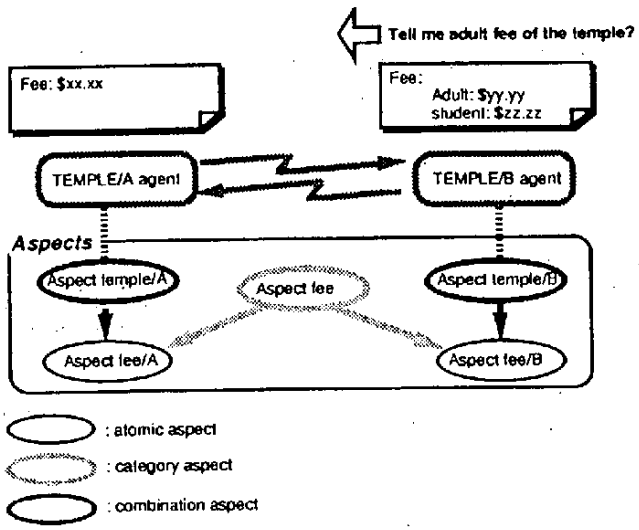


図5.3-3　エージェント間コミュニケーションの例

```
(temple ?x)
(name ?x ?y)
(fee ?x ?f)
(adult ?f ?f1)
(student ?f ?f2)
(fee-value ?f1 ?v1)
(fee-value ?f2 ?v2)
(< ?v1 500)
(< ?v2 400)
```

(a) The given message

```
(temple ?x)
(name ?x ?y)
(fee ?x ?f)
(adult ?f ?f1)
(student ?f ?f2)
(fee-value ?f1 ?v1)
(fee-value ?f2 ?v2)
(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
(temple-fee-elm ?f1)
(temple-fee-elm ?f2)
```

(b) Adding class definitions

```
(temple ?x)
(name ?x ?y)
(fee ?x ?fee)
(fee-value ?fee ?value)

(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
(temple-fee-elm ?f1)
(temple-fee-elm ?f2)
```

(c) Applying a translation formula

```
(temple ?x)
(name ?x ?y)
(fee ?x ?fee)
(fee-value ?fee ?value)

(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
```

(d) Removing unnecesary literals

図5.3-4　変換例

5.3.3　知識コミュニテイ KC_0における実装

知識コミュニテイ KC_0ではアスペクト間の変換機構を導入することで、メッセージ仲介機構［武田ほか 94a］をより効果的に利用することができる。仲介とアスペクト変換を組み合わせた場合のメッセージのやり取りの例を図5.3-5に示す。ここでは、前項で述べた変換機構はtranslatorというエージェントによって行なわれる。また、オントロジーはアスペクトを単位としてontology serverと呼ばれるエージェントで管理され、概念の定義、そのアスペクト、それを用いるエージェントなどの情報を提供する。

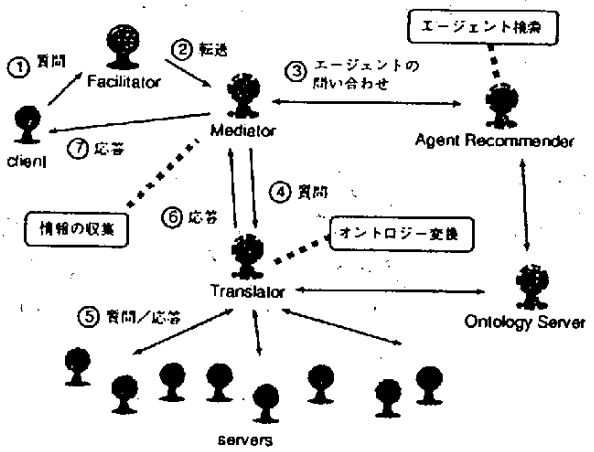


図5.3-5　質問の場合の仲介

5.4　実世界オントロジーにおける多重性

ここまでで議論してきたオントロジーは主に情報エージェントにおけるオントロジーであったが、これとは別にロボットなどの実世界に関係するエージェントにおけるオントロジーについても研究を行なっている（［武田ほか96］［寺田ほか96］参照）。

実世界エージェントは共同して動作するためには情報を共有する必要がある。ここではそれを概念の共有として扱っている。概念とはある事象に関する認識、記号、行動をまとめたものと捉えている。共有すべき概念としては、対象、空間、行為の3種類の概念にわけることができる（図5.4-1参照）。すなわち、対象・空間・行為の3つの共有オントロジーが必要となる。

ところが各実世界エージェントはセンサの有無・性能、アクチュレータの有無・性能からくる物理的機能と情報処理における機能から、それぞれ理解可能な概念は異なる。ここでは、この共有性と個別性を以下の方法で実現した。

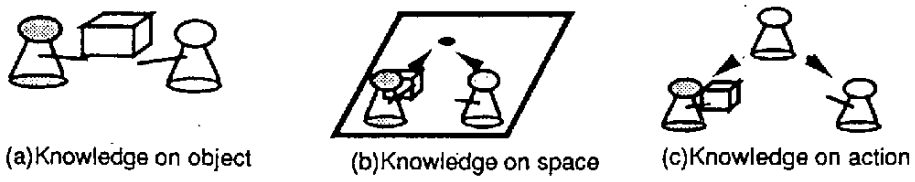


図5.4-1　3種類の概念の共有

対象に関する概念　共有すべき対象に関する概念は対象の分類（taxonomy）と対象の属性の概念の2種類である。しかし、エージェントの物理的構造や情報処理能力の違いによって、理解できる対象は異なってくる。対象分類においてはis-a関係を用いて対象を階層的に記述する。ここで抽象度の高い対象は多くのエージェントに共有されるが、より具体的な対象の理解はエージェントの能力に依存する。共通な対象の属性として、位置、色、重さなどがある。このうち位置としては現在位置とよくある場所（デフォールト位置）がある。対象の属性も同様に異なる抽象度で記述する。

空間に関する概念　空間に関する概念も共有すべきものであるが、これもエージェントの能力によってその記述が異なってくる。例えば、地図に示された経路をもとに移動するエージェントにとってはその経路が空間の記述であるし、相対的な位置しか理

解しないエージェントもありうる。ここでは、共有のための空間概念として以下の２つを用意した。

●前置詞表現：対象と前置詞で対象からの相対的位層を示す。ここでは、at, on, in, in-front-of, behind, to-the-right-of, to-the-left-of の７つを用意した。例えば、エージェントAの前ならば（in-front-of　agent-a）と表現する。

●行為表現：特定の行為が実現できる位置を、行為と対象の組合せで表現する。
例えば、エージェント１とエージェント２が出合う場所を
(meetpoint agent-1, agent-2)
と表現する。実際の位置は行為の主体のエージェントによって異なるが、行為の達成という意味では十分である。

それぞれは各エージェントがその表現を解釈する時に実際の位置が決定される。エージェントの個別性はこのとき反映される。

行為に関する概念　行為概念は行為の名前と主体や起点や終点などの行為に関わる属性で定められる。また行為間の関係は行為の分解関係として記述される。同様にその表現をエージェントが解釈する時にそのエージェントの能力に合わせて行なうことにより、個別性を実現する。

実装においては、共有オントロジーから個別オントロジーへ変換を行なう処理をしてから、各エージェントに送っている。

以上のオントロジーを用いてタスクの仲介を行なう機構を作成している（図５．４-２参照）。

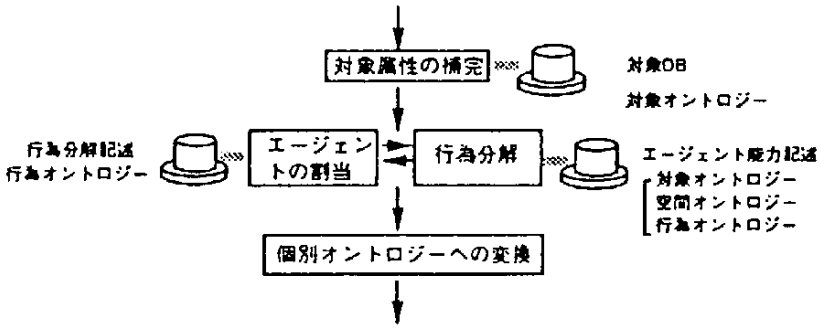


図５．４-２　タスク仲介の流れ

この結果の例を図５.４-３に示す。ここで、最終的には各エージェントのオントロジーになった行為（タスク）がエージェントに渡される。

ここでの多重性の表現と利用は次のようにまとめることができる。

●個別オントロジーは共有オントロジーと概念を共有

●個別性は詳細度と解釈によって実現

●多重性の利用は共有オントロジーから個別オントロジーへ変換する形で利用

●仲介は変換可能な部分だけを共有オントロジーから個別オントロジーへ変換

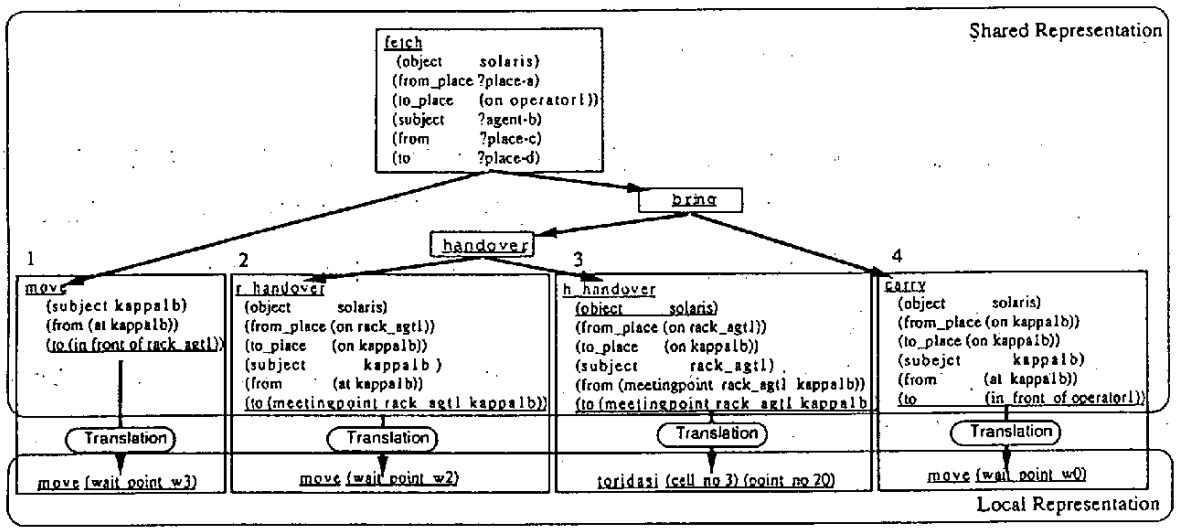


図５.４-３　タスク仲介の例

5.5　多重オントロジー作成の支援

ここでは別々に作られたオントロジーの共通性や差異を統計的に計算することで、多重オントロジーの作成を支援する方法について述べる［鷹合ほか96］。

ここでは複数の人間が独自にオントロジーを書いた時に、いかに相互のオントロジー間の共通性や差異を発見できるかについて統計的方法を使って調べている。図5.5-1に例を示す。ここでは、各概念のスロットの類似性を用いて計算をしている。「移動経路」のように異なる同名の概念が近くなっている。また、名前の異なる概念（例えば「行事」と「催し物」）が近くに配置される場合もある。このような類似性を調べることで、概念のカテゴリー化を進めることができると思われる。

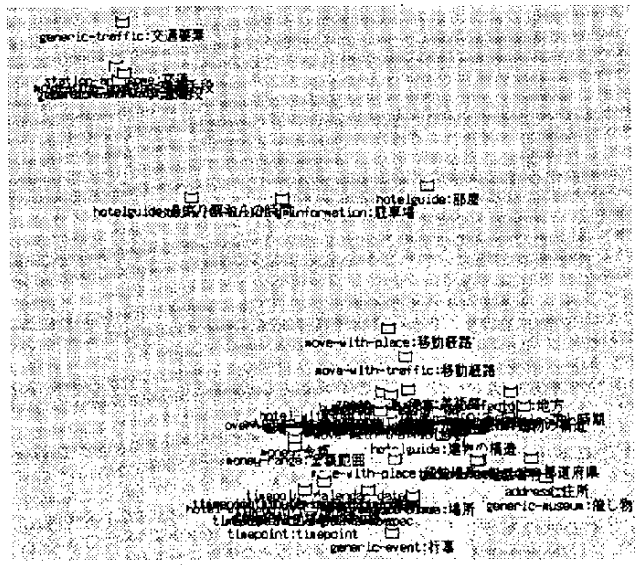


図5.5-1　多次元尺度構成法による統合

5.6　まとめ

本研究では柔軟なオントロジー構築を目指して、アスペクトという単位によるオントロジーの構築とその利用について述べた。アスペクトによるオントロジー構築は (i) 範囲を明示的に限定したオントロジー構築による構築の容易性と信頼性の向上、(ii) 構成的にオントロジーが構築可能、(iii) 例えば概略的定義と詳細定義の混在など、概念の多重性を許したオントロジーも可能、などの利点がある。

しかし、今回の研究では、(i) アスペクト変換記述および変換方法が必ずしも一般的

ではない、(ii) 変換記述の作成が容易ではない、などの問題がある。今後はより一般的な変換記述やその自動獲得などを研究していく必要がある。

参考文献

[Gruber 92] T. R. Gruber. Ontolingua: A mechanism to sup-port portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[Gruber 93] Thomas R. Gruber.Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.

[Nishida 93] T. Nishida and H. Takeda.Towards the knowledge-able community. In Proceedings of International Conference on Building and Sharing of Very Large-Scale Knowledge Bases '93, pp.157-166, Tokyo, 1993. Japan Information Processing Development Center.

[武田ほか 96] 武田英明, 岩田浩司, 鷹合基行, 沢田篤史, 西田豊明. 共有知識に基づく実世界エージェントの協調機構. 人工知能学会全国大会（第10回）論文集, PP.453－456, 1996.

[武田ほか94a] 武田英明, 飯野健二, 西田豊明. 知識コミュニティkc0における知識共有メカニズム. 人工知能学会全国大会（第 8 回）論文集, PP.279-282, 1994.

[鷹合ほか 96] 鷹合基行, 武田英明, 西田豊明. マルチエージェント系のためのオントロジーの分散開発支援環境. 人工知能学会全国大会（第10回）論文集, PP.139-142, 1996.

[寺田ほか 96] 寺田和憲, 武田英明, 西田豊明. 実世界エージェントのためのオントロジーの構築. 人工知能学会全国大会（第10回）論文集, PP. 203-206, 1996.

6．生産スケジューリング・システム構築における ドメインオントロジーの構成とその再利用

6. 生産スケジューリング・システム構築における ドメインオントロジーの構成とその再利用

6.1 はじめに

本稿では、生産スケジューリング・システム構築のための再利用部品群の構成とその再利用性について、実システム開発の経験を踏まえて報告する。オブジェクト指向の概念に基づくソフトウェア開発は、分析から実装へ至る反復的な過程を、実世界における概念を反映した共通のモデルを前提として行なえるものとして発展が期待されている。さらに、ソフトウェア再利用にあたって個々のクラスを単位とするのではなく、類似のアプリケーションに共通するクラス群の振舞に着目するオブジェクト指向フレームワークの概念［Johnson 88］が注目されている。しかしながら、再利用を前提とした対象領域の分析において、実世界のどのような側面に着目し、どのような観点からモデル化を進めるべきかは必ずしも自明ではない。したがって、フレームワークの設計においても類似の問題領域に共通して見い出される概念を整理しボトムアップに再構成していく過程が不可欠である［Beck 94, Johnson 94］。

筆者等はこれまでスケジューリング問題をジョブ割付問題として定式化し、部品合成によるプロトタイプ開発環境の構築［Hori 94］、ならびに問題解決手続きの部品化に取り組んできた［Hori 95］。さらに、ジョブ割付問題を前提とした部品抽出の経験を踏まえて、生産スケジューリング・システム構築のためのフレームワーク SCOOP（Scheduling Components in Object-Oriented Paradigm）をC++を用いて実装した［Hori 96］。本報告では、SCOOPの適用例を示すとともに、その経験を通して得られたフレームワーク設計の指針を示す。以下、6.2ではSCOOPの概要について特に生産現場をモデル化するためのドメイン・オントロジーの構成を中心に示す。6.3ではフレームワーク中のサブシステム（モジュール）を適宜組み合わせることによって開発されたアプリケーション例を紹介する。最後に、知識処理システムを対象としたフレームワークの設計過程について考察する。

6.2 スケジューリング・フレームワーク

このフレームワークは、生産設備や生産工程といった製造分野に特徴的な構造を表すスケジュール・モデル、計画立案の機能を担うスケジューリング・エンジン、さらにユーザインタフェースに関するモジュールの3つのサブシステムから構成される（図6.2-1）。したがって、サブシステム相互の組合せ、ならびにサブシステム内でのクラス部品の選択といった2レベルでのカスタマイズが可能となっている。以下では、スケジュール・モデルとユーザインタフェイス・サブシステムの構成について述べる。スケジューリング・エンジンの詳細については文献［堀95］を参照されたい。

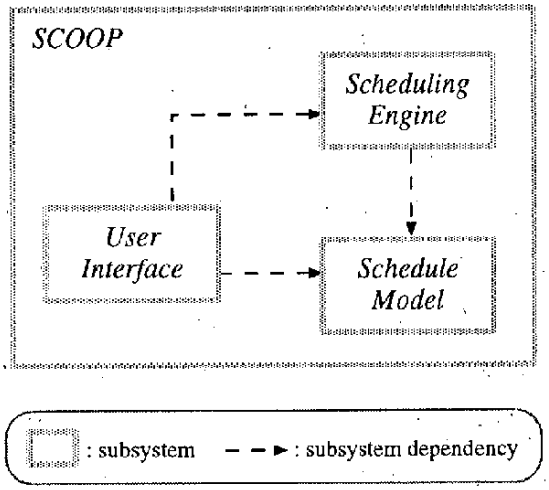


図6.2-1．スケジューリング・フレームワークの構成

(1) スケジュール・モデル

スケジュール・モデルはスケジューリングの観点から生産現場をモデル化したもので、図6.2-2に示すように資源（Resource）、生産工程（Process）、製品（Product）に関する3つのサブシステムから構成されている。ただし、本報告での図表現はOMT［Rumbaugh 91］の記法に基づいている。

資源サブシステムは、工作機械などの生産設備に関するもので、同一タイプの資源集合である資源グループ（ScRscGrp）、個々の資源（ScRsc）、および資源の休止時間（ScRest）からなる。一連の休止時間はその雛型である休止パタン（ScRestPattern）に基づいて生成される。一方、生産工程はあるタイプの資源において実施される最少の作業単位であるユニット（ScUnit）、ある製品を完成させるために順序付けられた一連のユニット全体に相当するジョブ（ScJob）からなる。ユニットは個別のロットに

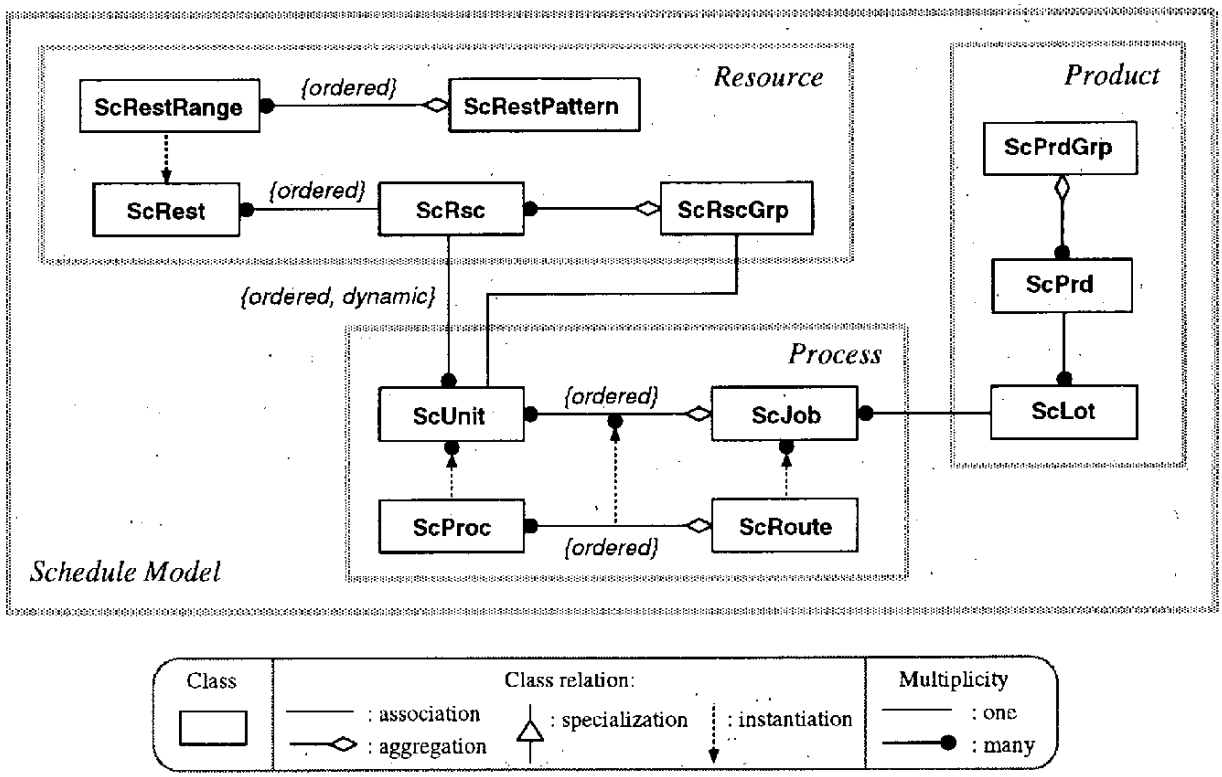


図6.2-2　スケジュール・モデルの構成

対応付けられた具体的な作業系列であり、製品の種類ごとに定められたルーティング情報（ScRoute）と工程情報（ScProc）に基づいて生成される。さらに、製品については、製品グループ（ScPrdGrp）、各製品（ScPrd）、および納期や数量の定められた受注単位であるロット（ScLot）からなる。

一般にスケジューリング問題は、割付対象としてのユニットと割付先である資源との間に多対一関係を動的に設定するものとして特徴付けられる。スケジュール・モデルではユニットと資源の多対一関係に加えて生産工程をはじめとする製造分野に特徴的な問題領域の性質がモデル化されている。それらの関係や構造的特徴は、問題解決（スケジューリング・エンジン）や内部状態の視覚化（ユーザインタフェース）に関わる他のサブシステムにおける前提を明示的にするもので、ドメイン・オントロジーとしての役割を果たす。

(2)　ユーザインタフェース

生産スケジューリング問題では、作業の基本単位であるユニットを特定の資源に割

り付けるとともに、作業の開始／終了時刻を設定しなければならない。したがって、ユニットの開始／終了時刻に関する属性値ならびに資源とユニット間の関連をスケジュール・チャートとして提示することは、ここでのユーザインタフェースに求められる主要な要件の１つである。ユーザインタフェース・サブシステムでは、スケジュール・チャート（ガント・チャート）に相当する２種類のクラスが定義されている（図６.２-３）。１つは各資源ごとに関連付けられた複数ユニットを時間軸上に表示するもの（UiRscSchChart）で、もう一つはジョブごとに関連付けられた複数ユニットを時間軸上に表示するもの（UiJobSchChart）である。いずれも一連の作業として関連付けられたユニットを時間軸上に配置するものであり、このレベルでの属性や操作は共通する抽象クラス（UiSchChartBase）から継承される。さらに、２種類のスケジュール・チャートには、画面の詳細表示の異なるもの、あるいは編集機能が追加されたものなどがサブクラスとして定義されている。

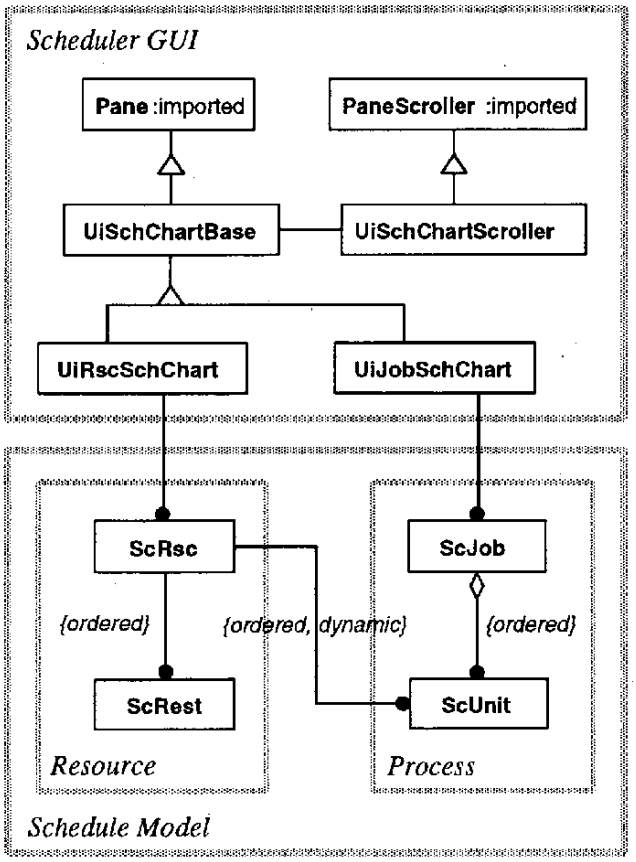


図６.２-３　ユーザインタフェース・サブシステムとスケジュール・モデルの関連

図６.２-２のスケジュール・モデルの構成からも明らかなように、複数のユニット

を直接取りまとめる単位となり得るものは資源（ScRsc）とジョブ（ScJob）しかない。したがって、スケジュール・チャートの基本的な構成としては上記の２種類以外にないと言うことができる。逆に、資源あるいはジョブ以外の概念を介して複数ユニットを配置するスケジュール・チャートが必要ならば、その概念はこの対象領域にとって本質的な役割を果たすものであり、スケジュール・モデルに追加されるべきであると言える。

資源に対してユニットを配置するUiRscSchChartは、資源に関連付けられた休止時間も表示することが可能であり、生産設備の稼働状況を把握する上で有用な情報を与えることができる。一方、ジョブは顧客からの注文に相当するロットを生産の単位として分割したものであり、ジョブからロットへの関連を介して注文の納期や最早着手可能日といった属性を参照することができる。したがって、ジョブごとに関連するユニットを配置するUiJobSchChartでは納期等の情報も提示することができ、出荷前の作業状況を確認する上で有効な情報を与えるものとなっている。実際の生産現場においても、生産設備の稼働率を向上させることを業務目的とする部署と、注文を期限までに顧客に出荷することを目的とする部署があり、それぞれの部門のユーザにとって異なる観点から生産の進捗状況を把握する場合が一般的である。

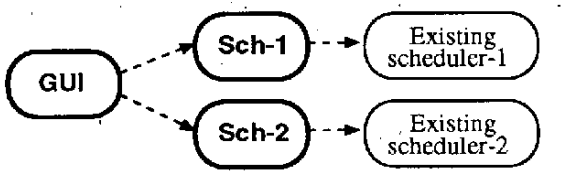
6.3 フレームワークの適用

(1) サブシステムの組合せ

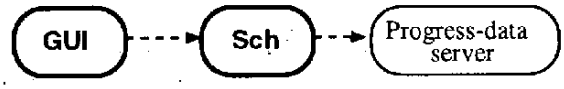
フレームワークを用いた個別アプリケーションの開発は、アプリケーションに固有の属性や操作をサブシステム中の該当するクラスのサブクラスとして定義するとともに、各サブシステムを組合せることによって行なわれる。3つのサブシステムのうち、スケジュール・モデルは前提となる生産現場をモデル化するもので、どのようなアプリケーションにおいても少なくとも1つ必要である。他の2つのサブシステムは、いずれも特定のスケジュール・モデルを前提とするもので要求仕様に応じて適宜用いられる。これらのサブシステムはいずれも10～20個程度のクラスからなり、単独のクラスよりも大きな単位である。したがって、所与のアプリケーションの構成に含めるべきか否かの判断はシステムの設計者にとって容易である。典型的な組合せとしては図6.2-1の各サブシステムを1つずつ組み合わせるものであるが、実際にはシステムの利用状況や運用形態の進化に伴いその構成は様々に変化する場合が一般的である。

図6.3-1に本フレームワークを用いて開発されたアプリケーションにおけるサブシステム構成を示す。図6.3-1(a)は、既存のスケジューリング・システムが異なる2つの工場で運用されている状況で、そこから得られた予測値と生産現場から収集された実績値の相違を提示するスケジュール・モニタの構成である。このシステムは、実際の要求仕様に基づいて構築されたもので、2つの工場での進捗を同一ウインドウ上で画面を切替えることによって表示するものとなっている。

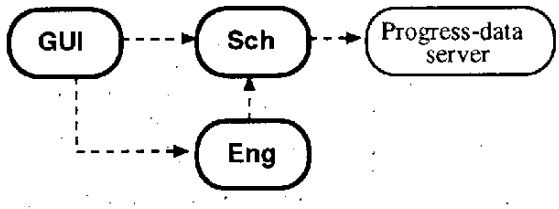
図6.3-1(b)(c)は、段階的なアプリケーション開発過程におけるサブシステム構成の推移の一例で、実際の生産現場で現在稼働中のものである。(b)の構成は、一定の時間間隔で最新の実績データをスケジュール・チャート上に表示するもので、おもに生産現場での作業指示が遵守されているかどうかの確認とその改善のために用いられている。(c)では、既に作業に着手しているジョブの残りの工程を全て完了するまでの時間を予測するために、スケジューリング・エンジンのサブシステムが統合されている。さらに、生産のサイクル・タイムが長期にわたる場合、生産工程をいくつかの部分工程に分割したり、生産ライン全体について異なる観点からスケジューリングを行なうことが考えられる。そのような場合には、複数のスケジューリング・エンジンが1つのスケジュール・モデルを介して協調する分散スケジューリングの形態を取る［吉田95］。



(a) Schedule monitor



(b) Progress monitor



(c) Online scheduler

Sch : Schedule model　**GUI** : Graphical user interface
Eng : Scheduling engine　--→ : Subsystem dependency

図6.3-1　サブシステム構成の例

(2)　フレームワークの進化

フレームワークを用いたアプリケーション開発ならびにフレームワークの拡張作業において、各サブシステムのコーディングとデバッグに要した時間の比率を図6.3-2に示す。各開発サイクルの時間は一様ではないが、ここでは各サイクルに要した時間を正規化している。また、フレームワークの開発と適用には一人の開発者が一貫して関わっている。

図6.3-2(1)は図6.3-1(a)のスケジュール・モニタを開発した際に、共通ユーティリティクラス群（Com）、スケジュール・モデル（Sch）、およびユーザインタフェース（GUI）のコーディングに要した時間の比率を示している。さらに、各サブシステムに相当する矩形において斜線で示された部分は該当するアプリケーションに固有の部分、すなわちフレームワークには含まれない部分の開発に要した時間の比率を表している。例えば、図6.3-2(1)のスケジュール・モデルに関するコーディングは全てアプリケーションに固有の属性や操作を追加するものであり、フレームワークとして提供されたモデルを一切変更する必要がなかったことを表している。図6.3-2(2)はスケジュール・モニタの開発終了後、フレームワークの拡張を行なったサイクルであ

る。図6.3-2(3)のi), ii)は図6.3-1(b)に示したアプリケーションの開発に関するもので、ii)のサイクルで行なわれた実データによるテストとデバッグの完了後、現場に導入され現在実運用されている。さらにiii)は図6.3-1(c)の開発に関するものである。

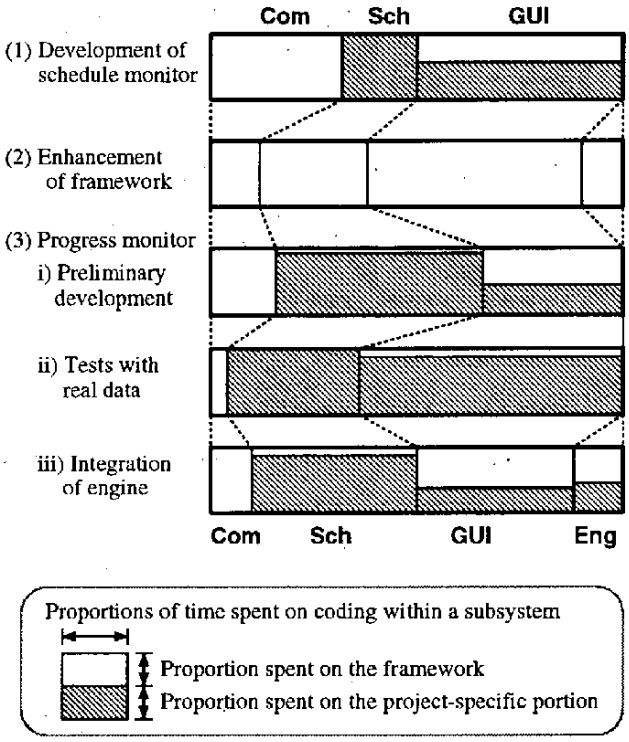


図6.3-2　フレームワークの拡張と適用の経緯

スケジュール・モデルに関する開発時間は、フレームワーク拡張のサイクルに相当する図6.3-2(2)を除けば、95%がアプリケーション固有の部分に関するものであった。フレームワークの適用にあたっては、各々の生産現場をスケジュール・モデル（図6.2-2）との関係は容易に対応付けることができ、新たなクラスは全てサブシステム中のいずれかのクラスのサブクラスとして定義された。このことから、本フレームワークにおけるスケジュール・モデルは生産現場のドメイン・オントロジーとして再利用性の高いものであると言える。一方、ユーザインタフェース・サブシステムについては、図6.3-2(2)のサイクルを除いた開発時間のうちアプリケーション固有の部分に関するものは47%に過ぎず、フレームワーク自身に機能拡張の余地が残されていると言える。

6.4 フレームワークの設計

(1) 設計パタンの利用

前節の適用例ではフレームワークの利用者がその開発者と同一であったが、実際には再利用対象物の提供者と利用者は異なる場合が一般的である。そのような状況では、再利用対象物の背後にある意図や設計理由を明示的にすることが重要となる。フレームワークについては、その前提を文書化しフレームワーク利用を容易にすることを目指して設計パタンの収集と整理が進められている。

本稿で述べた生産スケジューリング・フレームワークの背後にある設計理由の一部は、設計パタンをあてはめることによって補うことができる。ここでのサブシステム構成に特徴的な性質のいくつかは、文献［Gamma95］に示されたパタンを用いて以下のように説明することができる。例えば、生産進捗状態の変化はスケジュール・チャートに適宜反映されることから、スケジュール・モデルとユーザインタフェース・サブシステムの依存関係はObserverパタンによって説明される。一方、スケジューリング・エンジンは様々な計画手順の詳細な差異をユーザから隠し、手順の使い分けを容易にするためにサブシステムとして独立しているが、これはStrategyパタンの利用例となっている。その他にも将来的な変更や拡張を前提とした典型的な計画手順の雛型を与えるTemplate Methodパタン、スケジュール・モデルにおいて複雑に関連し合ったクラスへの参照を容易にするために窓口となるクラスを導入するFacadeパタン等から設計理由を補うことができる。

設計パタンは様々なフレームワークに共通して見い出される断片的な性質を、より一般的な原則として再整理したものとなっている。したがって、設計パタンはフレームワークよりも相対的に細粒度の構成要素である。一方、フレームワークが想定すべき抽象度については、対象領域の性質を反映したものと、より汎用的なものを再利用性の観点から明確に区別すべきであるとの指摘もある［Rumbaugh 93, Jacobson 95］。しかしながら、再利用性と有用性のトレードオフの中で妥当な均衡点を見い出すためには、対象領域の特性を踏まえてフレームワーク設計の指針を提示する必要がある。

(2) 抽象レベルの設定

図6.3-2は第1版のフレームワーク完成後、引き続いて行なわれた詳細設計・実装の作業を対象としている。また、それに先立って行なわれた対象領域の分析では、再利用範囲を設定するまでに異なる観点ならびに抽象度でのモデル化が試みられた。

第1版のフレームワークが開発されるまでの主要な段階におけるモデル化の観点と抽象度の推移を図6.4-1に示す。

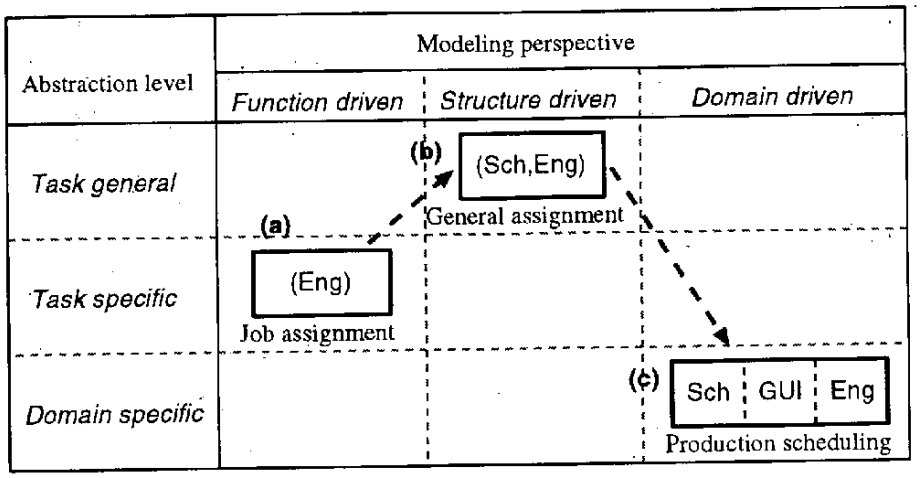


図6.4-1　フレームワークの設計プロセス

(a) ジョブ割付問題

図6.4-1(a)の段階ではスケジューリング問題をジョブ割付問題として定式化し、部品合成によるプロトタイプ開発環境をCLOSを用いて構築した［Hori 94］。ジョブ割付問題は、旅客機の搭乗ゲートへの割付、製鋼工程の生産計画、レストランでの人員配置など様々な分野におけるスケジューリングを対象とするもので、「種々の制約条件を満たした上で、与えられたすべての'ジョブ'を'資源'と'時区間'から構成される割付平面上に配置する」ものとして捉えられる。これらの分野に共通する断片的な機能単位は再利用可能な11個の問題解決部品として抽出された［Hori 95］。部品の記述には集合操作をはじめとする12種類の基本操作が用いられたが、一般的なデータ構造を前提としていたためにアプリケーションに特徴的な操作対象の構造が明確ではなかった。

(b) 一般割付問題

ジョブ割付問題は資源列と時間軸から構成される割付平面にジョブを配置するもので、割付先については時間軸を伴う2次元空間が必然的に仮定されていた。そこで、割付対象であるジョブや割付先の資源や時区間についても、より抽象的なレベルでそれらの構造的性質を考えることによってジョブ割付問題の背後にある多様性を捉えることができる。

図6.4-1(b)の段階では、「種々の制約条件を満たした上で、与えられたすべての'割付対象'を'割付先'に割り当てる」ものとして一般割付問題を想定し、その問題解決手順が前提とし得る操作対象のモデルをマクロ構造として抽出した［堀94］。そして、マクロ構造を拠り所として製鋼工程の生産スケジューリングと電気回路の配線設計における問題解決手順の間に類似性を見い出すことができた［堀94］。なお、配線設計システムのプロトタイプはC++を用いて構築され、ある設計業務における実システムの基本設計として採用された。

マクロ構造は操作対象の構造を著しく抽象化したものであることから、再利用性の観点からはより広い問題領域を対象としている。しかしながら、抽象化された問題解決手順を所与のアプリケーションに適合するようにカスタマイズする労力が大きく、その過程を支援する有用な指針を得ることは容易でないと考えられる。

(c) 生産スケジューリング問題

(a), (b)の段階を経て、それぞれ経験的な問題解決手順ならびにその手順が前提とする操作対象の構造が明らかとなった。しかしながら、それらは依然として特定の応用分野あるいは産業分野に依存しない抽象レベルでの記述であり、類似のアプリケーションに共通するクラス群と典型的なメッセージ交信の形態を規定するフレームワークとして実装するには対象分野をさらに具体化する必要があった。

ジョブ割付問題は、人員配置、生産計画、旅客機の搭乗ゲートへの割付、タンカーの桟橋への割付など様々なスケジューリング問題を包含するが、図6.4-1(c)では特に生産スケジューリング問題を対象とした。生産計画業務は各現場に固有の条件が少なくなく、また生産設備の変更や増強にも継続的に対応する必要があることから、ソフトウェアの再利用によって多様な要求に柔軟に対処していくことはシステム開発部門にとって重要な課題となっている。

生産スケジューリング・フレームワーク中の3つのサブシステムのうち、スケジューリング・エンジンの構成はおもに(a)の段階での経験に基づいて設計された。またスケジュール・モデルについては、(b)の段階で考察したマクロ構造を参考にするとともに、既に多くの導入実績がある生産計画システム・パッケージの開発者にもヒアリングを行なうことによって(c)の段階で詳細設計を行なった。ユーザインタフェースについては、(c)の段階で既存のパッケージとスケジュール・モデルの構成を考慮して設計された。第1版のフレームワークの開発には、図6.3-2に示した作業全体の約3倍の時間を要したが、その多くは初期の分析結果に明示的に現れないシステム的なオブジェ

クトについてデータ構造の設計と実装に費やされた。

(3)フレームワーク設計への指針

上記の分析／設計の過程は、スケジューリング問題以外にも対象領域に固有の経験的な知識や手順を必要とする知識集約的問題を対象としたアプリケーション・フレームワークの設計に有用な指針を与えると考えられる。ただし、処理の流れが明確な定型的業務、あるいは原子力プラントや電子回路の故障診断のようにモデル化の対象物に対してその振舞を説明する理論的な動作原理が存在する場合には、この方針が適さないと考えられる。以下にその概略を示す。

まず、類似のアプリケーション例に対して共通する問題解決手順を見い出す。この段階は経験的な処理手順、すなわち機能的側面に着目しておりその意味ではオブジェクト指向的ではないという見方もできる。しかしながら、特定のアプリケーションに依存しないレベルで機能的側面に着目することの有効性は経験的に実証されており、その成果として診断／計画／設計など様々な領域における問題解決手順が既に知られている［Clancey 85, McDermott 88, ティヘリノ 93, Breuker 94］。

次に、そのような経験的処理手順の操作対象に着目することによって、処理手順の背後にある構造的側面を明らかにする。ここで得られた構造はフレームワークの骨格を与える。最後に、業務分野を設定し、そこで特徴的な性質を反映させながらフレームワークの設計と実装を行なう。特に、フレームワークの全体的な構成としては、業務分野のモデル、問題解決手順、ユーザインタフェースに対応するの3つのサブシステムを前提に設計を詳細化することができる。

ある程度具体的な業務分野を想定することによって再利用性が損なわれる部分もあるが、このことは関連アプリケーションとの相互運用性も考慮しながらフレームワークを進化させていく上で有用であると思われる。生産スケジューリング・フレームワークの場合には、製造業に特化することによって資材所要量計画、在庫管理、工程管理といった関連アプリケーションとの連携を考慮しながらフレームワークを拡張させることが可能となる。

参考文献

[Beck 94] K. Beck : Patterns and software development. Dr. Dobb's Journal, February, pp.18-21, 1994.

[Breuker 94] J. Breuker and V. Van de Velde, Eds.: "CommonKADS Library for Expertise Modelling: Reusable Problem Solving Components". IOS Press, 1994.

[Clancey 85] W. Clancey : Heuristic Classification. Artificial Intelligence, Vol.27, pp.289-350, 1985.

[Gamma 95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides : "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley, Reading, MA, 1995.

[Hori 94] M. Hori, Y. Nakamura, and T. Hama : Configuring problem-solving methods: a CAKE perspective. Knowledge Acquisition, Vol.6, No.4, pp.461-488, 1994.

[Hori 95] M. Hori, Y. Nakamura, H. Satoh, K. Maruyama, T. Hama, S. Honda, T. Takenaka, and F. Sekine : Knowledge-level analysis for eliciting composable scheduling knowledge. Artificial Intelligence in Engineering, Vol.9, No.4, pp.253-264, 1995.

[Hori 96] M. Hori and T. Yoshida : SCOOP: An object-oriented framework for production scheduling systems. Proceedins of the 6th IFIP International Conference on Advances in Production Management Systems, pp.389-394, Kyoto, Japan, 1996.

[堀 94] 堀,中村 : 汎タスクレベルを介した問題解決知識の再利用性について. 第8回人工知能学会全国大会論文集, pp.85-88, 1994.

[堀 95] 堀,吉田,吉川 : 問題解決手続きの構成とドメイン・オントロジーの関連について. 第9回人工知能学会全国大会論文集, pp.383-386, 1995.

[Jacobson 95] I. Jacobson and M. Christerson : A confused world of OOA and OOD. Journal of Object-Oriented Programming, Vol.8, No.5, pp.15-20, 1995.

[Johnson 88] R. Johnson and B. Foote : Designing reusable classes. Journal of Object-Oriented Programming, Vol.1, No.2, pp.22-35, 1988.

[Johnson 94] R. Johnson : Why a conference on pattern languages? ACM SIGSOFT, Software Engineering Notes, Vol.19, No.1, pp.50-52, 1994.

[McDermott 88] J. McDermott : Preliminary steps toward a taxonomy of problem-solving methods. In S. Marcus, Ed., "Automating Knowledge Acquisition for Expert Systems", pp. 225-256. Boston: Kluwer Academic, 1988.

[Rumbaugh 91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson : "Object-Oriented Modeling and Design". Prentice Hall, Englewood Cliffs, NJ, 1991.

[Rumbaugh 93] J. Rumbaugh : Objects in the twilight zone. Journal of Object-Oriented Programming, Vol.6, No.3, pp.18-23, 1993.

[ティヘリノ 93] ティヘリノ, 池田, 北橋, 溝口 : タスクオントロジーと知識再利用に基

づくエキスパートシステム構築方法論 ---タスク解析インタビューシステムMULTISの基本思想---. 人工知能学会誌, Vol.8, No.4, pp. 476-487, 1993.
[吉田95] 吉田, 堀 : スケジューリング・フレームワークの半導体製造工程への適用. 計測自動制御学会 第21回システム情報関連合同シンポジウム論文集, pp.321-326, 1996.

7．　設計対象の変更とその根拠を記述する言語系Engineering Scriptの設計

7. 設計対象の変更とその根拠を記述する言語系 Engineering Script の設計

7.1 まえがき

本報告では、設計生産知識の体系化の一環としての設計根拠の計算機記述化の方法について述べる。設計過程は仕様を満足するように設計対象を変更してゆく過程と見做すことができるが、設計対象の変更は、感覚的に行われるものでもなければ特殊な場合を除いて予め定まった手順をなぞったり計算するものでもなく、関連する事象に関する考察に基づく根拠の説明を持つ。従来、このような設計根拠は設計支援システム記述されることがなく、時空間的に隔たった場合の設計変更において、設計根拠で仮定された前提条件の侵犯によってさまざまな事故を生じている。この問題は自動車業界などでは以前から知られており、トラブル対策集や設計変更記録集などが蓄積されるようになっているが、充分に利用されているとは言えない。

もし、このような設計根拠がその意味処理を伴って計算機記述されるならば、実用的な価値は大きいものがある。一方、それが関連事象に関する認識を表現することから、工学理論の原型と考えることができるのが重要である。すなわち、計算機記述される設計根拠を集成し、新たな観点で体系化し直せば、従来の理学的あるいは科学的理論に代る、実用問題を解くことのできる工学的体系の構築が可能となろう。

本報告では、技術者の持つ主観的な理解を重視した設計根拠の計算機記述化について、その基本的な考え方を示し、設計根拠の説明のための新しい言語系Engineering Scriptの設計を示す。

本研究が最終的に目指すのは設計のための工学知識の再体系化であるが、体系化については冨山らの研究［冨山］がある。また、設計根拠の記述は第13回設計シンポジウム講演論文集（'95-7-12, 13, 東京）Reverse Engineeringと深く関係し、上野の研究［上野］とその目的において関連が見いだせる。さらに、内容的には田浦らの研究に２つの意味で近接性が見いだせる。一つはプロセス的知識の記述［田浦94］としてであり、他方は深層知識の表現法の探求［田浦95］においてである。筆者らは、設計の

著　者：伊藤公俊, 川野邊衛

標　題：設計対象の変更とその根拠を記述する言語系 Engineering Script の設計

出版物：第13回設計シンポジウム講演論文集, 1995.7.12-13, 東京

ための深層知識はドメインの再調査によって新たに構築するべきと考えているが、それには情報処理あるいは人工知能からの接近が不可欠である。情報系の研究が領域固有の意味内容の処理を指向するべきであるとの主張であり、溝口の提唱する「内容指向AI」［溝口］の一部と深く共鳴するところである。

7.2 工学知識の体系化

ここでは、工学知識とは主に設計および生産のために用いられる技術知識を言うことにする。理学あるいは科学、また文学、政治学および経済学など人文系の学問に比して、従来地位の低かった工学の存在理由を特に主張するために工学知識という用語を用いる。ただし、工学知識についても、その体系化についても明確な定義は存在しない。しかし、自ずとそれ以外の知識とは異なる性質をもち、それが工学知識の体系化の目的に反映することになる。

工学知識の体系化の利点として、以下の２つが考えられる。

(1) 技術開発の重複を避けるための技術知識の時空間的な伝達の実現
(2) 新たな技術開発の啓発

(1) において、空間的伝達とはいわゆる技術移転であり、時間的伝達とは将来からみれば技術の蓄積、現在からみれば技術の継承となる。これらの目的を達成するためには、工学的知識が体系化されている必要があるのは広く認識されている。一方、量的な制約などから、この目的に対しては計算機の利用が不可欠となる。

(2) の目的は、技術がただそのまま伝達されるだけでは不十分であることを主張している。技術移転や技術継承において、この問題は既に認識されていると言えるが、現状の情報処理技術では人間に新たな技術開発を期待せざるを得ず、啓発とは人間を啓発することである。したがって、体系化を計算機システム化することとして、ブラックボックス化したり自動化してしまうことは誤りである。

次に、上の工学知識の体系化の利点を実現するためには、以下の体系化の条件が必要である。

(a) 人間の理解が平準化されなければならない
(b) 説明［Wright］が与えられること

(a) は、多義性の解消を意味する。伝達可能であるためには明らかなように多義的であってはならない。しかし、一方でなんらかの啓発が可能であるためには問題解決のための多義的な理解が必要であり。(b) 説明が行なわれなければならない。これに対して、近代科学的方法は、特に主観と客観の分離および機械的還元論に特徴付けられ、

(a) の条件にしたがって(1) の目的だけが追及されてきたといえる。それに対する本報告での主張は、主観の積極的導入と、還元論の根底にある因果的決定論の再考である。すなわち、工学では、因果則の全ての条件を述べ立てることは不可能であり、常に「不測の事態」が生ずる危険性をはらんでいる。それにも拘らず人間は設計生産活動を（不完全ながらも）行ってきた。人間の理解は、理学的規範からは正しいとは限らないが、実用上有用であったということが重要である。

7.3 CADシステムのタスクとしての設計根拠の記述

従来、CADシステムは設計業務の効率化と設計品質の向上の支援を目的として開発されてきた。それは、主として自動化による人間の行為の代替の形で実現されてきたが。ここでいう効率化あるいは品質とは、短期的企業間競争における利潤によって測られるものであった。しかし、吉川が指摘しているように、工業が短期的視野で効率化を追及したり、品質を向上させても地球的あるいは長期的視野において破綻を来すことになる［吉川］。その直接的な例が、PLでありリサイクル問題であろう。従来は製品の欠陥による社会的損失に対する生産者の責任が軽く、gadgetのような新機能の追加のために設計生産性の追及が求められていたが、今後は製品に対する生産者の責任が増大すると考えられる。すなわち、

(1) 設計がその時点で知り得る問題をどこまで考慮していたかが問われる。
(2) 設計ミスによる損失補償が営利活動にとって重大な障害になる。

また、IMSで重視されている技術のDiffusionや社会構造の変革に対する要求に従って技術および技術者の流動化の必要性が高まっている。

以上のような新しい状況に対して、従来のCADシステム開発の局所的かつ短期的な指向性は変革が必要であると考えられる。単に処理の自動化を達成するだけではむしろ有害であり、対象技術の進展に対応してCADシステムに組込まれる知識が改訂される必要のあることは徐々に理解されるようになってきた。しかし、設計技術の継承あるいは移転を考慮すれば、CADシステムは実行知識が編集可能であるだけでなく、その背景となる設計技術を明示的に表現しなければならない。なぜなら、システムに組込まれる実行知識の編集が、今後は異なる設計技術者によって行われる機会が増大し、技術者同士の対象理解の差異が知識の運用に障害を生ずる危険性が高い。教育などによって理解の平準化を計る努力とともに、可能な限り計算機表現化しておくことによって、経時的地域的に安定した技術継承あるいは技術移転が可能となる。また、計算機記述化を計ることは、設計者に自らの行為を明確化させることを強いることであるが、これは野中が企業活動のために暗黙値の外在化の必要性を主張していることとも整合すると考えられる。

本報告では、CADシステムの指向する目標として、設計技術の記述化を提唱するものである。技術の記述化がCADシステム開発の目標の一つとなる。しかも、人間の柔

軟な設計行為の代替は現在のいかなる形の計算機にも最も不得意とするところであり、むしろ機械的演繹能力あるいは多量の情報蓄積能力を生かすべきであると考えられる。設計根拠の記述化は、技術の体系化の一環あるいは前段であるが、体系の創造がやはり人間の行為であるのに対して、より計算機向きのタスクである。本研究では、認知科学的主観性を重視する。これは、筆者らの設計対象モデリングシステムCONMOTOを設計者のメンタルモデルと捉えていることとよく整合する。

次に設計図書との関係を考えよう。従来、上記の設計根拠は原則的には設計図書に記述されることになっている。現実的な理由から必ずしも常に実施されていたわけではないが、実行しさえすれば有効であると考えられてきた。しかし、紙上に書かれた図書だけでは不十分である。この問題は、最近のマルチメディア化の中でも見落されていることのように思われる。明らかなように、図書による伝達は人間の理解に依存しており、それは文脈あるいは背景知識が共有されていなければならない。最近は、技術者の実経験機会の減少や国際分業、製品の短寿命化などによって技術的文脈の共有が困難になってきている。しかも、文脈が共有された場合でも語義の多義性の問題は解消されない。筆者らは、これらの問題を対象モデルを参照するAbStruct記述によって解消しようとしている。紙上に書かれた図書のもう一つの欠陥は、図と文書の束縛が実現されないことであり、実務上しばしば不整合が生じることが知られている。明らかなように、この問題の解決には図情報と文書情報とを拘束することであるが、本研究では、AbStruct言語によってEngineering Scriptの語彙と設計対象モデルが束縛されるようになるのが特長である。

7.4 AbStruct 言語による技術記述の語彙定義

記号処理に基づく現在の情報処理、特に人工知能関連の研究は、常に記号の意味解釈の問題、すなわち外界と記号の対応付け、一部にはsymbol Groundingと呼ばれる問題を抱えている［Harnard］。機能や属性を記述する基本vocabulary の定義を明確にシステム上に行わなければ、機能や属性を処理したと主張することはできない。一方、従来からシステム工学の応用として、機能語を利用して形式知を操作する手法、例えば、value Engineering（VE）、品質機能展開手法（QFD）、ISMなどが企業を中心に利用されてきている。これらの方法は、計算機の導入によって技術開発においてさらに有用性を増すものと考えられ、例えば、TQCC活動の属性モデリングへの応用が論じられている［中島］。しかし、これらの活動でも、人工知能などと同様に機能や性質に関する語彙は定義されない。確かに、これらの活動は人間が行うものであるから、全てを人工知能研究と同様に扱うことはできないが、前述の技術の記述化からは、ある段階からはvocabularyが明確になっていることが必要である。

筆者らは、設計対象モデルCONMOTOの実現構造の属性あるいは共同である目的属性を定義する言語系AbStructを開発している［伊藤92］。もし実現構造が製造過程を経て実物となることを承認すれば、AbStruct は実物（人工物）の性質に名付けを厳密に与える言語と考えることができる。さらに、対象世界が設計対象モデル表現できるならば、VEやQFDなどの語彙もAbStruct言語による表現が可能である。

名付けが完了すれば、それ以降は定義された言語ラベルを用いて自由に言語的記述が可能となる。したがって従来の設計言語は、本来ならば名付けの終了後の設計対象の操作などを記述していたことになり、言語を利用する人間側に対象理解の共通化という厳密には非常に困難な要求を課してきたことになる。筆者らの方法によれば、モデル一般が持つ演繹性と指示内容の厳密さを備えた言語的記述が得られ、名辞の参照は常に対象モデル（実現構造）との対応において行うことが可能となる。ところで、AbStruct のような方法はベースとなるモデルの意味論には依存しないから、機械設計用であるCONMOTOの実現構造以外のモデルをベースとするように変更することが可能であり、応用範囲は広いものと考えられる。

7.5　主観的な因果説明

7.5.1　工学における理解と納得：深層理解

ルール型知識を始めとする表層知識の限界が指摘され、深層知識の重要性が指摘されてから相当な時間が経過した。しかし、大変に難しい課題であることは事実としても、知識工学関連の研究で、表層性を脱却した例は存在しない。設計関連ならば、短期的効果が設計計算に求められているに過ぎない。深層知識とは、原理や法則などであるとする向きがあったが、直観的にも、力学の法則を知ったからといって問題が解けるとは限らないのは明らかである。何を深層知識とするべきかについての議論の欠落が指摘できる。

工学における深層知識に限定すれば、それは設計（工学）問題の解法に寄与しなければならない。しかし、設計では結果に対する社会的責任が問われるという性質があり、最終的に人間は、自らの設計を納得し［森］、その責任を負わなければならない。したがって、工学における深層知識とは、対象とする事象についての人間が納得可能な説明でなくてはならない。

7.5.2　心的説明のための因果性

一般社会では対象のメカニズム（機序）の解明の必要性がしばしば言及される。メカニズムの解明とは、因果的に対象を説明することである。因果説明とは、認知科学研究で指摘されているように、原因と結果を反転させることによって望ましい結果を得るための手段を与えることから、設計のような問題解決にとって本質的であることである。波多野は領域知識の本質であると主張している［波多野］。

機械振興協会技術研究所では「加工技術データファイル」と呼ばれるデータ集を蓄積および刊行している。このデータファイルは、欧米の類似名称のデータ集と異なり、数値データ集というより自然言語による仮説的事象説明が大きな部分を占めるとともに、これが最も有用であるという。ここでの説明とは、検討の余地は多く残ってはいるが。因果論的説明であるように見える。しかし、機械加工（特に、このデータ集が対象としているようなサイズの）の現象は複雑であり因果決定論的に説明するのは困難であろうと予想されるものでもある。それでも、人間の問題解決（この場合、加工のパフォーマンスの向上）には有用であるということは、類推など人間の思考のダイ

ナミズム［村上］が因果的説明　――一般に多義的である――　を与えることによって発揮されるということを強く仮説させる。因果的説明とは、人間の心的過程のためにあり、説明の持つ多義性が、人間の知的活動の豊かさを生む。

一方、因果的決定論は、定性推論研究の一部では、所与の原理として無反省に取り入れられてしまっている。すなわち、因果的推論結果を現象との対比によって理論的に検証するといった不用意なことが行われることがある。しかし、例えば力学の理論化をみると皮肉にも理論体系から因果関係、したがって「力」の概念を排除する傾向にある。因果関係とは、直接的に実証されるものではなく、人間の心的認識と捉えるべきであろう。西洋医学的な機械論的説明がつかない理論を持つ東洋医学が、ハリや指圧などによって有用性を持つことが一つの例になっている。

7.5.3　法則および理論の「検照」

知識工学では、獲得される仮説的知識は事実に照らして検証されるものと捉えられている。しかし、工学が対象とするような複雑系では、その法則あるいは理論は排中律的あるいは二元論的に検証することは不可能である。複雑な系は多義的に解釈可能であり、そもそも、事実を知るということが人間の持つ価値負荷の下にある。したがって、仮設からの演繹結果と新しい事象の認識（計測など）結果が大きく異ならなければ、とりあえず仮説を正しいものとするという姿勢を採用することが工学問題には適していると考えられる。このような姿勢の下で、仮説と事実の認識を対照することを、検証（verification）と対照（contrast）を組み合わせて「検照」と呼ぶことにする。

実験やシミュレーションは従来検証のための重要な過程であった。しかし、それらは人間の与えておいた視点設定に依存しており、決して理論的な意味での検証ではない。本研究において、設計根拠は設計過程あるいは使用される過程において検照され変更されてゆくものであると考える。その様子を図７.５-１に示す。

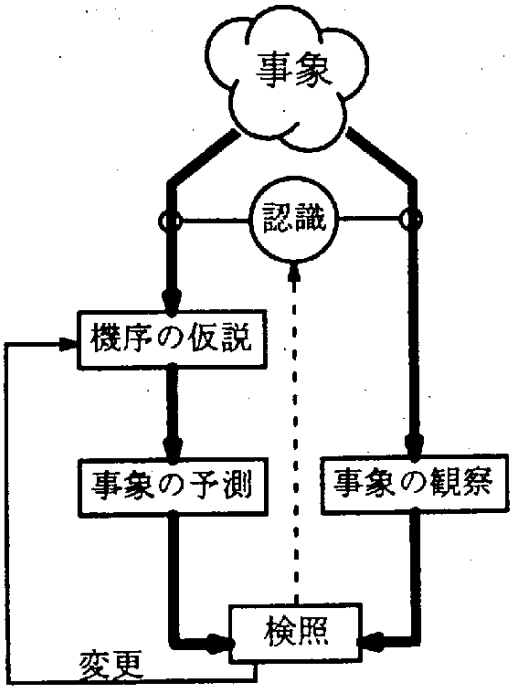


図7.5-1　検照のプロセス

7.6　Know-Why の提唱：第3のKnowing

Know-How は、言うまでもなく工業界で非常に重視される知識の一種であり、これにはKnow-What が対応するものとされている。前節までに述べてきたように因果説明が問題解決の豊醸を生むとすれば、Know-Why の方が工学にとって本質となるはずである。Know-Why は、それ自身新しい考えではなく、How／What 二元論ではWhat に分類されてきた。しかし、人間が介在するならば、HowはWhy 説明から生み出され、通常その逆ではない。How だけの知識は表層的であり、順序などの柔軟性が乏しい（図7.6-1）。また、宣言的な形をしたWhat 知識は、明らかなようになんらかの解釈を行ってHow に変換しなければ問題解決には有効とならない。したがって。Know-Whyを第3の知識の形態としての地位を与えることを提唱したい。

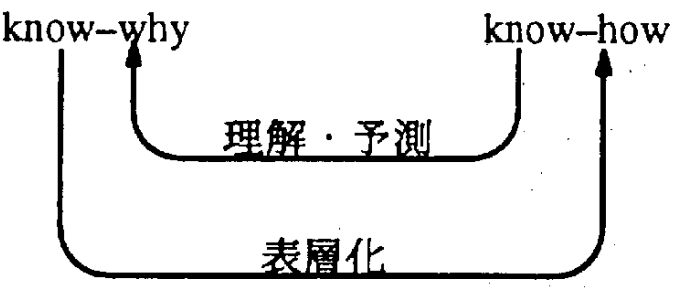


図7.6-1　know-howとknow-why

Why型の知識の重要な一例が物理学や電子物理学などで知られる「効果（effect）」がある［高尾］。特に興味深いのは、因果決定論が成立しない量子力学の分野でも工学的応用を行うときには因果的な「効果」が生み出されてきたことである。「効果」は、しばしば詳細を隠蔽したある抽象度および粒度を持ち、定性的であることも多い。また。効果は、通常基本的な法則に基づく説明が与えられるが、必ずしも法則群の機械的適用にはなっておらず、目的指向性を持つのが特徴である。そこからは問題解決の手段が自然に導かれ。その手段のための問題も明らかとなる（ただし、その解決が容易とは限らない）。

7.7　設計根拠の記述と設計プロセス

設計対象モデルは、それが設計結果のみを表現するために設計変更に弱く、それゆえに設計結果の再利用が阻害されている。設計変更に弱いとは、設計変更によって元となる設計において考慮されていた暗黙的前提条件が侵害されるために予想外の副作用を生じることであって、相当以前から、設計意図や設計根拠を設計対象モデルに表現することの必要性が指摘されてきた。設計根拠は、従来、対象表現としての図面とは別途設計文書として記述されるか。技術者の常識として図面から解読されることを期待されてきた。

設計根拠をCONMOTOの枠組みに従って分類すると以下のようになろう。

(1) 目的属性に寄与する実現構造の集合
(2) 着目する実現構造の説明
(3) 対象操作の説明
(4) 既存の理論・法則の適用

(1) は、以前から筆者が提唱していた最も表現容易な情報であり。例えばどの部品が集合となってリンク機構を構成するかをいい、その結果としての運動（目的属性）が現れる。(2) については既定義の実現構造をどのように分節するかの問題がある。(4) は、力学法則などのように大変強力であるが人間の理解という観点では強く寄与せず、むしろ設計計算問題としての利用価値が高い。

機械設計では、（統計的）少数の事例についてのみ実験や実物が存在するだけで、関与する現象や性能発揮の機序が必要な抽象度、粒度では判明していない（Ex. 運動精度、工作精度）ため、要求仕様から演繹的に設計解を求めることは期待できない。したがって生成検証法に頼ることになるが、設計案の生成を組織的に行うことは同じ理由によって不可能である上に、設計案検証すなわち設計計算の正当性が得られるとは限らない。そのため、以前から遠藤は類似設計法を提案している［遠藤］。類似設計法は、原型図を可能な限り流用し、本当に必要な箇所のみを変更することによって新しい図面を作成するとともに、同様に遠藤の創案になる組織的検図法によって検証する方法である。最初に選択される原型図は生産実績のある製品のものであり、新製品はすべて幾つかの原型図の変型された子孫としてそれらの実績を継承する。さらに、不具合の発生時には、継承関係に従ってフィードバック（およびフィードアサイド）

が行われる。

旧図の編集は図面CADでしばしば行われるが、遠藤の類似設計法は旧図の過剰な変更が有害であることに基づいた設計指針である。これには、生産設備を共通化させるためのような個別性の高い条件から、寸法変更の効果のオーダが一定でない（例えば面積と重量は、それぞれ２乗と３乗のオーダである）といった一般的条件までが関与する。ここで注意するべきことは、原型図の選択は機能的類似性に依存するとは限らないことである。類似性は設計者の認識によって定まり、設計仕様は全く異なる場合がある。いずれにせよ類似設計法によれば、新しい設計はすべてあるプロトタイプの変型であり、それ自身が別の設計のためのプロトタイプとなる。

設計対象の変形には以下の三段階が考えられる。

① 実現構造が内在する変形法
　（例）CONMOTOシステムの概念ベクトル、変数
② ①および部分構造の交換による変形法
　（例）ユニットの置換
③ 全体的な変形法

①は見かけ以上に重要であり、多くの設計計算問題がこれに該当する。設計対象は、基準の計画構造（追い寸など）によって自身の変形法を内在していると見ることができる。従って、この場合類似設計法に従えば、設計対象モデルは全てが変形可能なプロトタイプ（Transformable Prototype）であると考えることができる。CONMOTOシステムでは動作記述をも包含したプロトタイプとなる。次に、定型設計以外の設計では、設計仕様とその変更に対する設計対象の変形法とを記述しておく必要がある。もし類似設計法に従った②、③のような変形法を記述することができれば、この変形法を含めたTransformable Prototypeを定義することができる。すなわちTransformable Prototypeとは、原型とその変形法を組合せたものをいう。類似設計は、その変更履歴によってネットワークを構成することが知られているが、設計対象の変更をEngineering Scriptによって表現するとしたときの様子を図７．７-１に示す。また、Transformable Prototypeは、上記の段階に応じた内部表現を持つ。本研究の場合、これらは全てCONMOTOシステム上に統合的に構築される。筆者らは、この基盤構造を属性モデリングプラットフォームと呼んでいるが、全体の関係は図７．７-２のようになる。

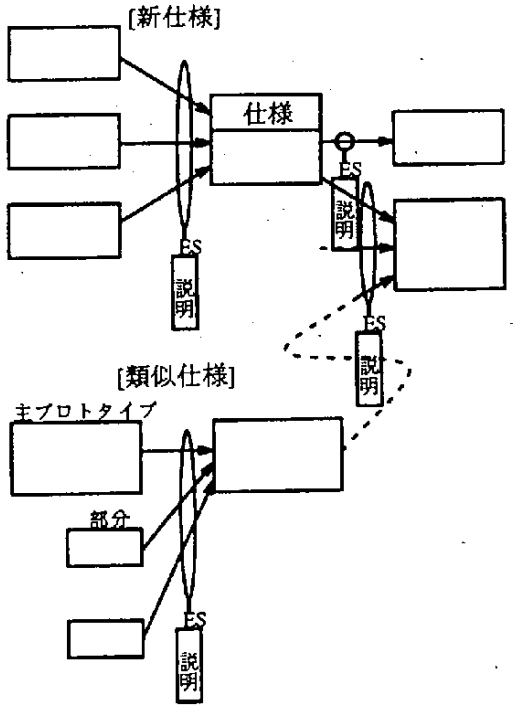


図7.7-1　類似設計ネットワークの生成

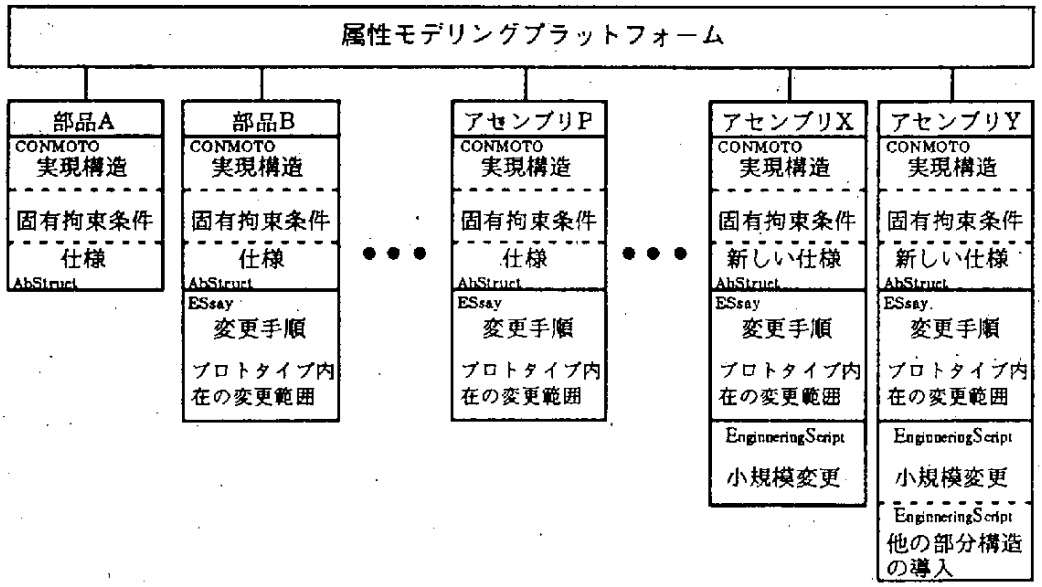


図7.7-2　プラットフォーム上のTransformable Prototype

7.8 Engineering Script

Engineering Scrip（ES）は、設計対象の変更操作とその根拠の目的論的説明［Wright］を記述するためのCAD用言語系であって、上記のKnow-Why型の知識をretrospectiveに計算機表現するものである。ただ、明らかなようにKnow-Why型の知識の全てを表現可能とするものではなく、新しい法則や理論を直接表現するものでもない。しかも、CADシステムに組込まれるにも拘らず、設計支援の直接的効果を持たない。しかし、設計対象は、必ず根拠を持つべきであり、その説明は設計のための深い知識となる。

ES設計の基礎とする観察の一つに、設計知識は設計変更時に明かとなるという仮設があり、機械設計がプロトタイプを出発点とする仕様変更に応じた類似設計のネットワークを構成することから、このネットワークが設計知識を担うと考えることができる。なお、設計根拠のうち定量的関係が存在するものは、因果を逆転させることにより設計計算問題とすることができる場合があり。その表層性を意味するものと解釈される。明らかなように、この方法でSymbol Groundingを理論的に完全に解決することは不可能である。ベースとなるモデルのGroundingが理論的には不可能であるからである。設計対象モデルをベースとする場合も、加工法などの一部が自動製造装置として実現される手続き的意味論において疑似的に実現されるのみである。さらに、モデ

```
ＥＳ ≡
        名称 スコープ定義 ES_Begin 対象操作文列 ES_End ;
スコープ定義 ≡
        AbStruct定義文          (*構造変数宣言，構造値宣言*)
対象操作文 ≡
        操作 説明 or_else(%) 説明 or_else 説明 . . . (*多重説明*);
操作 ≡
        構造変数 := 構造値 ;   (*構造変更*)
      | 関係式 ;               (*拘束条件参照*)
      | 構造変数 ;             (*構造参照*)
説明 ≡
        if_not  説明スコープ定義  因果記述文列(*反事実的説明*) fail      (*回避問題*)
           quantified(%) (式)  after（引用）;
      | 説明スコープ定義 target（着目変数） as 因果記述文列              (*改善問題*)
           quantified(%) (式)  after（引用）;
説明スコープ定義 ≡ (*ModelOnModelの定義*)
        view_from(%)  (視線) with AbStruct定義文;
      | in_view_of(%)  (視界) with AbStruct定義文;
因果記述文 ≡
        出現変数定義(%) 説明変数 := 説明値   by(%) 説明変数   in(%)  条件;
                                (*結果*)       (*原因*)       (*因果成立条件*)
引用  ≡
        出典文書の参照
(%)は，当該項の省略可を意味する。
```

図７．８-１　Engineering Scriptの文法

ル間の共約不可能性の問題も存在する。一方、工学的には、理論的な完全性は必ずしも不可欠ではない。特にベースモデルの意味解釈は、一般的な枠組みを与えることから。通常の教育などを介した人的文脈に担われていても不都合はないと考えられる。むしろ問題となるのは個別的な名辞の差異であり、これが定義可能となることの意義は大きい。

図7.8-1に、Engineering Scriptの暫定的な文法を示す。ESは、Knuth［Knuth］に従って代入文（Assignment）を主要な構成要素としており、設計対象操作（変更）の記述と事象の推移の記述の2種類の代入文からなる。図では、代入を強調するために代入記号”：＝”を使用している。

佐藤［佐藤］によれば、意志決定に関する問題構造は一般に、以下のように分類される。

発生型問題	探索型問題	設定型問題
逸脱問題	改善問題	回避問題
未達問題	強化問題	間発問題

ここでは、設計対象操作が行われる問題構造を、佐藤の問題構造のうち、

(1) 回避問題

(2) 改善問題

の2種類であると仮定し、それぞれの説明を行う構文とした。

ESの記述は、人間の希望する粒度あるいは抽象度で開始する。しかし。場合によっては、そこで用いられる語の意味は上記の意味では定義されていない。この定義は、後からAbStructなどを用いて行うこととする。したがって、ESは階層化された（しかし、簡単な）ハイパーメディアとなる。以下に、ESの性質などについて箇条的にまとめて列挙する。

【ESの目的】

(1) 設計手順と設計対象を相互に関連付けたまま保存する。

(2) 技術の透明性の実現

異分野への展開（社会ニーズの多様化）

時間的継承

地域的拡散

(3) 理学的記述の困難な総合用（Synthesic）知識の記述

(4) Symbol Grounding

多義性の解消、intrinsic性の向上

【ESの設計方針】

(1) （主観的）因果の記述 ⇒ 検照

(2) 多重説明（一つの対象に複数の説明が存在すること）の実現

(3) 素朴性または直感性の反映 ← 「設計の源泉」

(4) 代入文形式の採用 → 組み込み述語の削減、技術的意味論の分離

(5) Executable（実行可能）性 ← 記述のInsentive

【ESの用途】

(1) 設計事例の構造的記述（事例記述のための構造的方法の提供）

(2) 領域理論前駆体の記述

【ESによる簡単な記述例】

図7.8-2（a）のような鍔付き車輪を両輪とするレール走向子を考える。これが外部の熱の影響を受けやすい場所に置かれる場合、片側車輪を鍔なし車輪に変更する必要がある。CONMOTOシステムでは、概念ベクトル木の付け替え操作がCONMOTO内部変数の代入として記述できるので、車輪の付け替え操作は直接ES記述できる。従って、車輪の付け替え操作の説明は下記のようになる。

```
WbeelR：＝FlangeLessWheel（*鍔なしへの付替操作*）
  IF_NOT（*説明系列*）
  AxisLen：＝AxusLen＋AxisLen* σ BY ExternalHeat；
    （*軸が伸びる*）
  RailClearance：＝- RailClearance * τ ；
    （*クリアランスが負になる*）
  FAIL
  QUANTIFIFE σ＝H * △t ；
```

【現状の問題点】

(1) 記述のためのInsentive

Design Reviewなどへの併合

PLなどの外部強制

実行可能（Executable）性

→　編集時のトラップ

自動採集　←　操作履歴からの抽出

(2) ベースモデルの表現能力

(3) Groundされない情報の残留

(4) 記述の規範の欠如

(5) 類例検索および利用の枠組み　→　類似性の定義

(6) 概念化の方法論

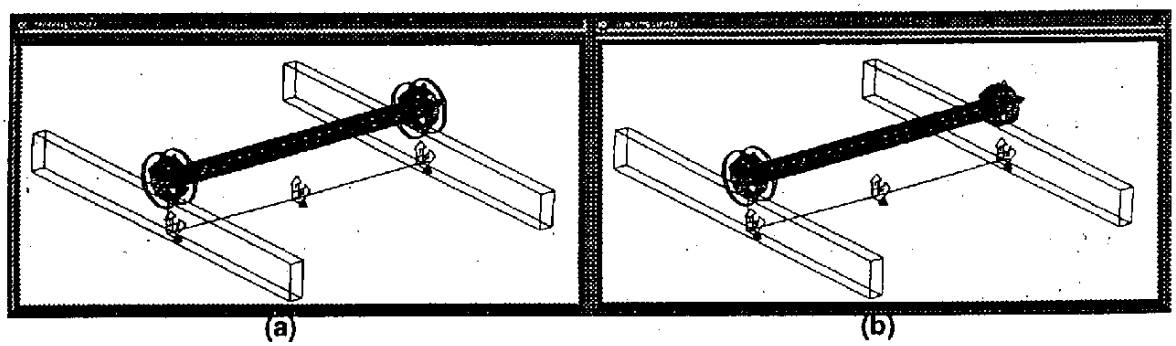


図7.8-2　レール走行子の設計変更

7.9 あとがき

設計根拠を記述するための言語系Engineering Scriptの構想と、前提となる考察を述べてきた。Engineering Scriptは、機械設計用設計対象モデルCONMOTO上に抽象実体定義言語系AbStructを用いて定義される語彙を持つことを特長の一つとしている。現状では。実装が完了しておらず、構想段階に留っている。今後、早期に実装を完了し、記述実験を行って洗練化を計るとともに、実例の蓄積を行ってゆく予定である。なお、ESは、設計対象の変形操作とその根拠を記述するように開発を進めているが、それを設計対象の挙動の因果的記述に応用することは自然な拡張であるので、この方向での考察も進めてゆきたいと考えている。

支援を受けている。記して御礼申し上げる。

参考文献

[Harnard] S.Harnard: The Symbol Grounding Problem, Physica D, 42 (1990) 335.

[Wright] G.H. von Wright（丸山ほか訳）: 説明と理解, 産業図書 (1985).

[村上] 村上陽一郎: 科学と日常性の文脈, 海鳴社, (1979).

[Holland] J.H. Holland ほか（市川ほか訳）: インダクション, 新曜社 (1991).

[伊藤92] 伊藤公俊, 川野邊衝: 抽象実体記述言語によるTransformable Prototypeの構成, 第10回設計シンポジウム講演論文集, pp.11-20 (1992).

[伊藤95] 伊藤公俊, 川野邊衝: 抽象実体定義言語AbStructに基づく属性モデリングプラットフォームの構築, 日本機械学会設計・システム工学部門講演会論文集 (1995).

[遠藤] 遠藤健児: 設計情報のまとめ方, 税務経理協会 (1984).

[冨山] T. Tomiyama et al.: Systematizing Design Knowledge for Intelligent CAD Systems, Proc. PROLAMAT92 (1992).

[上野] 上野晴樹: リバースエンジニアリングとその課題; 電子情報通信学会技術研究報告書 AI94-71 (1995).

[中島] 中島裕生: 設計における知識表現とモデリング, 人工知能学会誌, Vol.7, No.2, (1992).

[田浦94] 窪田敦之, 田浦俊春: 設計における行動知識の表現に関する研究; 第12回設計シンポジウム講演論文集, 203 (1994) 25.

[田浦95] 田浦俊春, 小山照夫, 川口忠雄, 芝田京子, 叶迎春: 自然法則データベースの研

究: 本講演会論文集 (1995).
[高尾] 高尾ほか: 理工学における効果の事典; 東京電機大学出版局 91973).
[佐藤] 佐藤允一: 問題構造学入門; ダイヤモンド社 (1984).
[溝口94] 溝口理一郎: 知識の共有と再利用研究の現状と動向; 人工知能学会誌, Vol.9, No.1, (1994) 3.
[Knuth] D. Knuth: Algorithmic Thinking and Mathematical Thinking; American Mathematical Monthly (1985) March. {bit. Vol.18, No.4 (1986).}
[森] 森　政弘（監修）: 納得の工学; 開発社 (1977).
[波多野] 波多野誼余夫: 理解と教授の相互作用; 人工知能学会誌, Vol.10, No.3, (1995) 3.

8. DODDLE: 計算機可読型辞書に基づく法律オントロジー構築支援環境

8. DODDLE：計算機可読型辞書に基づく法律オントロジー構築支援環境

要　旨

本稿では、開発済みの計算機可読型辞書（MRD）に基づく法律オントロジー開発支援環境DODDLEについて述べる。本支援機構は、最初に入力として与えられた法律に関する語彙との照合を通して、MRDから関連情報を抽出し、不要な情報を除去することにより初期モデルを獲得する。次に初期モデルに対して、法律固有性を付加することにより、法律オントロジーの開発支援を試みる。本稿では、入力語彙との照合に基づくMRDからの関連情報の抽出と、不要な情報の除去による初期モデル構築方法を述べる。さらに、初期モデルを法律オントロジーとして利用するための法律固有性の付加方法について述べる。

最後に、本支援環境を用いた実験結果を述べ、その有効性について検討する。

8.1　はじめに

大規模エキスパートシステム（ES）を構築する際に、我々はいくつかの知識ベース（KB）を構築し、それらを統合する必要がある。それらKBを個々に構築し、それらの間の一貫性を維持することは、多大な時間と開発労力を要する。その労力を軽減するために、「ある領域知識を記述するために必要な概念定義や概念階層」である領域オントロジーを導入することが考えられている。さらに領域オントロジーを用いたKB開発支援機構は、ESの規模が大きくなるにつれて、より必要性が増すと考えられる。しかし、領域における語彙が膨大であることや、領域固有性の表現方法が困難であること、推論時における知識の利用方法の考慮が必要であることなどから、独自に領域オントロジーを構築することは困難であり、ユーザに依存するところが大きい。そのため、ユーザへの負担の軽減を図る領域オントロジー構築支援システムが必要となっ

著　者：山口高平

標　題：DODDLE：計算機可読型辞書に基づく法律オントロジー構築支援環境

出版物：平成８年度科学研究費補助金重点領域研究

「法律エキスパートシステムの開発研究」研究成果報告書, pp.189-199

てきている。

一方、近年、CYC［Guha 94］やWordNet［Miller 90］、EDR電子化辞書［EDR 93］［Yokoi 93］のような大規模な知識を持ち、あらゆる領域の基本的な語彙が含む計算機可読型辞書（以下、MRDと省略）が利用できる環境が整えられつつある。しかし、これらMRDは対象とする領域が異なる点から領域オントロジーとして直接利用することは困難であり、何らかの変換が必要である。

以上のような背景から、我々は、MRDを利用した法律オントロジー（法律分野を対象とした領域オントロジー）構築支援環境LODEを構築してきた［Kurematu 95］。LODEは、ユーザが構築した初期法律オントロジーとMRDを照合し、位数、深さ、概念間の構造関係から比較評価し、有意な差異をユーザに示すことによって法律オントロジーの改訂をうながす。通常、開発コンテキストの差異から、MRDと領域オントロジーの間には共通する概念はほとんど存在しないと考えられる。しかし、法律分野においては、法令に出てくる語彙は法律固有の概念であるが、判例などに出てくる語彙の多くは、日常世界の語彙である。そのことから、法律オントロジーと様々な領域の基本的語彙を含むMRDは、対象としている領域が近く、共通する語彙が多く存在していると考えられる。その点から、法律オントロジーの構築にMRDを利用することは有用であり、膨大な語彙の記述に対する一つの解決方法になるととらえている。

さらに、MRDのみでは法律固有の概念に関する支援が困難であることから、LODEにケースオントロジーを利用する機構を追加した［Aoki 96］。ここで、ケースオントロジーとは、判例を記述するために必要な法律固有の概念とそれに関連する概念の概念定義と概念階層であり、LODEでは、判例群からボトムアップ的に構築したものを用いる。本支援環境では、ケースオントロジーについてもMRDと同様に、ユーザが構築した初期法律オントロジーと照合し、位数、深さ、概念間の構造関係から比較評価し、有意な差異をユーザに示すことによって法律オントロジーの改訂をうながす。

現在まで我々は、上記の支援環境を実装し、評価実験を通してその有用性を確認してきた。しかし、入力として与えられる初期法律オントロジーは、ユーザが構築しなければならないため、依然としてユーザへの負担が大きいのが現状である。この負担を軽減するために、現在我々は、入力として初期法律オントロジーではなく、法律語彙を与え、MRDとの照合結果に基づいた法律（ドメイン）オントロジー構築支援環境DODDLEを検討している。DODDLEは、入力として与えられた法律に関する語彙と、MRDを照合し、関連情報（概念階層）を抽出する。次に抽出された関連情報から不必要と考えられる情報を除去し、それを初期モデルとして獲得する。初期モデルは、こ

の時点では法律固有性が反映されていないため、法律オントロジーとしては利用できないため、DODDLEは、ユーザとインタラクションをとりながら、法律固有性の付加を試みる。本稿では、MRDからの関連情報の抽出方法を述べ、法律固有性の付加方法について述べる。さらに、現在実装を行っているDODDLEについての実験結果について述べ、最後に今後の方針を述べる。

8.2　領域オントロジー構築支援環境

8.2.1　基本設計

法律オントロジーを構築するのには、多くの開発労力と時間を必要とし、ユーザへの負担が大きい。DODDLEは、MRDを利用することにより、法律オントロジー構築に関するユーザの負担の軽減を試みる支援環境である。DODDLEの概観を図8.2-1に示す。

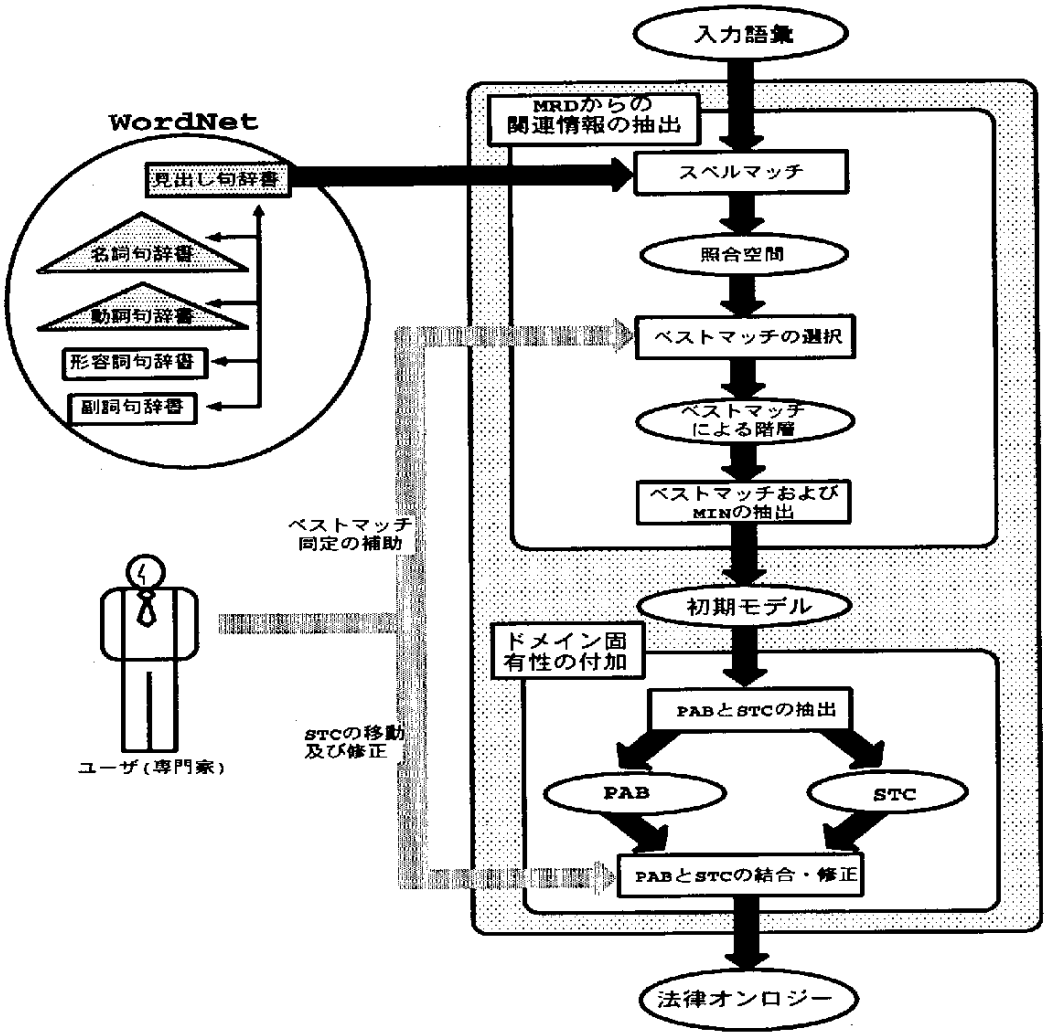


図8.2-1　DODDLE：法律オントロジー構築支援環境

DODDLEは、入力として与えられた法律に関する語彙と、MRDの照合を行い、関連

情報（概念階層）を抽出する。さらに不必要であると判断した情報を関連情報から削除し、初期モデルとして獲得する。獲得された初期モデルに対し、法律オントロジーとして再利用可能な部分を同定し、法律固有性の高い再利用不可能な部分を修正することを、ユーザとのインタラクションを通して行い、法律固有性の付加を図り、法律オントロジーの構築支援を試みる。なお本稿では、MRDとして評価の高いWordNet［Miller 90］を利用する。

8.2.2 入力語彙

構築目的である法律オントロジー中に含まれる語彙をユーザが決定し、それを入力語彙として、DODDLEに与える。基本的に与える語彙は単語である。また、ユーザの決定した語句よりも意味的に妥当なMRD中の語句があれば、その語句を用いる。また、ベストマッチの選択時に利用するために、ドメイン名である法律（law）をキーワードとして入力語彙に加える。

8.2.3 WordNet

本稿において利用するMRDは、プリンストン大学で開発されたWordNetである。WordNetは、名詞句辞書、動詞句辞書、形容詞句辞書、副詞句辞書、および、見出し句辞書から構成されており、約9万個の語彙を保持している。各々の辞書に記録されている見出し数および概念数を表8.2-1に示す。

表8.2-1　WordNetの辞書サイズ

辞書名	見出し(語彙)数	意味(概念)数
名詞句辞書	107424	60557
動詞句辞書	25761	11363
形容詞句辞書	28749	16328
副詞句辞書	6201	3243
見出し句辞書	119217	91519

見出し句辞書は、見出し句、意味情報としての概念ID、辞書編集情報、品詞情報などから構成されている。概念IDは、残りの4種類の辞書と共通のIDであり、これによって、見出し句辞書と各辞書の間にそれぞれリンクがはられている。

名詞句辞書および動詞句辞書は、概念ID、辞書編集情報、対応する見出し句リストから構成されており、概念IDは階層構造を形成している。また、一部の概念IDには、反対概念の概念ID、part ofやmember of、substance ofの概念の概念ID、グロッサリーなどが与えられている。

形容詞句辞書、副詞句辞書も同様に概念ID、辞書編集情報、対応する見出し句リストから構成されているが、階層構造はもたない。

現在はこれらの辞書のうち、見出し句辞書、名詞句辞書、動詞句辞書のみを利用する。

8.2.4 法律オントロジー構築支援プロセス

DODDLEは、MRDから抽出した関連情報から不要な情報を除去することで初期モデルを獲得し、それに法律固有性を付加することで、法律オントロジーの構築支援を行う。以下、それぞれの処理について説明する。

(1) MRDからの関連情報の抽出

最初に入力語彙の各語句に対し、WordNetの見出し句とスペルマッチを行う。スペルマッチに失敗した語句については、保存しておき、最後にユーザが法律オントロジーに追加する。

スペルマッチによって獲得された語句に対し、名詞句および動詞句の概念IDに対して、それぞれの辞書より、各語句の概念IDからルートまでの概念階層を抽出する。

次に、語句によっては複数の概念ID（意味）を持つ場合があるため、それらの語句に対して、以下の方法でベストマッチを半自動的に選択する。

① 対象とする語句の各概念IDに対し、以下の方法で関係値を計算する。

1) 対象とする概念IDと上位下位関係以外（たとえばpart of）のリンクが張られている概念IDのうち、入力語彙の概念IDになっているものの総数。

2) 対象とする概念IDの全ての下位概念の概念IDのうち、入力語彙の概念IDになっているものの総数。

3) 対象とする概念IDとその辞書のルートまでの各パスごとに出現する概念IDのうち、入力語彙の概念IDになっているものの総数の最大値。

以上、3つの総和を関係値とする。

② 各概念IDに対し、関係値が最大になる概念IDをベストマッチとしてユーザに提示する。ユーザがその概念IDをベストマッチとして認めれば、それをベストマッチとし、認めなければ、次に関係値の高い概念IDをユーザに提示し、ベストマッチを決定する。
また、関係値が同じ概念IDが複数ある場合は、それらをすべて提示し、ユーザがその中から選択する。

スペルマッチで求まった照合結果から、以上の操作で求まったベストマッチの概念IDに関する情報のみを抽出し、これをベストマッチの階層とする。
最後にベストマッチの階層から、必要な中間節点のみを抽出する。本稿では、必要な中間節点をMIN（Minimal Internal Nodes）と呼び、以下のように定義する。

MINの定義： ベストマッチではない中間ノードのうち、ベストマッチかMINである下位ノードを２つ以上もち、それらのうち、子孫関係が成立していない下位ノードの組が存在するならば、その中間ノードもMINである。

図８.２-２において、白抜きのノードがベストマッチにあたり、楕円で囲まれたノードがMINである。ルートを除く斜体で書かれたそれ以外の中間ノードは除去される。
ベストマッチの階層からルートノード、MIN、ベストマッチのみを抽出した階層を、MRDから抽出された法律に関連する情報とし、初期モデルと呼ぶ。
初期モデルは法律固有性を含んでいないため、法律オントロジーとしては利用できない。そのため、初期モデルに法律固有性を付加することを試みる。

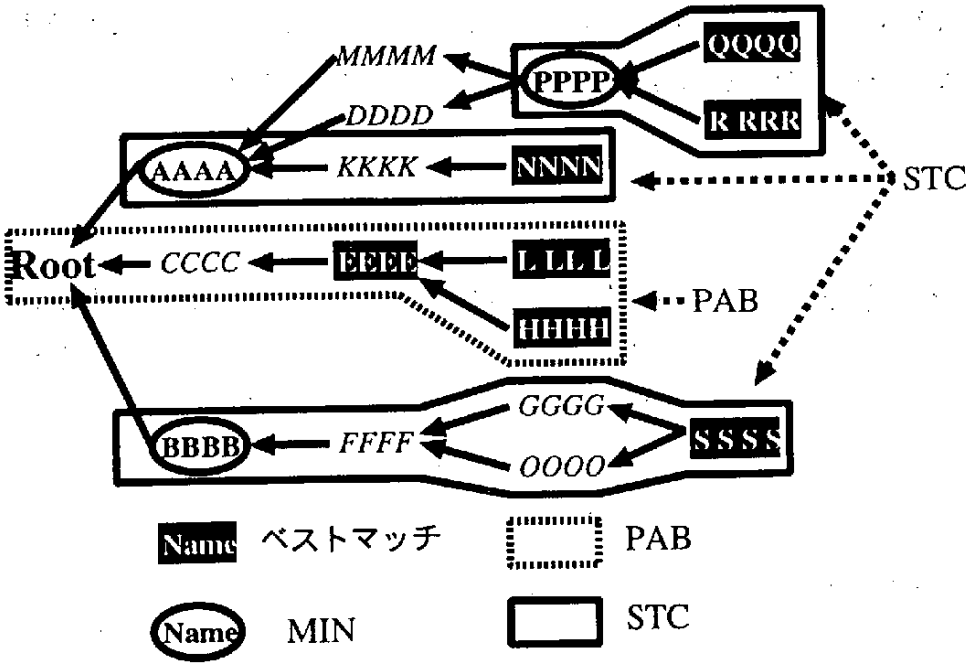


図8.2-2　MIN，PAB，STCの例

(2) 法律固有性の付加

法律オントロジーとMRDを比較したとき、両者の間には異なる箇所と共通する箇所とが存在する。前者は、法律領域に固有な箇所であり、concept driftが生じている部分である。一方、後者は、法律領域に固有ではない、いわば汎用的な箇所であり、MRDから再利用可能な箇所である。そのことから、汎用的なMRDから再利用可能な箇所に、法律領域固有の箇所を付加すること、すなわち、法律固有性を付加することにより、法律オントロジーの構築が可能であると考えられる。またMRDから法律オントロジーの一部を再利用するため、その箇所の構築に必要な開発労力を軽減でき、ユーザへの負担の軽減が図れると考えられる。

以上の考えに基づいて、DODDLEでは最初に、獲得された初期モデルから再利用可能な汎用的な箇所と、再利用不可能であり法律領域に固有な情報を付加すべき箇所を、以下の定義に基づいて抽出する。ここで、前者をPAB（Paths with All Bestmatches）、後者をSTC（Subtree To be Changed）と呼ぶ。

PABの定義：ルートからベストマッチが複数個連続しているパス。

STCの定義：MINをサブルートとし、その下位ノードがすべてベストマッチで構成されている部分木

図８.２-２において、破線で囲まれた部分がPABであり、実線で囲まれた部分がSTCである。

次に、PAB（再利用可能な部分）を固定し、STC（再利用不可能な法律に固有な情報を付加すべき部分）を移動することにより、初期モデルに法律固有性を付加し、法律オントロジーの構築支援を試みる。ここで、STCの移動先についてはユーザが決定し、移動する必要がないと判断した場合は移動しない。さらに、STCの移動のみでは対処できない箇所についてはノード単位でユーザが修正を行い、法律オントロジーを構築する。また、スペルマッチに失敗した語彙についても同様にユーザが初期モデルに付加する。

以上のように、DODDLEは、不完全ではあるがMRDに基づいて階層構造を自動的に構築し、法律固有の箇所について複数の概念をまとめて処理することで法律オントロジーの構築支援を行う。この点から、語彙から直接法律オントロジーを構築するより、ユーザの開発労力の軽減を図れると考えられる。

8.3 法律オントロジー構築支援実験

8.3.1 実験の概要

本項では、DODDLEによる法律オントロジーの構築実験の結果について述べる。ユーザは、国際売買法（以下、CISGと略記）第２部［曽野93］に含まれる法律用語群から46語句を抽出し、これを入力語彙として、DODDLEに与える。入力語彙を表８.３-１に示す。DODDLEを用いることにより、この語彙から法律オントロジーを構築する際にユーザの負担が軽減できれば、すなわち、MRDから構築した初期モデルに多少の変更を加えることでユーザが満足できる法律オントロジーが構築できれば、本システムは有効であると考えられる。

表８.３-１　入力語彙

承諾	acceptance	修正	modification
行為	act	申込	offer
付加	addition	申込者	offerer
住所	address	被申込者	offeree
同意	assent	当事者	party
状況	circumstance	支払	payment
通信	communication system	人	person
行為	conduct	営業所	place of business
契約	contract	価格	price
反対申込	counteroffer	申入	proposal
日	day	質	quality
遅延	delay	量	quantity
配達	delivery	拒絶	rejection
不一致	discrepancy	回答	reply
発送	dispatch	居住地	residence
効果	effect	取消	revocation
書簡	envelope	沈黙	silence
物品	goods	陳述	speech act
休日	holiday	電話	telephone
表示	indication	テレックス	telex
意志	intention	時間	time
誘引	invitation	伝達	transmission
手紙	letter	撤回	withdrawal

8.3.2 実験結果

(1) 関連情報の抽出

入力語彙46語句のうち、offererとoffereeの２つの語句はWordNetとの照合に失敗した。それらの語句は保存しておき、残りの44語句の全ての概念IDに対し、WordNetから図8.3-1に示すような照合空間を獲得する。

さらに、先述の方法にそって、ユーザとインタラクションをとりながら、各語句の概念IDのベストマッチを同定し、図8.3-2に示すようなベストマッチの階層を抽出する。図において長方形に囲まれたノードがベストマッチである。獲得されたベストマッチの階層から不要な中間ノードを除去し、図8.3-3に示す初期モデル（ルートノード、MIN、ベストマッチからなる階層）を抽出する。図中において各矢印上の数値は削除された中間ノード数の最大値であり、長方形に囲まれたノードが選択されたMINであり、本実験では15個のMINが抽出された。

次にこの初期モデルに対して法律固有性の付加を行う。

(2) 法律固有性の付加

初期モデルから前述の定義に基づいて、PABとSTCを抽出する。図8.3-4において長方形に囲まれたノード及びそれらを結ぶパスから構成される部分木がPABであり、二重の長方形に囲まれたノードが各STCのサブルートになる。今回の実験では、13個のSTCが抽出された。

次に、ユーザはSTCを移動することによって、法律オントロジーを構築する。今回の実験では、７個のSTCを移動し、図8.3-5に示される初期法律オントロジーを得る。図中の長方形に囲まれたノードが移動したSTCのサブルートである。さらに、STCの移動のみでは対処できない箇所を修正するとともに、スペルマッチで失敗した２つの語句を追加する。本実験では、27箇所が修正され、図8.3-6に示すような法律オントロジーが構築された。ここで長方形で囲まれたノードがSTCの移動のみでは対処できず修正を加えたノードである。それらのうち、修正前の親ノードと修正後の親ノードが５親等以上離れているノードは11個で、修正した全ノードの41%にあたる。

DODDLEは、現在、PerlとTcl-Tkを用いて、エンジニアリングワークステーション、S-4/model51上で実装されている。表8.3-2に現時点でのDODDLEの仕様を、図8.3-7に実行画面を示す。

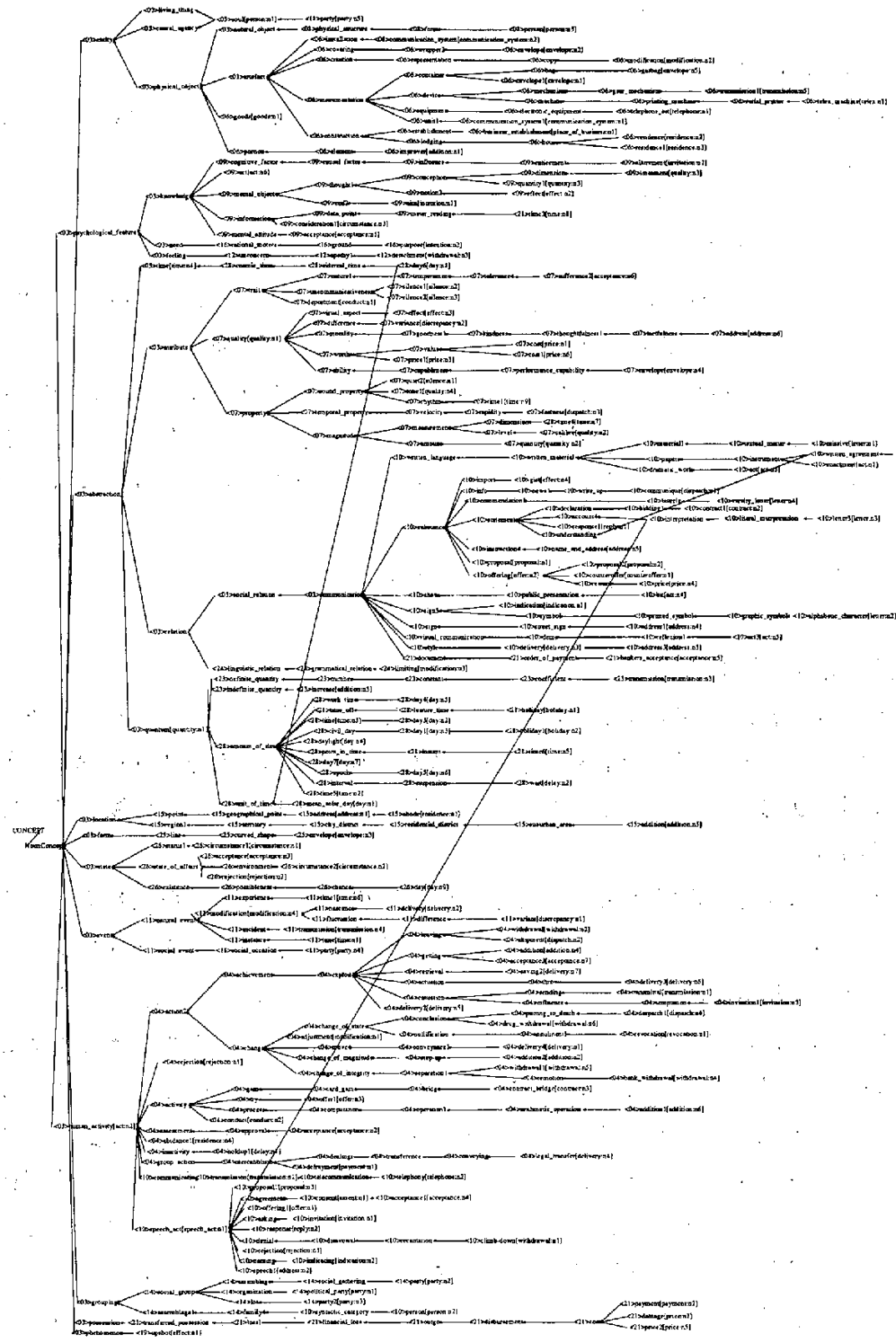


図８．３-１　入力語彙とWordNetの照合結果

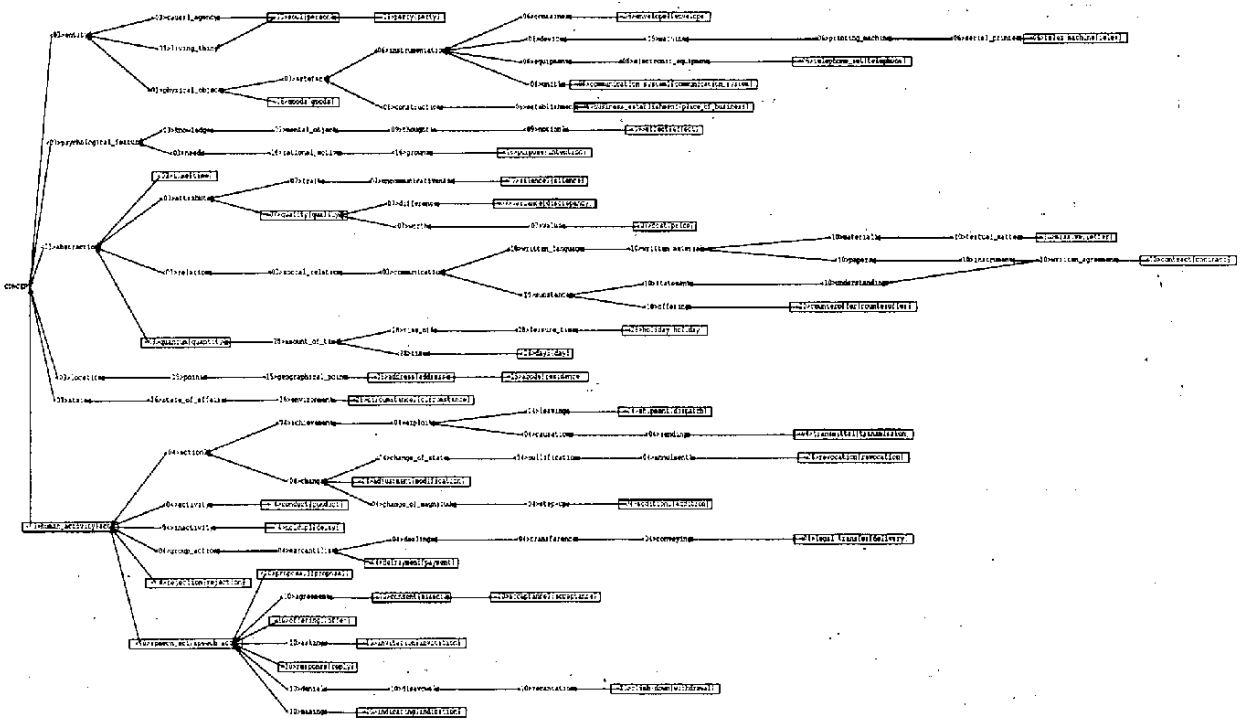


図8.3-2　獲得されたベストマッチの階層

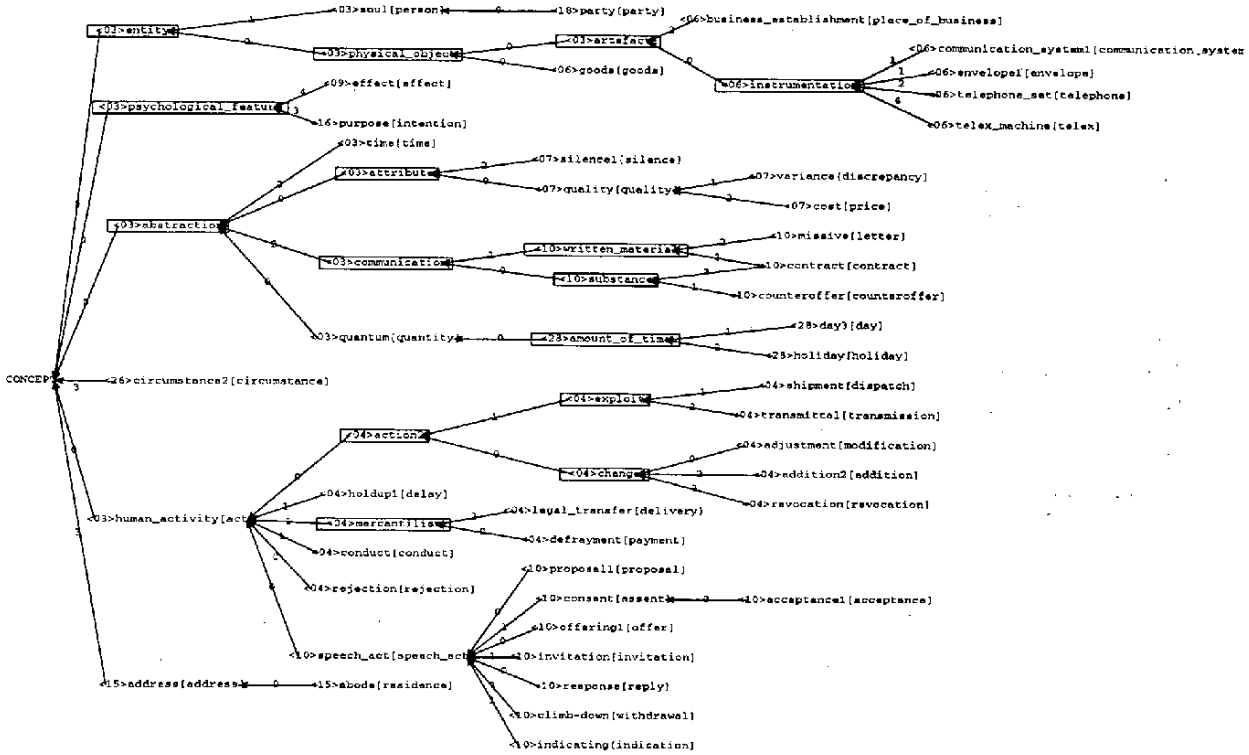


図8.3-3　獲得された初期モデル

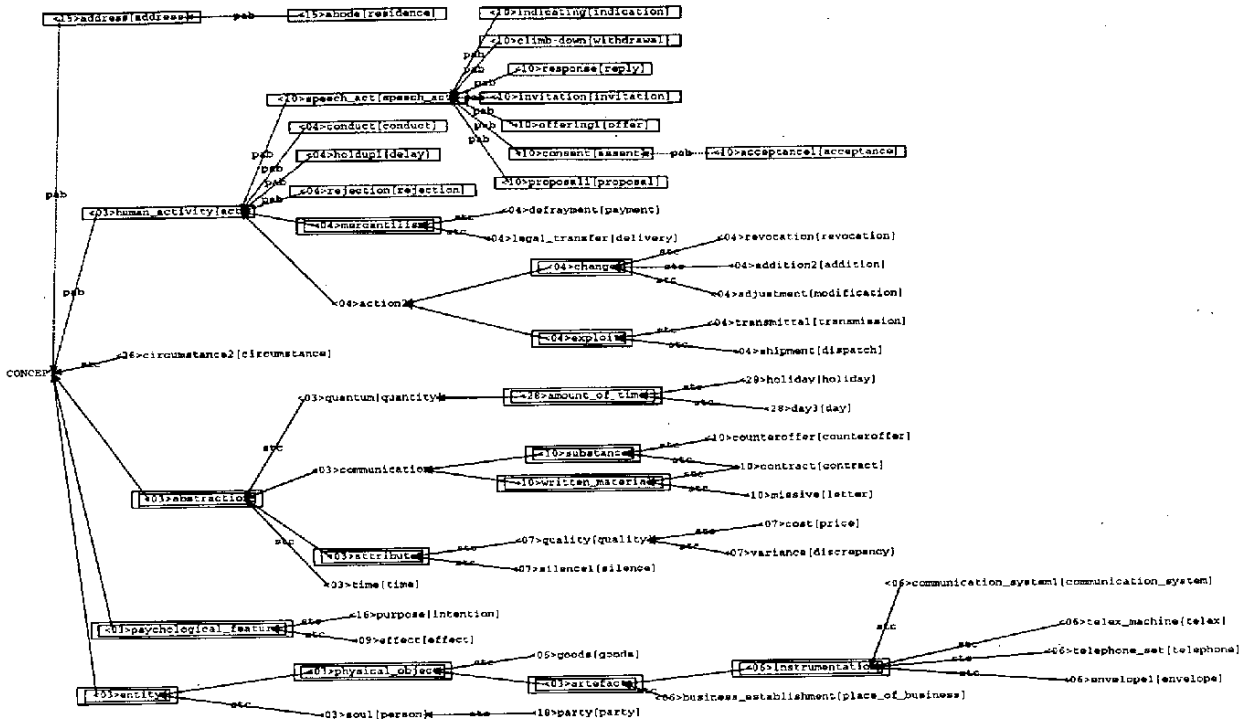


図８．３-４　PABとSTCに区別した初期モデル

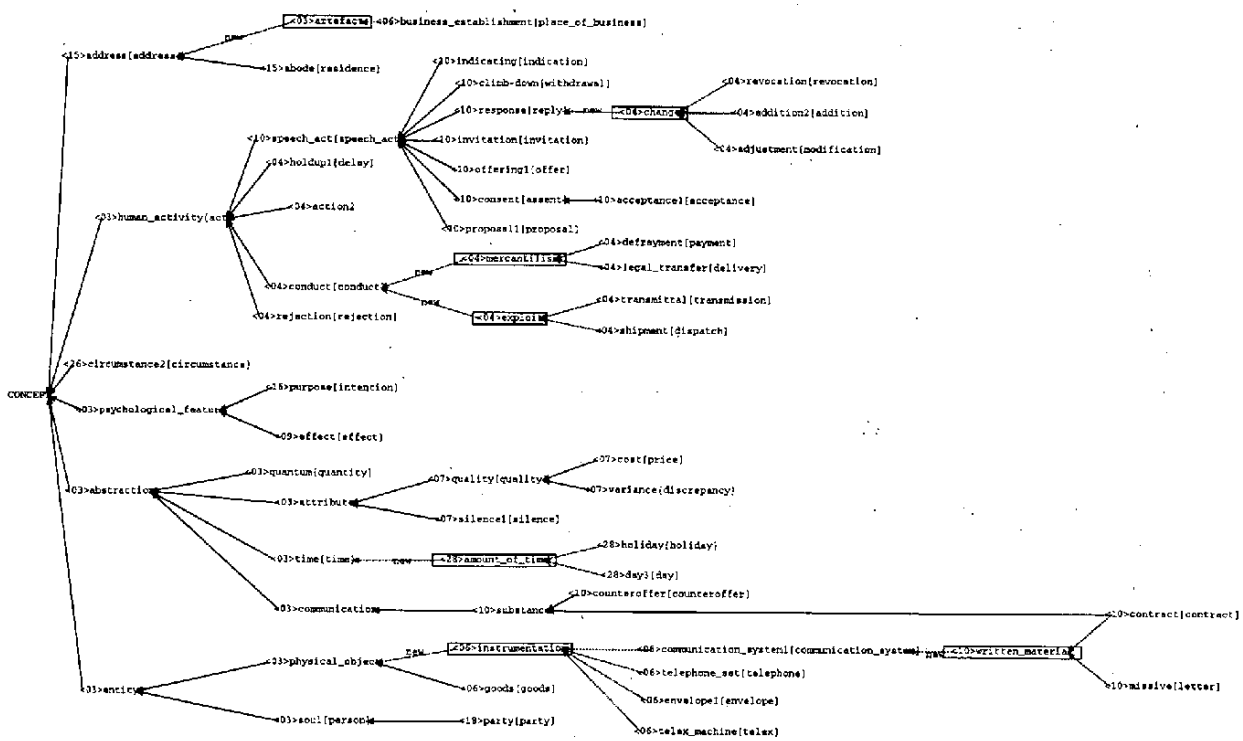


図８．３-５　STCの移動のみで構築された法律オントロジー

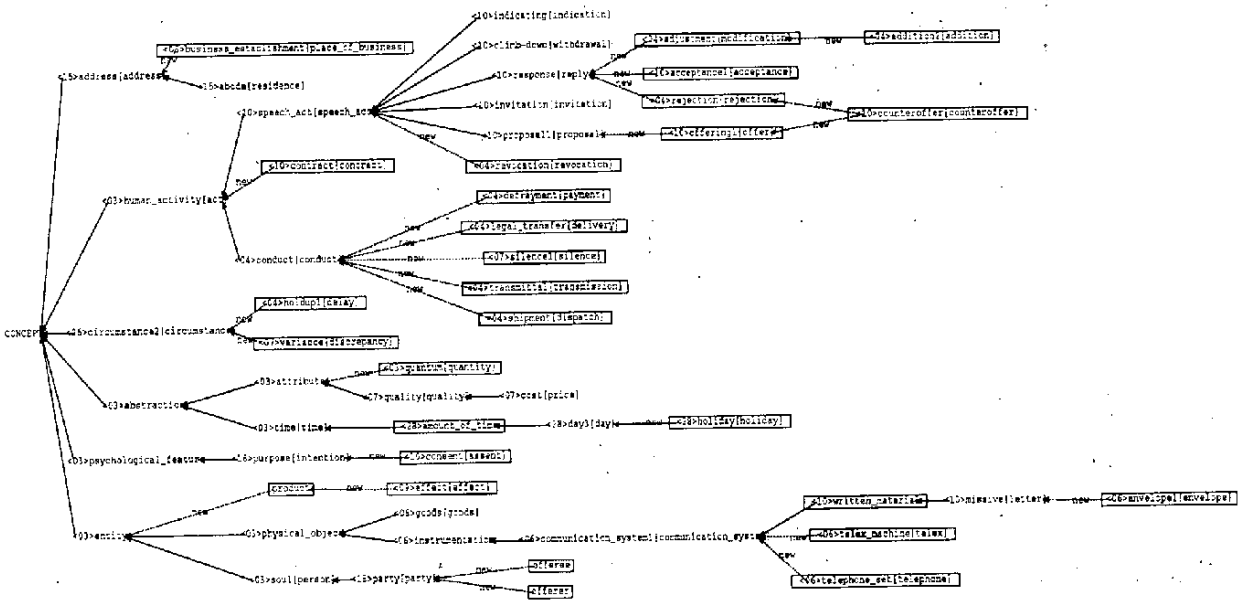


図8.3-6　構築された法律オントロジー

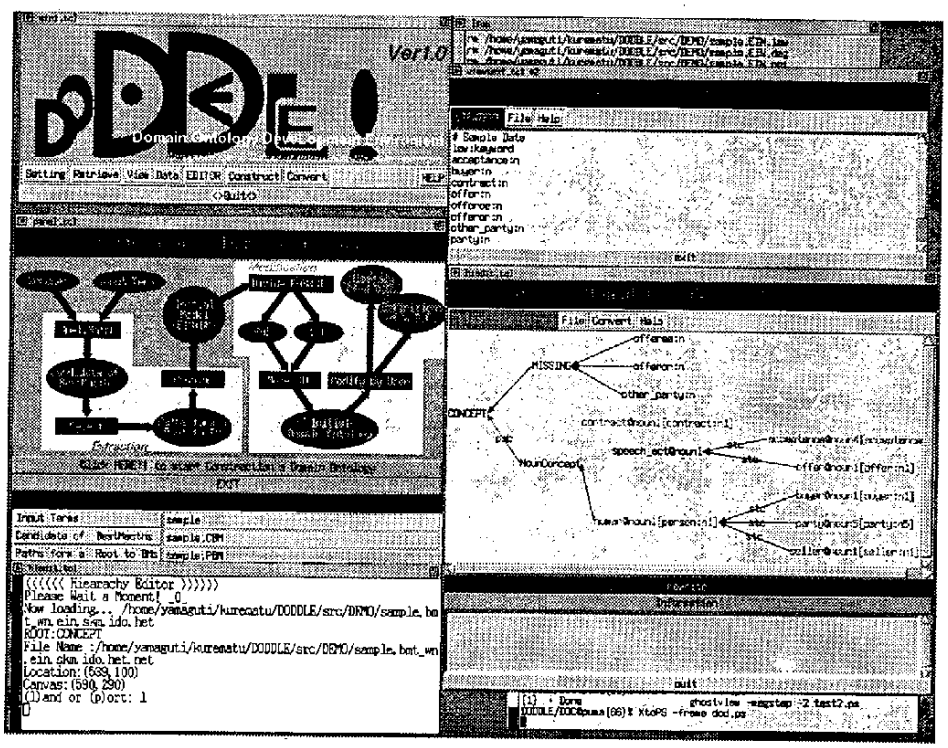


図8.3-7　DODDLEの実行画面

表8.3-2　DODDLEの仕様

知識ベース＆モジュール	プログラミング言語	サイズ(KB)
入力語彙	テキスト	.5
WordNet(利用部分のみ)	テキスト	16359.6
初期モデル構築モジュール	Perl&Tcl-Tk	50.8
法律固有性付加モジュール	Perl&Tcl-Tk	31.7
表示モジュール	Tcl-Tk	48.5

(3)　評価・考察

今回の実験の結果、MRDから獲得した初期モデルを変更することにより、妥当な内容の法律オントロジーを構築することに成功した。初期モデルから法律オントロジーを構築するために変更した箇所のうち、STCの変更だけでは対処できなかった部分は、27箇所であり、それらについてはノードレベルで変更を行っている。それらの変更のうち、MINの除去によるものは6個で22%をしめ、修正前の親ノードと修正後の親ノードの距離が4親等以内のノードは、10(16)個で37(59)%である（()内はMINの除去も含めた数）。さらに、修正前後の親ノードの距離が、5親等以上の変更があったノードは11個で41%である。全体のノードのうち、修正前後の親ノードの距離が5親等以上の変更があった箇所は全体の20%であり、その値から、STCの移動時点で若干の修正はあるが、妥当な法律オントロジーが構築できているといえる。以上のことからはDODDLEは法律オントロジー構築支援に対して有効であると考えられる。

本実験において、MINの除去以外でノードレベルで修正を行った箇所は、PABとSTCという識別方法では同定できなかった部分であり、法律固有性が非常に高い部分であると言える。それらの法律固有性の高い概念は、MRDの開発コンテキストの中には一般的に含まれていないと考えられることから、MRDのみを利用する現在の方法では、その同定は非常に困難であると考えられる。また、STCの移動先と移動距離の決定については、ユーザに依存しており、この点に関してはユーザへの負担が依然として大きい。これらの解決手段として、現在、我々は、判例群やケースオントロジー、法ルール、法定理群などより法律固有性の高い情報源を利用することを検討している。

また、現在DODDLEは概念階層の構築のみを支援し、概念記述の構築についての支援を行っていない。オントロジーは前述の定義通り、概念階層と概念記述より構成さ

れているため、概念記述の構築の支援も行う必要がある。概念記述の構築支援に対しては、EDRのような概念記述を持つMRDや判例群、法ルールなど、他の情報源の利用を検討している。

8.4 おわりに

本稿では、計算機可読型辞書（MRD）に基づく法律オントロジー構築支援環境、DODDLEについて述べた。DODDLEは、入力として与えられた法律に関する語彙と、MRDを照合し、関連情報（概念階層）を抽出し、不必要と考えられる情報を除去し、初期モデルを構築する。次に、ユーザとのインタラクションを通して、初期モデルに法律固有性を付加することにより、法律オントロジーの構築を支援する。本稿では、CISG第2部の語彙を対象に実験を行い、この環境の有用性を示した。今後の研究としては、法律固有性の付加に関するユーザの負担の軽減を図るため、判例など、より法律固有性の高い情報源の利用を検討すること、また、現在行っていない概念記述の構築支援に対して、EDRのような概念記述を持つMRDや判例など現在利用していない他の情報源の利用を検討することがあげられる。さらに、法律専門家に実際に利用してもらい、そこで得られるコメントを参考に、DODDLEを向上させることである。

参考文献

[Aoki 96] C. Aoki, M.Kurematu, and T.Yamaguchi: "*LODE : A Legal Ontology Development Enviroment*", Proc.of the Pacific Knowledge Acquisition Workshop 96, pp.190-209 (1996)

[EDR 93] EDR電子化辞書使用説明書, 日本電子化辞書研究所 (1993)

[Guha 94] R.V. Guha and D.B. Lenat : "*Enabling Agents to Work Together.* ", Communications of the ACM, Vol.37, No.7, pp.127-142 (1994)

[Kurematu 95] M.Kurematu, M.Tada and T.Yamaguchi: "*A Legal Ontology Refinement Environment Using a General Ontology*", Proc.of Workshop on Basic Ontology Issues in Knowledge Sharing, IJCAI'95 (1995)

[Miller 90] Miller,G. 1990. WordNet : "*An on-line lexical database*", International Journal of Lexicographer, Vol.3, No.4 (1990)

[曽野 93]曽野和明, 山手正史 : "国際売買法", 青林書院, (1993)

[Yokoi 93] T.Yokoi: "*Very Large-Scale Knowledge Bases Embodying Intelligence Space*", Proc. International Conference on Building and Sharing of Very Large-Scale Knowledge Bases '93, pp.11-20 (1993)

第3部　海　外　動　向

1.　米国動向調査

第３部　海外動向

１．米国動向調査（報告）

（１）調査要領

① 調査目的：オントロジー工学に関連する技術について、米国の研究動向を言語学の観点から調査する。

② 調査期間：平成９年２月１６日（日）～２月２３日（日）

③ 調査者　：外池俊幸（名古屋大学 言語文化部　助教授）
（平成８年度　オントロジー工学調査委員会　委員）

④ 訪問先　：Stanford 大学 Center for the Study of Language and Information
Xerox 研究所（Palo Alto）
SRI International

（２）調査概要

２月１６日（日）午後、Stanford大学に近いPalo Altoに到着、帰国まで同じところに滞在した。翌日、１７日の月曜日はアメリカでは祝日で、到着してから１７日までは、訪ねたいと考えていた人たちに電話などで連絡を取ることに費やした。

２月１８日（火）Stanford大学のCSLI（Center for the Study of Language and Information）に在外研究で滞在していた、友人でもある国立国語研究所の加藤安彦氏とCSLIを訪ねる。

事前に約束を取りつけていたわけではなかったのだが、幸い会って話しが聞きたいと考えていた旧知の Stanley Peters 氏とサバティカルで毎日は来ていないというIvan Sag氏の二人に会うことができた。

Stanley Peters氏は言語学科の学科長で非常に忙しく、約束もしないまま会ってもらえて話しができたのは運が良かったと後で他の人にも言われた。加藤安彦氏と３人で、主に私が共同して研究を進めたいと考えていた日本語と英語の生成的辞書の構築に関して意見交換をした。興味深いプロジェクトだと評価してくれた。

その後、CSLIの建物を中をまわったら、Ivan Sag氏が同じく訪ねたいと考えていた研究員のDan Flickinger氏と話しをしていた。そこで、やはり旧知のIvan Sag氏と同じく

共同研究に関して話しをした。現在CSLIで進めている言語学関係のプロジェクトに関する話を聞いた。

①Verbmobile：こちらは継続して進められている、英語とドイツ語の翻訳を目指したプロジェクトである。当初は携帯できるくらいの大きさの端末上で英語とドイツ語の翻訳ができることを目指したものであった。
②インタフェースとしての自然言語：現在コンピュータをはじめ、様々な機器を使いたい人は数多いが、例えばキーボードを思うように使えない人も多い。現在提供されているコンピュータをそのままでは利用しにくい人たちを支援するために、自然言語をどう活かせるかの研究が進行中であった。これは大きな課題であり、また重要な方向で、自然言語に興味を持っている人が大きな貢献をできる領域だと考えられる。

２月１９日（水）Palo Alto にあるZerox の研究所へ加藤安彦氏と行き、Mary Dalrymple氏の研究室を訪ねた。彼女はStanfordで言語学を専攻してPhDを終えた人で、CSLIとも関係を保ちながら主にLFG（Lexical-Functional Grammar）の枠組みで、言語に関する研究を続けている。

２月２０日（木）午前中に、Stanfordで日本語を教えている松本善子氏の研究室を加藤安彦氏と訪ねた。私も一昨年来た時に話しを聞いてもらっているので面識があり、加藤さんも既に会ったことがあったので、打ち解けた感じで最近の日本語研究に関し意見交換を行った。

続いて、昼にオントロジーに関する研究会があるから来るといいとメールをもらったKnowledge System Lab.（ビル・ゲイツ・ビルディングと呼ばれている新しい建物にある）のBob Engelmore氏の研究室を訪ねる。早速、昼食を取りながら、１０人くらいの研究者、院生、学生が集る中で、院生らしい学生が話しをしたのを聞いた。知識表現に関して簡単な枠組みを仮定して学生が話し、かなり根本的な問題に関して熱心に議論していた。特に印象的だったのは、Bob Engelmore 氏で、アメリカ人らしい研究者の鑑だという印象を受けた。気さくで、親切で、まじめで、と三拍子揃った人だと感じた。

その後、研究員でお昼の研究会にも参加していたIwasaki Yumiさんの研究室を訪ね、彼女の研究について話しを聞いた。物を設計するために書く絵が、設計を進める上にどういう影響を与えるかを、CSLIにいる心理学が専門の人などと共同で研究を進める予定だということであった。私にも興味が持てる内容で１時間くらいがあっという間

に過ぎてしまった。

その後、Palo Alto にあるSRI Internationalの研究所を加藤安彦氏と訪ねた。研究員で旧知の亀山恵氏を訪ね、話しが聞きたいと考えていたJerry Hobbs 氏に会えた。Jerry Hobbs氏はオントロジーにも興味があり、関連する論文を２件の提供を受けた。後で亀山氏の研究室にも立ち寄り、日本語に関しての研究について話しをした。CSLIに戻り、その後、CSLIの研究員で旧知のDan Flickinger氏と会い、私が考えている共同研究案に関して時間を掛けて話しを聞いてもらった。

２月２１日（金）CSLI に行き、Ivan Sagと会って、再度共同研究の話しをする。面識が無かったCSLIの研究員であるAnn Copestake氏に紹介してもらい、その後、かなり時間を掛けて共同研究案に関する意見交換を行った。

(3) まとめ

１週間という短期間ではあったが、会いたいと考えていた人のほとんどに会え、共同研究の話しを進められたことは大きな収穫であった。成果は次の三つにまとめることができる。

① 日本語・英語などの自然言語を使い、様々な領域の知識を書き出すことの意義を確認できたこと。
② 日本語と英語を対象とした生成的辞書の構築を目指した共同研究を進める準備ができたこと。
③ コンピュータを始めとした様々な機器、さらには公共施設でのサービスに関して、それを多くの人が利用できることを支援するためのインタフェースとしての自然言語という観点が重要であるということを、具体的な研究が進められている話しを聞いて確認できたこと。

2. 概念スキーマモデル機能(CSMF) の標準化

2. 概念スキーマモデル機能（CSMF）の標準化

2.1 はじめに

本稿では、1994年にスタートしたISO（国際標準化機構）のCSMF（Conceptual Schema Modelling Facility：概念スキーマモデル機能）標準化プロジェクトを中心に、その背景、考え方、現状を解説する。

2.2 概念スキーマモデル機能とは

「概念スキーマ」は、データベースの対象世界の論理的な構造を表現するものとして、ANSI／SPARCによって1975年に提案された概念［ANSI/X3/SPARC 75］であり、今日では、データベース分野に限らず、情報システム分野全般で使われている用語である。「概念スキーマモデル機能」は、この概念スキーマを記述するための１組の概念の集まりを指している。

モデル機能（Modelling Facility）は、概念を記述する概念であるため、言葉で説明しようとすると混乱をきたしやすい。そこで、RMDM（データ管理参照モデル）［ISO/IEC 10032 95］のレベル対を用いて、例示的にその位置付けを示したものが、図２.２-１である。

現実世界の個（この例では、山田太郎と呼ばれる個人）と１対１に対応する記述が、レベル１のインスタンスレベルであり、それらを類型化したもの（この例では、従業員）が、レベル２のスキーマレベルである。このスキーマレベルにも似たようなものがたくさん登場してくる。会社をモデル化する場合であれば、従業員、商品、注文、倉庫などがでてくる。これらを類型化したものがレベル３のモデル機能レベルである。P. Chenの実体関連モデル［Chen 76］では、「実体」や「関連」が、このモデル機能レベルの概念であり、SQLでは、「表」や「列」がこのレベルの概念である。

従来、データベース分野では、このレベル３の概念の１組の集まりを単に「データ

著　者：佐藤英人

標　題：概念スキーマモデル機能CSMFの標準化

出版物：情報処理学会学会誌, Vol.37, No.7, pp.646-650, July 1996

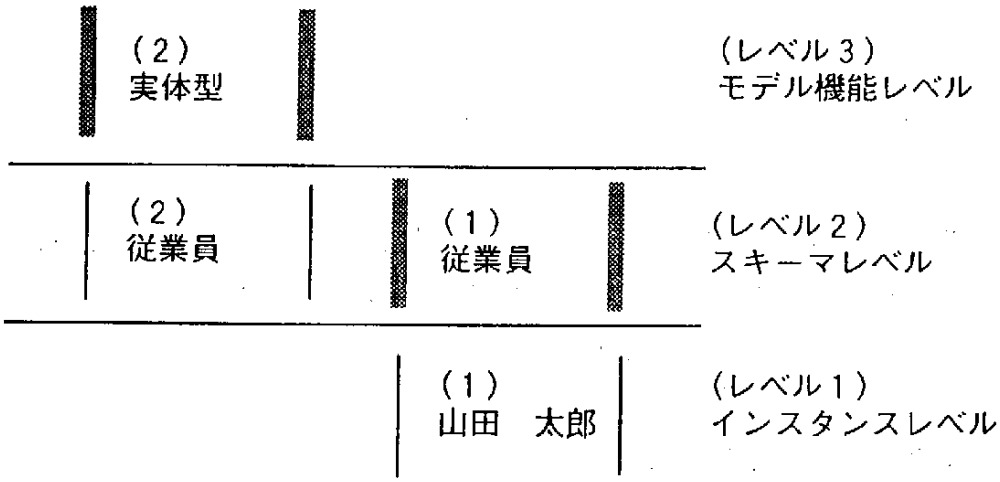


図2.2-1　レベル対とモデル機能

モデルと呼ぶことが多かった。しかし、レベル2のスキーマ概念の集まりも、「○○業務のデータモデル」と称することが、現実には多い。そこで、標準化に当たって、混乱を避けるために、レベル3の概念の集まりを「モデル機能」と呼ぶことにしたのである。

なお、「概念スキーマモデル機能」は、このモデル機能の中でも、SQLの表や列といった情報の表現を対象とするのではなく、実体や関連といった現実世界の対象に関する概念を扱うものである。従来、データベース分野で「意味データモデル」と呼ばれてきたものとほぼ同じものであると考えて差し支えない。また、オブジェクト指向モデリングで、概念モデルと呼んでいるものは、レベル2の概念の集まりであり、それらを記述するために用いられるオブジェクト、性質、状態、イベントといった概念は、レベル3の概念である。

2.3　概念スキーマモデル機能標準化の目的と経緯

先に述べたANSI／SPARCによる3スキーマの提案以降、そこで規定された概念スキーマを記述するモデル機能として、多くの意味データモデルが提案された。P. Chenの実体関連モデルやG. M. NijssenのENALIM（今日のNIAM）［Nijssen 77］などは、その代表的なものである。

1980年代のはじめにISOは、これらの概念スキーマのための概念・用語を標準化すべくプロジェクトを起こした。このプロジェクトは、1987年にTR9007と呼ばれるテクニカルレポート［ISO/TR9007 87］を公刊して解散したため、TR9007プロジェクトと呼ばれている。

このレポートは、概念スキーマに係る問題を深く考察したもので、学術的価値の高いものであるが、結論的には3種の概念スキーマモデル機能（実体関連モデル、2項関係モデル、述語論理モデル）を併記し、拘束力のないテクニカルレポートに止まったため、標準化という意味では実世界に大きな影響を与えることはなかった。

その後、1990年代に入って、別の意味で概念スキーマモデル機能の標準化が求められるようになってきた。それは以下のような情報システム分野の新しい動向に対応するものである。

(1)　データ交換上の必要性

商取引の電子化であるEDI（Electronic Data Interchange）の普及・拡大に伴い、交換するデータの意味を記述する情報の標準化が必要となってきた。EDIは単にデータの送受信に関するプロトコルだけでなく、取引の形態や取り扱い商品の性質に応じたビジネスプロトコルを必要とする。このため、交換されるデータの意味や交換後の取り扱いを規定できる概念スキーマモデル機能が必要になる。

(2)　CASEツール統合上の必要性

ソフトウェア開発を自動化するCASE（Computer Aided Software Engineering）ツールの普及に伴い、ツール間の連携をとり、定義情報を共有するための標準化が必要になってきた。CASEツールは、概念レベルの上流ツールから実装レベルの下流ツールに至るまでの多くのツール群で構成される。各レベルのツールは、それぞれより上位のツールの出力を入力として受け取る。ツール相互間の互換性を高め、円滑な連携を可能とするためには　定義情報の意味を規定できる標準的な概念スキーマモデル機能が必要

である。

(3) 概念モデリングの普及

業務ソフトウェア開発におけるデータベース利用の一般化、CASEツールやリポジトリの登場、オブジェクト指向概念の普及に伴い、ソフトウェア開発における概念レベルのモデル化は不可欠のものとなってきた。この結果、概念モデリングに関する種々の方法論が登場し、それぞれが独自の概念スキーマモデル機能を提案するに至っている。

(4) 国際標準間の重複

(1)～(3)のような背景から、［穂鷹96］で詳しく述べられているように、ISOをはじめとする標準化組織において、それぞれの分野で概念レベルを含むモデル機能の標準化が進行している。しかしそれらは、多くの共通点をもつにもかかわらず、それぞれの分野の対象に適した固有の概念・用語を用いており、標準そのものの統合や融合あるいは共存が困難になっている。

以上の状況の中で、ISOでは、1989年11月のイタリア・フローレンス会議を皮切りに、概念スキーマモデル機能標準化の新プロジェクトの検討が始まり、1993年6月の横浜会議で新プロジェクトの提案が行われた。投票の結果、5ケ国（日、米、カナダ、ベルギー、フィンランド）の賛成が得られ、1994年7月の英国サウザンプトン会議でSC21 WG3の下の活動として、CSMFプロジェクトが発足した。

2.4　CSMFプロジェクト

2.4.1　標準化作業の現状

CSMFプロジェクトは、発足と同時に作業を開始し、現在、1996年5月に米国カンザスシティで開催されるラポータ会議でワーキングドラフトを完成させるべく、作業を進めているところである。

ラポータはカナダのBaba Piprani、エディタは米国のJohn K. Sharpと日本の穂鷹良介が勤めている。常時参加している国は、カナダ、米国、日本、イギリス、オランダ、フランスの6カ国であり、他にスウェーデン、オーストラリア、韓国、ブラジルが1～2回参加している。

作業は、米国が中心となってワーキングドラフト案を用意し、主として、日本、イギリス、オランダがこれに対し対案や修正案を提案するという形で、進められている。CSMFは対象が抽象的なものであるだけに、各国代表間の相互理解に時間がかかったが、ようやく大筋について合意が得られつつあるところである。

2.4.2　標準CSMFの位置付け

2.3節で述べたように、CSMFの応用領域と想定されているEDIやCASEなどの分野では、既に多くの国際標準が制定されている。標準CSMFはこれらの既存の標準のモデル機能に取って代わることを意図するものではなく、これら相互間の関係を記述する手段を規定することにより、異なるモデル機能間のスキーマ情報の交換、更には、インスタンスデータの交換を容易にすることを目的としている。

この関係を図示したものが図2.4-1のアーキテクチャである。この図は現在審議中のワーキングドラフト［ISO/SC21 WG3 N190996］に収録されているものであるが、まだ完全な合意が得られているものではない。今後字句の変更などが施される可能性は高いが、大筋は変わらないものと思われる。

この図で、最下層のApplication Modelと書かれたレベルには、「○○会社のスキーマ」「△△業務のスキーマ」といった個々の応用スキーマが位置付けられる。先の図2.2-1でいえば、レベル2のスキーマレベルに相当している。その上のMeta Modelと書かれたレベルは、応用スキーマの定義言語を規定するレベルで、図2.2-1のレベル3のモデル機能レベルに対応している。最上位のMeta-meta-Modelレベルにある

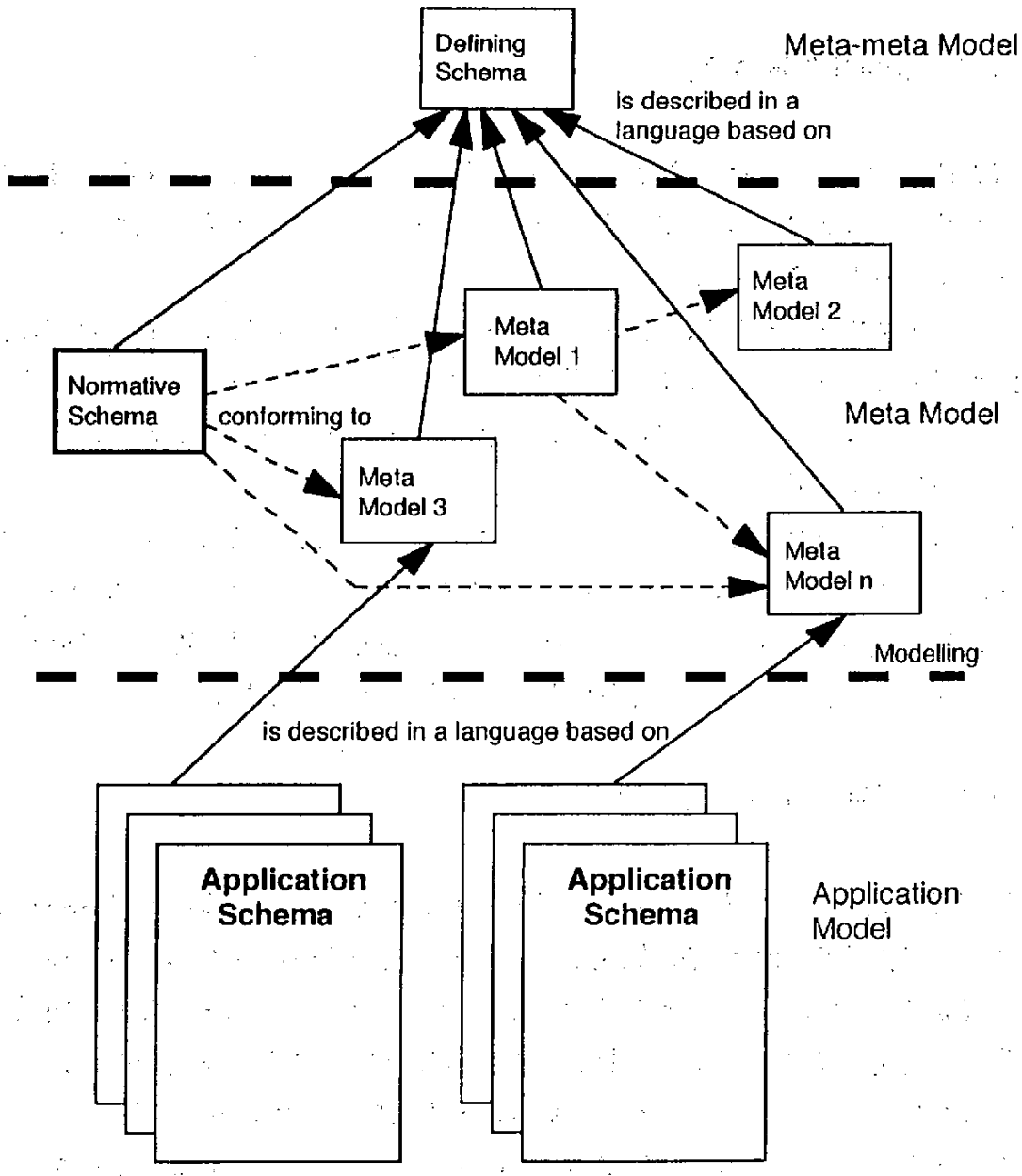


図２.４-１　CSMFアーキテクチャ（暫定案）

Defining Schema は、個々のモデル機能を定義する言語の規定であり、自然言語で補完された一階述語論理のようなものが想定されている。

Defining Schemaが提供するものは、概念定義のための構文に過ぎない。そこで、概念の意味を規定するための最小限のプリミティブを集めたものがMeta Modelレベルに書かれているNormative Schemaである。標準CSMFはDefining Schemaで規定される構文

（Syntax）とNormative Schemaで規定される意味定義（semantics）で構成される。

個々の応用分野のモデル機能は、この図のMeta Model 1、Meta Model 2、…である。これらのモデル機能は、標準CSMFとは独立に、それぞれの分野の応用スキーマを規定できる。しかし、それぞれのモデル機能の構成要素をDefining Schemaの構文とNormative Schemaの要素を使って定義することにより、異なるモデル機能間でのスキーマ情報の同一性の判定や（可能な限りの）スキーマ変換ができるようになる。

2.4.3　標準CSMFの具体的内容

上で述べたように、標準CSMFは、モデル機能を定義する構文（Defining Schema）と基本的なモデル概念（Normative Schema）で構成される。以下、それぞれについて、これまでに行われた議論の概要を紹介する。

（1）　抽象概念スキーマ言語

モデル機能を定義する構文の規定には、述語論理の活用が有益であるという点については、各国とも異論がない。この面における具体的な提案としては、米国提案の「抽象概念スキーマ言語」（ACSL：Abstract Conceptual Schema Language）がある。これは、人工知能分野の知識交換を目的としてスタンフォード大学が中心となってまとめたKIF（Knowledge Interchange Format）［Genesereth 92］をもとに、それを簡略化したものである。内容としては、述語論理をLISPの構文で書けるようにしたものである。

LISP構文であるため、慣れないと読みにくいものであるが、電子的情報交換が容易であるという利点がある。標準CSMFにこのACSLを採用するか、それとも、通常の述語論理の記法をそのまま用いるかは今後の議論の課題である。

（2）　規範的構成要素

構文だけ定めても概念を定義することはできない。これは文法があっても単語がなければ文を構成できないのと同じである。そこでCSMFプロジェクトでは、概念スキーマモデル機能を定義する上で必要最小限の基本的な概念の規定を進めている。この基本的な概念のことを規範的構成要素（Normative Constructs）と呼んでいる。このとき、何を規範的構成要素と見るかについて、これまでに３つの異なる意見がでている。

① 概念の原始的意味要素への分解

CSMFプロジェクトが正式発足する前に、米国はJ. F. SowaのCG（Conceptual Graph：概念グラフ）［Sowa 84］を用いた標準CSMFを強力に提案した時期がある。このCGは、述語論理に基礎をおく意味ネットワーク型の概念モデル機能で、概念間の関係記述をC. J. Fillmoreの格文法（Case Grammar）に基づく約40の原始的な関係およびその合成に制限するところに特徴がある。

このCGを用いて定義された概念間では、意味的な共通性と差異を機械的に抽出できる。つまり、スキーマ変換の自動化が（ある程度）可能である。このアプローチは理論的に優れたものであるが、個々の概念の原始的な関係への分解は、人手に依らざるを得ないところがあり、実際的かどうか疑問が呈されている。

② 既存モデル機能の概念の体系化

第２のアプローチは、既存のモデル機能で用いられている概念を網羅的に集め、それらを意味的に分類し、体系化しようとするものである。CGを取り下げた後の米国提案はこれに近いものであった。このアプローチは、公平ではあるが、規範的構成要素間の重複や規範的構成要素の矛盾した複数定義を生み、その規範性が疑われるものになりかねない。また、作業的にも、どこまでやれば十分か明確でないという難点がある。

③ モデル機能間の共通要素の抽出

世の中に異なるモデル機能が沢山あるといっても、そこで用いられているモデル概念には見かけはど大きな差はない。モデル化の対象を表現するのに、実体、実体型、オブジェクト、クラスなど異なる用語が用いられていても、その意味に大きな差がある訳ではなく、若干の機能的な差があるに過ぎない。属性、性質、関連、リンク、参照などのモノとモノとの結び付きについても同様である。

そこで、多くのモデル機能間で共通に見られる概念を抽出し、それを標準化すれば、他のモデル機能については、それとの対比で、同一性と差異を定義することができる。これが第３の立場であり、主として、日本が主張してきたものである。このアプローチは、①ほど理論的ではないが、実際的である。簡単に対応関係が付けられないような概念相互の変換が実務上必要であるケースは稀であると考えられるからである。

CSMFプロジェクトの議論は、①から始まり、②を経て、現在③に近いところでま

とまりつつある。昨年の暮れに開かれた中間会議で、一応の合意が得られた規範的構成要素の案は、図２．４-２に示す通りである。

しかし、これらの要素はまだ単に列挙されたレベルに過ぎず、その具体的な仕様は今後の課題である。仕様の具体化に伴って、これらの要素が変更される可能性は高い。この現時点の規範的構成要素は、伝統的なER（実体-関連）やオブジェクトの概念に基づくものではなく、ファクト（事実）をベースとする色彩が強いものである。

Normative Constructsの案 (SC21 WG3 N1909)

1) Data concepts/Enterprise data
entity-type, supertype/subtype, entity-instance, fact-type, role, attribute-type, attribute-instance, uniqueness-constraint, value-constraint, mandatory-constraint

2) Process concepts/Enterprise activities
process, event, trigger, state, agent, input/output

3) Temporal consideration/Enterprise events
relative-time, causal-dependency

4) Business rules
algorithm

5) Other
universe of discourse, conceptual schema

図２．４-２　規範的構成要素（暫定案）

２．４．４　今後の予定

1995年７月のオタワ会議で勧告されたCSMF標準化の予定は、以下の通りである。

WD（ワーキングドラフト）	96年１月
CD（委員会草案）	96年７月
DIS（国際標準草案）	97年７月
IS（国際標準）	98年７月

しかし、1996年３月時点で、まだWDが固まったといえる段階ではない。この予定通りに標準化プロセスを進行させる事は難しい状況にあるといえよう。

2.5 むすび

以上、現在進行中のISOのCSMFプロジェクトについて、その背景と現状を紹介した。CSMF標準は、他の情報関連の標準と異なり、直ちに具体的な製品に結び付くような性質のものではない。しかし、情報関連の種々の分野で概念レベルの標準化が進行しつつある現在、その早期の標準化が求められている。まず、必要最小限の部分に的を絞って、早期実現を目指すという日本の立場を今後も堅持して、プロジェクトに協力して行きたいと考えている。

参考文献

[ANSI/X3/SPARC 75] ANSI/X3/SPARC: Study Group on Data Base Management Systems: Interim Report, FDT (Bulletin of ACM-SIGMOD), Vo1.7, No.2, 1975.

[ISO/IEC l0032 95] ISO/IEC l0032: Reference Model of Data Management, 1995.

[Chen 76] P.P.Chen: The Entity-Relationship Model-Toward a Unified View of Data, ACM Transactions on Data base Systems, Vo1.1, No.1, 1976.

[Nijssen 77] G. M. Nijssen (ed.): Architecture and Models in Data Management Systems, North-Holland, 1977.

[ISO/TR9007 87] ISO/TR9007: Information Processing Systems -- Concepts and Terminology for the Conceptual Schema and the Information BaseI, 1987.

[穂鷹 96] 穂鷹良介: データベース関連技術の標準化, 情報処理, Vol.37, No.6, 1996.

[ISO/SC21 WG3 N1909 96] ISO/SC21 WG3 N1909: I nformation Technology -- Conceptual Schema Modelling Facilitie, 1996.

[Genesereth 92] M. R. Genesereth and R. E. Fikes, Knowledge Interchange Format Version 3.0 Reference Manual, Stanford University, 1992.

[Sowa 84] J. F. Sowa: Conceptual Structures, Addison-Wesley, 1984.

大規模知識ベースに関する調査研究報告書

発　行　平成９年３月
発行所　財団法人　日本情報処理開発協会
東京都港区芝公園３丁目５番８号
機械振興会館内
電話（03）3432-9390

08-R 008