

フレーム表現に基づく機械工学分野の 知識ベース管理システムについて*

長坂保美*¹, 大滝英征*², 石川義雄*²

Management System of Knowledge Base for Mechanical Engineering based on Frame Representation

Yasumi NAGASAKA, Hideyuki OTAKI and Yoshio ISHIKAWA

When we develop an expert system, knowledge bases which were built individually can be merged easily or connected organically, and can be used effectively. We have developed a total management system for knowledge bases based on frame representation. This paper describes a method by which to represent knowledge for the mechanical engineering field and by which to manage knowledge bases with different knowledge representation, and emphasizes the effectiveness of extended functions of frame representation, system slot and inheritance. As examples, using this management system, we show the connection of an existent CAD system and an existent program "synchronized meter".

Key Words: Production System, Modelling, CAD, Expert System, Frame Representation, Management System

1. 緒 言

機械構造物の故障診断や知的設計システムの構築にエキスパートシステム(ES)の導入が図られている。しかし、そこで構築される知識ベースは、専門家の経験と知識を基に独自の知識表現と推論機構が採用されているため、互いに利用されることはない。そこで、この独自に構築される知識ベースを有機的に結合することができれば、ESの構築が容易になり、知識ベースの一元化も図れる。

そのためには、知識ベースをモジュール化し、それを有機的に結合するための機能が必要となる。このような機能を実現するための知識表現としては、従来の知識表現を包含でき、しかも知識表現の記述性や推論機構の柔軟性に優れているフレーム表現がまず挙げられる。しかし、従来開発されてきたフレーム表現のKEE⁽¹⁾やART⁽²⁾は、異なるESツールで構築された知識ベースや既存システムを、有機的に結合するための機能を全く具備していない。そこで、知識ベースを有機的に結合するための管理機構や、統合的に活用す

るための拡張機能を持つツールの開発が必要となる。

著者らは、上記の機能を満足するツールの構築と、それに基づく知識ベース構築を目指している。ツールの構築では、KEEの知識ベースのデータ構造「ユニット-スロット-ファセット-バリュー」を準用し、ユニットを管理するシステムスロットの拡張、「ファセット」に代わる「アトリビューション」という概念を導入することとした。アトリビューションとは、スロットに関係する付属情報を意味し、従来のスロットの値やタイプといった固定的なものを意味するものではなく、STROBE⁽³⁾のようにユーザがその付属情報を自由に定義できる。また、その付属情報はフレーム表現内の記述に限定されず、ルール表現や従来の言語表現でも自由に記述できるのが特徴である。これらの拡張機能を用いれば、知識ベースを呼び出す際に、知識表現の異なる知識ベースが混在する場合でもその結合関係を自動的に生成することができ、知識ベースを有機的に結合することが可能である。

本研究は、このような考え方を基に、機械工学分野における適切な知識表現を検討し、それによって構築された知識ベースを用いたシステムの管理機構と、拡張したシステムスロットとインヘリタンス機能の有効性について述べる。

* 原稿受付 平成3年8月26日。

*¹ 正員, 埼玉大学大学院 (〒338 浦和市下大久保 255)。

*² 正員, 埼玉大学工学部。

2. 機械工学分野における知識表現

機械工学分野における知識ベース構築の問題は、材料の機械的性質のようにある程度確立された知識と、現場現場で生起している現象を見て判断する知識が混在していることである。

そこで、著者はこれらの知識を時間的要因を基に、一度設定されると容易に変更されることのない知識（静的知識）と、時々刻々変化したり規格化し難い知識（動的知識）とに大別し、これら知識の適切な知識表現についてまとめた。すなわち、

静的知識の知識表現：操作マニュアルや JIS 規格など、既に確立化された知識の知識表現は、フレーム表現が適していると考えられる。マニュアルや規格などは、項目別に階層構造で整理されており、フレーム表現の抽象-具体 (ISA) 関係で容易に表現でき、記述性に優れているためである。

動的知識の知識表現：機械加工時のドリルの回転数や送り速度など、その場での状況を判断しながら設定していかなくてはならない知識の知識表現は、プロダクション (ルール) 表現や述語論理表現が適している。これらの知識表現は、もともと推論機構を持っており、専門家の知識を容易に構築できることに加え、システム構築後の追加修正時における知識の無矛盾性を TMS⁽⁴⁾ 機能により容易に管理できるためである。

図 1 は、静的知識と動的知識の知識表現例を示したものである。図中の (a) は、「金属材料」知識の階層関係とその内容を示しており、記述性のある表現であることがわかる。

また (b) は、「機械加工条件」知識の述語論理表現の例で、M1 や M2 の条件を基に切込みや回転数が選択され、動的にその知識が決定される。このように、機械工学分野の知識を図 1 のような適切な知識表現で知識ベースを構築し、これらを有機的に結合して、利用者の利便性を図ることができる。

3. 知識ベースのシステム管理機構と

拡張機能

3.1 システム管理機構の概要 図 2 は、本報で提案する知識ベースを統合的に管理するシステム管理機構の骨格を示したものである。このシステム管理ユニット群には、「知識管理」、「知識モジュール」、および条件ユニットの枠組みを示す「フレーム表現」などから構成されている。

「知識管理」ユニットは、知識ベースの組込みに関係する拡張機能を管理している。知識ベースを組込むた

めに拡張したシステムスロット

@has-representation

@has-knowledge

と、知識ベース間を統合的に結合するために拡張したインヘリタンス機能

@ext-inherit

などが管理されている。

それでは、このシステム管理機構のメカニズムの動きを、順に追って述べる。

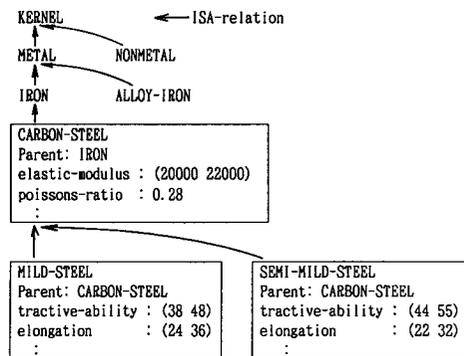
(1) 図 2 中のシステム管理ユニット群 (a) を、新たな知識ベース内の最上位ユニット (KERNEL) の下に組込む。

(2) 必要な知識ベースと知識表現を、「知識モジュール」内のスロットに記述する (b)。ここで、

@has-representation により条件ユニットが「知識モジュール」の下位に自動的に生成される (c)。

(3) 知識ベースと有機的に結合するための条件を、この条件ユニットに記述する (d)。ここで、@has-knowledge

により、この条件ユニットの内容を基に該当する知識ベースが組込まれる (e) か、知識ベースを呼び出すための関数が自動生成される (f)。



(a) Frame-representation of static-knowledge "METALLIC-MATERIAL"

```

:: Knowledge of Milling Cutter
((CUTTER D125) (diameter 125) (edges 8))
((CUTTER D160) (diameter 160) (edges 8))
:: Knowledge of Machining Conditions
((S45C 250BHN)
  (M1 ((CUTTER D125)
      (M2 (grain-depth 3) (grain-depth 2) (grain-depth 1))
      (M3 (rotary-speed 270) (rotary-speed 200)
          (rotary-speed 180))
      ((CUTTER D160)
      (M2 (grain-depth 3) (grain-depth 2) (grain-depth 1))
      (M3 (rotary-speed 200) (rotary-speed 180))
      )
  )
)

```

(b) Predicate-Logic-representation of dynamic-knowledge "MACHINING-CONDITION"

図 1 静的知識と動的知識の知識表現の例

このように、モジュール化された知識ベースを半自動的に容易に結合することができ、知識ベースを有効的に活用することができる。

3・2 知識ベースを有機的に結合するための条件ユニット 図3は、モジュール化された知識ベースとして、フレーム表現、述語論理表現、およびC言語による例を示したものである。知識表現がフレーム表現(静的知識)の場合と、フレーム表現以外(動的知識)の場合とは、知識ベースの組込み条件や呼び出し方法が異なる。すなわち、

モジュールがフレーム表現の場合：条件ユニットの下位ユニットに、必要なユニット群が組込まれる。条件ユニットでは、モジュール名、ユニット名、および上位概念の有無 (ISA 関係の整合性を図るための条件) などが記述される。そして、@has-knowledge 機能により、知識ベースが組込まれ、新たな知識ベースと結合される。

例えば、図中の(a)は「金属材料」知識ベースより、「炭素鋼」以下のユニット群を新たな知識ベースに組込んだものである。

モジュールがフレーム表現以外の場合：動的知識の知識ベースのため、時々刻々その状況が異なることを考慮する必要がある。そこで、条件ユニットでは、知識

ベースを呼び出す際の条件や返値の条件などが記述される。例えば、述語論理表現の場合は起動コマンドや返値の条件が記述される。そして、@has-knowledge 機能により、新たな知識ベースの付加手続きとして呼び出される Lisp 関数が自動的に生成される。

例えば、図3(c)のC言語のための条件ユニットでは、必ず三つの Lisp 関数が自動的に生成される。図中の(A)は、付加手続きとして直接呼ばれる関数である。(B)は、(A)で使用されている関数(STRENGTH)が外部記号であることを登録している関数である。(C)は、関数(STRENGTH)の入出力関係を定義している関数である。

このように、条件ユニットで記述された内容は、そのまま Lisp 関数として自動的に定義される。つまり、知識ベースの内容を全く理解していなくとも、条件ユニット内の指示に従えば利用できるという特徴がある。このことは、知識ベースをモジュール化する必須の条件であり、本手法によりそれを解決できる。

3・3 知識ベースモジュールの一元化のためのインヘリタンス機能の拡張 図2の管理機構では、フレーム表現における知識の無矛盾性を保証するため、知識ベース間の知識の一元化を図る機能が必要である。この機能は、拡張インヘリタンス機能@ext-inherit

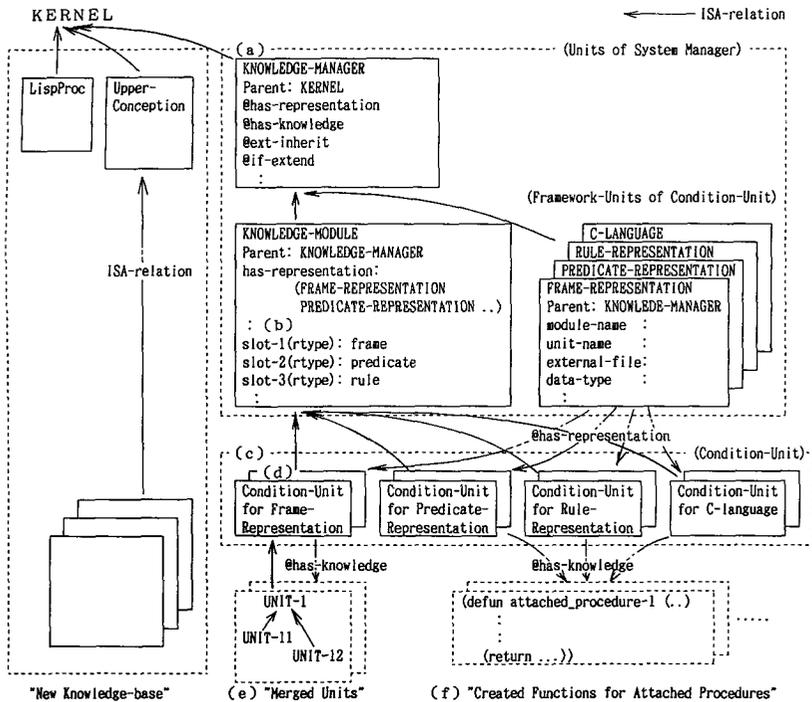


図2 知識ベースシステムの管理機構

によって実現されている。この機能は、スロット内のアトリビューションとして、

(ext-inherit スロット名)

形式で表される。スロット名は継承先のユニット名を格納している名称を指している。

図4は、この拡張機能の例を示したものである。この例は、ユニット「並目ねじ」のスロット「引張強さ」にメッセージを送り、その評価値を探索する過程を示している。「並目ねじ」には該当するスロットが存在しないので、ISA 関係を利用して上位ユニット「ねじ」へ移り、スロットの参照を図る。ここでは、該当するスロットは存在するものの未定義状態にある。しかし、拡張インヘリタンス機能 (ext-inherit) が定義されているためこの機能が起動する。この拡張インヘリタンス機能は、まずユニット「並目ねじ」に戻り、スロット「金属属性」の値を参照する。次に、このスロットが示すユニット「軟鋼」内で該当するスロット「引張強さ」の参照を図る。「軟鋼」内には該当するスロットが存在し、しかも値を持つ。そこで、このスロット値(38 48)を評価値として返す。もし、このユニット「軟鋼」にも該当するスロットがなければ、ISA 関係によってユニット「炭素鋼」へと参照が移る。

このようにスロット値を取り出すことができるの

で、知識ベースの知識内容を重複して構築する必要がなく、容易に探索できるため、知識ベースの一元化と有効利用に極めて有効な拡張機能である。

4. 知識ベース管理機構の適応例

4.1 システム管理機構を用いた推論の適応例

図5は、フレーム表現の「機械要素」と「金属材料」知識ベース、およびC言語で書かれた強度設計プログラムを用いて、フレーム表現の付加手続きにより強度解析を行った例を示したものである。

「機械要素」知識ベースは、ボルトやナットなどの機械部品の形状要素と機能などに関する知識が格納されており、静的知識として扱われている。機械部品はなんらかの材料で作成されているわけで、「金属材料」知識ベースとの関係を表しているのがスロット「金属属性」である。そして、その関係づけは前述の拡張インヘリタンス機能で実現している。図中の点線は、付加手続き (SCREW-STRENGTH) を起動した際の探索経路を示している。

ここで、図2の知識ベース管理における推論機構の動きを理解するため、知識ベース中の付加手続きを起動し、その値を取り出す経路について述べる。

(1) 付加手続き (SCREW-STRENGTH) を起動

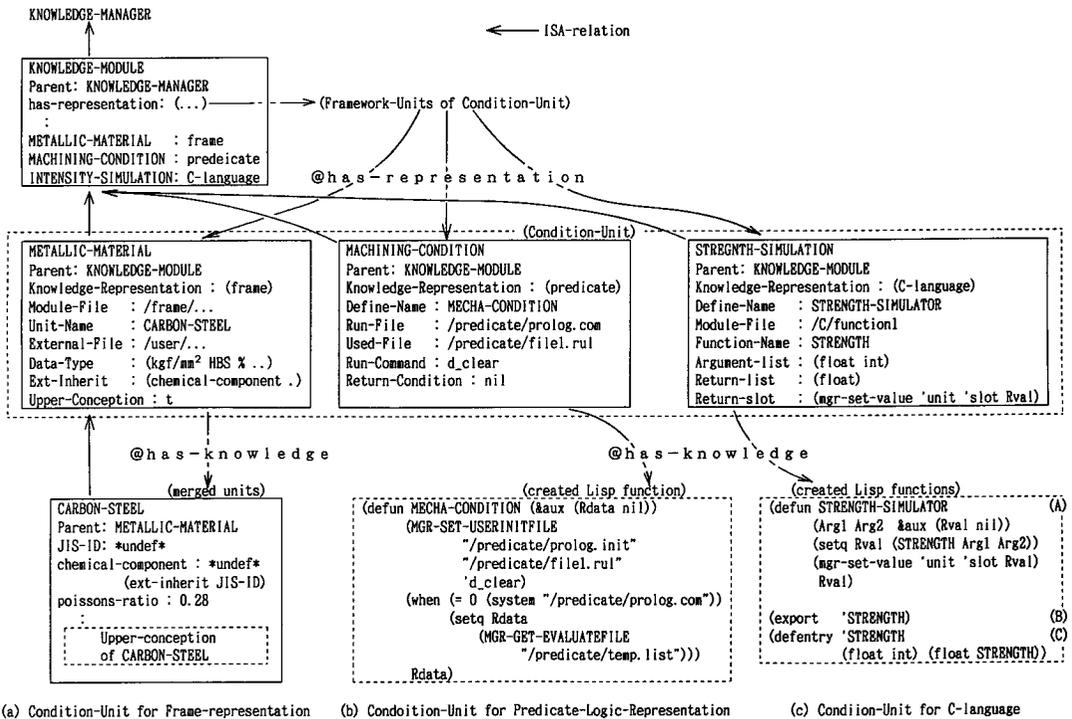


図3 知識ベースを有機的に結合するための条件ユニット

する。

(2) 第一引数の評価が行われ、ユニット「M8並目ねじ」のスロット「内径」にメッセージが送られ、その値 7.74 が返される。

(3) 第二引数の評価が行われ、ユニット「M8並目ねじ」のスロット「引張強さ」にメッセージが送られ、3・2 節の拡張インヘリタンス機能により、(38 48) が評価値として返され、引数として最初の値 38 を採用する。

(4) 二つの引数が評価された後、C 言語で書かれた「ねじの強さ」の強度設計プログラムへ渡され

$$(\pi/4) * (\text{内径})^2 * (\text{引張強さ}/6)$$

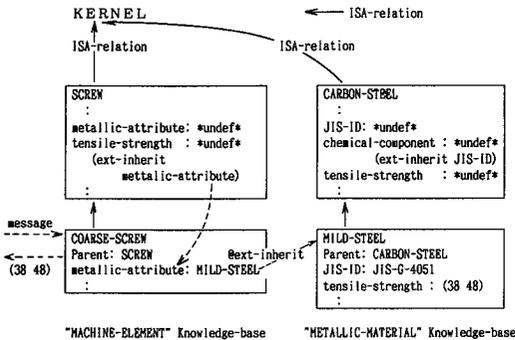


図 4 拡張インヘリタンス(@ext-inherit)機能の例

評価される。

(5) 評価された値は、あらかじめ指定したスロット値に格納され、この付加手続きを終了する。

このように、図 2 の管理機構では従来の強度設計プログラムとも容易に結合できる。そのため、従来の資産も有効に活用でき、知識ベース構築の際にも極めて有用な管理機構である。

4・2 既存システムとの結合の適応例 図 6 は、著者の開発したフレーム表現の ES ツール ZES (UNIX 上の KCL 言語で開発) 上に、図 2 の管理機構により知識ベースを有機的に結合した例を示したものである。(a)は、述語論理表現の「ハノイの塔」知識ベースを結合した例である。(b)は、著者らが C 言語で構築した「メータ表示」の例である。また、(c)は市販の CAD システム (CADKEY) を有機的に結合した例である。

図中(b)は、メータ表示部分を C 言語で表示させ、メータの動きをフレーム表現のスロット値で制御している。また、メータ間の制御は、拡張したインヘリタンス機能を用いて同期をかけている。また、(c)は CAD システムで描かれた図面の内容を、自動的にスロットの値にセットし、4・1 節の強度解析などに応用している。

このように、図 2 のシステム管理機構は、知識ベースの知識表現や言語系を意識することなく容易にシス

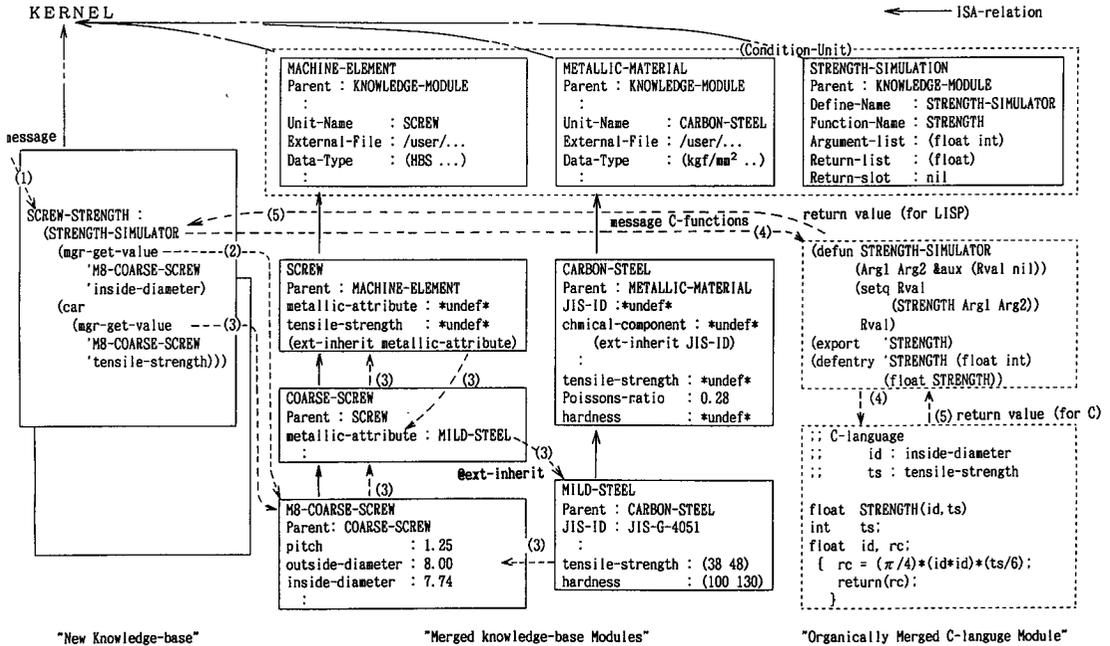
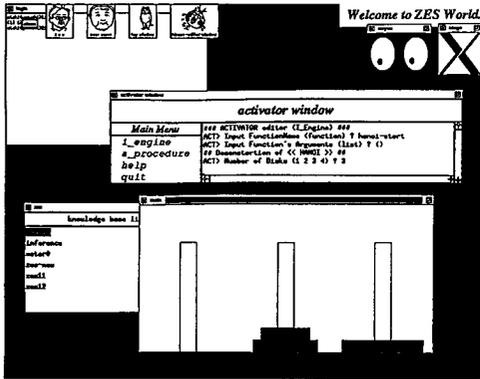
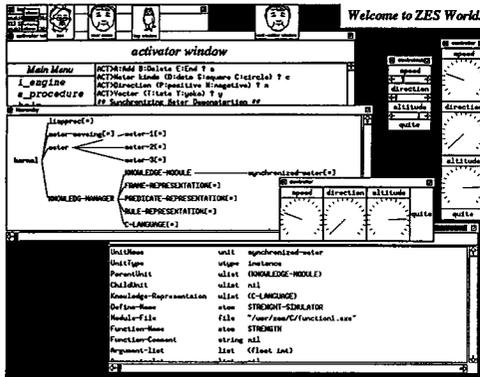


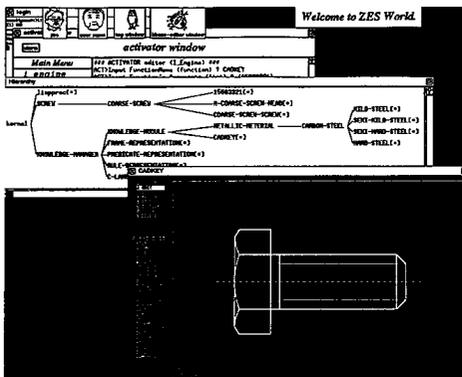
図 5 システム管理機構を用いた推論の適応例



(a) Example of "the Tower of Hanoi"



(b) Example of "Synchronized Meter"



(c) Example of CAD (CADKEY) system

図 6 既存システムと有機的に結合した例

テム内に呼び込むことができる。特に、図中の(b)の

例が示すように、C言語で書かれた内容とフレーム表現の知識とが共存して実行できることである。このことから、従来のFORTRAN言語などとも、C言語を介することにより容易に結合できる。

以上のことから、本システム管理機構は、従来の豊富な知識ともいべき解析シミュレーションや解析結果のデータを、有効的にかつ効果的に活用することができ、知識工学分野との有効的な橋渡しとなる。

5. 結 言

ここで提案した知識ベースの管理機構およびその活用は、個々に構築されている知識ベースを統合的に運用する上で有効なものと考えられる。本研究をまとめてみると以下ようになる。

- (1) 機械工学分野における知識表現を静的知識と動的知識に区分し、静的知識はフレーム形知識表現が適しており、動的知識は述語論理表現あるいはプロダクション表現が適していることを示した。
- (2) 図2の知識ベースシステムの管理機構により、知識表現の異なる知識ベースを統合的に、かつ有効的に活用できることを示した。特に、知識ベースモジュールの組込み条件を自由に定義できるので、知識ベースを有効的に活用でき、既存の知識ベースや既存のプログラムなどの資産も有効に活用することが可能である。
- (3) インヘリタンス機能の拡張により、知識ベースのモジュール化と一元化が図れる。

文 献

- (1) Kehler, T. P. and Clemenson, G. D., KEE, Knowledge Engineering Environment for industry, *Artificial Intelligence*, (1983), 573-581.
- (2) Clayton, B., *ART Programming Primer Inference Corp.*, (1985), 21-148.
- (3) Laufue, Gilles M. E, A Modular Tool Kit for Knowledge Management, *IJCAI*, (1985), 832-839.
- (4) Doyle, J., *A Truth Maintenance System Artificial Intelligence*, 12(1979), 231-272.
- (5) Minsky, M., A Framework for Representating Knowledge, Haugeland, J., *Mind Design*, MIT, (1981), 95-128.
- (6) 馬場, 機械工学必携 (第7版), (1986), 249-364, 三省堂.
- (7) 上野, 知識工学入門, (1985), 84-187, オーム社.