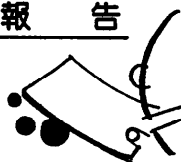


報 告



パネル討論会

エキスパート・システムの課題†

パネリスト

石塚 満¹⁾, 小山 照夫²⁾, 高木 茂³⁾
 辻井 潤一⁴⁾, 元田 浩⁵⁾
 司会 上野 晴樹⁶⁾

上野(司会) エキスパート・システムは実用システムも少しずつ出始めており、いよいよ実用化の時代へ入ったと言えますが、一方では現在の技術はいまだ不十分であり、限界も見えてきたから、マスコミの大げさな報道に浮かれていてはならない。期待への反動として過熱が間もなく冷めてしまいかねない、という見方もあります。そこで今日は、我が国のこの分野における第一線の研究者に集まっていただき、それぞれの経験に基づいて、冷静な目で、エキスパート・システムの直面している課題について議論していただくことになりました。

まず議論の手がかりとして、エキスパート・システムとは何かを簡単にまとめてみます。「エキスパート・システムは、問題領域の専門知識を利用して、十分に複雑な問題を、エキスパートと同等の能力で解決する、知的プログラムをいう」と定義できるのではないのでしょうか。ポイントは、エキスパタイズつまり熟練された知識を利用すること、これはエキスパート(専門家)から獲得する必要があること、そして、エキスパートでなければできない問題を対象とすること、そして、そのシステムの能力はエキスパートと同等であることを目指していること、などです。簡単にできる問題ならばほかの手段でも実現できるでしょうし、あるいはエキスパート・システムの形態をとっていてもデモとしてしか動かないシステムは、エキスパート・システムではなかろうというわけです。ですからエキスパートがどの程度の能力を持つかということを想定して、それと比較できる能力を持つ必要があるわけです。最初のエキスパート・システムとして有名な

DENDRALの開発者で、かつ知識工学の提唱者でもあるスタンフォード大の Feigenbaum が、77年に「能力は知識の中に潜んでいる。だからたとえ推論のメカニズムが簡単であるにしても知識だけは質の高いものを十分準備する必要がある。」ということを指摘したが、この考え方は今でも基本的には生きていると思われる。

さて、いくつか似て非なるシステムであると考えられるシステムをあげてみます。1番目がテキスト・システムで、これはテキストを見ながら作ったものをいいます。デモすると知らない分野の人にはエキスパート・システムに見えるが、エキスパートから見るとおもちゃであるというものです。2番目はチューデント・システムといい、学生が作ったシステムです。学生はけっしてエキスパートではないから、そこで作られたものはやはり使いものにならない。トイ・システムというのが3番目ですが、最初から実用的な応用を考えないで単に勉強のためあるいは趣味で作ったものです。最後はデモ・システムです。これは、デモンストレーションのために作ったシステムでデモ効果は高いが、知識が極端に片寄っており使いものにはならない。これらは、すべてある意味でエキスパート・システムのだけれど、エキスパート・システムではない。こういうことをやっても、ほんとに使えるエキスパート・システムにはならないであろう。というのが、よく指摘される問題です。

では混乱の原因であるエキスパート・システムという言葉の使われ方といえば、実態は次のようなことであります。一つは、エキスパート・システム開発ツールを使って作られたプログラムを即、エキスパート・システムということが多い。上の定義からするとそうではないが、一応そうであるように言われています。これは、LISPやPROLOGを使って何かやっていると人工知能の研究をやっているような錯覚に陥ることと

† 日時 昭和60年11月8日(金)

1) 東大, 2) 浜松医大, 3) NTT, 4) 京大, 5) 日立, 6) 電機大

同じであるわけです。たとえば、プロダクション・システムによるプログラムはなんでもエキスパート・システムだということになっているのですから、外の研究者から見ますとエキスパート・システムというのは AI ではない、ああいうことをやっても人工知能は研究できないのだと指摘されてしまう。しかし、この分野の学問的研究をやっている人は、エキスパート・システムを人工知能の研究としてやっているのです。

そこで、二つ目が、本来の概念に基づく知的問題解決支援システムとしてのプログラムです。ここでは、能力がエキスパートのレベルに達するためにはどうしたらよいかということが基本的目標となるので、それを達成するために、専門家の認知行動、複雑な問題における体系的知識の表現、効率の良い推論制御技法、知識の獲得と検証などの課題をさぐりながらやっているというわけです。すなわち、AI 研究そのものであります。

さて、1 番目のエキスパート・システム開発ツールの利用という点から AI ビジネスが起きているわけですが、これは AI に光と影があるとすると光の源その 1 だといえます。AI ビジネスの人たちはこの“光”を商売にしているという感じがします。AI はもっと広い学問分野ですが、AI=エキスパート・システムとして、単なるイメージアップのために使われ過ぎており、このことが、エキスパート・システムは学問ではないなどという批判の原因となっているものと思われます。しかも、実際にはエキスパート・シエル・ビジネスであり、もっと辛辣に言いますとエキスパート・システム教育キット・ビジネスであるわけです。つまりツールが手元にないと実験ができないので、高いお金を出して買うわけですが、実際には単なるプログラムではデモ・システムにしかならない。

では次に、エキスパート・システムを研究開発をする立場から整理してみます。まず、1 番目は、エキスパート・システムはプログラムであるという考え方です。知識表現言語はプログラミング言語である。これはつまり、FORTRAN や COBOL によるプログラミングのアプローチでは、エキスパート・システムをなかなか引き出せないが、知識表現言語を使うとスムーズに引きだせる。したがって能力の高い問題解決プログラムの開発ができる。そういうのに向いている言語である。こういう考え方では、結局はエキスパート・システムはプログラムなんですから、人間に分からないことはプログラムできないわけです。このところがポイン

トである。良く分からないからエキスパート・システムを開発するのは、したがっておかしいわけです。

2 番目は、エキスパート・システムは実用システムである。実用システムを目指しているのだということです。そこまではないかと本当の意味でのエキスパート・システムといえない。現場で日常的に使われる必要がある。そのためにはどうすればよいかという問題が出てくるわけです。一つは、開発のプロセスから眺めてみますと、まず概念設計をし、次にインプリメンテーション、つまり知識ベースを構築することになります。次に、インフォーマル・エヴァリュエーションをする。つまり、いくつかの典型的なケースを使って、うまく知識が働くかどうか、能力がどの程度にあるかということなどを検証しながら改善していく。それがうまくいった次の段階で、フォーマル・エヴァリュエーションをする。つまり、ケースの数を増やしていった、いろいろな汚いケースのデータを使いながら能力を厳密に査定していく。これでうまくいくという見通しがでて初めてフィールド・テストになる。フィールド・テストに入ると、今度は設計者以外の人が実際に使う。そこで実際の問題がうまく対応できるかということが評価されることになる。ここでうまくいったときに、はじめて運用のシステムになってくる。そこまで行くのが本来のエキスパート・システムですが、インプリメンテーションのところが、学会発表や新聞発表で非常に多いわけです。そのために誤解していることが多いのではないかと思います。よそがやったという、うちもやったというのを新聞発表せざるを得なくなりますね。それで誤解が続いている。Davis が 1982 年のレポートの中で、これらは“アイディアが動いた”(Idea worked.) という類であり、それをいかに脱却するかというのが重要だということを指摘していますがまさにそういうことだと思います。

3 番目はエキスパート・システムは研究対象であるという立場です。これは、先ほど言いましたように、より専門家の能力に近づける努力をするわけで、深層知識システムだとか、認知モデルなど、研究者の理解度をモデル化してそれをシミュレーションで検証することになります。AI (Artificial Intelligence) は IA (Intelligence Amplifier) であるかといわれますが、こういうアプローチによって自分の Intelligence がより明瞭となる。したがって、この立場では、エキスパート・システムの研究はそのまま AI の基礎研究であるわけです。時々ビュー AI と称して昔やられていたよ

うなトイ・モデルをいじりまわして満足している人を見掛けますが、AI の歴史を勉強してほしいものです。

このような三つの考え方があってそれを交ぜこぜにしますと論点がぼけてしまったり、誤解を生むということになると思います。

最後に、いくつか課題をあげてみますと、一つは能力をどうやって高められるかという問題です。これは深い推論だとか、真に意味のあるエキスパタイズをどうやって引きだすか、などと関係します。次が、実用システムをどうやって実現できるかという問題です。ユーザ・インタフェースや実行効率をいかに早めるかという問題とか、在来のシステムとどうリンクするかという問題です。3番目が知識獲得のボトル・ネックをどうやって解消するかという問題です。本格的な開発をやろうとすると必ずぶつかる問題で、エキスパートから知識を獲得するところが細いボトル・ネックになっている。これを抜ける努力をしなければならない。それを助けてくれる人がノーリッジ・エンジニアです。もう一つは知識獲得支援ツールを整備すること。そして知識表現そのものをその分野の専門知識が表現しやすい形式にすることです。これは非常に重要な問題です。最後に、これが不可欠な問題ですが、エキスパートが存在しなければならない。よくエキスパート・システムを開発したいという相談を受けるとき、どうしてですかと聞くと“うちにエキスパートがいなくて困っているからせめてシステムをつくってそれに代えたい”と言われる。どこかでエキスパート・システムを買ってくる時はそれでいい。将来の AI ビジネスがそこまで行ってほしいということですが、現在ではそういうことはできない。エキスパートが身近に存在するということが重要である。こういう問題になりますと、当然人材不足をどう解決するかということになる。皆さん悩んでいらっしゃるわけです。AI ビジネス側が AI 教育までやり始めて、目がとび出るほどのお金でも仕方がないから参加しようということになりかねない。教育体制が不備であるからそういうビジネスが成り立つわけです。

次が、“Look USA”をどうやって抜け出すかということ。二言目にはアメリカという話が出てくるわけですがそれを解決しなければならないわけです。そして最後が、“AI の冬の時代”にどう対処するかという問題で、今までも冬の時代というのがあって夜明けになったというわけですが、実態は根無し草的なところがあって、あまり期待が大きすぎると 2, 3 年後

にはまた冬の時代がもう 1 度くるのではないかということを実際に指摘している人が多いわけです。私もこういうことを心配している一人です。

さて、次のパネラの話に有益につながるために、もし質問がありましたらちょっと議論をしたいと思います。

志村(東工大) さっき AI ビジネスはエキスパート・ビジネスだというようなことをおっしゃったけれども、将来はどういうふうになると思っていますか。

上野 現在シェルを売っている人たちは必ずしも AI の専門家ではない。大部分がアメリカから買ってきて売っているわけですが、なんでもできるシェルというのは存在しないわけです。どういう性質の問題はという推論のやり方とか知識の表現が向いているからこのシェルが良いとか、そういう判断が重要になるわけです。ところが今は売りこむために、我が社のシステムがベストであるなどという売り方をしていて混乱に陥っているということがあるわけですね。自動車でも家庭用掃除機でもいろんなタイプのものがある。ユーザは目的に合わせてそれを選択するわけです。シェルもそうでなければならない。しかし、今は、何が分からないか分からない。だからとにかくどれかを使ってみようということになる。このような試行錯誤を経て、少しずつポイントが分かってくる。その次に本格的な開発の手続きに移っていくであろうと思いますが、そうやって初めて AI ビジネスが軌道に乗り始めると思います。それには後 4, 5 年かかると思われます。将来ある望ましい姿は、ツールを作っている人とエキスパートとが協力してアプリケーション・パッケージを作ってそれが普及することではないかと思っています。

たとえば、昔はコンピュータのハードウェア・リソースが共同利用されて TSS というリソース・シェアリングがあった。今はそれより一歩進んで、データそのものの共同利用であるデータ・シェアリングが進んできている。次のステップになり、専門が細分化していきますと、データだけでは意味が分からないので、解釈もやはりサービスしてほしい。それはノーレッジ・シェアリングだと思うんです。そういうのが、作りやすいところから少しずつ出てきて、次のステップのノーレッジ・シェアリング・システム (KSS) というのが普及していくのではないか。それが本当の意味の AI ビジネスのサービス・ビジネスであるのかなと思

うのです。それは 90 年代になって少しずつケースが出てくるのではないかと考えています。

では、次は小山先生よろしくをお願いします。

小山 ただいま上野先生がエキスパート・システムの本質的な問題について非常にきれいに整理されたわけですが、私は今の上野先生のお話からすると



エキスパート・システムとは呼びにくいシステムについてもっと考えても良いのではないかとことをお話ししてみたいと思います。というのは、先程のお話にも出てまいりましたが、最近エキスパート・シェルなどが整備されていくことによって、エキスパート・システム的なものを構築するための環境ということでは、確かに整備されてきていると言っているのですが、それでは実際に本当の意味のエキスパート・システムが構築できるかどうかという問題を考えると、特に医療ということを考えると、現状ではなかなか難しいことが多いように思われるのです。確かにエキスパート・シェルのようなツールを利用することにより、ある程度定式化された演えきの推論に基づいて、これまでのシステムでは実現しにくかったような有用な機能を実現できるということはあると思います。しかしながら本当のエキスパートの能力の実現ということになると、既存のツールが提供しているような枠組みの中で、必ずしもきれいに定式化できないような問題がどうしても出てくるように思われるのです。医療の問題でいえば時間経過の取り扱いですとか、概念間のさまざまな関係、あいまいさの取り扱いなど、これから検討していかなければならない問題がたくさんあるわけで、これらをきちんと定式化し、実際にインプリメントしてからでないと本当の意味でのエキスパート・レベルのシステムにはならないのではないかと思います。よくエキスパート・システムの一つの実用形態がコンサルテーション・システムだということが言われるわけですが、これは私の考えではそのようなシステムがエキスパート・レベルの能力を達成できていない証拠であろうと思うのです。なぜコンサルテーションという形を取るのかといえば、どうもその根底には、システムの結論を人間が完全には信用していないから、そこで人間が一つのフィルタの役割を果たしてシステムの結論を取捨選択してやろうというニュアンスがあり、人間が判断したのだからシステムの責任ではないという逃げの姿勢があるように

思われます。ところがこのような取捨選択が適切にできるということは、それ自身かなりのエキスパート化を必要とする作業であり、変に素人に使わせると益よりむしろ害のほうが大きいようなシステムになっているわけで、このようなシステムが実用システムとして議論されていること自体、結局現時点では本当のエキスパート・レベルの能力を持つシステムの構築が困難であることを示していると思います。

ところでそれではこのような、エキスパート・レベルの能力としては問題のあるシステムといったものは全く役に立たないのかといえば、実はそうでもなくて、素人には使わせたくないけれども、専門家が自分の仕事を行っていく上での道具として考えた場合には、特に秘書的な機能の提供ということを中心に考えれば、使い方によっては十分実用的なものになる可能性もあると考えられるわけです。そこで実用システムということを考える場合には、本当の意味でのエキスパート・システムを実現しようという試みのほかに、本来の意味ではエキスパート・システムと呼びにくい、演えきの推論の利用により現時点で実現可能であるような機能の中に有用な物があるならば、それらを利用したシステムを作ってみることも考えられて良いと思うのです。ただこの場合にもいくつかの問題があると思うわけで、まずそのようなシステムが能力的には当然制約があるものとして、それに何をさせるかというシステム仕様を明確な形で決めてやる必要があります。従来エキスパート・システムを構築するにあたっては、最初からあまり明確な仕様を決めるということとは行なわれていないのですが、実用システムということを考える場合にはシステム仕様の早期収束を図る必要があるわけです。またここでの仕様ということは、必ずしも本当の意味でのエキスパートの能力を期待するのではなく、むしろ医師の日常の仕事の手助けになる機能を中心に考えるという意味で、従来のエキスパート・システムということで考えられていたものとはかなり発想の異なったものとなるでしょう。次に別の問題として、そのようなシステムができたとして、システムが長期にわたって利用されていくためのメンテナンスができるかどうかという問題があります。知識ベースシステムということ考えると、システムのカバーする範囲を広げようとすると、どうしても非常に特殊な症例に限ってのみ利用される、もしかしたら一つの症例に用いられた後は二度と利用されないかもしれないような知識まで含めていかなければなら

くなる傾向があると思うのですが、このような場合にたとえば新しい医学的知見を円滑に組み込むことができるのかなどの問題も考えておく必要があるでしょう。それから実用システムということになると、そのようなシステムの利用される環境ということも考えておく必要があります。日常の仕事を行いながらその中で利用するシステムということを考えますと、医療という世界の中では患者を相手にしているその場でシステムが利用でき、適切な情報支援を行ってくれるという態勢を作っておかなければならないということもあるわけです。以上をまとめますと、現在エキスパート・システムと呼んでおりますものが、本当の意味でのエキスパート・レベルの能力を達成しているかどうかは、特に医療の場合には問題が多く、またごく近い将来そのような能力を持つシステムが実現されるということも期待しにくいという事情があると考えられるのですが、しかし能力的にそういう限界のある技術であっても、そのような限界を認識して、可能な範囲内で実用的システムを作りうる可能性はあるのであり、その場合の問題についても検討することが重要であると考えているわけです。

上野 どなたか発言はありませんでしょうか。

中間 カシオ計算機の間中と申します。何人かのお医者さんと個人的におつきあいをしていて、雑談的に聞いた話ですが、一般の開業医の方はそういう機械を導入して診断することに抵抗があるということなのですが、小山先生は、専門医の方とおつき合いが多いと思いますが、その点専門医の方のお気持ちはどうなんでしょうか。

小山 医療の分野で本当のエキスパート・システムを作るのは難しいという話をしたわけですが、もしも本当にそのような能力を持つシステムが実現できたならば、それに対する見解は当然変わってくるものと思います。現時点では不完全な能力しか持たないシステムを見せて、それを完全に近い能力を持つものとして使うことをどう思うかということを知っているわけで、これはそのようなシステムは使い物にならないという結論が出てもしかたがないと思うのです。また逆に専門家の方の見解として、どうせ非常に高度の能力については達成されないだろうということで安心してある面もあると思います。

上野 ほかに発言はないですか。

高田 日本タイムシェアの高田ですが、一つお聞きしたいのですが、医学関係のエキスパート・システム

というのは鳴りもの入りで沢山でてきても全部だめになったという話を聞きますが、そのだめになった理由ですが、たとえば要するに、100% 当たらないということなんです、100% 当たらないければ使い物にならないでしょうか、聞くところによると、病気の85% くらいはほおっておいても治ってしまうとのこと、すると85% くらいはちょっとさわればたいてい治ってしまうということで……これはまあ冗談なのですが、本来85% くらいしか当たらないものだと頭において使えばそれはそれなりに使い物になるという気がするのですが、まずその第1に……

小山 医療診断システムの診断精度ということに關しましては、それが何を表わしており、医療上どのような意味を持つかについてはさまざまな問題があります。現在言われているところの診断精度というのは、特定のサンプリングの行われた患者に関して集められた、特定のデータを利用するという条件のもとで、システムが導く結論と専門医の結論とがどの程度一致するかを議論しているわけですが、このようにして導かれた統計的数値が実際の現場で特定の個人の患者を扱う際にどのような意味を持つのかということになると、非常に不明確なものであると言わざるを得ません。そもそも医療ということを考えると、個人の患者について診断が当たったとか当たらなかったかは結局最後になるまで分からないと言うこともできるわけで、当たった当たらなかったよりはむしろ医療を行っていく各段階で医者がベストな選択をしたかどうか、医師の責任問題とも関連して問題とされるであろうと思うのです。そこで MYCIN などの、適用範囲に制限の大きい現在のエキスパート・システムの能力を完全に信用して、その提案をそのまま実行したとすると、それは多くの場合ベストな選択ではないであろうと考えられるのであり、そのようなものを頼りにするくらいなら自分自身の判断と責任で結論を出した方がよいということになるのは、ある意味で当然であると思います。

上野 では次に、NTT の高木さんに話してもらいます。

高木 NTT の高木です。私は LSI 設計のエキスパート・システムを作っています。エキスパート・システムは分野によってとらえられ方が違うと思うのです。何もできていない分野で作ればエキスパー

ト・システムと……、ところが、LSIのような従来からDA(設計自動化)システムの研究が盛んな分野でエキスパート・システムを作った場合には、厳しい見方をされます。

たとえばAIの分野では、診断知識のモデル化法として、症状と原因を表層的なルールで結びつける方法と、深い意味という因果モデルを使う方法があって、因果モデルの方が発展した形態といわれています。LSIの分野でも診断問題があります。因果モデルに該当するものとして、構造・動作モデルというのが考えられます。装置はどういった部分で構成され、各部分がどういう機能を持っていれば全体がどういうふうに動くという原理的な知識を入れておけば、その知識を利用してどこがこわれているか診断できる。また、こういうモデルというのはモデルをたてた段階で、暗黙の前提が入ります。たとえば障害は一カ所とする、あるいは障害は間欠的に発生したり直ったりというのではなく固定的に発生するなどの前提です。ところが従来の診断アルゴリズムも、構造・動作モデルとは陽にいいませんが、原理的にこれと類似の方法を使い前提もほぼ同じものを使っています。したがって原理、前提が類似のものを別な方法で作っても、多分DAの分野の人はあまり感心してくれないと思います。今までより何が良くなったかという話が期待されているわけです。エキスパート・システムの研究を見てもみると、推論機構や表現方式などの発表はたくさんあります。ところが問題解決のモデル、先ほど言えば知識の浅いレベルとか深いレベルとかの議論は割り合い少ないですね。そういったものを使ってどれだけ効用があったかという論文はさらに少ない。そうすると従来のDAと何が違うのかということがよく分からない。

それでこれは非常に独断と偏見に充ちているのですが、LSI関係のエキスパート・システムの意味を無理やりに評価しようとするとなればこのような段階になるのではないのでしょうか。第一番目はシステムの構築の容易さ、柔軟性、マンマシンの向上を狙ったものです。方法論としては従来からだいたい分かっていたが、プログラミングがなかなか大変だったものを、AIのプログラミング技法を使って容易にしようとするアプローチです。オブジェクト指向のシミュレーション、記号シミュレーション、オブジェクト指向のマスクパターン生成、マルチウィンドウを使ってマン・マシン・インタフェースを良くしようというのが該当す

るでしょう。第2番目は従来の延長ではあるが、もう少しきめ細かくやって設計品質をあげようというものです。たとえばLSIでは素子間の配線問題というのがあります。従来の手法ですと、一つのアルゴリズムにたよりますから98%から99%までは配線できるのですが最後の1%の配線ができない。未解決の部分は人間がやらざるを得ない。その部分をルールでうまくやろうというシステムがあります。しかしまだ完全には自動化できなくて設計者とインタラクトしながら解決していく。システムが配線の込み具合などを計算して、概略配線の経路を候補としてあげてくると、設計者が大局的に判断して候補の承認/次候補の要求を行う。要するにデータ作りと、候補案の作成はシステムに任せ、重要な判断ポイントは人間がする。こういう共同作業は現実的であり、またLSI設計におけるエキスパート・システムの現状かと思います。しかし今の話で人間の大局的な判断とは何かというのが興味深いところです。大局的な判断というのは設計の場合には割合たくさんある気がします。そういうものがどこか一カ所明らかになれば他の部分も分かってきて全体がうまく自動化できる気がします。

第3番目は従来のDA手法とは違った何か新しい視点によって設計品質を向上させるとか、計算時間を短くするというものです。たとえば、与えられた機能仕様を満たす演算器を合成する問題の場合、従来でもやろうと思えば非常にロジカルな世界ですから、論理式の公理、定理などを持ち出してきて合成することは可能です。しかしそういう手法でやるとできたものはよくないし計算時間も非常にかかります。それに対し人間は非常に効率的に設計している。この効率的、あるいは大局的にやるという設計知識というのは漠然として分からない。そういう漠然としたものをここでは大きなアーキテクチャみたいなサブ構造を想定して、機能仕様とサブ構造を関連づけるルールで定式化している。こういう方法をとると計算時間は短縮され、設計品質も人手設計とほとんどかわらなくなる。別の例としてLSIの配置問題というのがあります。従来ですと基本セルをまず初期配置します。次にそのなかの二つのセルに着目し、それを交換したときの配線長を計算し、配線長が短くなるようであれば、交換することにより配置を次第に改善していく。この方法ですと素子の数が増加すると計算量が急激に増大する。それに対し、エキスパートの場合、割合バサッ、バサッと物を置いてできあがりというふうにする。こういった知

識を明らかにできれば、処理時間が激減できる可能性もある。また新機能の追求もこの分類に属するでしょう。すなわち、従来の DA システムでやっていないこと、たとえば仕様変更に対する自動対処、設計事例の再利用、再設計、あるいは知識の自動獲得などに知識工学の手法が導入されればインパクトは大きいでしょう。

現在エキスパート・システムで使っている言語としては、LISP、OPS、Prolog、フレーム……といろいろあります。表現力、記述のスタイル（宣言的記述／手続きの記述／両方）、推論機構の豊富さ、スピード、スケールアップの容易さ、プログラミング環境、知識獲得のしやすさ、などの評価尺度でいろいろの優劣があってどれがいいか一概には言えない。しかし LSI のような大きな問題を扱うには、スピードと容量が重要な評価基準になると思います。

LSI 設計分野でエキスパート・システムを実用化するための短期的研究課題としては次のことが考えられます。まず大規模データが扱え、高速処理できる知識表現系の確立です。また当然言語の標準化が強く望まれます。それから既存ソフトウェア資産とのリンクも重要です。

中長期的な研究課題は、問題の新しい定式化法を見つけることです。たとえば、論理仕様として明記する以前の曖昧な設計思考過程の明確化、事例の活用、部分修正、仕様変更にたいする解決モデルなどです。また現在の DA システムでは設計の階層ごとに閉じたアルゴリズムを使っています。ところが実際の大規模設計ですと、設計階層にまたがった処理が非常に重要です。たとえば、トップダウンに設計を進めてきて、最後にコスト、速度条件などが満たされないことが分かったときに、どこまでさかのぼって違った設計をするかなどです。こういうことを実現するには、設計戦略、設計計画、設計階層間の情報の対応付け、設計プロセスの記憶メカニズムなどを解明することが重要です。

上野 少し質疑をしたいと思いますがどなたかどうですか。高木さんはトイ・リサーチよりはかなり進んだ研究をされているように思えますので……。ゲートの数も 1 万ゲート前後の LSI を実際に作って動かしている……。

高木 正確に言うと全部はまだ完成していません。たとえばこれは乗算器の設計知識で、任意の長さの乗算器は自動的に設計できます。しかし、乗算器だけで

はシステムはできません。他の設計知識を組み込んで、全体をドッキングする必要があります。

上野 医学エキスパート・システムと似たような問題がある……

高木 少し違うような気がします。こちらは、この部分とこの部分を作って、組み立てれば全体が動くというのは割合見えています。問題は各部分を高品質に設計できる方法がよく分からない。たとえば、抽象的な仕様記述から正しくマスクパターンに落とす原理的な方法は分かるが、品質のよい結果を得る方法がよく分からない。各サブ問題ごとに、解決法を研究し積み上げているということです。

上野 分かりました。どなたか発言ないでしょうか。では時間が遅れぎみですので、また後ほどまとめて議論していただきたいと思います。今度は京都大学の辻井先生から機械翻訳にまつわる問題提起をしていただきます。

辻井 京都大学の辻井ですけど、今日は機械翻訳と知識処理というタイトルでお話させていただきます。

機械翻訳は狭義のエキスパー

ト・システム、知識工学のシステムではないわけですが、知識工学が直面している問題と機械翻訳がかかえている問題は共通点があります。多分大規模な知識工学システムを作ったときに、やはり同じような問題が生じてくるのではないかとということで、どういう所でわれわれ機械翻訳研究者が困っているかを、お話しさせていただきますと思います。話の結論を先に言ってしまいますと、機械翻訳、あるいは、知識システムの場合には、プロトタイプシステムというものと実用システムの間、非常に大きなギャップがあるということです。

自然言語処理というのは、現在の機械翻訳のシステムを作る以前からかなり長い間、大学、あるいは研究所で研究されてきていました。その中では、たとえば積み木の世界に限って会話をするというシステムだとかかなり深い理解を示すシステムが作れるとか、あるいは機械翻訳でも、少し対象分野を限定すると非常に自然な翻訳が実現できるということが分かっていた。こういうプロトタイプシステムができるのだったら、それを大規模化すれば実際に実用的な機械翻訳システムができるだろうという雰囲気があったわけです。そういう直感からプロトタイプシステムから実用



システムへ移るという仕事を始めたわけですが、その結果分かったことは、できるということと、本当にできるということの間にはものすごいギャップがあるということです。どういうことかと言いますと、プロトタイプシステムでできるというのは、「こういう知識を入れてやって、こういうふうなフレームを用意しておいてやれば、しかも、うまい具合に予想していたような入力文が入ってきてくれれば、非常にうまく翻訳できる」ということで、原理的に不可能なことではないという話なのですね。それで本当にできるというのは、医療の分野と同じような話なのですが、それがいろんな分野にまたがって、そういう膨大な知識を本当にシステムの中に入れられるかという話です。しかもそれが整合的に動作できるという、そういうシステムの話なわけですが、ここの間には、われわれが思っていた以上に実際には差があったという感じがします。どういう差があったかということですが、それは二つに分けて考えることができます。

まず第1に、多分これは多くの知識工学者の方も感じておられると思いますが、狭い分野を対象としたシステムを足し算的につなぎ合わせるだけでは、広い分野をカバーするシステムは作れない、ということです。機械翻訳システムの場合でも、電気工学の中の計算機工学のしかもデジタル回路の中の文章だけを対象にするようなシステムを作るとうまく翻訳できる。次にちょっと別な分野の所でもシステムを作るとその分野の翻訳ができる。もしそれが可能だったら足し算をしてやれば広い範囲の分野を覆う機械翻訳システムもできておかしくないはずですが、実際にやってみますと、これがうまくいかない。機械翻訳でそれがどういう形で出てくるかということですが、まず分野が広がったということで、単語の数が増えます。これは量的な側面で足し算で解決できる問題に近いということになりますが、もう一つ質的に違う側面が出てきます。これはウィノグラードの挙げている例ですが、たとえば“dark”という単語を考えてみます。その場合、分野限定を非常に強くしておくと、この単語の意味はほぼ一意にしばらくすることができます。たとえば、自動車のデータベースに対して自然言語でやりとりしている場合ですと dark car というところかというのと深い色の自動車、極端な場合には、dark car というのはたとえば紺色か黒かブルーの自動車に限られるとか……そういう意味で dark という単語の意味は、分野を狭くすることによって非常に強く限定することが

できるわけです。ところがいったん分野を広げますと、dark の中には光の量が少ないとか、あるいは気分的に憂鬱な状態になっているとかそういう意味ももちろん出てきますし、それからダークカラーという意味で深い色という意味ももちろん出てくる。こういう問題は、機械翻訳の中で実際の翻訳文中で、つまり、相手言語の表現を選択するときに、どういう単語を使うかという単語の訳し分けの問題になってくるのですが、分野を限定していた場合には潜在していて見えなかった機械翻訳の原理的な問題があらわれてくるということです。この原理的な問題を解決するためには、「文の深い理解」という、現在の機械翻訳システムではほとんど対処していない問題に直面することになります。現在の機械翻訳と「言語の理解」との関係を示す例として、日本語の「おくる」の訳語選択を考えてみましょう。ある分野に翻訳対象を限定している場合、たとえば、「おくる」は、「メッセージ」や「物」を「おくる」場合にのみ使われていて、send と訳しておけば良い、といったことがよくあります。ところが、ちょっと分野を広げて、「人をおくる」という使い方も出てくるとなると、「おくる」は必ずしも send ではなくて take の方を使わねばだめだということになる。「おくる」という単語を send と訳すか take と訳すかというのをなんらかの知識として入れておいてやらなければいけないという事になります。現在の機械翻訳は比較的単純にやっています。物をおくる場合は send を使い、人をおくるときは take を使うという知識（規則）を入力することになるわけですが、少し例文を考えてみますと、そういう単純な規則ではとてもいけないことが分かります。つまり「協議の必要が生じたので太郎を東京におくった」という場合ですと、なにか相談の必要があるから大阪に会社のある人が自分の部下の太郎という人を東京におくったという意味で、「人を送っている」のだけれども、この場合 take と訳してしまうと大変おかしくなってしまう。一緒についていったという感じになってきますから……ところが「夜が遅くなったので、花子を家におくった」という場合ですと send ではおかしい。つまり危険だから家までついていった。だからこの場合は take なのです。こういうのは、ある意味で現在の機械翻訳では全くできない話なのです。これは何を示しているかというと、分野が広がったことによって、「おくる」に send か take かという二つの訳し分けが出てきたんですが、それを分けようと思うとかなり深い理解が必要

であるということになります。

よくエキスパート・システムで shallow knowledge と deep knowledge という議論があるわけですが、機械翻訳の場合も同じでして、現在の機械翻訳というのはあくまで shallow knowledge でやっていて、つまり、「おくる」とどういう単語がつながっているかを調べる程度で訳を決めている。ところが広い分野に適用できて、しかも本当に質の高い機械翻訳を作ろうと思うと、こういう状況の理解という所まで進まないといけない。これに関しては、こういう状況に関する知識をどう表現しておいたらよいかすら分かっていないというのが現状なんじゃないか、それが多分、分野限定のエキスパートを足し算しても広い分野を作れないということ、深い意味でつながりがあるんじゃないかというふうに感じています。これが一つの質的な側面といえますか、狭い分野を対象にしたプロトタイプシステムを広い分野に広げたときの問題点の一つなのですが、もう一つは量的な側面でも難しさが生じてきます。

プロトタイプシステムと本当のシステムのギャップの二つ目は、システム中の知識の網羅性の問題です。現在のエキスパート・システムというのは多分、一言で言いますと deep knowledge というのはとりあえず取り扱わないという現在の機械翻訳とほぼ同じような立場で研究が進められていると思うのですが……。たとえば、機械翻訳の場合で言いますと、「夜が遅くなったので……」というのと「協議の必要が生じたので……」ということとで記述されている状況の差を認識するような知識というのは多分現在の知識表現技術では無理なので、そういう deep knowledge を回避して、一応不十分だけれども、言語の表層の表現に近いレベルで翻訳モデルを作る。そういう表面上にあらわれる手掛りを規則化してシステムの中に入れておいてやれば 80% のパフォーマンスでいいからある程度の機械翻訳をやってくれる。あるいは医療診断をしてくれるということになると思うんですけど、ここでも現在の技術というのはある意味で限界があるかという、そういう話です。

第一の問題というのは、現在の技術をかなり越えてしまっていて当分できないという話なんですけど、こちらの方は現在できるように思われているんだけど、実際にやろうと思うと、まだまだ大変な所がある、そういう話です。実際にある程度のパフォーマンスを持つような機械翻訳システムを作るには、どれくらいの

規則、あるいは、知識があるかをまず考えてみます。まず、知識として思いつくのは、文法規則ということになりますが、現在われわれが作っているシステムですと、3000 くらいのルール数になります。これはまあ、プロダクション・システムのルール数が 3000 くらいだと思っていただいたら良いのですが、これだけだとそれほど大きい知識ベースという感じがしないのですが、実際の機械翻訳を難しくしているのは辞書の部分なのです。辞書の部分がたとえばわれわれのシステムですと 10 万語くらいの単語を入れてやる必要があります。10 万語というとそれほど多くないという印象を持たれる方もあるいはおられるかと思うのですが、この辞書の中に実は各単語ごとの細かな知識というのが符号化されているわけです。ですから辞書とは言いながら、ルールとほぼ同じような役割を果たす。そういうデータがかなり大量にある。たとえば、「部屋に入る」という言い方を英語に訳したときに go into を使うか enter を使うかということがありますが、たとえば go を使った場合は日本語の「に」というのは into という前置詞になるわけですね。簡単な例ですけど……。ところが、enter を使った場合には into は不要で enter the room でよろしい。これは結局 go とか enter の意味をいくら考えても go の場合には into がいるとか enter の場合は into がいないということとは出てこないわけです。そうしますと結局は go や enter という単語ごとに、「into を使え」、「into はいらない」ということをなんらかの形で形で書いておかざるを得ないわけです。同じように、英語の場合ですと、marry という動詞に対して目的格の関係で Hana-ko というのがつけばいいのですが、日本語で「花子を結婚する」というのはちょっと気持ちが悪くて、やはり「花子と結婚する」とする必要がある。これもやはり「結婚する」という日本語の単語が英語の方では object 格で表現されるような、普通だったら「を」で表現されるようなものでも、「と」という表現をとらないとだめだということになるわけです。これは辞書にどういう知識が入っているかということの一端なのですが、言いたかったのは、個人の単語ごとにかかなり細かな情報の記述があるということと、それが 10 万というオーダーで必要になります。そういう数の増大というのが現実はどういうような問題をひき起こすかということなのですが、これもある意味でできることと本当にできることの差になるわけですが、現在の知識工学の議論を聞いてますと、ツールの方にかなり重点

を置いた議論がされるわけですね。こういう知識表現を使えば良いとか、プロダクション・システムで書けば、とかフレームで書けばという議論があるわけですが、そういうツールの話と、それから、それを実際に使って知識を encode できるかどうか、あるいは必要となる大量の知識を網羅的に、しかも、安定に書けるかどうかという問題とは別の話です。ある分野をとったときに、その分野のある仕事をするのにどれだけの具体的な知識が必要かを系統的に整理する方法論がいまのところ全くないわけです。

十分な知識を与えたと思って翻訳をやらせてみると、実際にはいろいろなところで翻訳に失敗する。うまく翻訳できないという所が出てくるわけですが、それがちょうど例外的な症例というお話がありました。が、それと同じことでして、たとえばこの翻訳がうまくいかないのは、この単語にこういう知識が足りなかったからだ、それを入れたらできるようになるという形で、いくらでも例外的な症例もカバーできるようにすることはできます。つまり実験をやってみて、おかしいことが起こると、原理的にできないということではないですから、それ専用の知識を入れることで翻訳はできるようになるわけです。ただその単語がそういう状況で出てくるのは、次のたとえば10万くらいセンテンスを流したときにまたもう1回出てくるだけかもしれない。つまりそのセンテンスだけは訳せるようになったけれども、それがどれだけの範囲をカバーしているかという保証はほとんどないわけです。ですからツールの議論というのは、そのツールを使えばこれだけできるという議論でして、実際にそのツールを使ってどれだけ知識が書けたかとか入っているか、あるいは現実の現象をカバーするのがどこまでカバーできているかとかってというのは別次元の話なわけですね。ですから、そういう意味ではツールの議論というのは「適切な知識を入れれば、原理的にできる」という議論であって、本当にできるという議論にはなっていないわけです。この問題は、知識獲得の問題とか、知識システム用のメンテナンス・ツールの問題とかになるかと思えます。ここでも、やはりそれ用のツールを作れば問題は解決すると、われわれ計算機屋は考えるわけで、われわれも機械翻訳システム開発用、あるいは、メンテナンス用のツール（主としてデバッグ・ツール）を symbolic 上のマルチウィンド機能を使って、実際にどういう過程で翻訳が進んでいくのかというのを表示するというシステムを作ったわけです。大

学の方にいろんな見学の人があるわけですけど、見学の人にこういうデバッグ用のツールを見せると非常によろこばれるわけですね。しかし、実際に機械翻訳を作っている側からしますと、こういうツールはまずほとんど役に立たない。まず画面が非常に小さい。一つの文章を解析しますと、本当にテキストに出てくるような長い文章をやりますと、tree の形がものすごく大きくなって、1部分切りだしてみても、その中の情報を見ているくらいだと使いものにならない。結局デバッグするときもラインプリンタにいっぱい出して、それを丹念に見る方が早い。ですから覇ツール議論で僕自身はかなり気になっているのは、ぱっと見たときには役立ちそうに思えるんだけど、それを使ってみたときにほんとに開発用のツールとして適切かどうかということはかなり問題があるのではないかということです。

あといくつか言いたいことがあったのですが、時間もないので、結論だけ言わさせていただきますと、まずプロトタイプ・システムというのと実用システムの間にはかなり大きなギャップがあるのだということを認識してかかる必要があるのではないかとということ。また、知識工学のシステムであろうと、機械翻訳のシステムであろうと、いずれのシステムでも deep understanding というのはある意味で回避していて、100% 完全性の保証はないシステムになっていること。そのようなシステムにおいては、いくつかの例に対してうまく動くということは、それが最後まであらゆる例に対してうまく動くという保証にはならない。これは、知識ベースシステムが多分今までのソフトウェアとか、今までのシステムとはかなり性質が変わっているということで、たとえば今までのシステムですと、一応処理モデルをプログラムに移すと、そのアルゴリズムが間違いでなければそのプログラムはほぼ100% 確実に動く。あるいはテストケース・ジェネレーションくらいをやってそれでテストしておけば、100% の保証がある。ところが現在の機械翻訳システム、あるいは知識工学システムというのはその保証がきわめて少ないシステムになっているということです。つまり100個くらいの例を見てうまく動くということと、それがほんとにいろんなケースに対してうまく動くという、そのつながりがないということですね。そういうシステムですから結局システムの開発と運用、保守というものが基本的には区別できなくて、知識を実際にコーディングしていく過程と、運用して保

守していく過程とが切り離せない。それから先ほどから言っていることですが、表現できるという機能をもつということ、実際に知識をその中に入れられるとか、整理できるということとはまた別だろうということ。

最後になりますが、これは今までの議論では全然していなかったのですが、小山先生のお話と同じで、現在の翻訳システムというのは、普通の人々が使うようなシステムでは全くない。そういう意味では翻訳のプロが、自分より能力の低いシステムを使ってただ単に処理のスピードを上げるためにだけ使うものです。粗訳を計算機で作ってもらって、それをもとにして、実際の翻訳結果を作る。そういう意味ではプロの翻訳家の知識がすべて入っているとかいうおこがましいシステムでは全くなくて、奴隷のように翻訳家に奉仕するだけのシステムになっている。人間の翻訳家がやっているような翻訳結果を出すまでは、先ほどの「おくる」の例で少し言ったように質的なギャップがものすごくあるというふうに感じています。

上野 どうもありがとうございました。質問ありますか？ あるいは問題指摘などどうぞ。

質問 (最初の方はマイクの状態が悪くてはっきり聞きとれないので要約すると) ……われわれが日常機械翻訳を必要とする場合を考えると、80~90% くらいはだいたい自分の力でできるのであって、本当に必要とするのは残りの10% くらいである。今の send か、take か手紙を書くときにどちらを使ったら良いのだろうかということを知りたいのであって、最後の10% くらいだけ与えてくれるシステムがあればわれわれは残りは全部必要ないわけである。そういうものが本当のエキスパート・システムというべきものだと思うのですが、そういうものが本当に可能なのか、あるいは……(聞きとれない) とそういうように考えてよろしいのかどうかちょっとお聞きしたいのですが。

辻井 私自身は現在のシステムの延長では全く不可能だと思っています。どういうことかと言いますと、機械翻訳システムはある意味で誤解を受けやすいシステムでして……。といいますのは翻訳という言葉が非常によく分かる言葉なんです。医療の話になりますと一般の人にはよく分かりませんから、議論ができなくなってしまうのですが、翻訳というのは普通の人々が日常的にある種の必要性を感じているし、自分なりの翻訳に対するイメージを持っているわけですね。そうすると機械翻訳という言葉が出た途端に自分のイメージ

に合った翻訳を考えて、その「翻訳」をやってくれる機械というふうに思ってしまうわけです。つまり、「機械翻訳」という言葉で、かなり性質の違う翻訳システムがイメージされているという現状があります。私自身、二つかなり性質の違う機械翻訳システムというのがありえると考えています。一つは言葉を換えて言うと、英作文の支援システムみたいなものです。それは実際に情報を作る人、つまりこれから英文を書きたいのだけれど、直接英語を書くのはいやなので、日本語で少し文章を作っていくと英語を出してくれるようなシステム、そういうシステムが一つのタイプですね。英作文支援システムと呼んだ方がよいだろうというタイプのシステムです。それからもう一つは、恒常的に翻訳をしているシステムがあるわけです。たとえばある製品を買ったときにマニュアルがこれくらいある。それを翻訳したいというとき、それを1人1人の翻訳家に割り当てると用語の統一もとれないし、あるいは全体の訳ができて上がるのは1年後になってしまうとか……そういう話があるときに、それをバッチ的に翻訳したい。これは、英作文の支援システムとは全然違うタイプの機械翻訳システムなんです。つまり情報をこれから発信したい人がその場にはいない。すでに書かれた英語のマニュアルや日本語のマニュアルがあってその翻訳をある期日までに仕上げたい。そのときに機械をどうやって使えば、人間だけで翻訳しているより速いかという、そういうことが問題になるようなシステムですね。現在われわれが開発しているのはどちらかと言うと後者の方です。つまり情報の生成者は翻訳の現場にいない。そういうタイプの機械翻訳なのですね。で、今ご質問にあったのは前者、英作文支援システムだろうと思うのですが、その場合にはシステム設計の考え方がかなり変えられると思うのです。これから英語を書こうと思っている人は、システムがいくら質問を聞いてきても丹念に答えられるわけですね。たとえば1時間計算機とやりとりして、計算機が分からない所、これはどういう意味で言っているんだと質問するとそれをパラフレーズしてやるとか、自分の意図まで含めて計算機に教えてやるとか……そういうことをやることができる。これは時間がかかりますが、でも一応はちゃんとした英語を出してくれる。そういうシステムというのは、今言ったみたいなのわれわれが作っている機械翻訳システムでなくて非常に interruption の激しい、しかも作る日本語の文章というのは必ずしもちゃんとした日本語の文章になって

いなくてもよくて、ある意味で自分の意図さえ表現してあればいいようなものですね。しかも対象分野を少し狭めるという、そういうシステムの可能性だったらあると思います。ただ現在市販されているシステムだとか、現在メーカが作っているシステムだとかはそういう用途には全く合わないという感じですね。今言ったような英作文支援システムができるというのは、まだ誰も研究していませんし、どういう形でそれが実現できるというのは分からないのですが、そういう発想にたったシステムならひょっとしたらできるかもしれない。それは、原文の準備段階で人間の援助をうけるという前提のシステムであると思うんですけど……。

上野 どうもありがとうございました。次は日立の元田さんをお願いします。

元田 私は現在、日立の基礎研究所で AI 基礎研究に携わっています。以前に原子力の研究に従事しておりまして、いくつかのエキスパート・システムを開発したことがありますので、そのときの経験を基に話させていただきます。すなわち、computer scientist



というよりは nuclear engineer として、種々の難しい問題を抱えておりましたので、そのとき感じたことも盛り込みながら意見を述べたいと思います(表-1に課題を要約)。

先ず第1番目に感じていることは、理解の度が非常に浅いということです。私が知識工学を勉強し始めたきっかけは、事前に想定し得なかったような事故や異常が発生した場合にでも正しく対処してくれるようなシステムを構築するのに、この技術が使えるかもしれないと考えたからです。たとえば、有名な TMI の事故ですが、事故当時オペレータは自分では正しいと判断して種々の操作をしたわけですが、結果的には間違っていました。ところが時間をかけて原因を調べてみると当然のことながら、理解できないようなことはなにも一つなかったわけです。

すなわち、プラントに関する深い知識や物理の基本原則からときほぐせるような仕組みがあれば、未経験の事態に対処可能なはずですが、これは表層的な知識のみを用いた推論と区別して深い推論 (deep reasoning) と呼ばれているものです。私の研究は深い推論のはしりのようなものでしたが、最近、関連した技術と

表-1 エキスパート・システムの課題

- 1) 理解の度が浅い
 - 表層的な知識を使った問題解決 (「浅い知識」対「深い知識」)
 - 人間の理解は定性的推論によるものが主
 - 知識の多層表現による対象世界のモデル化
- 2) 扱える専門領域の幅が狭い
 - Knowledge-rich, Problem-solving-poor
 - 不完全な情報からの推論, 信念の翻意に弱い → 非単調論理, 知識の整合性保持
 - 常識がない → 1) である程度カバー

人間との協調 ~ 人間: 非専門家, 常識人,
機械: 専門バカ, 非常識
- 3) 自己成長能力がない
 - 経験から学んで知識を増やし体系化する自己組織化能力がない (学習, 帰納)
 - 知識の系統的収集法が未確立 (人間相手は難しい) → 発想を促すような人間機械協調システム
- 4) 使用していると飽きがくる
 - 同じ質問に対していつも同じ反応 → ユーザモデルの構築 (ICAI)
 - 自然言語処理 (理解)
 - Direct Manipulation
 - 計算機を意識しないで直接問題に入り込めるインタフェース
- 5) 記号処理に重点が置かれ過ぎている
 - 記号処理と数値処理の融合
- 6) 処理速度が遅い
 - 工夫した PS (別弁ネット, Rete Algorithm) でもアルゴリズム的な記述に比べて 2 桁遅い
 - 並列処理とハードウェアの性能向上に期待
 - 人間の情報処理とは本質的に違う?

どうすればよいか?
- 1) 地道な基礎研究
 - 人工知能から知識工学 → 反省 → 知識工学から人工知能へ
 - ポイントを抑えた小さな良い問題の発掘 → 要素技術の研究開発
- 2) 良い問題を選んだ応用研究
 - 従来技術との組み合わせ

して定性的推論がよく話題に上がります。われわれ人間があることを理解しようとする場合、ものの動作原理を支配する法則を定性的なレベルで捉えて判断していると考えられます。数値レベルまで落とさないで人間が判断できるということが、問題解決のスピードを早めています。しばしば経験したことは事象と対応操作を直接関連づけて表層的な知識として効率的に利用できる形で記述し、他人への説明や、動作原理の理解には前述した定性的な推論に適切な形（たとえば、定性的な微分方程式）で記述し、具体的な大小関係などの定量的な情報まで使用しないと分からない場合には数値解析に適した形（たとえば、偏微分方程式）で、また、もっと根本的な物理法則まで遡ってときは必ず必要のある場合には関連する物理法則(first principle)を記述した、多層の知識表現モデルが必要であると考えています。問題解決というのは一層の方から使える知識をどんどん使い、分からなければ自然に下層に落ちてより深い知識を使うのが自然であるし、表層的な知識は、逆のプロセスで、下層の知識をコンパイルして生成されると考えることが自然です。

第2番目は扱える専門領域の幅が狭いということです。これは「知識こそ力なりき」が強調され過ぎて、knowledge rich であるが problem solving poor なシステムになっているということです。すなわち、これだけは分かりますといったことに関しては強力な問題解決能力があるけれども、ちょっと外れたことは分からないということです。これに対しても深い知識を使用すればある程度はカバーできると思いますが、いわゆる常識を埋め込むのは非常に難しいと思います。一つの考え方として、人間は非専門家であるが常識はある、機械は常識はないが専門バカであると割り切り、人間と機械が協調して問題を解決していく方向を探るのも面白いと思います。また、常識に関連して、人間は情報が不足していても適当な仮定を置き推論を先に進めることができます。後で、不具合が発見されれば、過去の結論を翻意することはよくあることです。実際の問題においてはこのような非単調な推論をせざるを得ないことが多く、設計などはこの典型的な例です。この問題に関し、変電所のレイアウト設計で知識の整合性を自動的に保持する機構を作ったことがあります。

第3番目は自己成長能力がないということです。人間は生まれてから20年かかり多くのことを学びます。その過程で知識を増やし、自立的に体系化していきま

す。脳細胞の応答速度はミリ秒のオーダーですから、処理速度という観点からは計算機の方が格段と優れており、人間の20年間の経験はあっという間に計算機では達成できてしまうことになります。しかし、現実にはそうはいきません。経験から学んで、知識を構造化していくことができないと、問題解決能力の向上は期待できません。また、自己成長は無理にしても、知識を系統的に収集する方法を確立する必要があります。私の仲間の例ですが、リスクマネジメントのエキスパート・システムを開発した際に、社内の有識者から知識を集めることに大変苦労しました。人間が相手ですと、失敗したことや、不利になることはなかなか出たくないし、フォーマットを指定しても書いてくれません。発想を促すような人間機械協調システムの研究も重要だと思います。よく知識は断片的に書けばよいといいますが、勝手に書いても決してうまくいきません。どれとどれがどうマッチするということをすべて考慮して書かないと正しく働きません。

第4番目は使用していると飽きがくることです。よいゲームはなかなか飽きませんが、普通のゲームだとすぐ飽きてしまいます。ほとんどのエキスパート・システムは使用する側がどの程度のレベルかということを考えていないので、同じ質問に対しても同じ反応をします。人間は相手の目を見て話します。ICAIに関連してユーザモデルの構築も必要です。言うまでもなく、自然言語によるインタフェースもきわめて重要な要素です。またインタフェースに関し、今後、非常に重要になると考えられるものにダイレクト・マニピュレーションと言われているものがあります。これはノーマンが提唱しているものですが、計算機を意識しないで直接問題に入り込めるインタフェースの重要性を説くものです。すでにパソコンのゲームでこれに該当するものは（たとえば、ピンホール・ゲームの設計）ができています。

それから第5番目は記号処理に重点が置かれ過ぎているということです。すでに、われわれは数値シミュレーション技術に関しては相当の蓄積があります。問題解決においては記号処理による推論に加え、これらの技術を駆使させる必要があります。原子炉の故障診断システムではこれは不可欠で、リスブ処理系に手を加え解決しました。

最後に第6番目として処理速度が遅いということです。弁別ネットやリット・アルゴリズムを用いて工夫をしても、処理手順が分かった後、フォートランのよ

うな手続き型言語で記述すれば2桁速くなります。ハードウェアの性能向上と並列処理技術に期待できるのですが、その場合でも人間の情報処理の仕方とは本質的に異なっていると思われます。

以上、6点ほど私が課題と感じていることを述べてまいりましたが、それではどうすればよいかというと、やはり地道な基礎研究を続けていく必要があると思います。人工知能の歴史は30年くらいあるわけですが、知識工学という言葉が世に広まり、大き過ぎる期待に技術が追いつかなくて反省すべき点多々あるように思います。人工知能の基礎に戻って、深く腰を落着けて未解決の問題に再度挑戦すべき時期がきています。計算機環境の方もそれが可能な状況にやってきました。そのためには小さくてもポイントを抑えたよい問題を捜すことが大切です。あまり大きな課題を取り上げて泥沼に入り込むだけなのでこの点はよく考える必要があります。それではこれらの課題が解決されるまで役に立つものはできないのかというと、そんなことはなく、よい問題を選び従来技術とうまく組み合わせれば期待される効果を出すことができます。私のかつての同僚の研究である3次元配管のルーティングの問題などそのよい例です。具体的な事例については、ディスカッションのときに時間がありましたらご紹介いたします。

上野 どうもありがとうございました。パネラの最後となりますが東京大学の石塚先生お願いいたします。

石塚 エキスパート・システムという場合、エキスパートの能力が注目されがちですが、私はもう少し広くとらえて新しい情報システムを作っていく上での中核技術と考えようと思います。その意味では「知識システム」といった方が適切だと思っています。実用に近づけようとすると、表面的な知識だけで不十分が多々あります。今日は、課題を話せということなので、自分がやっていることに関連して話したいと思います。だいたいの内容は予稿にあるように、次のようなものです。

- 1) モデル、深い知識の必要性
- 2) 直感が欲しい
- 3) 常識をもたす方法
- 4) 学習機能
- 5) 設計問題のパラダイム

6) 認知科学は期待できるか？

1) に関しては、個々の問題領域、知識というようなきれいな部分で終わることなく、ドロドロした部分にまで立ち入らなければならないと思います。私は、3次元のコンピュータビジョン関係もやっていますが、どの方向から見ても分かるようにするためには、見ようとする対象のモデルというものをもち必要があります。ではそれをどう表し、どう利用するのが問題になり、これはコンピュータビジョンの世界に立ち入ると、永年の課題になっている事柄です。一般的に深い知識、モデルというのは簡単でも、個々の領域で効果的に表現、利用する技術となるといろいろ課題が多いというのが実感です。3次元ビジョンのモデリングでは、グラフィックのモデラの使用が直接的であり、それも重要と考えていますが、現在私たちが利用しているのはもう少し概念的という抽象レベルの、物体表面のトポロジ的なモデルです。それをどのように覚えさすかということ、フレーム的な表現を基礎にしています。主に、構造表現に適すると理由によります。ですから、モデル表現自身は個別の領域のかなりドロドロした部分に立ち入って考えなければならないけれど、その表現には知識工学における知識表現の考え方、手法も役立つ部分が多いというのがわれわれの立場です。それと、深い知識に立ち入ると要するに細部まで考えなければならないということで計算量、処理量も多くなって大変だが、そこを要約したような浅い知識、大局的な浅い知識も組み合わせることによる効率化が重要ということで、われわれもそのような考えからビジョンシステムの開発を進めています。

原子プラントは元田さんの方が専門なのですが、ICOTで元田さんとも一緒に次期原子プラントに必要なAI技術というものを検討しました。このときも深い知識の利用を課題としてあげており、原子炉シミュレータみたいなものと浅い知識をどう結合するか、さらに深い知識を記述すると浅い知識を自動的に生成する、すなわちコンパイルする技術が必要だということ指摘しました。深い知識をコンパイルして浅い知識にするというのは、通常は浅い知識で動いた方がずっと効率が良いからです。そういう技術が必要だというのが一つの観点で、もう一つ私が強調したいのはknowledge-guided simulation technologyです。私のいる研究所にはシミュレーションをやっている先生もいますが、大きなシミュレーションが走るだけではなく、細かい断片的なシミュレーション・モジュール



が知識にドライブされて走りたい技術というのはおもしろいだろうと言われてますし、私自身もおもしろいだろうと思います。定性的推論 (qualitative reasoning) が話題になっていますけれど、どうも浅い知識によるシミュレーションという気がします。研究の状況を見ていると、単に結果を説明する程度の役にしか立たないような気がします。近似はあるとしても深い知識による Knowledge-guided simulation の方が本道のような気がします。

2番目の直感が欲しいというのは、要するに三段論法だけにより答えが出える今の知識システムでは、当たり前な答えだけが出てくるだけなので、飛躍するとか、困ったときに気転をきかすような知的メカニズムがないかということです。一つのアプローチに類推の研究がありますが、われわれのところのフェジィ論理に基づく prolog である prolog-ELF による例をお見せしようと思います。これは世界の 35 都市の緯度、経度、高度、海からの近さ、年平均気温のデータが入っており、いくつかのデータは欠損データです。たまたま横浜の年平均気温が欠損していますが、それが必要のときどのように推測するかというと、近い都市、たとえばワシントン市の緯度が横浜と近い、海からの距離も近いということで、その年平均温度から横浜の年平均気温を推測します。ただし、ぴたりと同じ緯度やその他の条件が一致するものがないので、緯度が離れたり高度差があると、推測の真理値が低下してくるようになります。緯度や高度に依存して温度がどう変動するかという一般知識を入れておくと、困ったときにバックグラウンドの知識として働いてくれる訳です。

常識についてはマッカーシー先生はcircumscriptionを主張しており、最近、1階述語の枠内で計算可能な範囲というのも見い出され、少しずつ実用に活かす目途が見えてきたように思います。しかし、マッカーシー先生の circumscription は消極的というか、保守的な常識の生成というところがありますが、もう少し常識の範囲内で知識を拡大解釈する、一種の帰納推論的な常識のメカニズムも必要と思っています。(後に ICOT の有馬氏より、そのようなメカニズムの一種として ascription の話を聞いた)。

診断とか解析問題のパラダイムは大大分かってきた気はしますが、まだ知識型設計はこれだと言えるような見本となるパラダイムは見い出せません。もちろん、設計というのは何かを生成してそれを検証するサ

イクルがある訳で、検証には演繹的推論を有効に使えるのですけど、それに生成を含めてどうやって包括的に形式化するかということです。なんとかその確立へ向けてアプローチしたいということで、われわれのところでも LSI パターンを対象にして知識型 CAD を行っています。われわれが現在とっているのは機能ライブラリを中心としたものです。要するに設計というのはやはり基礎となるのは部品を集めた良いライブラリがないとだめだと思うのです。画像処理のプログラムを自動合成する研究が進んでいますが、実用的なレベルでいい線が出てきているのは、前提として SPIDER というサブルーチン・パッケージの存在があります。われわれも部品を揃えたライブラリを基礎にし、“頭がいい部品群”を作るということで研究を行っています。もう少し具体的に述べると、成功例を具体的な部品としてライブラリに蓄えるだけでなく、いくつかの部品を拡大解釈可能なパターンデータ形にして蓄える。具体的にはいくつかの値を変数化して記述する。通常はデフォルト値が使われるが、必要に応じて変形が許される。あまり勝手な変形は困るが、それは制約条件に相当する設計ルールでチェックすることになっています。このような頭のいい部品群による NMOS-LSI セルパターン機能ライブラリというものを作っております。オブジェクト形式でできていますので、寄せ集めて大きなパターンを作っていくというスタイルをとります。

最後に“認知科学は期待できるか?”ということですが、私の立場はあまり認知科学には期待していないでやっていこうというところですが、これについては上野先生の方からなにかあればまた議論があると思います。

上野 どうもありがとうございました。まず全体の議論の前に、最後の2人のパネラに関する議論がありましたらしていただきたいのですが、どなたかございますか。それでは先ほど反論があると言われていた辻井先生にお願いしたいのですが。

辻井 特に今のコメントに対する反論ではないのですが……。もともと心理学と工学というのはパラダイムが違うのですから、おのおの違った方向を目指しているのは明らかなわけですね。ただ、最後の2人の方の議論にあったように、知識工学がデッドロックというところですが、行きづまり状況にあるのが分かったら、つまりどこかで基本に戻らないとだめだとか、より基礎的な問題で難しい問題を解くべきだとい

うお話がちょっとあったと思うのですが、そういうときに心理学の人が持っている知見というのは、聞くべきものがあるだろうと……。それを直接工学として評価したら、工学としての意義はないということになると思うのですが、われわれがどちらの方向に進むかといったときに、たとえば知識工学から人工知能にもう1度戻る必要があるというコメントもあったと思うのですが、その方向とある意味で似た方向だと思うのですが、もう1度、心理学的な知見を見直してみる必要があるのではないかという気がしています。もう少し具体的な話では、自然言語処理なんかやっているのと、特に人間は素晴らしいという感じがするのですが、少し視点を変えると、計算機の方がものすごく、あらゆる可能な分の解釈を探し出すことができる、とも考えられます。人間の場合には、どこかで考えられる可能性があることでも考えないで済ましてしまう部分がある。そういう意味では人間が素晴らしいという言い方をするかわりに人間が不完全だという言い方も当然できると思うのです。心理学の人たちは、どういう点で人間の情報処理に偏りがあるかも考えているわけですね。私自身は、このような研究成果は、工学的なシステムにも積極的に入れるべきだろうと、そういう感じをもっているわけです。ですから直接今の反論ということでないのですけど、どの方向に進んでいくかというときに示唆を与えてくれるという意味では認知科学というのもそれほど悪いものでもないのではないかという気がしてます。ただ彼らの理論を直接工学として評価するかどうかというのは別問題だと思いますが……。

諏訪(電総研) 今日のパネルというのはつまり知識工学というものがどういう位置にあって今後どうするべきかという議論だと思うのですが、それを考えると、人間と計算機の役割、その果たすべき持ち場がなんであるかというのを明確にする必要があって、そこから話が始まるのではないかと思います。そうしたときに、今日の議論の中でもエキスパート・システムというのは不完全でいろいろな問題があるという指摘がありましたが、そういうシステムというのは、完全であればあるに越したことはなくて、もしもすべての可能性をサーチできるのであれば、すべて調べてみてその中でベストのものをとるに越したことはないし、それから扱う対象のモデルというのがきちんと記述できれば、それをきちんと記述しておいて、その上でロジカルに結論を引っ張り出すようなメカニズムをきち

んと作っておくに越したことはないわけです。ところがなぜ知識工学というのが出てきたかということを考えると、扱おうとしている問題をきちんとブレイクダウンをして、計算機にのせることが非常に難しい問題があり、これをどう取り扱うかというところに問題意識があるのではないか。そうは言ってもそういうシステムを作るのは結局人間である。じゃそういうシステムを作るのを人間がどういうふうサポートしたらよいだろうかと、そういう見方になると思うのです。そうすると表層的にはツールを作るとよいということですが、もっとAIの立場に返ってみると、知識の表現だとか、人間がコーディングする作業とか、やりやすい方法を考えなければならない。

私も認知科学に非常に興味を持っていて、問題をブレイクダウンしてシステムを作るのは人間である。そこで計算機と人間との役割分担、そして計算機が人間を助けるというようなメカニズムを作ろうとしたときに、やはり人間のいろいろな認知的な過程というものを知っていなければならない。そういうものを知ろうというのが、工学的な発想からの認知科学への期待であり、それを満足するような方法論を探すのが、われわれの役割だと思うのです。

心理学者は心理学者でやはりモデルを作ろうということで、最初の石塚先生の議論にちょっと戻りますけど、あまりにも表面的に認知科学というものをとらえていらっしゃるような気がするのですけど。いや、それは日本の認知科学者が悪いのかどうか知りませんが、やはりそこにはご指摘のようにきちんとしたモデルというものを作らなければいけないという認識はもちろんありまして、そういう方向でいっていると思います。じゃ、モデルをどうやって作るのかという問題がきつと出てくると思うのですけど、まだ良い方法は提案されていないわけですが、たとえばプロトコルアナリシスなんていう従来の方法の限界というの指摘されている。そうすると先ほどの上野先生が、人間の発想を促すような役割の話もありましたけれど、逆にあるモデルを計算機の上で作ってみて、それと人間との対比において、そこでモデルを精細化していくようなプロセスで、言葉では表せない、プロセスをモデル化するような試みもこれからなされていくだろうと思いますし、やはりこれからの人工知能、あるいは知識工学の基礎を作る上で、認知科学のアプローチというのは、大きな柱の一つではないかというふうに私は思っております。

上野 認知科学についてほかにまだ発言があるんじゃないでしょうか。

小山 認知科学とディープレベル・ノーレッジという話も出てきておりますが、これに関連してコメントを述べさせていただきたいと考えます。そもそもディープレベル・ノーレッジが重要であるということで議論が進められておりますが、この問題を考えるにあたって、たとえば一つの表層知識があるとして、単純にそのディープレベル・ノーレッジを探すというようなアプローチで良いのだろうかという疑問があるわけです。おそらく単純にそのようなことをやりますと、現状でさえ表層知識の管理にきゅうきゅうとしているところに、さらにこれに数倍する膨大な情報をすべて管理していかなければならないということになりかねません。そこでディープレベル・ノーレッジというものが本当に有効なものであるためには、そこに一種のモデルとしての働きがあり、単一の表層知識にだけ結びつくのではなく、かなり広範な現象を説明できるだけのバックグラウンドとなり得るものでなければならぬという感じを持つわけです。そこで認知科学といいますか心理学との関連ということを見ると、ここで行われておりますさまざまな現象に係わる事実の収集ということの重要性は理解できるとしても、どうもここでの興味の中心は合理的な推論というよりはそれ以前の問題にあるような感じを受けるわけです。ところが先ほど申し上げた問題意識からすると、現在の知識工学の中でディープレベル・ノーレッジについて検討すべきことは、まさにこの合知的推論のモデルというものに関するものであり、ここでの興味の対象という点からすればむしろ哲学に近いところにありそうな気もするわけで、将来的には人工知能ないし知識工学と認知工学との関係は重要であるとは思いますが、現時点ではまだ係わりとしてはそれほど密接なものとはなり得ないのではないかと思うわけです。

辻井 同じようなコメントになる可能性があるのですが、このパネルでいろいろ議論されたときに、深い知識というのが「ある」という前提があるのですが…。たとえば回路とかある種の制御系みたいなものを記述するような数学モデルですとか、かなり定式化できるような知識がどっかにあって、それを実際には使いたい。しかしそれを直接使うと、ある意味で計算的な不可能性とかそういうやっかいな問題があるので、少し浅いレベルで議論するという、そういう論の立て方というのがかなりあったと思うんですが、自然言

語、多分医療の方もそうだと思うんですが、そういう深い知識そのものがない、あるいは定式化できてないという、そういう分野もあるという気がします。知識工学のそもそもの出発点もそこにあったわけですね。それが広がって、いわゆる深い知識もあるんだけれど、そこで困っている問題を知識工学で取り扱うようになってきたと言うだけで、深い知識は確かに非常に大事なわけけれど、そこに確固として存在して、そこには行けないから浅い知識で、という発想だけではできないような、そういうシステムが知識工学が対象とする分野にはあるのではないかという気がしています。それから議論がちょっと変わってしまうのですが、認知科学についての僕の好意的なコメントは、すぐに役に立つという意味ではありません。小山さんがおっしゃられたように、合理的な推論からはかなり離れたようなものに心理学が興味をもつというのは自然なことですから、つまり合理的な推論でもって説明できないような人間の心の動きという所に興味をもつというのが自然ですから、それがすぐに現在の知識工学の問題を解決するようなものを与えてくれるという、そういう感じは僕自身もあまりしません。ただある意味で広いマンマシン系を考えると、計算機が僕らと全く違った思考形態で推論していたら困るわけですね。そういう問題とか、あるいは自然言語処理なんか考えてみると、本来的に人間の内的な処理ですから、そういう点での期待があると言っただけで、たとえば5年後にある認知科学の成果が出て、それでもって知識工学の方がドラスティックに変わるといった、そういう性急な変わり方を期待すると、むしろ間違っているのではないかという気がします。

元田 自然現象に関してはディープノーレッジはあると思います。ただ、物理学の歴史を見ても仮説と検証の繰り返してして、具体的にどう表現するかが問題です。いったんモデルができると、そこからは演繹的にいろいろな事実をとき明かすことができます。しかし、そもそも与えたモデルが悪いと正しい推論はできないわけです。そのディープノーレッジを与えるのは今の所人間ですが、システムが自分の振る舞いを見ておかしいと思ったら自分で自分を修正するようになるところまで成長して初めて真のディープノーレッジが扱えることになると思います。

高木 普通ですと工学の場合モデルがあるのですが、実際に設計をしている人はそのへんはもう無意識でやっていると思うのです。私が最初知識工学がいい

なと思ったのは、その無意識にやっていることを無理やりクリヤにしようとする態度です。もともとモデルがないところをクリヤにするのは知識工学でも無理ではないかと思います。クリヤの仕方にどういうタイプがあるかということをたくさん揃えて、「あ、私は今までこういう枠組みがあったのにそれを飛ばして考えていた。飛ばしていた枠組みをきっちりうめれば計算機はちゃんと動く」と……そういうのが知識工学の役目ではないかと思います。

ほかにコメントありませんか。……志村先生いかがですか。

志村 僕の考えでは、ディープノーレッジというのはだいたい定義がどうだっていう問題がまずあると思うんですね。人間がシステムに与えられるものは、ディープと言わなくて普通の知識と考えてよいのではないかと思うのですが、それともう一つ、このパネル全体を通してかなり問題が哲学的になっている気がするのです。バスに乗る人は運転技術を知らなくてよいわけですし、運転する人は別にエンジンの作り方や修理の仕方を知らなくてもいいわけです。それと同じかどうか分かりませんが、知識工学でエンジニアがすべきことは、たとえば医者の特権を犯すからどうかとかこうだとかいう話をしてもしようがないのではないかという気がするわけです。そうすると、エンジニアができることで1番大事なことというのは、そういうことを踏まえた上で、十分なツールを提供し得るかかどうかということ、これをまず考えなければいけないのではないかという気がするのです。今日のパネルの話を聞いてみますと、どうもツールなんかどうでもいい、ただ知識をどうやって埋め込むとか深い知識をどうするかというふうな議論になってしまっている気がするのですが、その辺はどうなのかを、ちょっと上野先生代表して一言答えていただければ……。

上野 私は工学というものは最終的には普遍的な手段を提供することですから、こういうような概念についても、ツールの形にまとめ上げて提供することだと思ふのです。ですから私はツールをかなり努力しながら作っています。ツールの開発をしてみますと、いわゆるAI的でない所までやらなければならないことになる。くだらないと思われる所まで……。ところがそこまでやらないと使いものにならない。よく石塚先生は日本の大学では使えるツールを作らないとおっしゃってますけど、私のところでは使えるツールを作るためにかなりの努力をしているわけです。私は、ツール

を作ってアイディアを実験しないと、地についたアイディアの評価ができないという考えを持っています。それが私のアプローチであるんです。

とにかくツールの形にしなければ、人は使えないわけですね。ほかの工学技術だって誰でも使える形態に持っていってそれで技術が向上してきたわけですね。そのプロセスの中で分からないことをいろいろ考えながらつきつめていくことになります。ある人は認知的心理学的側面をヒントとして考えますし、ある人はすでに存在するいろいろな技法をうまく組み合わせて使えるような環境を作る。少なくとも全体を作ってみないことには、優れていると思ひ込んでいる技法の正しい評価は絶対にできないし、ツールとしてまとめあげるには多くの機能を切り捨てる決断を必要とし、ユーザからのフィードバックをとおして、技法ばかりでなく概念そのものに対するいろいろな知見を得ることが出来ます。一般に、日本の大学ではこのような努力が欠けていると思います。いずれにしろ、最後には汎用で有用なツールにならないと工学とは言えないと私は思います。

先ほど石塚先生が最後に話があると言われましたので、どうぞ。

石塚 上野先生は最初にエキスパート・システムの定義を話されたのですが多少異議があります。上野先生はエキスパートの知識が含まれているというのを相当強調していたと思いますが、私はそれはエキスパート・システムを開発する人に誤解を与え、ディスカレッジするのではないかと思うのです。私も建築物の被害診断のシステムを最初はエキスパートの知識に依存して作成したが、実用に近づけよう（たとえば保険のための査定システムとする）には、基礎となるテキストの知識、スケジュール的な知識が欠かせない。エキスパート・システムといっても、9割以上はテキスト的知識だと思うのです。ですから、定義を聞いたときだいたい異論があったわけです。

上野 実は最後のつもりだったのですが……。私がどうしてああいうことを強調しているかといいますと、少なくとも初めの頃のエキスパート・システムの考え方はあれであったと思うのです。その後だんだん変わってきたことは認めます。先ほど言われました石塚先生の、大部分がテキスト・ノーレッジでやっているのだというのは私も同感でして、私のところで作っているものとか、あるいは共同研究者がやっているシステムも、テキスト・ノーレッジ・レベルのものが

ノーレッジの9割ぐらいです。問題によるのでしょうけど……、つまりうまく問題を選べば、うまく働くようですね。ですから決してほかのやり方がダメだということは言うつもりではなかったのですが、あまりにも手法万能的に宣伝され過ぎているから、本来の考え方はこういうことであるということを強調したわけです。

また、そういうことをいつも頭に描いて、今やっているのはそうしたことであるというのを理解しながらやらないと、たとえばツールを手に入れて何かやろうとしたときに、期待と違う結果となったときそれがむしろ発展の障壁になる。そんなのでいいのだという容易な考えが出過ぎますとかえって進歩が阻害されると、むしろ私は逆の意味の進歩の阻害を心配していて、基本的な問題を何回も指摘し続けているということなんですけどね。

石塚 その前提になっているのは、エキスパートの知識か、エキスパートではないテキストの知識かというのがあるのですけれど、その区別というのは私なんか画像理解やっているとよく分からないですね。画像なんか人間誰でも分かるわけで、自分が画像理解のエキスパートだと思っているわけではない。ですから、私はエキスパート・システム誌のエディタなんかも引き受けているのですが、エキスパート・システムというのはなにか今までのシステムにはない柔らかな感じがかり名前が非常に魅力的だったから拡がったけれど、そう狭くとらえるのでない方が適切だと思います。形式から言えば知識システムと言えると思います。

上野 少し曖昧になってきましたが、まあ手法が非常に有用であれば、本来のエキスパート・システムの概念と違うところに使われたって別にかまわないとは思いますが……。しかし画像では、たとえば、胃癌の検診のための画像理解システムは完全な意味のエキスパート・システムですね。医者でないと分からない知識で理解しますから、ところが一般の画像を認識するというのは、それは明らかにエキスパート・システムでない。ただし、それをめくじらをたててこれは違うというほど堅い考えを持っているわけではありません。

石塚 一つは初期の人工知能から知識工学へ移ったときの契機が、エキスパートの持っている知識が比較的陽に書けるだろうという感じがあったと思います。ですから、画像とか自然言語の処理はむしろ難しいと

とは確かだと思います。エキスパートの知識でないから表現がしやすいかというと、そうではない部分があるというのは注意しておく必要があると思います。そういう意味で、エキスパートよりノンエキスパートの知識を入れるのは本当は難しい話であるというのが一つのコメントです。

高木 難しいと言った場合、科学の難しさ、人間の難しさと、その真中に知識工学の難しさがあって、その三つがゴチャゴチャになっている。たとえば医者の診断の難しさは科学の難しさだと思うのです。科学の難しさに認知心理を用いても多分解けない。そのへん人によってばらばらになっているので、もっとクリヤにしたほうがいい。

もう一つはツールの話です。エキスパート・システムのツールは言わばハードウェアなのですね。そのツールをどういうふうに使ったならば、どういふことができるかという難形がありません。そういった使い方（ソフトウェア）がそろわないと、（多分その辺が一番難しいでしょうが）普及はなかなか難しいと思います。

小山 つまらないことなのですが、先ほどの高木さんのお話で医療の分野におけるサイエンスの難しさというお話があったのですが、この問題は確かにあると思います。しかしそれだけではないという気もするわけで、サイエンスの話と言って良いのかどうかわからないところを医者が実際にやってしまっているということがあるわけです。ここにはやはり何かがあると考えざるを得ないところがあって、そのあたりの所が現在考えているような知識の表現と運用という枠組みからはみ出しているような気もするわけです。

上野 今日の議論というのは皆さんご存知のようにだいふ本音のところの議論に近いものになったと思います。最後は多少対立しましたが、新しくいろいろな可能性を持った学問分野では、むしろ研究者による考え方の違いがあって当然だと思います。本日の討論の趣旨の一つはここにもありまして、むしろ意見が合う方がおかしいと思います。これからこの勢いで議論すれば、もっと勉強になると思います。

少し尻り切れトンボでもありますが、まあ何かの参考になったであろうと信じています。今日はこれで終わりたいと思います。どうもありがとうございました。