

ニューラルネットワーク内包モジュールによる状況認識システム

島川 博光 宇佐美 照夫

三菱電機(株) 産業システム研究所

鉄鋼プラントのような大規模プラントをより少人数で運転できるようにするためには現在何が起きているかが正確に認識されなければならない。状況認識によりこれが達成できる。状況とはプラントにおける機器の状態遷移である。状況認識は状況を引き起こしている本質を同定する作業である。本論文では、状況を形式的に表現するための新しい枠組が提供される。擬人化された計算単位であるモジュールが分散型の状況認識活動のために提案される。これらは我々が現在開発中の状況認識専用言語SOULの基本概念を形成している。SOULでは状況認識システムを構築するために3種類のモジュールが用意されている。なかでも専門家モジュールは、特定の状況がおこったかどうかの判断基準を表現したニューラル・ネットワークを内包することができる。このニューラル・ネットワークはオペレータが経験上している現場特有の知識を反映するようトレーニングされる。

Situation Recognition System
Based on Modules Containing Neural NetworksHiromitsu Shimakawa and Teruo Usami
Industrial Electronics & Systems Development Laboratory
Mitsubishi Electric Corporation
1-1, Tsukaguchi-Honmachi, 8-Chome, Amagasaki, Hyogo, JAPAN

What is going on must be recognized so that a large scale plant like a steel plant can be operated by less persons. Situation recognition is proposed for this sake. A situation is the state transition of machines in the plant. The situation recognition is an activity to identify a substance which causes the situation. This paper presents a new framework to formally represent situations. Modules which are personified calculation units are proposed for the cooperative situation recognition activities. We develop a new programming language SOUL. SOUL is specific to the situation recognition, and is based on the framework and the modules. In SOUL, three types of modules are prepared to construct a situation recognition system. Especially, specialist modules can contain criteria to judge whether specific situations have occurred. The criteria are represented by five layered neural networks, and can be trained so as to reflect field specific knowledge.

1. はじめに

鉄鋼プラントのような大規模プラントシステムの生産性を向上させるためには各機器をもっとも望ましい状態に保つことが大切である。これは状況認識により達成できる。状況認識により、異常なプロセス徴候や機器のふるまいといった状況が正確に認識でき、したがって、故障やオペレーション・ミスといったようなこれら徴候の本質的原因を同定することができる。

このような状況認識システムを構築するには2つの問題がある。第一に、複数の機器が同時に観測されているときには情報が多すぎ認識に要する作業はあまりに複雑になってしまう。第二に、現場における各機器はその使用形態に応じてさまざまな特異性を呈する。特定の状況が発生しているかどうかの判定基準はこの特異性を反映するものでなければならない。

本論文では、まずデータ抽象化により状況を形式的に記述する新しい枠組を提供する。複雑な状況認識の作業は分割され複数の主体によって実行されるべきである。このような共同作業の主体として専門家モジュールを提案する。専門家モジュールはニューラルネットワークで表現された判断基準を内包することができる。現場でニューラルネットワークをトレーニングすれば、判断基準は各機器の特異性を反映するようになる。

2. 状況の形式的記述

2. 1. 概念とインスタンス

プラント機器のような観測対象物は時間に依存し

て互いにいろいろな関係を持つ。これら物の状態はそれが持つすべての関係により定義される。物の特徴はそれがとるべき状態の集合である。

概念とは共通の特徴をもつ物の集合である。本手法では、概念の記述は3つの部分からなる[1,2]。特徴記述とはその概念に属するすべての物に共通の特徴を示している。特徴記述中ではこの特徴を表現するために特徴変数が宣言される。分割記述はひとつの概念がどのように複数の部分集合に分割されるかを示している。新しい概念は一つ以上の上位概念の積集合として宣言される。これら上位概念が帰属記述の中で指定される。下位概念は上位概念におけるすべての特徴変数を継承する。

インスタンスは物を表す抽象データである。インスタンスはある概念の要素として宣言される。インスタンスはその概念のすべての特徴変数を持っている。インスタンスの状態は特徴変数の値により表現される。

2. 2. 場面と状況

観測対象物は時間の流れの中でさまざまな状態をとる。時間の流れは時点と区間の連結である時間軸で表現できる。ある時点もしくは区間における物の状態を表す抽象データを場面と呼ぶことにする。場面は時点もしくは区間のいずれかと、インスタンスの状態との組合せである。

時間の流れに伴う物の状態遷移を表す抽象データが状況である。状況は時間軸と場面の組合せである。状況は、各時点・各区間においてインスタンスが指定された状態にあることを示している【図1参照】。

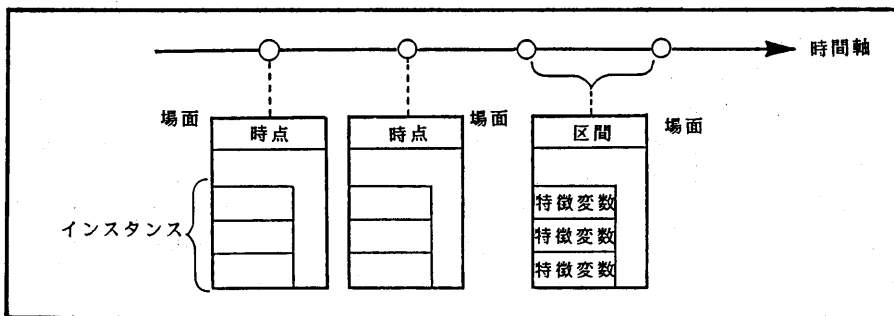


図1 状況

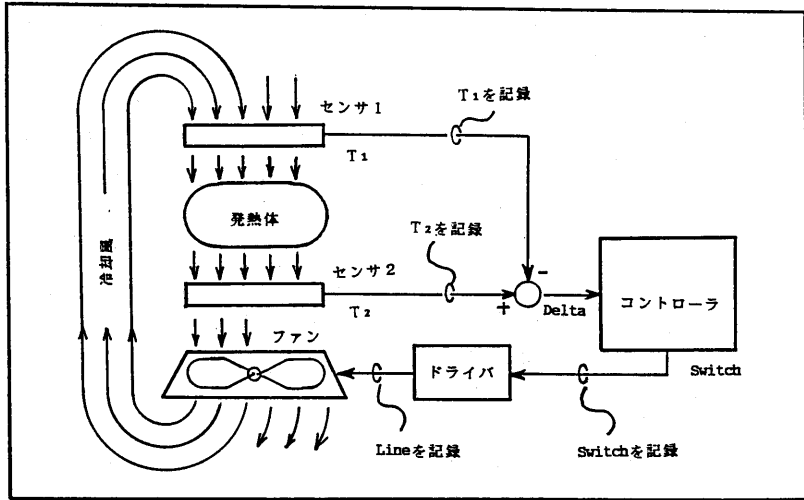


図2 クーラ・システム

2. 3. スケルトン

状況を引き起こしている本質を同定するためには特定の条件を満たす状況を他の状況から区別しなければならない。この条件は制限と制約によって表現される。制限は特徴変数の値の範囲を示す条件である。制約は特徴変数の値の範囲についての包含関係を示している。

特定の状況を示すために2種類のスケルトンを導入する。概念スケルトンは制限と制約のあつまりである。これは制限と制約を満たすインスタンスの集合に対応する。状況スケルトンは特定の条件を満たす状況の集合にあたる。これは、時点・区間の順序を示す条件と各時点・区間での概念スケルトンの2つの部分からできている。時点もしくは区間と概念スケルトンとの組合せは状況スケルトン中の場面である。

スケルトンは故障や操作ミスのような異常状況の本質を同定するのに用いられる。各サンプル時点における機器の状態が観測された場面として記録される。観測された場面の並びは観測された状況である。これは、特定の本質によって引き起こされる状況に対応する状況スケルトンと比較される。この両者がマッチしたとき本質が同定されたことになる。

2. 4. SOUL

SOUL (Situation Oriented Unification Language) は状況認識に特化されたプログラミング言語

である。SOULは概念、インスタンス、場面、状況が形式的に表現できるように設計されている。

SOULの能力を説明するために図2に示したようなクーラシステムの例を挙げる。このクーラシステムではコントローラはセンサ2とセンサ1からサンプルされた温度データの差が 800°C を越えるときドライバをONとし、差が 500°C を下回るときドライバをOFFとする。プログラム1はコントローラがドライバをONからOFFにしたことを示す状況スケルトンである。概念コントローラは2つの特徴変数 delta , switch を持っている。状況スケルトンではインスタンスの状態遷移は場面の並びで表現される。インスタンスの状態を示している概念スケル

```

situation SwitchOFF(ControllerSkeleton)
skeleton Contoller ControllerSkeleton;
{
    axis has concept Transition point;

    scene before(point)
        ControllerSkeleton such that {
            delta : >500, <800; switch == ON;
        };
    scene at(point)
        ControllerSkeleton such that {
            delta == 500; switch == OFF;
        };
    scene after(point)
        ControllerSkeleton under {
            switch == OFF ==> 0 < 500-delta;
        };
};

Program 1. Situation Skeleton

```

トンが各場面中で指定されている。SOULでは制限はキーワードsuch thatに続き、制約はキーワードunderに続く。

3. ニューラルネットワークを用いた状況認識

3.1. 場面を分別するニューラルネットワーク

ニューラルネットワークはパターンを分別し、また、望ましい分別ができなかった場合は例題を用いて望ましい分別をするようこれをトレーニングすることができる。本手法においては、観測された場面が状況スケルトン中のいかなる場面に当てはまるかを判断するのにニューラルネットワークを応用する。

1つのネットワークは1つの状況スケルトンに対応し、5層からなる。第1層は入力層であり、第5層は出力層である。状況スケルトン中の特徴変数はネットワークの入力層中のユニットに相当し、場面は出力層中のユニットにあたる。ユニットは隣あった層中のユニットにのみ結合を持つ。各結合には重みが付加されている。ユニット u_i への入力 x_i はユニット u_j の出力を y_j 、 u_j から u_i への結合の重みを w_{ij} として

$$x_i = \sum_j w_{ij} y_j$$

であらわされる。ユニットの出力値は次のような非線型関数により0から1までの実数値に決定される。

$$y = 1 / (1 + \exp(-x))$$

これを利用した場合、各ユニットの出力は[0, 1]の範囲にあることが仮定されている。入力層中のユニットの出力値は特徴変数の値に相当するが、この仮定のため[0, 1]の範囲に正規化されなければならない。そこでSOULではスケルトン宣言時だけでなく概念を宣言するときにも、特徴変数の値の範囲を明示する制限を指定することを記述者に強要している。概念宣言時に指定される制限を特に対応する特徴変数の定義域と呼ぶことにする。定義域はその特徴変数に対応するユニットの出力値を正規化するのに用いられる。特徴変数 x 、 y の定義域が $[x_0, x_1]$ 、 $[y_0, y_1]$ のとき、それぞれの正規化値 ξ 、 ν は

$$\xi = (x - x_0) / (x_1 - x_0)$$

$$\nu = (y - y_0) / (y_1 - y_0)$$

で表現できる。

観測された場面を分別するネットワークの構造は、状況スケルトンの場面中で指定された制限と制約を表現している。特徴変数はすべて正規化されているものとして、例えば図3(a)に示されたネットワークの構造は制限 $0.1 \leq x \leq 0.6$ に対応している。ここで、第2層中の2つのユニットが0.5以上の値を出力することはそれぞれ $0.1 \leq x$ 、 $x \leq 0.6$

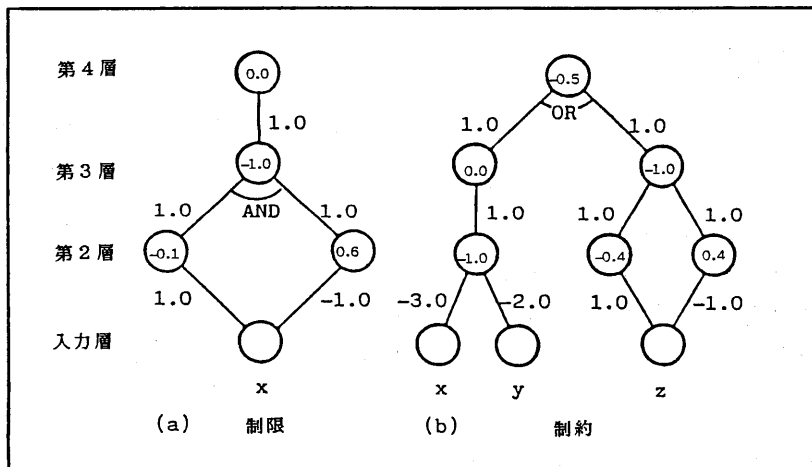


図3 ネットワークの構造

の不等式が成立するという事象に等価であることと、第3層のユニットが0.5以上の値を出力するかどうかはこれら2つの事象の論理積に対応していることに注意されたい。図3(b)に示したネットワークの構造は制約 $3x+2y+1 >= 0 \Rightarrow z == 0.4$ に相当する。先と同様、第2層の3つのユニットはそれぞれ $3x+2y+1 >= 0, z <= 0.4, z >= 0.4$ の不等式に、第3層のユニットは1つ以上の不等式の論理積に相当している。第4層のユニットの出力値が0.5以上であることは制約中の含意が成立するという事象に等価である。制限、制約に相当する2つの第4層のユニットは第5層の1つのユニットに結合される。この第5層のユニットが0.5以上の値を出力することは、状況スケルトンのうち前述の制限と制約の両方を含んだ場面に入力パターンが当てはまることと等価である。

ある状況スケルトンがS O U Lで書かれたとき、そこで指定された不等式・等式の中の特徴変数はその値が正規化されていない。このためネットワークの構造を決定するためにはこれら不等式・等式を書き換える必要がある。例えば前述の定義域をもつ特徴変数 x, y についての不等式 $ax+by+c > 0$ は ξ, ν を用いて

$$a(x_1 - x_0)\xi + b(y_1 - y_0)\nu + ax_0+by_0+c > 0$$

と書き直され、この不等式の係数からネットワークの構造が決定される。

3. 2. ユニフィケーション

状況を引き起こしている本質は観測された状況と状況スケルトンとの比較により同定される。観測された状況とは観測された場面のサンプル時点の順序に従った並びである。観測された場面は観測対象物からのデータを各サンプル時点ごとにインスタンスの特徴変数の値として記録することにより生成される。

ニューラルネットワークは、状況スケルトン中のどの場面に観測された場面があてはまるかを判断する。我々はこの判断を場面マッチングと呼んでいる。各場面マッチングの後で観測された場面が状況スケルトンに指定された順序どおりに現れるかが検査さ

れる。我々は場面マッチングと時間順序についての検査をあわせて、観測された状況と状況スケルトンとの間のユニフィケーションと呼んでいる。

3. 3. 現場特有知識の反映

現場特有知識とはプラントにおける各機器の特異性についての知識である。プラント内の各機器は同一の機種であってもその使用条件により特異性を呈する。プラント内のオペレータはこの特異性を経験上知っている。したがって彼らがプラント上に起こる状況を認識するための形式的記述を作成することが望ましい。しかし一般的にいてオペレータは、形式的記述作成のために不可欠な機器間の機能的依存関係をモデル化することに不慣れである。彼らが形式的記述をいちから作成するのは不可能に近い。一方、プラントの設計者はプラントを構成しているすべての機器の一般的特徴を知っており、これらの間の機能的依存性のモデルを持っている。設計者は形式的記述を作成することができる。問題はこのようなして作成された記述は現場特有の知識を反映していないことである。

状況認識システムを構築するさいには、まず設計者がS O U Lで形式的記述を作成する。認識システムの稼働時にシステムが異常であると判断した観測状況がオペレータに提示される。もし認識結果がおかしいと思えば、オペレータはおかしいと思う場面を指摘しその場面についての望ましいとおもう認識結果をシステムに与える。システムは望ましい認識結果にもとづいて内部のニューラルネットワークをトレーニングする[3, 4]。これにより状況認識システムは徐々に現場特有の知識を反映するようになる。オペレータは認識結果中でおかしいと思う場面とそれに対する望ましい認識結果を入力するだけでよく、機能的依存関係のモデルを持つ必要はない。

トレーニングはネットワークの第1層、第2層、第3層間の結合について行われる。これは制限・制約における不等式・等式の係数の修正に対応する。例えばプログラム1の最初の場面の制限の中で特徴変数 δ についての不等式の係数はトレーニングの結果 $\delta : > 500, < 850$ に修正されるかもしれない。

4. 状況認識のための専門家

4. 1. モジュール

プラント上の異常状況はいつ起こるかわからない。プラントを構成する機器をいくつかの主体が常に監視していなければならない。そのうえ、状況は多様で特異である。各状況を認識するためには多くの作業をしなければならない。このような作業は多くの副次的作業に分割されて、各作業に特化された複数の主体により解決されるべきである[5]。このような作業に特化された主体として我々はモジュールを提案する。モジュールには4つの特徴がある。

- (1) 自律性：モジュールは自己を制御する能力を持っている。
- (2) 並行性：モジュールは同時に動作することができる。
- (3) 規格化：モジュール間インタフェースが限定されている。
- (4) 通信：モジュールは通信により作業を協同してすすめる。

これら4つの特徴により、観測された状況の記録、観測状況と状況スケルトンとのユニフィケーション、ネットワークのトレーニングを同時に行うことが可能となる。自立したモジュールは常に機器を監視しているので異常状況の発生とともに開始される認識活動が実現できる。各認識作業は認識されるべき状況に特化された複数のモジュールに分散される。モジュールは全体の仕事がうまく進むように互いに通信することができる。

モジュールはUNIXオペレーティング・システムにおけるプロセスの機能を用いて実現される。プロセスの制御手順を示した手続きがモジュールの自律性を表現している。モジュールはその実行中に他のモジュールを生成したり、または自ら消滅することが許されている。モジュールは、遠隔手続き呼出し[6]にもとづいて実現される、メッセージ送受信による非同期通信とAdaのランデブによる同期通信の2つの通信機能をもっている[1]。モジュール間の相互作用はこれら2つの通信機能に限られており、モジュールのふるまいは通信機能によってのみ影響される。モジュールの内部構造は外部からは隠

蔽されている。

4. 2. 作業に特化されたモジュール

問題を分散的な手法で解こうとする人は、問題を解決するのに必要な作業が程よい複雑さでまとめられ、それらひとつひとつの作業を実行するのに適したモジュールが用意されていれば便利であると感じるであろう。このようなモジュールが用意されていれば、それにあわせて問題を分割できる。これらモジュールにはあらかじめ作業実行のための機能が用意されており、かつ、不必要な機能は削除されているので、保守性に富んだ問題解決のための記述が労苦なく作成できる。

本論文で説明してきた手法を用いて状況を認識しようとする場合、観測された状況の記録、ユニフィケーションによる状況の認識、ニューラル・ネットワークのトレーニングという3つの重要な作業が存在する。この3つの作業に対応してSOULでは3種類のモジュールが用意されている。クーラ・システムについてのこれらモジュール間の相互作用を図4に例示する。

4. 2. 1. 書記モジュール

書記モジュールの主な働きは観測された状況の管理である。書記モジュールは観測対象物からデータを断続的に取り込み、サンプル時点におけるインスタンスの特徴変数の値としてこのデータを記録することにより観測された場面を作成する。書記モジュールはサンプル時点の順序に従って観測された場面を並べ、この列を観測された状況として管理する。

図4においては4つの書記モジュールSensor1clerk, Sensor2clerk, Switchclerk, Lineclerkが存在する。

4. 2. 2. 専門家モジュール

専門家モジュールは状況認識活動に本質的なモジュールである。各専門家モジュールは数個の状況スケルトンを内部に持ち、特定の状況のみを専門に認識するように特化されている。専門家モジュールは認識に必要な観測された状況を書記モジュールから受け取る。専門家モジュールは自己の中の状況スケルトンと受け取った観測状況との間のユニフィケー

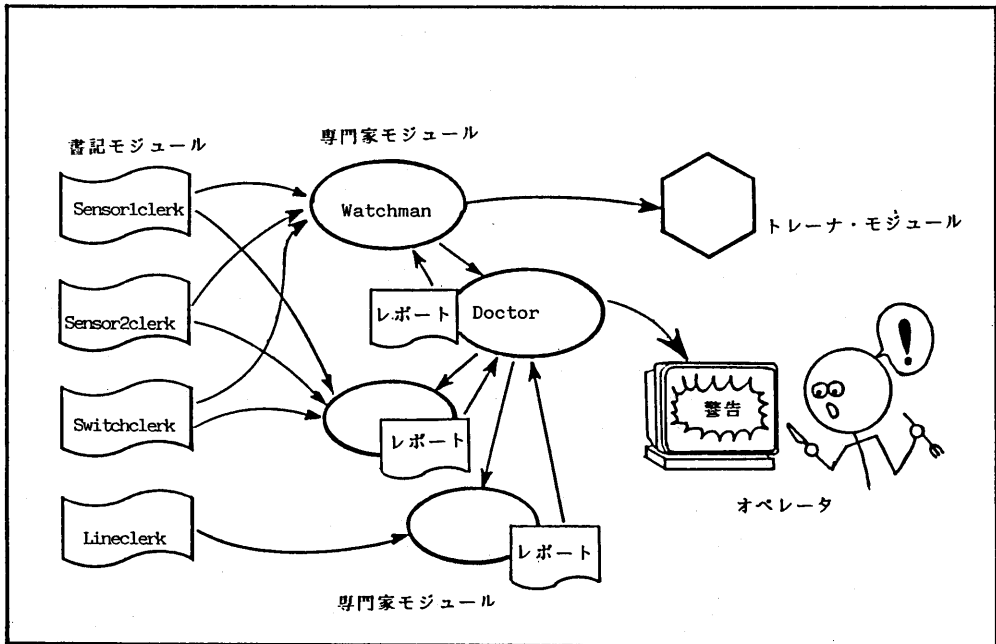


図4 モジュール間の相互作用

ションを試みる。専門家モジュールはユニフィケーションの結果といったような認識活動についてのレポートを作成する。

特定の状況を認識するための作業は複数のモジュールに分散される。各々のモジュールは自らの作業の一部を子のモジュールに依頼してもよい。状況は階層的に認識される。子のモジュールは作業結果のレポートを親モジュールに渡してもよいし、要求されればレポートの内容を自分でオペレータに説明してもよい。

オペレータが認識結果に同意しなかったとき、専門家モジュールは後述のトレーナ・モジュールにネットワークのトレーニングを依頼する。

図4では Watchmanなる専門家モジュールが、Sensor1clerk, Sensor2clerk, Switchclerkからもたらされた観測状況を使ってクラ・システム上に異常な状況が発生していないかどうかを常に監視している。Watchmanは内部に正常状態に相当する状況スケルトンを持っており、この状況スケルトンと観測状況とのユニフィケーションを定期的に試みる。ユニフィケーションが失敗したとき、Doctorなる

Watchmanの子の専門家モジュールが生成され、その原因を追求する。Doctorは特定の原因に対応した状況スケルトンを内包する専門家モジュールを自分の子として生成する。これらの子の専門家モジュールからのレポートを使ってDoctorは認識結果をオペレータに説明する。

4. 2. 3. トレーナ・モジュール

トレーナ・モジュールは専門家モジュールにより起動される。トレーナ・モジュールは専門家モジュールからネットワークの構造と観測された状況をうけとる。トレーナ・モジュールは観測状況中の各場面が状況スケルトンのいかなる場面に適合したかをオペレータに示す。ついで、トレーナ・モジュールはオペレータから彼が認識結果がおかしいと思う場面とそれに対する望ましい出力を受け取る。トレーナ・モジュールは学習アルゴリズム[3]によりこの場面に相当するネットワーク構造中の重みをトレーニングし、専門家モジュールにそのネットワークを返す。新しいネットワークは、オペレータのもつ現場特有の知識をより多く反映した構造を持っている。

5. 結論

本論文では状況を形式的に表現するための新しい枠組を提供した。自律性、並行性を持ち規格化され通信機能を有する計算単位としてモジュールを提案した。これらは状況認識専用言語SOULで実現される。状況認識システムを構築するために3種類の作業に特化されたモジュールを提示した。この中で専門家モジュールは内部にニューラルネットワークで表現された判断基準を持っている。このニューラルネットワークはオペレータがトレーニングすることができるので、専門家モジュールは今まで無視されてきた現場特有の知識を反映して現場に適応していくことができる。

現在我々はSOULのコンパイラを開発中である。現在までに状況スケルトンを構文解析しニューラルネットワークの構造を決定する部分がほぼ完成している。ひきつづき我々はモジュールの実行コード生成部を作成していく。

サンプル間隔を短くして変化する状況を素早く正確に認識できなければ状況認識システムは試作の域をでない。また分散方式をとる場合、認識作業の分割などを支援して状況認識のためのプログラムが容易に作成できるようにすることは状況認識システムの実用化に不可欠である。我々はこのような観点から、SOULコンパイラの開発とあわせSOUL実行系の高速化と、SOULによる状況認識プログラムの作成を計算機支援のもとでできるようなプログラミング環境についても研究を進める予定である。

[参考文献]

- [1] H. Shimakawa, S. Ikebata, The Structured Description of the Knowledge in the Situation Recognition, Proceedings of International Workshop on Artificial Intelligence for Industrial Applications, pp.576-581, Hitachi, Japan, 1988
- [2] 島川, 池端, 状況認識における概念と問題解決の知識の階層化, 人工知能システムの枠組シンポジウム論文集, pp45-54, 昭和62年
- [3] G.E.Hinton, Learning Distributed Representations of Concepts, Proceeding of 8th Annual Conference of the Cognitive Science Society. 1985
- [4] S.I.Gallant, Connectionist Expert Systems, Communication of ACM, Vol.31, No.2, February, 1988.
- [5] M.S.Fox, An Organizational View of Distributed Systems, IEEE Transaction on System, Man, and Cybernetics, Vol.SMC-11, No.1, January 1981
- [6] Sun Microsystems, Inc., Remoto Procedure Call Programming Guide, February, 1986