オブジェクト指向概念に基づく ファジィ集合処理の実現[†]

山本 創造^{*1} 井上 由文^{*2} 安信 誠二^{*3}

ファジィ集合の計算機上での表現と処理は、ファジィ理論を用いた知的システム構築の基盤として重要である。このファジィ集合はさまざまな形のデータ構造で表現することが可能であり、それらを統一することは困難である。本論文では、オブジェクト指向概念が複雑なデータ構造を抽象化し統一的に操作する力を持つことに注目し、オブジェクト指向概念に基づいたファジィ集合処理の可能性について検討し、具体的処理方法を提案する。プロトタイプシステムとして開発されたファジィ集合処理システム FOPS では、アレイ型ファジィ集合 (AFS) とペア型ファジィ集合 (PFS) の2つの基本的なファジィ集合表現の統一的な取扱いを実現し、さらにファジィ集合の数値としての取扱いを許すことにより、一般的なファジィ集合を容易に操作することを可能としている。これにより、オブジェクト指向概念のデータ抽象の力と多態性がファジィ集合処理に対して有効である事が示された。また、用いた内部データ表現と FOPS の有効性の評価についても示す。

キーワード：ファジィ集合、情報処理、オブジェクト指向概念、データ構造

1. はじめに

ファジィ理論は、ファジィ推論の制御への応用を発端として多大な注目をあびるようになったが、近年ではデータベースや社会科学などにも適用の範囲は広がっている。理論の展開に伴い、人工知能的な手法やニューラルネット技術とも融合し、より幅広い形で知的なシステムの構築に役立てられるようになってきている。

このような応用面での進展に反して、ファジィ理論を幅広く用いた知的システムの構築を支援するツールは研究段階にある。多くのファジィ理論応用システムでは、それぞれ独自にファジィ集合

表現を行なっているというのが現状である。また、本質的にあいまいさを許容し人間との親和性が高いコンピュータとして「ファジィコンピュータ」という概念が提唱されている^[1]が、この実現にはファジィ理論の応用に関する研究の進展を待つ必要がある。

ファジィ理論に基づく処理を計算機で行なう時には、ファジィ集合・ファジィ集合演算の計算機上での表現が基本となる。ファジィ集合の処理を行うシステムはいくつか提案されている^{[2]-[5]}が、ファジィ集合の表現方法を固定してしまっている点と数値や記号との統一的扱いが困難である点とに問題があった。これらの問題を解決し、ファジィ集合を統一的に取り扱う方法を本論文で検討する。

一方、コンピュータソフトウェアの分野では、オブジェクト指向の考え方が広まりつつある。これはオブジェクト指向概念^[6]が、ソフトウェアの開発・保守の効率化を実現するだけでなく、実世界のモデル化を行なうにあたって有効な概念であることによっている。オブジェクト指向概念で

[†] Fuzzy Set Manipulation Based on Object-Oriented Paradigm

Sozo YAMAMOTO, Yoshifumi INOUE and Seiji YASUNOBU

^{*1} 松下電器産業(株) 中央研究所 電子機器基礎研究所第6研究室
Central Research Laboratories, Matsushita Electric Industrial Co., Ltd.

^{*2} 花王(株) 文理科学研究所 知識情報科学研究室
Kao Institute of Knowledge & Intelligence Science, Kao Corporation

^{*3} 筑波大学 構造工学系
Institute of Engineering Mechanics, University of Tsukuba

は、システムを構成する要素をオブジェクトとして表現し、各オブジェクトは自らに対する操作の方法を持っているものとする。プログラムの実行は、オブジェクトに対するメッセージの送信によって行なわれる。ファジィ集合をオブジェクトとして表現することによって、ファジィ集合の内部表現とその操作を分離することが可能になり、ファジィ集合とその他の要素とを統一的に操作することが容易になると期待できる。

本論文では、最初に、オブジェクト指向概念に基づいたファジィ集合処理の可能性について検討する。次に、プロトタイプシステムとして我々が開発してきたファジィ集合処理システム FOPS におけるファジィ集合を表現するためのデータ構造とそれらの演算について述べ、オブジェクト指向概念のファジィ集合処理に対する有効性を示す。

2. オブジェクト指向によるファジィ集合処理

ファジィ集合の計算機上での表現において問題になるのは、その表現方法が多数存在し一つに確定していないことである。ファジィ集合の表現はアプリケーションによって様々な形をとり得る。例えばメンバーシップ値の行列によるファジィ集合表現は制御への応用などで広く用いられる方法である。また全体集合が数値軸でなく、名前の集合のような場合、あるいは全体集合が特定できない場合には、要素とそれに対応するグレードの値とを列挙することによって表現することが自然である。この他にも、数式による表現や折線の折点の並びで表現する方法、 α -cut の並びによる表現など、さまざまな形式がとられている。これらは、表現能力においても記憶・演算のコストについても異なった性質を持っており、1つの方法に統一することは難しい。ファジィ情報処理を支援するためのシステムには、これらのさまざまな表現を統一的に取り扱えるようにすることが求められる。

オブジェクト指向の枠組においては、オブジェクトは自身を操作するために必要な手続きを持った実体として取り扱われる。通常オブジェクト指

向言語では、数や文字列などのシステムに組み込まれている要素もオブジェクトによって表現される。このため、ファジィ集合をも同様にオブジェクトとして表現することにより、システム中の他の要素との統一性を保つことが出来る可能性がある。これはデータの集まりをオブジェクトという実体としてみるデータ抽象の機能が、ファジィ集合処理に対して有効であることを意味する。

また、オブジェクト指向には、メッセージやオブジェクトの持つ意味を状況に応じて変化させる機能である多態性(polymorphism)がある。このメッセージの多態性を利用することにより、ファジィ集合を数値など他のオブジェクトと統一的に取り扱うことが可能になる。

従って、オブジェクト指向概念に基づいてファジィ集合処理を記述することにより、ファジィ集合の利用上の制約が少なくなり、ファジィ理論を用いた知的情報処理が容易に記述できるようになる。

3. ファジィ集合処理システム FOPS の開発目的と課題

以上のような考えに基づき、われわれはオブジェクト指向のファジィ集合処理システム FOPS を設計した。開発の目的は、

- ・オブジェクト指向概念のファジィ集合処理に対する有効性の実証評価
- ・ファジィ集合の内部表現形式の検討
- ・ファジィ応用システムの研究開発用の実験的環境の実現

である。FOPS は代表的なオブジェクト指向言語である Smalltalk システムの上に構成されている。Smalltalk^[7]はシステム自体が Smalltalk で記述されている言語であり、パーサや演算子などシステムに関する変更が容易である。また、高度なプログラミング環境を備えており、実験用の環境の基盤として優れている。これらの理由から Smalltalk (Objectworks \ Smalltalk Release 4) をシステムの基盤に用いた。

このシステムの開発に当たっての中心的な課題

はファジィ集合を表現するためのオブジェクト(クラス)を設計することである。様々なタイプのファジィ集合表現に対応するためには、複数のクラスを並行して用いることになる。そこで、ここでは複数のファジィ集合表現を統一的に扱う方法を検討した。また、ファジィ集合に対するインタフェースについては、コードによる入出力を基本とし、ファジィ集合の記述を通常のプログラムの記述の中に含められるようにした。さらにメンバーシップ関数のグラフィック表示や専用のインスペクタなどのツール群を備えることにより、簡便なインタフェースを追求した。

4. ファジィ集合の表現

4.1 ファジィ集合の取扱い

複数の表現形式を持つファジィ集合を表現し、それらを統一的に操作するためには、それぞれの表現形式に対応した内部表現を持つクラスの設計と、統一性を保証する枠組が必要になる。ファジィ集合の表現形式としては、次の2つの基本的な形式がある。

- ・アレイ型ファジィ集合(arrayed fuzzy set)
全体集合に関する記述とメンバーシップ値の行列を基本とする表現形式
 - ・ペア型ファジィ集合(paired fuzzy set)
全体集合中の要素と対応するグレードとの組を列挙することによる表現形式
- アレイ型ファジィ集合はメンバーシップ値を行列の形で持つため、グレードを直接添字で参照できる。このため演算時には要素に関する処理が簡

略化でき高速な処理が期待できる。ペア型ファジィ集合は要素を参照するためにリストを順次たどる必要があるが、構造が自由であり、多様な構造を持つファジィ集合を一般的に表現するのに適している。

これら2つの表現形式に対応するクラスをそれぞれ設定する。アレイ型ファジィ集合を表現するためにクラス ArrayedFuzzySet(以下 AFS と記す)を、ペア型ファジィ集合のためにクラス PairedFuzzySet(以下 PFS)を設ける。それぞれのクラスは、後述の相互の型変換を容易にするため、数値全般を表現するための抽象クラスである ArithmeticValue のサブクラスとする。

クラス階層中の配置としては、集合のためのクラス Set などと同様に物の集まりを表現するクラスである Collection の下に置くことも考えられる。しかし、内部表現が特殊になるためメソッドのほとんどが継承できず、また数値としての扱いを重視するために、数値のための抽象クラスである ArithmeticValue のサブクラスとした。

また、ファジィ関係(全体集合が複数の集合の直積であるファジィ集合)も AFS, PFS の両形式を用いて表現できる設計とする。これは通常の(1次元の)ファジィ集合をファジィ関係の特殊なケースと見ることにより可能となる。詳しい内部構造については次節で述べる。

さらに、言語的真理値を AFS の特殊なケースと考え、クラス LinguisticTruthValue(以下 LTV と記す)を AFS のサブクラスとして定義する。これは言語的真理値がタイプ-2 ファジィ集合のグ

表1 ファジィ集合表現のためのクラス群

クラス名	スーパークラス	内容
Sequence	Array	ファジィ集合の次元の表現
GEPair	Link	グレードと要素の対
PairList	LinkedList	GEPair の並び, PFS が使用
PairedFuzzySet	ArithmeticValue	ペア型ファジィ集合(PFS)
GradeArray	Array	AFS 用のグレードの多次元配列
ArrayedFuzzySet	ArithmeticValue	アレイ型ファジィ集合(AFS)
LinguisticTruthValue	ArrayedFuzzySet	言語的真理値(LTV)

リードやファジィ真理値の表現などにおいて多用されるためである。

これらのクラスの設定により、ファジィ集合を抽象データとして取り扱うことが可能になり、ファジィ集合の操作が容易になった。

4.2 ファジィ集合の内部表現

ここでは FOPS における AFS, PFS, LTV の 3 つのクラスの内部構造について述べる。これら 3 つの他、表 1 に示すクラスが、ファジィ集合表現のために定義されている。以下、クラスとそのクラスのインスタンスとは、混乱のない限り区別せずにクラス名で示すことにする。

ファジィ集合の全体集合の次元を表現するために Sequence というクラスを定義している。Sequence は順序付きの集まりである Ordered-Collection のサブクラスであり、文字表現のみが異なっている。Sequence の要素は、' ' の代わりに '&&' で連結される。ファジィ関係の全体集合や要素における直積空間の指定はすべて Sequence を用いて行われる。

AFS は図 1 に示す構造を持つ。AFS は universe, gradearray, univsep の 3 つのインスタンス変数を持つ。universes は全体集合を記述するためのインスタンス変数であり、全体集合を要素

図1 アレイ型ファジィ集合の内部構造

とする Sequence を値にとる。全体集合は Interval (区間) や Array (行列) などの Sequenceable-Collection (順序付きのオブジェクトの集まり) であり、添字によりアクセスが可能である。Sequence の添字はファジィ集合の全体集合の次元に対応し、ファジィ関係の表現に対応する。メンバーシップ値はインスタンス変数 gradearray に格納される。gradearray はファジィ集合が n 次元で次元 i ($1 \leq i \leq n$) の全体集合 U_i の大きさが m_i であるとき、 $m_1 \times m_2 \times \dots \times m_n$ 個の要素を持つ 1 次元のアレイ (ベクタ) である。gradearray を 1 次元の行列としているために、演算時には gradearray のみを参照して高速に演算を行うことが可能である。

全体集合の要素とメンバーシップ値の対応は次のように行われる。各次元の全体集合の要素はそれぞれの全体集合によって添字と対応づけられる。すなわち U_i から添字空間 I_i への対応 ($U_i \rightarrow I_i$) が各全体集合により行われる。この添字の集合は gradearray の添字に変換され ($I_i (1 \leq i \leq n) \rightarrow I$)、この添字の空間 I がメンバーシップ値の空間 G に対応づけられる。univsep は自然数を値にとり、ファジィ関係をファジィな対応と見たときの定義域と値域の別を universes に対する添字により示す。これは主にファジィ関係の合成の演算に関係する。

一方 PFS は図 2 に示す内部構造を持ち、contents をインスタンス変数に持つ。contents のとる値は LinkedList (線形リストを表現するクラ

図2 ペア型ファジィ集合の内部構造

ス)のサブクラスとして設定されたクラス Pair-List のインスタンスであり、全体集合の要素とそれに対応するメンバーシップ値のペアを要素とする。すなわち、PFS ではファジィ集合に含まれる要素と対応するメンバーシップ値を数え上げる表現形式であり、全体集合は明示しない。多次元のファジィ集合を表現するためには要素の部分要素の Sequence とする。contents の各要素は各ペアの要素部分の順で整列されている。これは演算の高速化のためである。

LTV は AFS のサブクラスとして実現されており、内部構造は共通するが、全体集合が1次元でありしかも $[0,1]$ の区間に限定されるという点で異なる。また、演算が一部再定義されており、入出力の文字表現も異なっている。

4.3 ファジィ集合の入出力

プログラム中でファジィ集合を操作するための文法を設定した(図3)。例えば

$$\{0.1/2, 0.3/4, 0.5/6\}$$

のような自然なファジィ集合表現をプログラム中に記述できるようにするため、式の中で '{' と '}' で囲まれた部分をファジィ集合式(fuzzy set expression)として解釈し、このファジィ集合式の中では '/' と ',' の2つの2項演算子の優先順位と解釈を特殊扱いするようにシステムに変更を加えた。 '{' と '}' は Smalltalk では使用されていない記号であるため、ファジィ集合表現以外の部分については文法の変更はない。以下に現れるファジィ集合記述は全て FOPS 独自の文法に従うもので

図3 ファジィ集合記述の文法

あり、FOPS のコンパイラに理解される式である。

図3のように記述されたファジィ集合表現は内部的には通常のメッセージ式と同様に解釈され、評価の結果は AFS, PFS, LTV のいずれかになる。on: , ON: , from: to: の記述が含まれるものは AFS, ltv: , LTV: を含むものは LTV になる。その他のものは PFS である。大文字のメッセージ(ON: など)ではオブジェクト生成時にメンバーシップ関数の線形補間を行う。メンバーシップ関数の折点を順次指定することでファジィ集合を定義できる。

また、グラフィック表示機能も備えており、ファジィ集合のメンバーシップ関数の形を画面に表示することが可能である。

5. FOPS におけるファジィ集合の演算

5.1 数値としてのファジィ集合

前節で述べたようにファジィ集合は ArithmeticValue のサブクラスとして定義されており、これは数とファジィ集合との演算を容易に行うことを目的としている。数値軸の上に定義されたファジィ集合は数のファジィ表現であり、数値として演算を受けることがある。FOPS ではクラス Number とそのサブクラスに定義されている数値の表現(以下では単に Number と記す)とファジィ集合の間の演算を可能としている。Number とファジィ集合との間の演算は次の2つの場合に生ずる。

- ・ファジィ数と通常の数値との間の演算
 $2 + \{0.3/2, 0.7/2.5, 1/3, 0.7/3.5, 0.3/4 \text{ on: } (1 \text{ to: } 5)\}$
- ・タイプ2ファジィ集合のグレードの演算
 $\{0.4/'a', 0.7/'c'\} \text{ union: } \{\{0.5/0.8, 1/1\}/'a', 0.3/'b'\}$

ファジィ集合と Number とは相互に演算が可能であればファジィ数の演算に便利である。さらにタイプ2のファジィ集合のグレードの演算ではファジィ集合の数値としての演算が必要である。例えば上に示した例では要素'a'に対する2つのグ

レードの間に $0.4 \max: \{0.5/0.8, 1/1\}$ の演算が行われる。ファジィ集合と Number の間の数値演算を可能とすることによってタイプ2ファジィ集合の集合演算を容易に実現することができた。

この手続きは再帰的に適用されるので、より一般的にタイプ n のファジィ集合の集合演算が可能である。

5.2 普遍性の階層と型変換

ここで述べたような型変換の問題は数表現するオブジェクトにおいては一般的な問題である。例えば Smalltalk では Integer(整数)や Float(浮動小数点数)など数種類の数を表現するクラスがあり、相互に演算が可能である。これを実現しているのがクラスの普遍性(generality)の高さという考え方と強制的型変換(coersion)のメカニズムである(システム中で多用されるクラスでは高速化のため別の手法が用いられている)。generality は、あるクラスがどの程度表現の一般性を持っているかを示す指標であり、例えば Float は Integer よりも generality が大きく、Float の方が普遍性の高い表現であることを示す。

generality が異なる2つのオブジェクトの間に演算が行われた場合には、普遍性の低いオブジェクトがより普遍性の高いクラスの同一値のインスタンスに自動的に変換され、その後相互の演算が行われる。この変換が強制的型変換である。2項演算ではメッセージの解釈をレシーバとメッセージ引数の両方から決める必要があるが、この自動的な型変換により、内部構造によらず同一のメッセージによる演算が可能になる。この機構をファジィ集合に対しても用いることにより、メッセージの多態性というオブジェクト指向プログラミングの特長を生かすことができる。

FOPS では AFS, PFS, LTV の3つのクラスを設けている。これらのクラスと Number との generality は図4に示す関係を持つ。LTV は全体集合が $[0,1]$ 区間に制限されているためにファジィ集合の中では最も制約が多い。また AFS は全体集合の情報を持つため PFS よりも自由度が少な

い。Number < LTV < AFS < PFS の順で普遍性が高くなってゆくことになる。この階層の中で普遍性の低いクラスのオブジェクトはより普遍性の高いクラスに変換できる。Number に関しては $[0,1]$ 区間に含まれないオブジェクトは正しく LTV に変換できないが、先述のように Number と LTV の間の変換が行われるのは真理値に関する演算の場合であり、Number の側も $[0,1]$ 区間に含まれる値であるので正しい変換が可能である。

FOPS では AFS, PFS, LTV の 3 つのクラスを定義し、これらと Number とを同一の系列中に位置づけた。以上に述べた相互の演算の仕組みはユーザに解放されており、ファジィ集合のためのクラスを新たにユーザが定義することが可能である。新たなクラスを定義した場合には、相互に変換が可能かどうかによって適切な generality を設定し、普遍性の低いものから高いものへの変換のためのメソッドを定義すればよい。これにより新たに定義されたクラスが既存のクラスと統一的に使用できる。FOPS の拡張性については文献[8]により詳しく報告している。

6. 評価

6.1 データ型による処理速度の比較

AFS と PFS とのデータ構造の違いにより、演算の処理コストの違いがある。しかし、演算処理の部分とメモリ操作に依存する部分が絡むため、理論的には計算量を比較するのが難しい。また AFS の計算量が全体集合の大きさに依存するのに対して、PFS の計算量は非 0 のグレードを持つ要素の数に依存するので、対象により得失が変わるという問題もある。この処理コストの違いを実証的に評価する目的で、基本的な演算の処理速度を測定した。

2 つのファジィ集合の間での演算の中から、代表的な 3 つの演算を選び、AFS と PFS の各々について演算時間を測定した。選択した演算は次の 3 つである。

(a) fset 1 intersection: fset 2

図4 普遍性の順位と相互変換

(b) fset 1 min: fset 2

(c) fset 1 compose: frel 1

fset 1 と fset 2 は同じ全体集合の上のファジィ集合である。例えば 2 要素の場合は

AFS: $\{0.5/1, 0.5/2 \text{ on: } (1 \text{ to: } 2)\}$

PFS: $\{0.5/1, 0.5/2\}$

という値を選んだ。frel 1 はファジィ関係であり、

AFS: $\{0.5/(1\&\&1), 0.5/(1\&\&2),$

$0.5/(2\&\&1), 0.5/(2\&\&2)$

$\text{from: } (1 \text{ to: } 2) \text{ to: } (1 \text{ to: } 2)\}$

PFS: $\{0.5/(1\&\&1), 0.5/(1\&\&2),$

$0.5/(2\&\&1), 0.5/(2\&\&2)\}$

などである。

ワークステーション (Sun 4/370) で比較した結果を図 5(a) - (c) のグラフに示す。横軸はファジィ集合の要素数、縦軸は 1 演算あたりの計算時間 (システム時間を含む) である。図 5(a) は intersection 演算の結果であり、2 項のファジィ集合の論理演算の代表として示した。AFS では全体集合の要素一つ一つに関する演算が必要ないことから、PFS に比べて 4 倍程度高速に演算できる。

図 5(b) は min 演算の結果であり拡張原理に基づくファジィ集合の数値としての演算の例である。この場合はいずれの表現でも要素のすべての組み合わせについて演算を行う必要があるためあまり差はないがわずかに PFS の方が高速である。

ファジィ関係の合成 (図 5(c)) では要素数によ

って異なるが10倍あるいはそれ以上の差が見られる。これはAFSではファジィ関係の演算がグレードの行列の積として求められるのに対して、PFSでは一要素ずつ検索し、それに対するグレードの計算を行う必要があるためである。この差は要素数が増加するほど大きくなる。

以上述べたように拡張原理に基づく演算を除いてはAFSに大幅な高速性が見られた。ただし、ここでは両表現とも全体集合の全ての要素に対して非0のグレードがあるものとしており、PFSに不利な条件である。PFSの処理速度は非0のグレードを持つ要素の数に依存するため、グレードが0である要素が多い場合には、それだけ高速な演算が可能である。従って表現形式による得失はファジィ集合の種類にも依存する。

6.2 問題記述への適用

システムの効用を検証するために、問題記述に適用した例を示す。FOPSはそれ自体Smalltalkのひとつのアプリケーションであるが、その拡張性からファジィ理論応用システム構築の土台となり得る。FOPSの文法はSmalltalkの文法を一部拡張した形になっており、他のクラスを重ねて用いることにより、機能の拡張を図ることができる。

例題として、FOPSの上でファジィ推論を行うためのクラスを記述した。これによってファジィ推論を実行している様子を図6に示す。図中右中ほどの'InputPlane'のラベルがついたウィンドウでマウス指示により2次元の入力を指定し、右下方のウィンドウに推論結果のファジィ集合のメンバーシップ関数をリアルタイムで表示している。推論ルールやメンバーシップ関数はFOPSのファジィ集合の記法に従って容易に記述できる。一般的なMax-Min合成によるファジィ推論を実行するためのクラスが入出力のためのグラフィックインタフェースも含めて約5Kbytesで記述することができた。このコード量の少なさは、FOPSのファジィ集合記述の能力とSmalltalkのインタフェースに対するサポートによるものである。

(a) ファジィ集合演算

(b) 拡張原理演算

(c) ファジィ関係の合成

図5 表現形式間の演算時間の比較

謝 辞

本研究は技術研究組合国際ファジィ工学研究所ファジィコンピュータプロジェクトにおいて行われたものである。また大阪大学の馬野元秀助教授には、研究のさまざまな過程で大変有益な議論と助言をいただいた。

参 考 文 献

- [1] 安信：ファジィ工学，昭晃堂，154-174(1991)。
 [2] M. Umano：Fuzzy Set Manipulation System in Lisp, Preprints of 2nd IFSA Congress, 840-843 (1987)。
 [3] Z. A. Sosnowski：FLISP-A Language for Processing Fuzzy Data, Fuzzy Sets and Systems, 37, 23-32 (1990)。
 [4] J. M. Adamo：L.P.L. A Fuzzy Programming Language：1. Syntactic Aspects, Fuzzy Sets and Systems, 3, 151-179 (1980)。
 [5] J. M. Adamo：L.P.L. A Fuzzy Programming Language：2. Semantic Aspects, Fuzzy Sets and Systems, 3, 261-289 (1980)。
 [6] 竹内：オブジェクト指向の指向するもの，情報処理，Vol. 29-4, 295-302 (1988)。
 [7] A. Goldberg, D. Robson(相磯秀夫監訳)：Smalltalk-80：一言語詳解，オーム社，(1989)。
 [8] S. Yamamoto, Y. Inoue, S. Yasunobu：Object-Oriented Approaches for Fuzzy Information Processing, Proc. IFES '91, Vol. 1, 375-386 (1991)。

(1992年6月7日 受付)

(1993年1月29日 再受付)

[問い合わせ先]

〒570 守口市八雲中町3-1-1

松下電器産業(株) 中央研究所

山本 創造

☎：06-906-4849

☎：06-904-7252

E-mail：sozo@crl.mei.co.jp

著 者 紹 介

山本 創造 (やまもと そうぞう)

松下電器産業(株) 中央研究所 電子機器基礎研究所 第6研究室

1986年3月東京大学工学部産業機械工学科卒業。同年4月，松下電器産業(株) 入社。1988年度東京工業大学研究生を経て，1989年4月より国際ファジィ工学研究所へ出向。1991年12月松下電器産業(株) 中央研究所に帰任，現在に至る。ファジィ知的情報処理とそのためのソフトウェアに関する研究・開発に従事。日本ファジィ学会，情報処理学会，計測自動制御学会の会員。

井上 由文 (いのうえ よしふみ)

花王(株) 文理科学研究所 知識情報科学研究室

1966年生まれ，1988年 電気通信大学 電気通信学部 計算機科学科卒業，同年，花王株式会社入社，文理科学研究所 知識情報科学研究室 研究員，プログラミング言語，オブジェクト指向パラダイム，並列処理に興味を持つ。情報処理学会，ソフトウェア科学会，AAAI 会員。

安信 誠二 (やすのぶ せいじ)

筑波大学 構造工学系

'73年神戸大学工学部計測工学科卒業。'75年同大学大学院修士過程修了，同年 (株)日立製作所入社。'89年より'91年の間，技術研究組合 国際ファジィ工学研究所へ出向，第3研究室長。'91年 (株)日立製作所 システム開発研究所 第5部主任研究員。'92年 筑波大学 構造工学系 助教授，現在に至る。知的制御，ファジィ制御，ファジィ情報処理に関する研究・開発に従事，工学博士，昭和60年度計測自動制御学会論文賞受賞，第20回市村賞受賞，平成4年度科学技術庁長官賞受賞，計測自動制御学会，システム制御情報学会，電気学会，IEEEなどの会員。