

専門家の思考や判断の過程を記述できる意思決定支援システム

長谷川 利[†] 矢口 悠理[†] 遠藤 智絵[†] 荒井 淳[†] 有澤 博[‡]

[†] 横浜国立大学大学院環境情報学府 〒240-8501 横浜市保土ヶ谷区常盤台 79-7

[‡] 横浜国立大学環境情報研究院 〒240-8501 横浜市保土ヶ谷区常盤台 79-7

E-mail: [†] { toru, yaguchi, tomoe, arai }@arislabs.ynu.ac.jp, [‡] arisawa@ynu.ac.jp

あらまし 人間が行なう知識処理を計算機上で実現する取り組みは以前よりなされてきた。特に専門家自身の負担軽減を目的として、専門家による思考や判断の過程を計算機上で構築し意思決定を支援するシステムは既に実用化されている。しかし、これらは未だ利用者である専門家自身による十分な支持を得ているとは言い難い。その主な原因には、対象とする知識処理が複雑化した場合に、システムにおける思考や判断の過程の妥当性を専門家自身により確認する事が困難であるという点がある。特にマルチメディア処理を伴う医学画像診断等の場合、この問題は深刻となる。本研究では言語とデータ構造に着目した上で、システム構築に関わる人物の視点を明確に分離して記述する事ができ、かつマルチメディア処理に対応可能な意思決定支援システムを提案する。

キーワード 知識処理, マルチメディア処理, データの可視化, 言語とデータ構造

Decision Support System with the Description of Thinking and Estimating Process of Experts

Toru HASEGAWA[†] Yuri YAGUCHI[†] Tomoe ENDO[†] Jun ARAI[†] Hiroshi ARISAWA[‡]

[†] Graduate School of Environment and Information Sciences, Yokohama National University

[‡] Faculty of Environment and Information Sciences, Yokohama National University

79-7 Tokiwadai, Hodogaya-ku, Yokohama, 240-8501 Japan

E-mail: [†] { toru, yaguchi, tomoe, arai }@arislabs.ynu.ac.jp, [‡] arisawa@ynu.ac.jp

Abstract The attempt to achieve the knowledge processing that human does on the computer has been done since before. The system that supports expert's decision making aiming at the reduction of incidence of experts has already been put to practical use. However, these are difficult to say to still win enough trust by experts. When the targeted knowledge processing is complicated, thinking and estimating process on the system is difficult to understand by experts. For instance, this problem especially becomes serious at the medicine image diagnosis with multimedia processing. In this paper, it pays attention to the language and the data structure, and the aspect of the person who is related to system construction is separated clearly and described. And it proposes to decision support system corresponding to the processing of the multimedia.

Keyword Knowledge Processing, Multimedia Processing, Data Visualization, Language and Data Structure

1. はじめに

人間が行なう知識処理を計算機に代行させ実現する取り組みは、過去数多く行なわれてきた。このような取り組みの代表的かつ効果的な利用方法の一つとしてエキスパートシステム[1][2][3]があげられる。現在では、エキスパートシステムは医療診断や機械の故障診断等の様々な分野で応用されており[7]、その用途は多岐に渡っている。また、近年では計算機によって画像や映像、音声といったマルチメディアデータを扱うことが可能となっており、それらを用いたエキスパートシステムや知識処理が求められている。

しかしながら、これらのシステムは、未だ専門家自身による十分な支持を得ているとは言い難く、結果としてその多くが実用に至っていない。その主な原因は、知識処理において用いるアルゴリズムが複雑化した場合に、専門家自身によってシステムにおける思考や判断過程の妥当性を検証する事が困難であるという点にある。そこで我々は、知識処理システムとその利用をモデル化し、システム構築に関わる人物の視点を明確に分離して記述する事で、専門家自身が思考や判断の過程を容易に把握可能な意思決定支援システムの設計、提案を行なう。

専門家によるシステムの把握を実現するためには、以下の2点を満たす必要がある。

1. システム中のデータの把握
2. システム中のプロセスの把握

本研究では、この双方の条件を満たす為に、対象とする知識処理のアルゴリズムが複雑化した知識処理を行なう際、個々の処理過程において以下の3点を専門家に対して提示する。

1. ある操作における入力データ
2. 操作内容に関する概要
3. 操作後のデータ

その上で、データの流れに着目したシステムの妥当性の確認を行なう為の手法を提案する。

既存のプログラミング言語を用いても、データの流れに着目した上で、システムにおける専門家の思考や判断の過程を記述する事はできる。しかし、これらの言語は、データの流れとは関係の無い処理の記述も許容している。この事は専門家によるシステムの把握を妨げる原因となっている。この問題は、データの構造の変化のみを記述可能であるプログラミング言語を新たに提案する事で解決できる。

一方、本研究ではマルチメディアデータは、入れ子型の階層構造により十分表現できるとし、この考えに基づいたデータ構造の提案を行なう。このデータ構造は、入れ子型の構造を持つ Window の様な形式を用いて可視化して表現する事ができる。我々は、入れ子型の構造を持つ Window が、非常に直観的な GUI(Graphical User Interface)であると考えた。その上で、データ構造をビューアにより可視化する作業に関しても、研究の一部として行なっている。

本研究では、システムを流れるデータを、専門家自身が直観的な形で把握する事ができる様に可視化した上でデータ構造を表現したものである NEW(Nested Entity Window)、及び NEW を計算機上で表現する為のデータ構造 NEWeX(Nested Entity Window eXpression)、更に NEW の操作言語 MDPL(Multimedia Data Processing Language)を提案、作成している。

本稿では、2章で近年の知識処理のアプローチを紹介し、3章で知識処理システムのモデリング及び意思決定支援システム全体のアーキテクチャ、4章でデータ構造を表現したものである NEW 及びデータ構造 NEWeX、更にプログラミング言語 MDPL について記述し、5章でシステムの評価及び今後の課題について述べ、6章で全体のまとめを行なう。

2. 既存研究

2.1. 近年の知識処理のアプローチ

近年の知識処理の分野においては、データマイニン

グ[8]にとりわけ注目が集まっている。これは、限られた対象範囲のもとで、今まで人間が考えなかった様な知見を、計算機を用いる事で発見する手法であると言える。この様な手法においては、計算機を用いる事で得られた知識の妥当性を確認する作業が非常に困難である点が問題とされている。その結果、知識処理を用いた上でデータマイニングを目的として構築されたシステムは、その処理過程の妥当性を確認する作業が困難となる。

従って、この手法は少なくともシステムにおける信頼性が求められる場合には適切であるとは言えない。本研究においては、あくまでも人間が行なう様な知識処理を、計算機上で忠実に実現するという立場からの言及を行なっていく。

2.2. 既存のエキスパートシステム

現在、既存の一般的なエキスパートシステムにおいては、以下の手順によりシステムの構築を行なっている。

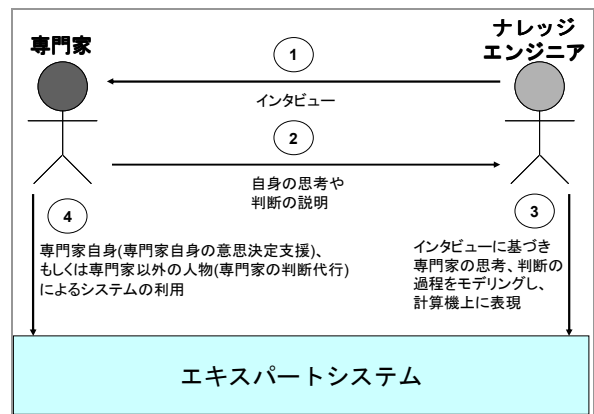


図 1:一般的なエキスパートシステムの構築過程

エキスパートシステムを構築する上で最も困難なのは、専門家の思考や判断の過程をモデリングし、計算機上に表現する事である。システム上に構築された専門家の思考や判断の妥当性は、ナレッジエンジニアが行なう対象世界のモデリングに一任されていると言える。従来のエキスパートシステムの構築過程では、ナレッジエンジニアにより行なわれた専門家の思考、判断のモデリングにおける正しさを専門家自身が確認し、システムの妥当性の向上を行なうという過程が含まれていない。この点が、エキスパートシステムにおける思考や判断の妥当性を確保する為の大きな障壁となっている。例えば、医療診断等のとりわけ間違いの許されない分野において、この種のエキスパートシステムは未だ専門家からの十分な支持を得ているとは言い難い。そして、専門家自身がシステムにおける思考や判断の過程を確認する事無しには、システムにおける妥当性の向上を行なう事はできない。

近年の計算機の性能向上により、計算機において取り扱えるデータ量は飛躍的に大きくなった。その結果、Raw データを計算機で取り扱える様になり、知識処理の分野においても、例えば Raw データから特徴量抽出処理等の操作を行ないたいというニーズは高まった。しかし一般的に、特徴量抽出処理等の複雑なアルゴリズムを非プログラマが理解するのは難しい。そして、アルゴリズムの概要提示を行なうという手法では、処理の流れが複雑になった場合に対応する事ができない。その為、非プログラマである専門家に、システムにおける処理過程の提示を行なう事は非常に困難となる。

そして、ナレッジエンジニアが構築した計算機上の処理過程が、他者にとってのブラックボックスとなってしまう事が、エキスパートシステムの妥当性を向上させる際の最大の問題点となる。その為、エキスパートシステムにおける思考や判断の過程の妥当性を向上させる為には、既存のシステム構築者の役割を再考した上で、対象世界をモデル化し直す必要があると我々は考えた。

本章では、これらの問題点を踏まえた上で、今回提案した専門家自身の支援を目的とする意思決定支援システムにおけるモデリング手法について述べる。

3. 意思決定支援システム

3.1. 専門家自身によるシステムの処理内容把握

ナレッジエンジニアが計算機上に表現した専門家の思考や判断の過程を、専門家自身の手で把握する際に大きな障壁が存在する事は既に2章で述べた。我々は、意思決定支援システムにおいて、データに着目した処理過程を記述する為の各種枠組みを提案する。図 2に、本研究で提案した意思決定支援システムの操作概念図を示す。

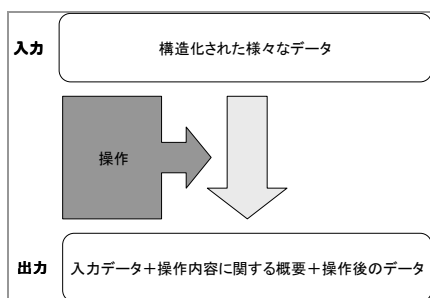


図 2: 意思決定支援システムにおける操作概念図

様々なデータが結びついている複合オブジェクトを用いて知識処理を行なう場合、専門家自身によりシステムの思考や判断の過程を把握する事が困難である事は2.2節で述べた。本研究では、この様なデータを用いる知識処理を行なう際に、個々の処理における次の 3 点を提示する手法を提案する。

1. ある操作における入力データ
2. 操作内容に関する概要
3. 操作後のデータ

我々はこの 3 点を提示する事で、対象とする知識処理のアルゴリズムが複雑化した場合にも、非プログラマである専門家に対してシステムにおける思考や判断の過程を分かりやすく提示可能であると考えた。

3.2. 意思決定支援システム作成に関わる人物

我々が提案した専門家の知識をもとに構築される意思決定支援システムにおいて、システム作成に関わる人物の役割は以下の 3 つに分けられる。

(1) 専門家

特定の分野において、適切な思考や判断のもとに、問題の解決を行なえる人物。

(2) ナレッジエンジニア

専門家へのインタビューを行ない、専門家の知識に基づいた思考や判断の過程をモデリングし、計算機上への表現を行なう人物。知識処理に関する十分な知識を有する。

(3) コンポーネントプログラマ

ナレッジエンジニアの指示に従い、システムに組み込まれる関数を作成する人物。

この役割を踏まえた上で、今回提案した意思決定支援システムの構築過程について図 3 に書き記す。

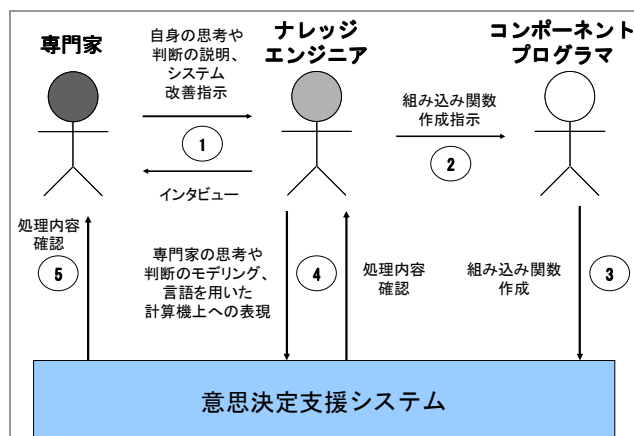


図 3: 意思決定支援システムの構築過程

以上に示した意思決定支援システムの構築過程においては、ナレッジエンジニアが専門家に対してシステムの改善に向けた指摘を求めるという点、及びナレッジエンジニアとコンポーネントプログラマの役割を明確に切り分けた点が従来と異なる。前者には、専門家による信頼を得るまでシステムの妥当性向上に関する作業を行なえるという利点があり、後者には、専門家の処理の表現と、計算機上での処理の表現を明確に切り分ける事ができるという利点がある。この様なモデリングを行なう事により、専門家により直接システム

における思考や判断を把握する事が可能となる。

3.3. システムにおけるデータの流れ

データの流れに着目した上で、複合オブジェクトの様な複雑化したデータを意思決定支援システムにおいて取り扱う為には、まず前述した役割分担をもとに、システムにおける記述の抽象度のレベルを明確に定義する必要がある。今回提案した意思決定支援システムにおける抽象度記述について図 4 に示す。

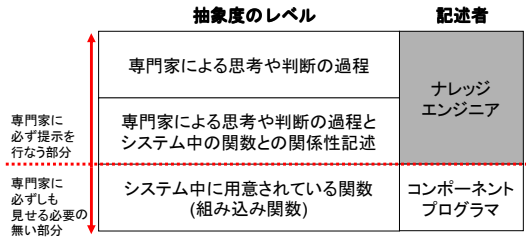


図 4: 記述に関するデータの抽象度

以上の様な抽象度の切り分けを行なう事で、専門家に対して提示する部分を明確に区別する事が可能となる。これにより、非プログラマである専門家が、計算機上における複雑なアルゴリズムの部分の把握する必要がなくなり、システムにおける処理の内容を直観的に把握する事ができるようになる。

ここで更に、本研究で提案したシステムにおいて、専門家に対するデータの提示を行なうまでのデータの流れについて図 5 に書き記す。

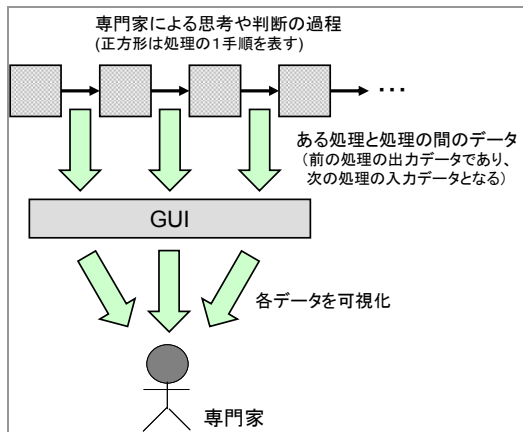


図 5: 専門家へのデータの提示

次に、意思決定支援システムにおける処理の抽象化について図 6 に示す。

我々は、この様に処理の流れを記述する事で、専門家に対するデータの可視化を十分に行なう事が可能であると考えた。本研究ではまず、複合オブジェクトの取り扱いを十分に行なうと共に、専門家に対しても把握し易い形式でのデータの可視化を実現するデータ構造を表現したものである NEW の提案を行なう。

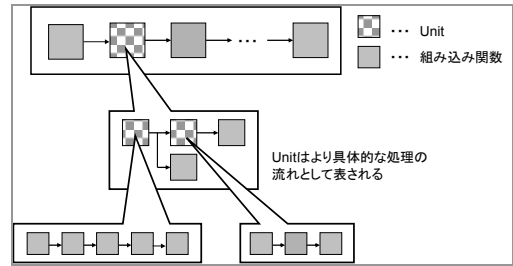


図 6: 意思決定支援システムにおける処理の抽象化

そして本研究では、システムにおいてナレッジエンジニアが専門家に提示を行なう部分と、必ずしも提示を行わなないコンポーネントプログラマが記述した部分を、明示的に分離して記述する事ができるプログラミング言語 MDPL の提案も行なう。MDPL には、複数の MDPL 文により記述された処理を更にまとめた処理単位として、Unit という概念が存在する。この概念は「専門家による思考や判断の過程」を明示的に MDPL 中で記述する為に提案されたものである。

これらの概念を踏まえた上で、今回提案した意思決定支援システムの記述手法を図 7 に示す。

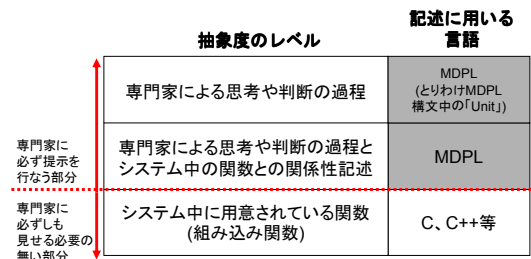


図 7: 意思決定支援システムにおける記述手法

この様な枠組みのもと、今回提案した意思決定支援システムは構築される。

3.4. システム全体のアーキテクチャ

ここで、今回提案した意思決定支援システムにおける全体のアーキテクチャを図 8 に示す。

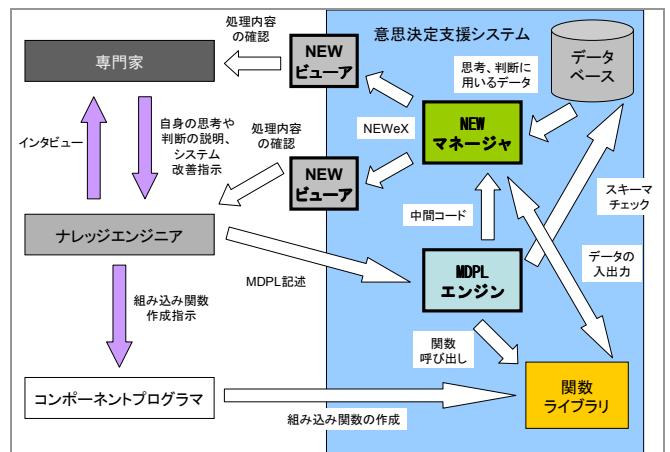


図 8: 意思決定支援システムのアーキテクチャ

図 8の構成要素中に、NEW ビューア、NEW マネージャ、MDPL エンジンが存在する。この部分が、専門家自身により思考や判断の過程を容易に把握できる意思決定支援システムを実現する為の主な提案事項となる。NEW ビューアは、NEW の構造を GUI により可視化する為のアプリケーションとなる部分である。専門家自身は MDPL により記述された内容を直接見るのではなく、NEW ビューアにより可視化された NEW の構造を見る事でデータの流れを把握する。NEW マネージャは NEW を計算機上で取り扱う為の処理機構であり、NEW ビューアへのデータ受け渡しの際に NEWeX を用いる。MDPL エンジンは MDPL の処理環境の事を表す。これらの構成要素に関しては次章において詳述する。

4. NEW と MDPL

4.1. データ構造の表現 NEW

NEW(Nested Entity Window)はデータ構造を表現したものであり、本研究室において提案された。NEW は数値や文字の様な単純なデータと、マルチメディアデータが混在する複合オブジェクトを取り扱う事が可能である。NEW に対しては、後述する MDPL を用いる事で、集合操作を行なう事ができる。NEW 及び後述する NEWeX は、当研究室における先行研究である OE(Object Expression)[4]の拡張仕様である。

NEW は木構造をなすデータ構造を表現したものであり、各ノードに特定の型を持つ。我々は、NEW のデータ構造を、長大かつ大容量という特長を持つマルチメディアデータを含む複合オブジェクトは階層構造となるという一般的性質を踏まえた上で考案している。

NEW における階層の要素は、特定の型に従うインスタンスのみを許容する。この特定の階層の事をランクと呼ぶ。ランクは NEW 特有の概念であり、ランクを単位とした上での集合操作が行なえる事が NEW の利点である。ランクは後述する MDPL の各種オペレータの操作単位としても用いられる。個々のランクを一意に指定する為には、ランク記述子という記法を用いる。ランク記述子はパス表現に基づいており、階層毎の属性名を'/'(デリミタ)で区切る事により、NEW における特定の階層を指し示す事ができる。

以下、NEW の構造制約について述べる。NEW におけるランクは親子関係を有し、root を除くランクは必ず 1 つの親を持ち、子は 0 を含む任意の個数持つ事ができる。ランクは順序付けられた Entity の列を持ち、Entity 自身も親子関係を有する。Entity とは、現実世界における事物、事象の代理物の事を表す。Entity は必ず 1 つの親 Entity を持ち、複数個の子を持つ事ができる。ランクと異なり、Entity は順序付けられる。特定のランク中に存在する個々の Entity の型は一定であ

る。従って、ランク記述子により指し示される NEW における個々の Entity は常に同じ構造を持つ。この NEW の特性を用いる事で、大量のデータ、及び同階層にある同型データの明示的な記述が可能となる。これは NEW の操作言語において常に何らかの意味を持つ操作が行なわれるという事であり、ランク中の個々の Entity に対しては同時操作性が確保される。

NEW を用いる事で、専門家に対するデータの可視化を容易に行なう事が可能となる。NEW が構造を持つ理由としては、以下の 2 点が挙げられる。

1. NEW で表現されたデータ構造は全て、入れ子型の階層構造を持つ Window という分かりやすい形式の GUI を用いて表現する事ができる。
2. GUI を用いたビューアを作成する事で、専門家が行なう論理的な操作を、直観的な操作に置き換えた上で、非プログラマである専門家に対して提示する事が可能である。

ここで NEW を、ビューアを用いて可視化した例を図 9 に示す。

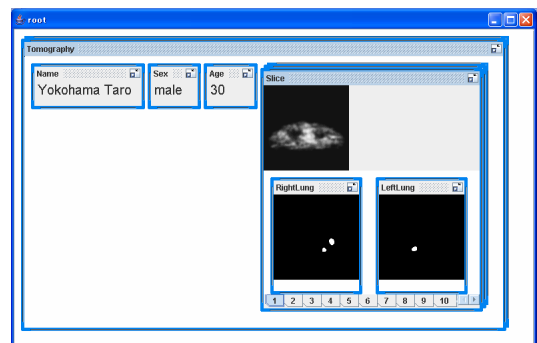


図 9:NEW ビューア表示例

図 9 で用いたデータには、CT(Computer Tomography)を用いて撮影された人体の肺部分のスライス画像や、被験者に関する情報が含まれている。

NEW は、ランクと Entity に関する操作において、以下の表 1 から表 4 で示す ADT(Abstract Data Type)を持つ。(以下の表群におけるメソッドの引数に関する記述は、E:Entity、R:Rank、CS:Character String、I:Integer、T:Type と略記を行なうものとする)

これらの ADT を NEW マネージャにおいて用意する事で、NEW を計算機上で実現する事が可能となる。

4.2. NEW の計算機上での表現 NEWeX

NEWeX(Nested Entity Window eXpression)とは、NEW を計算機上で表現する為の式表現であり、データ構造である。NEWeX で表現された情報は NEW で表されたものと等価である。NEWeX は、NEW ビューアを用いて NEW の表示を行なう際に中間コードの様な役割を果たす。図 10 に、前述した図 9 の NEW を NEWeX で書き表した際の表現例を書き記す。(一部省略を含む)

表 1:NEW の ADT (access 系)

メソッド名	処理内容
getParentEntity (E)	ある Entity の親 Entity を取得する。
getChildEntity (R, R)	あるランクに属する Entity と親子関係を有する子ランクに属する Entity の集合のみを取得する。
getEntities (R)	あるランクに属する Entity 全てを取得する。
getRootEntity ()	root の Entity を取得する。
getEntityRank (E)	ある Entity が属するランクを取得する。
getParentRank (R)	あるランクにおける親ランクを取得する。
getChildRank (R)	あるランクにおける子ランクの集合を取得する。
getRootRank ()	root のランクを取得する。
getRank (CS)	ランク記述子を引数とした上で、ランクを取得する。

表 2:NEW の ADT (述語系)

メソッド名	処理内容
isParentEntity (E, E)	2 つの Entity を引数とした上で、親の Entity であるか真偽の判定を行なう。
isChildEntity (E, E)	2 つの Entity を引数とした上で、子の Entity であるか真偽の判定を行なう。
isParentRank (R, R)	2 つのランクを引数とした上で、親のランクであるか真偽の判定を行なう。
isChildRank (R, R)	2 つのランクを引数とした上で、子のランクであるか真偽の判定を行なう。

表 3:NEW の ADT (add 系)

メソッド名	処理内容
createEntity ()	Entity を生成する。Entity は親を指定する事無く生成できる。
addEntity (E, E, R)	ある Entity を特定のランクに追加する。その際には属するランクと親 Entity を引数で指定する。
createRank (R, R, T)	引数で指定されたランクの子ランクとして新たなランクを追加する。root のランクを生成する為には親の引数を NULL とする。

表 4:NEW の ADT (remove 系)

メソッド名	処理内容
removeEntity (R, R, I)	ある Entity の子ランクに関して、引数で指定された順番の Entity を削除する。
removeAllEntity (R, R)	順番を指定せず、引数で指定されたランクの全ての子 Entity を削除する。(syntax sugar)
removeRank (R)	引数で指定されたランクを削除する。

実際に計算機上に NEWeX を記述する際には、上記

```
Tomography:
{
  [Name: "Yokohama Taro",
  Sex: "male",
  Age: 30,
  Slice:
  {
    <img alt="arrow pointing right" data-bbox="675 148 695 158"/>
    [RightLung: <img alt="arrow pointing right" data-bbox="675 168 695 178"/>,
    LeftLung: <img alt="arrow pointing right" data-bbox="675 188 695 198"/>],
    ... (以下、この部分にスライス枚数分の記述が続く)
  }
}]
}
```

図 10:NEWeX 表現例(一部省略を含む)

の画像部分はバイナリデータで書き表される事になるが、本稿の説明では簡単のために画像を直接 NEWeX の表現中に用いた。

4.3. プログラミング言語 MDPL

MDPL(Multimedia Data Processing Language)は、4.1 節で言及した NEW の為の操作言語である。MDPL を用いる事で、データの流りに着目した上で NEW に対する操作を記述する事ができる。MDPL は当研究室の先行研究である MMQL(MultiMedia Query Language)[4] の拡張仕様である。MDPL は、関数型言語[9]の特性を意識した上での設計がなされている。例えば MDPL の文法では、NEW という特殊なデータ構造の表現に対する入出力操作の記述のみを許容している。そして、MDPL において用いる事ができる組み込み関数は必ず何らかの値を返すものであり、予期しないデータ構造の書き換えを伴う記述を許容しない。この様に、MDPL では副作用が生じないという点に留意した上での文法仕様の策定がなされている。

MDPL の文法では、update、select、delete、reconstruct、retrieve の 5 種類のオペレータが用意されている。MDPL においては、オペレータで始まり、';'(セミコロン)で終了する文字列を操作記述と呼ぶ。操作記述は、4.1 節で示した様にランクをその操作の単位とする。表 5 に、各オペレータを用いて操作記述を行なう際の説明を示す。表中の単位操作とは、ランク中の特定の属性に対する操作の事を示す。具体的には組み込み関数を適用する場合等がこれにあたる。論理式とは、真偽のいずれかを返す式の事を指す。MDPL においては、select オペレータにおいて、残す要素のみが真となる様な条件式を、論理式の部分において記述する。

ここに示したオペレータ群は、意思決定支援システムにおいて必要となるデータに対する操作を、人間の論理的思考や判断の過程に従った上で、それに忠実に従った操作のみを行なう事を目的として定義された。よって、繰り返し文等の操作は、人間の論理的思考の形態として含まれないと考えられる為、MDPL の文法中には存在しない。ただし、組み込み関数を使用する事により、システムの形態に応じて複雑な操作を行なう事も可能である。その為、MDPL の言語としての表現力自体としては、特に不足は生じないと考えられる。

表 5:操作記述に関する説明

操作記述	意味
update<ランク記述子> 単位操作	ランク記述子で指定された列の全ての要素に対して、単位操作で指定された関数を用いて求めた値の代入を行なう。この場合、対象となる NEW 上に指定された属性が存在すれば、その部分の値を更新し、存在しなければ新しい属性を付け加えた上で、その部分に対して値を代入するという操作を行なう。
select<ランク記述子> 論理式	ランク記述子で指定された列の属性の中から、ある条件を満たす要素のみを残し、他の要素を消す。
delete<ランク記述子>	ランク記述子で指定された属性以下の部分を全て削除すると共に、属性自身も削除する。
reconstruct<ランク記述子>	ランク記述子による指定を行う事により、NEW の再構成を行なう。
retrieve<ランク記述子> 単位操作	ランク記述子で指定された既存の列に対して、データベース関数を適用し、その適用結果を新たな属性として NEW に付け加える。

ここで、実際に MDPL を用いて NEW に対する操作を記述した例を図 11 に示す。図 11 の記述例は、前述した図 9 において示した NEW ビューアの表示例、及び図 10 の NEWeX の表現例と対応している。なお、図中の左の数値は行番号を示すもので、MDPL 文には含まれない。

```

1: retrieve</Tomography>Tomography;
2: retrieve</Tomography>Name:TomographyName;
3: select</Tomography>{</Name> & "Yokohama Taro" };
4: retrieve</Tomography>Sex:TomographySex;
5: retrieve</Tomography>Age:TomographyAge;
6: retrieve</Tomography>Slice:TomographySlice;
7: retrieve</Tomography>Slice:Image:SliceImage;
8: unit DiagnosisLung
9: begin
10:   unit ExtractLung
11:   begin
12:     update</Tomography>Slice:Higharea:Threshold( </Img>, 1000 )
13:     ...(<省略>)
14:   end
15:   unit CheckLung
16:   begin
17:     ...(<省略>)
18:   end
19: end

```

図 11:MDPL 記述例

図 11 における一連の操作は、以下の 2 つのフェーズに分ける事が可能である。

1. データベースから、CT を用いて撮影された人体の肺部分のスライス画像と、被験者の情報を取得する。その過程で、特定の被験者以外のデータを消去する操作も行なう。(1~7 行目)
2. 肺のスライス画像から右肺、左肺それぞれの領域抽出を行なう。(8~19 行目)

1 のフェーズでは retrieve オペレータを用いて、データベースに対する問い合わせを行ない、各種データを取得する操作を記述している。データベースからデータを取得する過程において、図 11 の 3 行目では、select オペレータを用いて「Yokohama Taro」という人物以外のデータを消去する操作も記述している。

2 のフェーズでは、「DiagnosisLung(肺の診断)」という Unit 中に「ExtractLung(肺領域の抽出)」、「CheckLung(肺における異常チェック)」という 2 つの Unit を記述した上で、MDPL の操作記述を行なっている。図 11 の 12 行目の update オペレータでは、システムにおける組み込み関数である Threshold を用いて、肺のスライス画像に対してしきい値処理を行なっている。その他にも本来の肺の診断過程では、多数の組み込み関数を用いた上で処理を行なうが、本稿では簡単の為にその部分を省略した。

MDPL の大きな特性となるのが、専門家による思考や判断における一つの過程を、Unit という処理単位で明示的に記述できる点である。Unit を用いて MDPL 文を記述する事で、専門家が行なう思考や判断の一過程を、複数ないし単数の MDPL 文の操作記述をまとめた一つの処理として取り扱う事が可能となる。Unit の中には更に Unit を内包する事も可能であり、複数の抽象度に関する記述を、MDPL を用いて行なう事ができる。

Unit の特性として、一つの Unit における処理が終了した際のデータ構造が、Unit 中に存在する条件分岐処理等に関わらず、常に一定のものになるという点がある。これにより専門家が、Unit を単位とした上で NEW のデータの変化を逐次的に追う作業が容易に行なえる。そして我々は、複雑な専門家の思考に関しても、この様な MDPL の記述の組み合わせにより十分に表現でき、専門家自身による解釈が可能であると考えている。

5. システムに関する評価

本章では今回提案した意思決定支援システムに関する評価、考察を行なう。まず、本稿においてこれまでに述べた結果から、NEW、MDPL を、エキスパートシステムにおいて頻繁に用いられる幾つかのプログラミング言語と比較した場合の、特筆すべき比較点を表 6 に示す。なお、NEW はデータ構造を表現したものであるが、MDPL と組み合わせる事を念頭に置いている為、ここでは NEW と MDPL 全体をプログラミング言語との比較対象として取り扱う。

MDPL ではない他のプログラミング言語を用いても、他言語とのインターフェースを用いる等の手段により、複合オブジェクトを取り扱う事は可能である。しかし既存の言語は、データ構造との結び付きが希薄である為、複合オブジェクトに対して一括処理を行なう手段

表 6:NEW,MDPL と他言語との比較

判断基準	NEW, MDPL	C	Prolog	OPS/R2	KL1
記述内容の可読性確保	○	×	△	×	△
複合オブジェクトの取り扱い	○	△	△	△	△
データの流れてに着目した記述	○	×	△	△	△
計算機特有の概念記述	×	○	×	×	×
自然な並列性の確保	△	×	×	×	△

○:十分サポートされている △:可能である ×:実現し辛い

に乏しく、複合オブジェクトに対する集合操作を行なう事が難しい。

MDPL はデータの流れてに着目した上で、非常に限られた操作しかできない様に文法策定がなされている。実際、計算機特有の概念であるメモリやプロセスに対する処理を伴う手続き的な処理過程の記述を、MDPL を用いて行なう事はできない。しかし、本稿で繰り返し述べて来た様に、本研究においてはシステムにおける思考や判断の過程を専門家に対して提示する際には、データの流れてに着目する事が必要であると考えている。この様な視点から見ると、複雑なアルゴリズムを伴う手続き的な処理過程の記述を許容するプログラミング言語は、非プログラマである専門家に対して必ずしも把握しやすい処理過程を持つとは言えない。そして、同一のシステム構築において複数のナレッジエンジニアが存在した場合、MDPL においては記述された操作の内容自体が可読性に富むものである為、複数のナレッジエンジニアによるプログラムの修正作業を円滑に行なう事ができる。我々はこの様な点を、MDPL を他のエキスパートシステム構築に用いる言語と比較した際の優位性であると考えている。

MDPL におけるその他の利点としては、NEW の更新を行なう update オペレータにおいて、データの参照元、適用先を明確に記述でき、更に一つの操作記述において参照元となるデータが同じものである事から、その部分において並列性の自然な確保が可能である点があげられる。この様な自然な並列性の確保は、実際にシステムを構築する際に、速度向上を実現する上での有益な手段の一つであると言える。

現在当研究室では、NEW マネージャと NEW ビューア、及び MDPL のコンパイラ機構を中心に、並行して開発を進めている[5][6]。本論文提出時点では、部分的な実装は既に完了してはいるものの、未だシステムのプロトタイプに関しては作成過程である。

今後の課題としては、意思決定支援システム全体の構成要素を網羅する様なプロトタイプの完成、及び専門家とナレッジエンジニア相互の話し合いに基づいたシステムの妥当性の検証作業を行なう事があげられる。更に MDPL の操作記述自体は、データの複雑な構造変化を伴うものではなく、操作におけるデータの参照元、及び適用範囲が明確である事から、自然言語への変換が容易であると考えられる。この部分に関しても、今後の研究分野の一つとして考慮されている。

6. まとめ

本稿では、専門家による思考や判断の過程を記述できる意思決定支援システムを構築する為に、まず対象世界をモデリングした。その上で、システムにおける個々の利用者の視点に基づいた抽象度の定義を行ない、複雑化した知識処理を行なう際に、言語とデータ構造を新たに提示する必要性を示した上で、プログラミング言語 MDPL とデータ構造 NEW を提案した。更に、NEW の ADT を示す事、及び部分的な実装作業を行なう事でシステムの実現可能性を示し、最後に他言語との比較により提案事項の優位性を検証、考察した。

今後は、システムのプロトタイプ完成を早急に行なうと共に、医療画像診断等の分野における意思決定支援システムの実用化を目標とした上で、本格的なシステムの各種構成要素の構築、開発を行なっていく。

文 献

- [1] 菅原研次, 人工知能, 森北出版, 1997.
- [2] 飯島淳一, 意思決定支援システムとエキスパートシステム, 日科技連, 東京, 1993.
- [3] S.M.ワイス, C.A.クリコフスキ(共著), 森健一(訳), エキスパートシステムの設計, 近代科学社, 東京, 1987.
- [4] 富井尚志, 有澤博, “マルチメディアデータベースにおける映像モデリングと操作言語”, 電子情報通信学会論文誌(D-II), Vol.J79-D-II, No.4, pp.520-530, Apr.1996.
- [5] 荒井淳, 遠藤智絵, 有澤博, 鈴木晶子, 井上登美夫, “全身 PET を用いたガン診断のモデリングと自動診断システムの構築”, 医学情報処理システム研究会 MIPS2004 論文集, 2005.
- [6] 細金豊, “マルチメディアデータを扱う推論システムの設計”, 横浜国立大学大学院環境情報学府修士論文, 2004.
- [7] E.H.Shortliffe, “Computer Based Medical Consultations: MYCIN”, American:Elsevier, 1976.
- [8] Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, “Knowledge Discovery and Data Mining: Towards a Unifying Framework”, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Aug.1996.
- [9] J.Backus, “Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs”, Communications of the ACM, vol.21, pp.613-641, Aug.1978.