

# 論点提示による分類型グループ知識のインタラクティブな獲得手法

竹中 寿 啓<sup>†</sup> 楠村 幸 貴<sup>†</sup>  
土方 嘉 徳<sup>†</sup> 西田 正 吾<sup>†</sup>

ナレッジマネジメントにおける重要な課題として、暗黙知から形式知を導出し、それを体系化する知識獲得の問題がある。従来の知識獲得の手法として、事例に基づく帰納学習とユーザの領域知識による演繹的な説明付けを組み合わせた手法があった。しかし、この手法に基づく既存のシステムでは、グループ内の個々のユーザがそれぞれの知識を持ち寄り協力して知識獲得しようにも、システムは複数のユーザの知識を扱うことができず、個々人に応じた知識の修正案を提示することはできなかった。本稿では、2人のユーザが議論を行い分類型知識を獲得することを支援する手法を提案する。本手法は、事例から帰納学習して得られた知識と2人のユーザの経験的な知識の3者間で矛盾や相違を検出し、それに基づき知識を改善するための論点を提示する。専門領域としてグレ釣りを取りあげ、実際の釣りの専門家を対象に被験者実験を行い、その有効性を検証する。

## Classification Knowledge Acquisition Method for a Group by Presenting the Issues

TOSHIHIRO TAKENAKA,<sup>†</sup> YUKITAKA KUSUMURA,<sup>†</sup>  
YOSHINORI HIJIKATA<sup>†</sup> and SHOGO NISHIDA<sup>†</sup>

One of the subjects in knowledge management is knowledge acquisition to materialize the implicit knowledge and to formulate it. One traditional knowledge acquisition method is the hybrid method of inductive learning and deductive explanation. However the existing systems based on this method cannot provide questionnaires for improving the knowledge suited to each user by considering each user's knowledge. This paper proposes a method which supports two users to have a discussion to acquire classification knowledge. This method compares the knowledge generated by inductive learning from cases and two user's domain knowledge, and provides the point at issue to improve the knowledge. We conducted a user experiment in the domain of gure-fishing where real experts had participated in.

### 1. はじめに

近年のビジネスのグローバル化とオープン化にともない、組織としての知識をいかに運用するかというナレッジマネジメントの機運が高まっている。ナレッジマネジメントとは、企業の業務における知識をどう加工・利用してビジネスを俊敏で生産性の高いものにするかについての方法論である<sup>1)</sup>。ナレッジマネジメントの基本的な考え方としては、野中らのSECIモデル<sup>2)</sup>やDavenportらの組織論研究<sup>3)</sup>がある。前者のSECIモデルでは、ナレッジマネジメントの方法論として「知識」に重点をおいており、(1)知識には、明確な言語・数字・図表などで表現された「形式知」と、個々人が頭の中に持っているはっきりと明示されてい

ないモデルや技能などの「暗黙知」が存在する、(2)人間が知的活動を行う際には、これら形式知と暗黙知はお互いに作用し合い、形式知は暗黙知へ、暗黙知は形式知へと変化する、(3)組織の知は、異なったタイプの知識(暗黙知と形式知)と異なった内容の知識を持った個人が相互に作用し合うことによって創られることを前提とし、人間の知識運用における一種のライフサイクルを提案している。

このモデルの中でも特に、暗黙知からいかに形式知を導出するかと、いかに導出された形式知を整理し体系立てたものにしていくかが重要となる。これらの課題に対する具体的な方法論としては、エキスパートシステムにおける知識獲得の手法が有効な候補としてあげられる。知識獲得の基礎的な方法論としては、事例から帰納学習するもの<sup>4)-6)</sup>がある。事例とはシステムのログや各種センサなどから得られた、対象の分野において実際に起こったことを記録したデータを意味

<sup>†</sup> 大阪大学大学院基礎工学研究科  
Graduate School of Engineering Science, Osaka  
University

する．しかし，自動獲得した知識はそのままでは実用上満足いく内容にはならないことが多いため，多くのシステムではユーザにも領域知識と呼ばれる領域固有の知識を手作業で入力してもらい，それらにより演繹的な説明付けを行うような折衷的な方法論をとっている<sup>7)~9)</sup>．

この手法の代表例として，KAISER<sup>7)</sup>がある．KAISERは，事例から帰納学習して決定木を生成し，それに対してユーザの領域知識をもって，不都合と呼ばれる知識の矛盾や相違を検出する．さらに，不都合を解消するための質問をユーザに行う．しかし，KAISERは，組織に属するユーザが協調して知識の入力・洗練化を行うことを支援するような工夫は行っていないかった．つまりKAISERは，ユーザが1人で組織のための知識を構築することを想定しており，2人のユーザで知識の構築を行うには，あるユーザがその人の領域知識を基に知識を構築した後，その知識に対して別のユーザがその人の領域知識で検証を行わざるをえなかった．そのため，2人のユーザがお互いの知識を持ち寄り，協力して知識の構築を行おうにも，各々の知識を有効に利用することはできなかった．

本研究では，この問題に対して2人のユーザがリアルタイムに議論を行いながら分類型と呼ばれる知識（詳細は後で述べる）を獲得することを支援する手法を提案する．本手法では，2人のユーザが個々に持つ経験的な知識を明示化しておく．システムは2人の知識間の矛盾や相違を不都合として検出し，それらを解消するための質問を2人のユーザに個別に行う．2人のユーザはそれぞれの質問に基づき，間違っている点や不足している点などを議論し，知識の追加や改善を行う．また，事例からも帰納学習を行い，2人の経験的な知識とは別の知識を構築しておく．上述の不都合検出を，2人のユーザの知識間のみで行うのではなく，この事例から学習した知識も含めて行う．ここで，事例から学習した知識は，2人に第三者的な観点からの論点を提示する役割を果たし，より活発な議論を促進することを目的としている．このようにして，お互いがそれぞれ独立に持っていた経験的な知識の間の整合性を可能な限りとり，グループの中の知識として洗練化を目指す．

本稿の構成は，2章で本手法の概要を述べ，本手法の扱う知識の記述形式について述べる．3章では，本手法の中心となる不都合の定義と，不都合を提示する優先順位，提示先の選択基準について述べる．さらに，優先順位を付けながら不都合を検出するアルゴリズムを示し，発見された不都合に対応する質問文と不都合

となるルールの例を示す．最後に，実装したシステムの概要を示す．4章では，本手法を評価するために，実際の専門家を対象とした被験者実験を行う．その専門分野として，グレ（関東でいうメジナ）という魚を対象とした釣り（グレ釣り）を取り上げる．対象としては，レクリエーションの分野を取り上げているが，専門家の役割を演じた模擬被験者を用いるのではなく，実際の専門家に実験に参加してもらうため，実験を通して得られた知見はある程度価値があるものと考えている．5章では，知識獲得の分野における関連研究について述べ，本研究の同分野における位置付けを明確にする．最後に，6章でまとめを述べる．

## 2. 手法の概要と知識の記述形式

### 2.1 手法の概要

本手法の概要を図1に示し，その使い方を示す．本手法は分類型の知識を扱い，知識はエキスパートシステムで一般的なif thenルールで表現される（それぞれの記述形式と具体例は次節で示す）．2人のユーザは初めに領域に関する各自の経験的な知識を，互いに独立に，相談しないで入力する．以降本稿では，領域に関する経験的な知識を単に領域知識と呼ぶ．また，事例もシステムに入力される．システムは，事例から帰納学習を行うことで，事例に基づく知識を生成する．システムは，2人のユーザの領域知識と事例に基づく知識の間の矛盾や相違を不都合として検出し，それを解消する質問を各ユーザに生成する．ここで，問題になる可能性のある不都合をすべて列挙してしまうと，大量の質問をユーザに提示してしまう恐れがある．そこで，不都合にはあらかじめ種類ごとに優先順位を付けておき，検出された不都合とそれに対する質問はそ

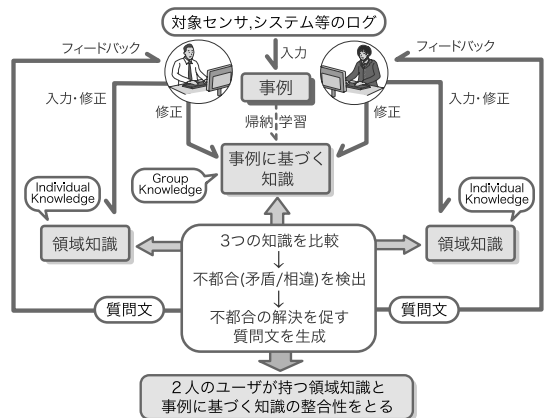


図1 システム概要  
Fig. 1 System outline.

の優先順位に従い並べ替えて提示される。

ユーザはシステムから提示された質問に従い議論を行い、最初に明示化しておいた領域知識を、自分専用の知識 (individual knowledge) として向上させていく。領域知識は、もともと自分が持っている知識を自分で明示化したものなので、後から見直したときに自分が理解するのに適しているといえる。またグループには公開したくなくても持っておきたい知識も含んでいると思われる。一方、事例に基づく知識は、1章で述べたようにユーザ同士の議論に第三者的な立場から論点を提示する役割を果たすのであるが、知識そのものは事例から得られた客観的なものとなっている。つまり、特定の個人に影響されていないニュートラルな知識であるといえる。そこで、事例に基づく知識に対しても、誤って学習されたものに対して修正・削除を行い、また足りない知識を追加していくことで、組織の他のメンバ共有の資産 (group knowledge) として残すこととする。つまり、事例から得られたニュートラルな知識に対して、両ユーザが自信を持って他のメンバに薦められる内容を残していく。このように、個々人の知識を尊重し、かつ確からしい知識を組織のために共有するという観点から、知識獲得を目指す。ここで本研究では、各個人が個別に持っている知識と、お互いが同意してグループに公開することに決めた知識を合わせて、グループが持つ知識 (グループ知識) と考えている。各個人が学習した知識も、後に再度知識獲得を行うときに生かされるものと思われる。

## 2.2 知識の記述形式

本システムは分類型と呼ばれる知識を扱う。分類型の知識というのは、対象とする事象に対して、判断であるクラス、その判断材料である属性、そして属性の選択肢である属性値によって表される知識のことである。属性値は、連続値は対象としないこととし、カテゴリ値のみを対象とする。これは、ユーザの暗黙的な知識を具体的な数値で表すのは困難と判断したためである。事例においてセンサなどから出力された連続値は、あらかじめ人手 (決められたルール) により、カテゴリ値に変換される。また、同じ属性に対して異なる属性値は同時には起こりえないものとする。図 2(a) に、5章で実験対象の領域とするグレ釣りの例を示す。この場合、属性は水温や天気など 11 種類あり、各属性に対して 2~6 種類のカテゴリ値が存在する。判断となるクラスは、「釣れる/釣れない」の 2 値となっている。この例では、真/偽の 2 値問題となっているが、クラスはその他一般のクラス (つまり N 個のクラス) でもかまわない。以下に、事例と知識の記述形式と、

それぞれのグレ釣り問題の場合の例を示す。

### (1) 事例

事例は、対象とする事象に関するすべての属性に対する属性値と、所属すべきクラスの対で表され、その記述形式は、

「(属性 属性値)…(属性 属性値) クラス)」  
となる。グレ釣り問題における事例の集合は、図 2(b) のようになる。

### (2) 領域知識

領域知識は、クラス・属性・属性値から構成される以下の if then ルールで記述される。

「if (属性 属性値)…(属性 属性値) then (クラス)」  
以降、if の部分を条件部、then の部分を判断部と呼ぶ。属性値はカテゴリ値のみを対象とするので、属性と属性値間の関係の記述は等号のみ扱い、不等号や NOT 演算子は扱わない。また、条件節 (属性と属性値の組) 間の関係は、AND のみ扱い、OR は扱わない。OR で記述したい場合は、複数のルールに分ける必要がある。また、判断部は 1 つのクラスのみを書くものとする。判断部において AND (「クラス A かつクラス B になる」) や OR (「クラス A かクラス B になる」)、NOT (「クラス A 以外のクラスになる」) といった、1 つにクラスが定まらなくなる演算子は記述できない。グレ釣り問題における領域知識の例は、図 2(c)、(d) のようになる。

### (3) 事例に基づく知識

事例に基づく知識は、事例から決定木により生成される。決定木とは、中間ノード (葉以外のノード) がテストされるべき属性を、枝がその属性値を、そして葉ノードがクラスを表している木構造のことで、この木構造は事例集合から帰納学習することによって得られる。決定木生成用のアルゴリズムとしては ID3<sup>5)</sup> や C4.5<sup>6)</sup> などが用いられる。決定木の 1 つのパスは、各中間ノードの属性に対する属性値と葉ノードにあるクラスから成り立っている。これは 1 つの if then ルールとみなすことができる。つまり、決定木は葉ノードの数だけある if then ルールの集合だと考えることができる。したがって、決定木のパスの記述形式は、  
「if (属性 属性値)…(属性 属性値) then (クラス)」  
となる。グレ釣り問題における事例に基づく知識の例は、図 2(e) のようになる。

## 3. 不都合の定義と検出方法

本研究における不都合検出の基本的な考え方は、一方のユーザの領域知識、もう一方のユーザの領域知識、事例に基づく知識の 3 つのルール集合から 1 つずつ

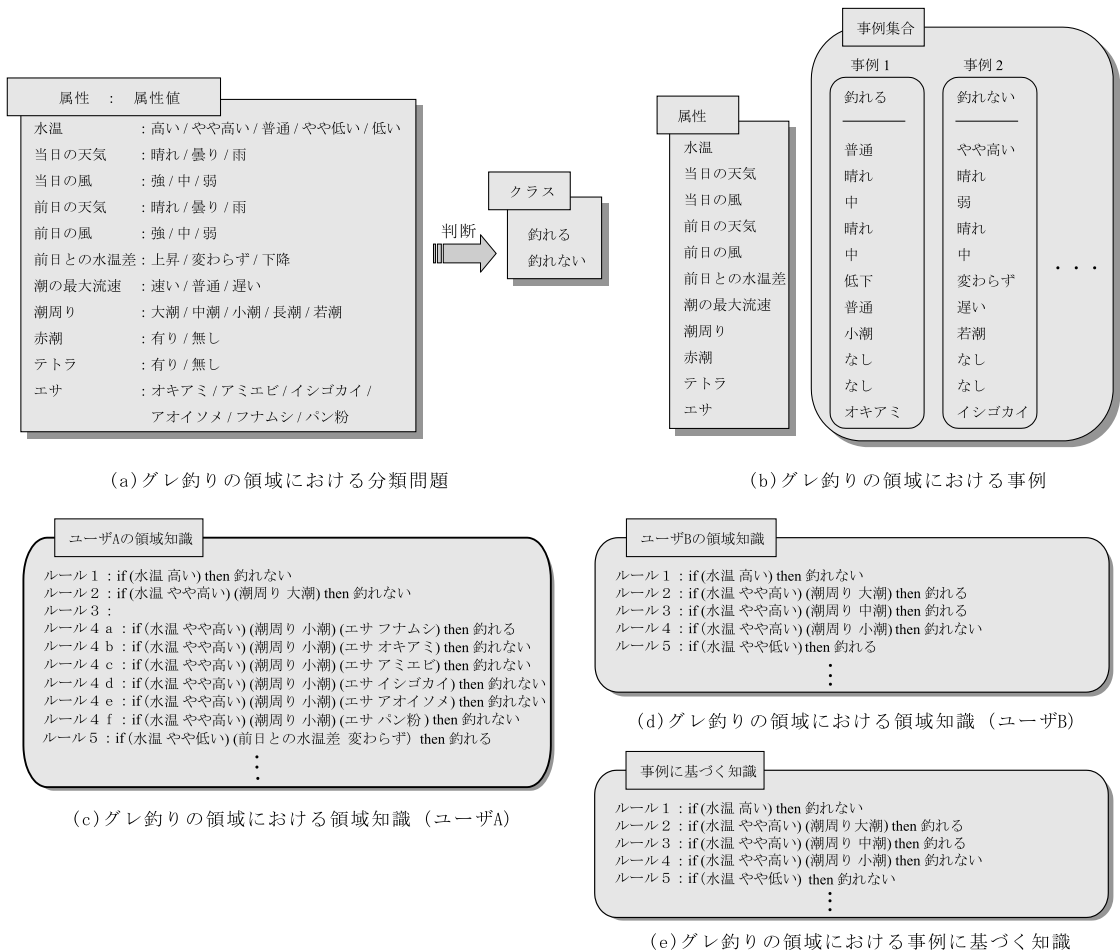


図 2 グレ釣りの領域における分類型知識の例

Fig. 2 Example of classification problems in the gure fishing domain.

ルールを取り出し、その組合せにおいて不都合がないかどうかを検出するというものである。これをすべての組合せに対して行う。さらに、検出された不都合とそれを解消するための質問を、適切なユーザに、ユーザにとって重要と思われる順序で提示する。本章では、まず不都合の定義と不都合をユーザに提示する優先順位について述べる。また、質問を提示するユーザの選択基準について述べる。次いで、それら不都合を優先順位を付けながら検出するアルゴリズムについて述べ、その際にユーザに提示する質問文の具体的な内容について述べる。最後に、不都合となるルールの例を示し、以上の方法を実装したシステムについて述べる。

### 3.1 不都合の定義

本研究の不都合として以下の3つを定義する。

- Type1 (矛盾)

3つのルールの組合せにおいて、2つ以上のルールで条件部が同じ(条件節の数も、その属性と属性値も

まったく同じ)であるが、判断部が異なるものがある。

- Type2 (条件不足・過剰)

3つのルールの組合せにおいて、少なくとも2つのルール間で、条件部に同じ条件節(属性も属性値も同じ)を持つが(2つ以上同じ条件節があってもよい)、片方のルールの条件部から見れば、相手のルールの条件部に自分が用いていない属性を持つ条件節がある(条件不足)。もう片方のルールの条件部から見れば、相手のルールの条件部に自分の用いている属性を持つ条件節が存在しない(条件過剰)。

- Type3 (欠落)

2つのルール集合に同じルール(条件部も判断部も同じルール)が存在するが、残りの1つのルール集合にそのルールが存在しない。

上記の3種類を不都合と定義したのは、以下の理由による。Type1は、それらのルールを書いた人以外の人間が、これらのルールを用いようとしたときに、そ

の条件部の場合にどちらの判断部を採用したらよいか分からないからである。Type2 は、本当に必要な属性を判断に入れていない（不足の場合）、本当は必要のない属性を判断の基準としている（過剰の場合）ことが考えられるからである。Type3 は、ルールが存在しない残りの1つのルール集合が、領域知識のものである場合には、ユーザが書き忘れている可能性やルールを知らない可能性があるためである。また、事例に基づく知識のものである場合は、たまたまそのようなルールを生成する事例が存在しなかったが、両ユーザが合意に達したため、グループにとって有益と思われるからである。

なお、Type2 の場合には、2つのルール間で2つ以上の条件節が異なっても、それらは条件不足・過剰とする。また、条件節の置き換えも含む。すなわち、同じ条件節（属性  $W$ -属性値  $w_1$ ）を持つが（2つ以上同じ条件節があってもよい）、片方のルールの条件部には属性  $X$  を持つ条件節があり（それ以外の条件節は持たない）、もう片方のルールの条件部には属性  $X$  とは異なる属性  $Y$  を持つ条件節がある（同様に、それ以外の条件節は持たない）場合である。つまり、2人のユーザは属性  $W$  が最も重要であることでは一致しているのであるが、判断に決定的な要素を一方では  $X$  として、もう一方は  $Y$  としている場合である。この場合、どちらかが決定的な要素を勘違いしているか、両方の属性が重要なかもしれない。

ここで、条件節の置き換えとなるのは、2つのルール間で異なる条件節において、属性が異なる場合に限られる。すなわち異なる条件節において、属性は同じであるが、属性値が異なる場合は、条件節の置き換えとは見なさない。図2の例では、ユーザBのルール2とルール3は（水温 やや高い）という条件節を共通として持つが、残りの条件節はルール2が（潮周り 大潮）でルール3が（潮周り 中潮）となっている。これらの条件節は、潮周りという同じ属性に対して、属性値が「大潮」、「中潮」とそれぞれ異なっている。異なる属性値が同時に発生することは想定していないため、これらは異なる条件を表しているものと見なす。

また、上記の説明では、簡単化のために互いに置き換えとなっている条件節が、一方のルールでは属性  $X$  に関するもので、もう一方のルールでは属性  $Y$  に関するもので、それぞれ異なっている条件節が1つずつであったが、条件節は複数個異なってもよいものとする。すなわち、もう一方のルールが属性  $Y$  に関する条件節と属性  $Z$  に関する条件節の2つを持っていても、これら2つのルールは互いに置き換えと見

表 1 不都合の付与先  
Table 1 Target to set flaws.

不都合の Type と関連するルール数	不都合の付与先
Type1 (3つのルールが関連)	3つのルールの組に対して
Type1 (2つのルールが関連)	2つのルールの組に対して
Type1 (2つのルールが関連)かつ Type2 (残りのルールが前記2つのルールに対して)	3つのルールの組に対して
Type2 (3つのルールが関連)	3つのルールの組に対して
Type2 (2つのルールが関連)	2つのルールの組に対して
Type3	2つのルールの組に対して

なす。

不都合は、関連するルールの組合せに対して付けられる（表1）。たとえば、ある3つのルールの組合せにおいて、2つのルールが Type1 の不都合で、残りのルールのルール集合に、これら2つのルールの条件部で使われている属性を用いているルールが存在しない場合は、条件部が同じ2つのルールの組に対して Type1 の不都合が付けられる。また、ある3つのルールの組合せで、2つのルールにおいて Type1 の不都合があり、残りのルールがそれら2つのルールに対して条件不足・過剰になっている場合は、これら3つのルールの組に対して、Type1 かつ Type2 の不都合が付けられる。

### 3.2 領域知識入力への制約

ユーザは2.2節の形式に従い各自の領域知識を入力することになるが、その後の知識獲得の効率化のために、以下の制約を設ける。

- 制約1：すでに入力したルールと同じ条件部を持つルールは入力できない。
- 制約2：すでに入力したルールと同じ条件部を含み、さらに別の属性を用いた条件節を追加した条件部を持つルールは入力できない。また、すでに入力したルールの条件部の一部の条件節しか持たない条件部を持つルールは入力できない。

制約1を設けた理由は、条件部が同じであるルールを持つということは、判断部が同じであればルールが二重に登録されていることになり、判断部が異なればルールに矛盾が生じるからである。制約2を設けた理由は、他のユーザの知識と不都合検出を行う前に、自分の知識に条件不足・過剰があると、他のユーザのルールが、上記のいずれかのルールに対して必ず条件不足・過剰となってしまうからである。したがって、常識となるルールが存在し、そのルールで使われていない別の属性がある属性値を満たすときには例外となる場合も、追加の属性についてその他のすべての属性値に関するルールを入力しなければならない。たとえば図2(c)のルール4では（水温 やや高い）潮周り

小潮)という条件では釣れないのが常識で、このときエサがフナムシだけ例外的に釣れるのであるが、その他のエサについても「釣れない」というルールを書いている。

### 3.3 不都合の優先順位

ユーザに提示する不都合の優先順位を考えるにあたって、以下の3つの基準を与える。

- (1) 条件部は同じであるが、判断部が異なるものを優先する。
- (2) 条件部が同じであるルールの数が多いものを優先する。
- (3) 条件不足・過剰があるものはないものよりも優先する。

また、(1)から(3)の順で重要度が高いと考える。この重要度の設定の理由は、次のようである。(1)は明らかなルール間の矛盾であり、知識を利用するユーザに混乱をきたす恐れがある。そのため(1)は、(2)、(3)に比べるとより致命的といえる。(2)は、誰もがその条件を重要と考えている可能性を表し、(3)を提示すればより確かな提案(何を追加・削除すればよいかの提案)ができる。(2)と(3)を比べると、(2)はルールの重要さを示す指標であるのに対し、(3)はその重要なルールをより確かなルールにするのに質問を生成するための指標と考えることができる。ここでは、まずは重要なルールを提示することが重要だとする立場をとり、(2)を(3)より優先させる。

この基準を用いて決めた不都合の優先順位を表2に示す。ここでは、不都合の場合の説明とその優先順位、さらにその場合に対応する不都合の定義のTypeについても示す。なお、不都合ではないが、各人の知識が独自に持っているルールも、すべての不都合の後に優先順位7として表示することとしている。これは、2人は重要と思わなかったが事例には多く存在する条件が、確率的には低い片方のユーザは重要と見なす条件であるため、そのルールを持たないユーザはそれを自分のルールにしたいと考えるかもしれないからである。

### 3.4 質問提示先の選択基準

不都合解消を促す質問を提示するユーザを選択する際に、ユーザの知識の確かさを多数決で判定する。本システムには、3つの種類の知識があるため、あるルールの組合せを選択した際に、そのうち2つの種類の知識においてルールが一致していれば、残りの1つのルールに誤りがあると判定して、そのルールを持つユーザに質問を提示する。そのルールが事例に基づく知識であれば、両ユーザに質問を提示する。ただし、

表2 不都合の優先順位  
Table 2 Priority of flaws.

優先順位	種類の説明	不都合 Type
1	3つのルールとも条件部が同じであるが、判断部が異なるものが存在する場合	Type1
2	2つのルールにおいて条件部が同じであるが判断部が異なる場合で、もう1つのルールが条件不足か条件過剰である場合	Type1 かつ Type2
3	2つのルールにおいて条件部が同じであるが判断部が異なる場合で、もう1つのルールが条件不足や条件過剰でない場合	Type1
4	2つのルールが同じで、もう1つのルールが条件不足か条件過剰である場合	Type2
5	2つのルールが同じで、もう1つのルールの属するルール集合に条件不足や条件過剰であるルールが存在しない場合	Type3
6	同じ条件部であるルールはないが、条件不足や条件過剰となるルールが存在する場合	Type2
7	同じ条件部であるルールも、条件不足や条件過剰であるルールも存在しない場合	

本システムでは多数決をとるのが、3つの知識間と数が少ないため、その判定が高い確率で正しいとはいえない。また、本システムは不都合提示後、リアルタイムな議論によりその不都合を解消するため、一方のユーザにのみ質問を提示すると、他方のユーザはなぜその知識を修正しようとしているのかが分からない。そこで、誤っていると思われるユーザに不都合解消を促す質問を提示し、他方のユーザにはそのような質問が提示された事実を伝えるメッセージを提示する。

### 3.5 不都合検出アルゴリズム

3.1節で定義した不都合を、3.3節で定義した優先順位を付加しながら検出するアルゴリズムを提案する。図3~図6にアルゴリズムのフローチャートを示す。本アルゴリズムの入力は、3種類のルール集合であり、出力は整合済みルールリストと不都合リストである。整合済みルールリストとは、3つのルールの組合せのうち、すべてのルールの条件部と判断部の両方が一致するものを記憶しておくリストである。不都合リストとは、発見した不都合の種類と、不都合が付与されるルールの組合せ、不都合の優先順位を記憶しておくリストである。本アルゴリズムは、3種類のルール集合からルールを1つずつ取り出して、不都合かどうかを確認する作業を、すべての組合せに対して行う(図3)。本アルゴリズムは3つのプロセスから構成される(図4~図6)。

プロセス1(図4)では、取り出した3つのルールの組合せに対して、すべての条件部が同じかどうかを確認する。すべての条件部が同じであれば、各ルールの判断部をチェックして、整合済みかType1の不都合かを判断する。3つのルールの条件部が同じではない場合、プロセス2を実行する。プロセス2(図5)

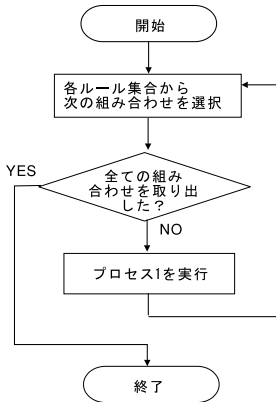


図3 不都合検出フローチャート(全体)  
Fig.3 Flowchart for detecting flaws.

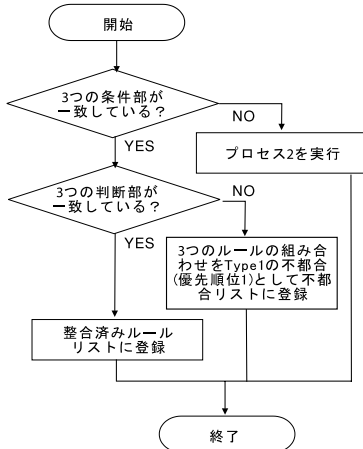


図4 不都合検出フローチャート(プロセス1)  
Fig.4 Flowchart for detecting flaws of Process 1.

では、2つのルールの条件部が同じかどうかを確認する。同じでなければ、プロセス3を実行する。同じであれば、まずその判断部をチェックしてType1の不都合か否かを判断する。さらに、残りのルールが条件部が同じルールに対して条件不足・過剰になっているかどうかを調べ、Type2の不都合か否かを判断する。Type1の不都合でない場合、Type3の不都合か否かを判定するために、条件部が一致していないルールが属するルール集合に対して、条件部が一致している2つのルールに対して条件不足・過剰なものがないかどうかを探る。なければ、Type3の不都合と判断する。プロセス3(図6)では、条件部が一致するルールはないため、条件不足・過剰がないかどうかをチェックする。なければ不都合ではなく、あればType2の不都合と判断する。

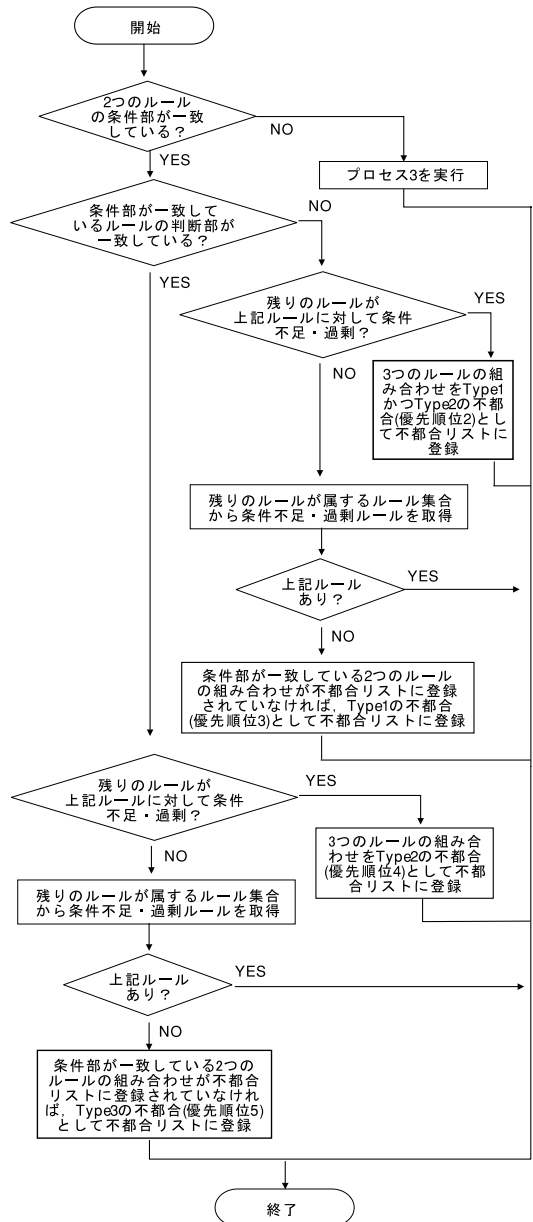


図5 不都合検出フローチャート(プロセス2)  
Fig.5 Flowchart for detecting flaws of Process 2.

### 3.6 メッセージ内容

3.1節で、不都合の付与先は関連するルールの組合せであることを述べたが、そのときにユーザに提示する質問文も、それらルールの組合せと不都合の種類に応じて生成される。質問文の内容を表3に示す。実際に不都合と質問文がユーザに提示されるときは、不都合に関連するルールの組合せと表3のメッセージがそのまま表示される。この表では、ユーザの知識に不都合がある場合の質問文を示しているが、事例に基づく

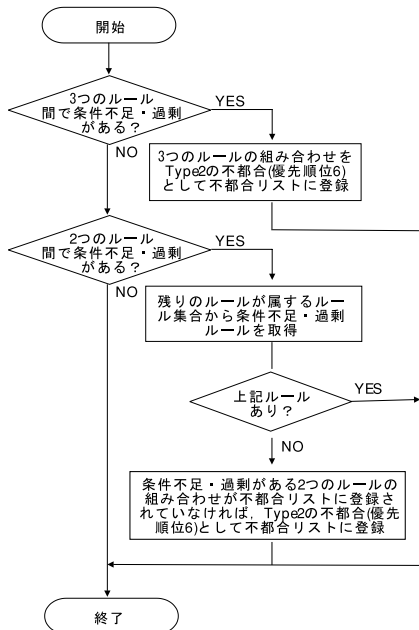


図 6 不都合検出フローチャート(プロセス 3)

Fig. 6 Flowchart for detecting flaws of Process 3.

知識に不都合がある場合は、両ユーザに事例に基づく知識に問題があるという質問文を示す。また、事例に基づく知識の所有者を「システム」と擬人化して、質問文を作成している。

表 3 で 3 つのルールの条件部が一致している場合について説明する。3 つのルールのうち 2 つのルールの判断部が一致している場合には、判断部が異なるユーザに判断が異なっていることを伝える。3 つのルールとも判断部が異なる場合は、全員の判断が異なることを伝える。

2 つのルールのみで条件部が一致している場合は、それら 2 つのルールの判断部が一致している場合と一致していない場合で質問が異なる。2 つのルールの判断部が一致している場合は、さらに Type3 の不都合である場合は、そのルールのないユーザに、ルールがないことを伝える。Type2 の不都合である場合は、ルールが一致していないユーザのルールが間違っていると質問を行う。ここで、不足、過剰、置き換えによってメッセージが異なる。自分のルールが相手のルールに対して条件不足な場合は、相手の使っている属性が判断に影響を及ぼすことを知らないか、知っているもそれほど決定的でないと考え、ルールを過度に一般化している恐れがある。相手のルールに対して、自分のルールが条件過剰な場合は、非常に特殊なケースを取り扱っているか、それほど決定的でない属性を

表 3 メッセージのパターン

Table 3 Patterns of messages to the flaws.

不都合種類	メッセージ
3 つのルールの条件部が一致している場合	
Type1	「あなたの判断だけ異なっています。」
Type1	「全員の判断が異なっています。」
2 つのルールが一致している場合	
Type3	「あなたには、このルールが存在しません。」
Type2	(不足の場合)「あなたのルールには の属性がありません。ルールが過度に一般化されている恐れがあります。」
Type2	(過剰の場合)「あなたのルールには の属性が余分に付いています。ルールが過度に特殊化されている恐れがあります。」
Type2	(置き換えの場合)「あなたのルールでは、決定的要素を の属性としています。 の属性が決定的である可能性があります。」
2 つのルールで判断部は異なるが条件部が一致している場合	
Type1	「あなたの判断は、 さんと異なっています。」
Type2	(不足の場合) 一方に「あなたのルールは、 さんと さんのルールに対して、 の属性がありません。」 一方に「 の属性は判断に関係のない恐れがあります。」
Type2	(過剰の場合) 一方に「あなたのルールは、 さんと さんのルールに対して、 の属性が追加されています。」 一方に「 の属性まで考慮しないと判断できない可能性があります。」
Type2	(置き換えの場合) 一方に「あなたのルールでは決定的要素を の属性としていますが、 さんと さんのルールでは、決定的要素を ×× の属性としています。」 一方に「判断を決める決定的要素は ×× の属性ではなく、 の属性である可能性があります。」
条件部の一致がない場合	
Type2	(不足の場合)「あなたのルールは、 さんのルールに対して、 の属性がありません。お互いにルールを確認してみてください。」
Type2	(過剰の場合)「あなたのルールは、 さんのルールに対して、 の属性が追加されています。お互いにルールを確認してみてください。」
Type2	(置き換えの場合)「あなたのルールでは決定的要素を の属性としていますが、 さんのルールでは、決定的要素を ×× の属性としています。お互いにルールを確認してみてください。」

決定的と考え、ルールを過度に特殊化している恐れがある。また、相手のルールに対して条件の属性が置き換わっている場合は、判断に決定的な属性を勘違いしている可能性がある。それぞれ、以上に述べた質問を行う。

2 つのルールの判断部が一致していない場合は、まずこれらのルールを持つユーザに判断が誰と異なるかを伝える。さらに Type2 の不都合である場合は、条件部が一致していないユーザのルールが、条件部が一致しているユーザのルール修正に参考になるものとして質問を行う。ここで、不足、過剰、置き換えによってメッセージが異なる。条件部が一致していないユーザのルールが条件不足の場合は、そのユーザには条件が他の 2 人のルールから見て不足していることを伝える。残りの 2 人のユーザには追加の属性が判断に影響しない可能性があることを伝える。なぜなら、その属性によって判断が難しくなり、結果として判断が割れている可能性があるからである。条件過剰の場合は、



そのユーザには条件が他の2人のルールから見て過剰であることを伝える。残りの2人のユーザにはその足りない属性が判断に影響する可能性があることを伝える。なぜなら、その属性まで見ていないために判断が割れている可能性があるからである。置き換えの場合は、そのユーザには条件が他の2人から見て置き換えになっていることを伝える。残りの2人のユーザには、判断を決める決定的な属性が異なる可能性があることを伝える。なぜなら、属性を変えて見れば、判断が一致するかもしれないからである。

いずれのルールも条件部が一致していない場合は、Type2の不都合のみありうる。この場合も、不足、過剰、置き換えによって質問文が異なる。この場合、どのユーザのルールに誤りがあるかは推定できないので、誰に対して条件が不足または過剰があるかだけを伝える。置き換えの場合は、決定的要素をお互いに確認するように伝える。

### 3.7 不都合になるルール例

図2(c)–(e)に不都合になるルールの例を示す。ルール1は、全員の条件部と判断部が一致しているので、不都合ではない。ルール2は、条件部は全員一致しているが、ユーザAのみ判断部が異なるため、Type1の不都合である。ユーザAに判断が正しいかどうかを尋ねる。ルール3では、「if (水温 やや高い) (潮周り 中潮) then 釣れる」というルールが、ユーザBの知識と事例に基づく知識には存在するが、ユーザAの知識には存在しないため、Type3の不都合である。この場合、ユーザAにこのルールが存在しないことを伝える。ルール4は、ユーザBの知識と事例に基づく知識で(水温 やや高い) (潮周り 小潮)なら「釣れない」ことで一致しているが、ユーザAはこれにエサまで含めて判断しようとしている。これは2つのルールが一致している場合のType2の不都合である。この場合、ユーザBの知識と事例に基づく知識が正しいと仮定すると、ユーザAはエサまで含めて考えることは、過度に特殊化しているといえる。そこで、ユーザAには、エサという属性は判断に関係のないことを、すなわちルールを過度に特殊化していることを伝える。ルール5は(水温 やや低い)では、ユーザBは「釣れる」として、事例に基づく知識では「釣れない」と意見が分かれている。しかし、ユーザAは前日との水温差まで考えており、実は(水温 やや低い)だけでは判断できず、前日との水温差まで考えることにより判断できるケースなのかもしれない。これは、条件部が同じで判断部が異なるルールがあるときのType2の不都合である。この場合、ユーザBに前

日との水温差まで考える必要がないかを尋ねる。

### 3.8 不都合検出システム

システムは、Windows 2000上で開発言語にMicrosoft VC++6.0を用いて実装した。本システムは、それぞれのユーザのPC上で独立に動作する。相互の知識の通信にはTCP/IP(具体的には、片方のPCがFTPサーバを兼ねており、一度ファイルに出力してから、FTPでファイルを交換する)を用いている。決定木にはID3を用い、学習の際には枝刈りを行う。

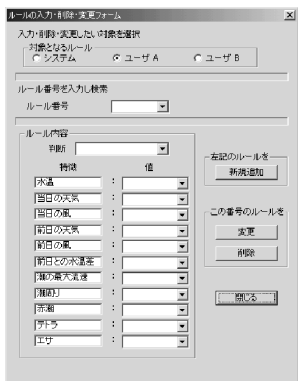
本システムのメニューは、「ルール入力」と「表示」から成っている(図7(b)–(d)のメニューバー)。「ルール入力」のメニューによって、ユーザはルールの入力・変更・削除を行うことができる(図7(a))。ルール内容の欄で、そのルールの内容を決める。判断と値は、リストボックスになっており、値を選択し組み合わせることによってルールを表現する。

「表示」のメニューによって、ウィンドウに表示する項目を変更することができる。表示できるのは、ルール一覧画面(図7(b))、決定木表示画面(図7(c))と不都合検出画面(図7(d))である。ルール一覧(図7(b))では、システム(事例に基づく知識)、ユーザA、ユーザBの3種類のルールを表示することができる。基本的には各ユーザが確認するルールは、自分のルールとシステムの持つルールであるが、相手のルールも閲覧することが可能のようにしている。決定木表示画面(図7(c))では、システムが事例から帰納学習したルールを見ることができる。決定木の各葉ノードの横には、その葉ノードに該当する事例の件数と割合が表示されている。件数と割合により、ユーザはそのルールがどの程度信頼できるものかの目安とすることができる。不都合検出画面(図7(d))では、不都合の検出結果を見ることができる。ユーザは、この画面の不都合を参考にルールの変更・削除・追加を行い、不都合を解消する。

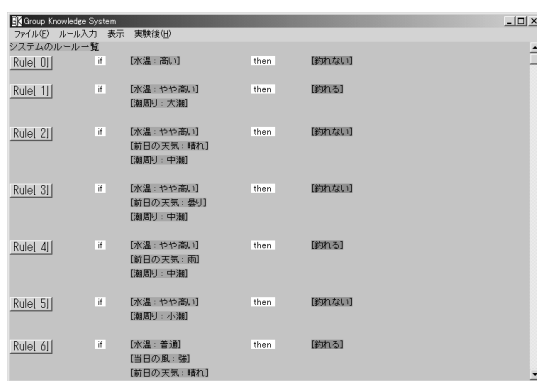
## 4. 評価実験

### 4.1 実験の目的

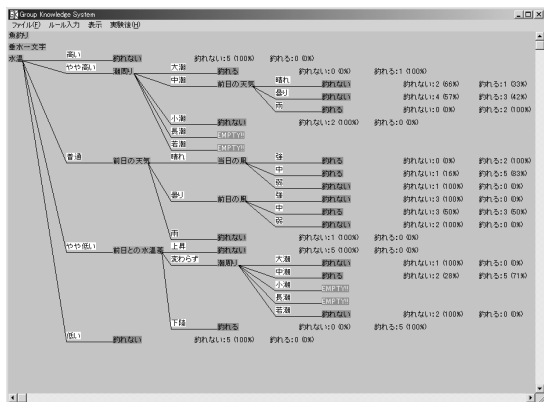
本手法の特徴は、事例に基づく知識と2人の領域知識を用いて不都合を検出し、各ユーザへの質問を不都合の優先度順に出力することにある。そこでこの評価には、まず不都合の分類に関する妥当性の検証と、不都合の優先順位(提示順序)に関するユーザの主観的評価の確認を行う。また、本手法に基づくシステムの最終的な目標としては、グループとしての知識獲得を支援することにある。そこで、システムを利用しない場合と利用した場合を比較し、お互いがより協調して



(a) ルール入力画面



(b) ルール一覧表示画面



(c) 決定木表示画面



(d) 不都合検出画面

図 7 システム画面例

Fig. 7 Screenshots of the system.

多くの知識を獲得できたか否かを評価する。

評価項目をまとめると以下の3点である。

- 評価 1 「不都合の分類の妥当性」
- 評価 2 「不都合の優先順位の主観的評価」
- 評価 3 「グループとしての知識獲得を支援できているか否か」

これらの項目の評価では、シミュレーションや専門家の役割を演じた模擬被験者や模擬課題を使った実験ではなく、実問題に対して実際の専門家に使ってもらい、実験を行う。評価1の「不都合の分類の妥当性」の評価は、検出された不都合のうち実験後に修正されずに残されたものの数を調べ、また被験者にアンケートによって各不都合を残した理由を尋ねることで行う。評価2の「不都合の優先順位」の評価は、被験者にアンケートによって不都合の提示順序に違和感がなかったか否かを調べることで行う。また、優先順位ごとの検出された不都合の数の分布を調べ、その評価に至った原因について考察する。評価3の「グループとしての知識獲得を支援できているか否か」は、本システム

を使用する場合と使用しない場合の2種類の実験を行い、ユーザのルールに対する修正操作回数を調べることで、アンケートによって主観的な評価を尋ねることで行う。

#### 4.2 実験の問題領域の設定

知識獲得システムが有効な問題領域としては、結果の原因にいくつかの条件が複雑に絡み合っており、人によって判断が微妙に異なるような領域が適している。そのような領域としては、工場や発電所の制御、また農業や漁業における収穫・出漁の意思決定などがあげられる分野であり、普段の日常生活からは離れた分野といえる。パソコンのネットワーク診断などは適当そうに見えるが、パソコンは人間が作り出した物であるために、ある条件に対して起こりうる問題がほぼ確実に定まってしまう。それに対応する知識を持っていれば誰が見ても判断が同じになり、結果として議論の必要がなくなってしまう。

実験のために実際の専門家を探すのは困難であるの

だが、本研究では筆者らの1人が魚釣りの専門家であり、そのコミュニティにも精通しているため、魚釣りの分野を対象として実験を行うこととする。魚釣り自体はレクリエーションの分野ではあるが、上級者に限って実験を行えば、各被験者はかなりのノウハウを持っているものと思われる。なお、魚釣りの中でも、本研究では「垂水一文字」と呼ばれる釣り場におけるグレという魚を対象とすることとする。垂水一文字を選択した理由は、潮の変化の激しい明石海峡出口付近にあり、自然条件が大きく変わるためである。グレを選択した理由は、グレはチヌなどの他の魚よりも、ある程度数が釣れることと、釣果（その数の差）が自然条件によりはっきりと差が出て、偶然性の少ないターゲットであるためである。

#### 4.3 問題設定

本実験で、獲得目標とする知識は、「どのような条件のもとで、グレが釣れるか否か」という知識である。クラスと属性、属性値は、図2(a)のように設定した。属性値の分類については、水温は1年分の実データから明石海峡付近でとりうる水温の範囲を均等に5段階に分けた。前日との水温差と潮の最大流速は、1年分の実データに対して、平均からの標準偏差で3段階に分けた。

各データのうち、水温データは神戸市立海釣り公園<sup>10)</sup>および兵庫県立農林水産技術総合センター<sup>11)</sup>の協力を得て獲得した。その他の自然条件に関するデータは、次の各ホームページから得た。天気データ：「気象庁・電子閲覧室<sup>12)</sup>」、潮の流速：「シマノメンバーホームページ<sup>13)</sup>」、赤潮情報：「大阪府立水産試験場<sup>14)</sup>」、潮のデータ：「全国の潮汐スクリプト<sup>15)</sup>」。

釣り座にテトラがあるかどうかと使用したエサのデータは、釣り師からの釣行データとともに得た。釣行データは、筆者らの1人が運営する「神戸波止グレ考釣会<sup>16)</sup>」というWebサイトにおいて、グレ釣りの上級者に直接呼びかけて獲得した。獲得したデータは、垂水一文字でのグレ釣りに限定し、日付とともにどれくらい釣れたのかを「まったく釣れない」「あまり釣れない」「普通に釣れる」「非常によく釣れる」の4段階で提供してもらい、提供されたものに対して、実験者が上位2段階を「釣れる」、下位2段階を「釣れない」とした。呼びかけにより、6人の釣り師から合計66件の事例を提供してもらった。新聞や雑誌などの釣果情報を使うのではなく、信頼できる一定の技術レベルの釣り師からのみ獲得しているため、可能な限りきれいなデータを集めているといえる。学習の結果、19個のルールが生成された。



図8 実験風景  
Fig. 8 Experimental environment.

#### 4.4 実験方法

2003年12月から2004年1月の期間で、グレ釣りの上級者12人（20代から40代の男性11人と女性1人で、釣り雑誌編集者、釣りのポータルサイト運営者、全国大会優勝者を含む）を被験者として実験を行った。実験風景を図8に示す。このように、2人並んでお互いに議論しながら知識獲得を行ってもらった。12人で6ペアできるが、このうち3ペアを不都合検出の機能を持つシステムで（グループA）、残りの3ペアを不都合検出の機能を持たないシステムで（グループB）、実験を行った。

実験の流れとしては、次のようになる。初めに実験者が実験の趣旨およびシステムの説明を被験者に行った。その次に、システムの操作方法の習得と知識獲得のプロセスの習得のために、「テーマパークが混雑するか否か」という練習課題に対して、各被験者に領域知識を入力してもらい、知識の構築を行ってもらった。その後、本実験となるグレ釣りにおける知識獲得を行ってもらった。本実験では、まず初めに個々の被験者の持つグレの釣れる/釣れないに関する領域知識を入力してもらった。次に、グループAはシステムの不都合検出機能を用いて知識の構築を行い、グループBはシステムの不都合検出機能を用いずにルール表示機能だけを用いて、知識の構築を行った。実験後、グループAの各被験者は実験後アンケートを記入した。

ここで実験後アンケートは、以下のことを4択式または5択式で尋ねている。

- (1) システムの質問がきっかけとなって、新たな知識が得られたか？(4択式)
- (2) システムの質問がきっかけとなって、自分の知識が洗練されたか？(4択式)
- (3) システムによる質問の順番は、知識を構築する際にやりやすかったか？(5択式)

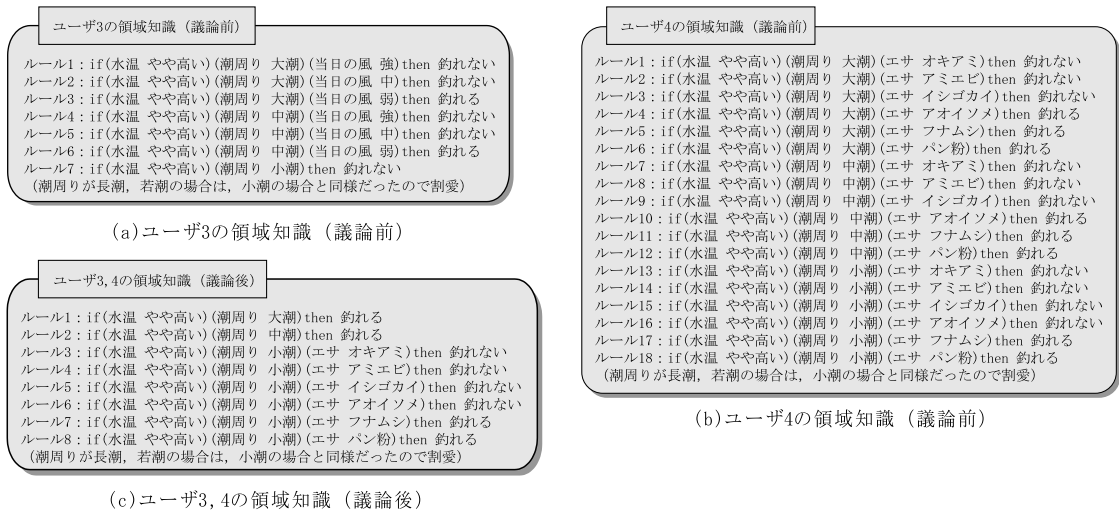


図9 議論による知識の修正前と修正後の例  
 Fig. 9 An example of rule modification after the discussion.

また、グループ A は知識獲得を終了したときに、システムがその時点で残っていた不都合を表示し、なぜそれらを残したのかを各不都合ごとに以下のような選択肢で尋ねている。

- (1) ルールの判断や条件の差は、どちらでもよいかから変更しなかった。
- (2) ルールの変更が面倒であった。
- (3) ルールの変更を忘れていた。
- (4) 意見が違うため変更しなかった。
- (5) 相手がルールを変更しなかった。
- (6) あいまいで分からないため、自分では判断しなかった。
- (7) そもそもこれを不都合と呼ぶのはおかしい。
- (8) その他

#### 4.5 実験結果

##### 4.5.1 不都合検出とルール変更の一例

まず、各評価項目の評価結果を示す前に、図9に典型的なルール変更の一例を示す。また、その変更をなぜ行ったのかを会話の内容から説明する。すべての被験者が作成したルールにおいて一致したのは、グレが釣れるかどうかを最も決定付ける属性は水温だということであった。また、どの被験者もルール記述に苦労したのは、水温がやや高いときに釣れるか釣れないかを判断することであった。図9では、ユーザ3とユーザ4の変更の結果を取り上げている。

最初ユーザ3は、水温がやや高いときというのは、餌取りが多すぎるため、特定の条件を除いて釣れないものと判断していた。その特定の条件というのは、潮がよく動くときで、風がないときである。つまり、

ユーザ3は潮がよく動くときは、泳ぎの下手な餌取りはあまり沖まで出ることができないため、またグレの活性が高くなるため、釣れると考えていたのである。しかし、風があるときは、仕掛けが遠くまで飛ばず、餌取りの遊泳範囲を越えることができないため、釣れないという判断していたのである。これに対しユーザ4は、水温がやや高いときに、決定的な属性となるのはエサであると判断していた。すなわち、餌取りがあまり好まないエサを使えば、グレの口までエサを届けることができると判断したわけである。

不都合検出の結果、条件不足・過剰が示されたため、ユーザ3とユーザ4は次のような議論を行った。ユーザ3は「水温が高いときにグレが釣れるのは、潮がよく動くときで、当日の風が弱いときだけだ」と主張していたが、ユーザ4は「当日の風が強いときというのは、釣り辛いのは事実だが、海が適度に荒れるためグレの活性が高くなり、また餌取りの遊泳範囲が狭くなる。したがって、大きなウキを使い、ウキを沈めるような対策を行えば、風に関係なく釣れる」と主張した。また、「潮周りが小潮のときに釣れないのは、エサに餌取りに弱いオキアミやイシゴカイを使うからで、餌取りに強いパン粉やフナムシを使えば釣れる」と主張した。ユーザ3はパン粉やフナムシといったエサを使っていなかったため、そこまでの知識がなかったのであるが、その意見を聞き、エサを変えることで釣れるようになるのはもっともであろうと考えて、ユーザ4に合わせてルールを変更した。ユーザ4も、潮周りが大潮(中潮)のときに、オキアミでも釣れると主張したユーザ3の意見を聞き、「確かに潮変わりなどグレの

表 4 不都合ごとの検出数・残存数・解消率

Table 4 The number of detected flaws, and the solution ratio in each type of flaw.

Type	ペア 1	ペア 2	ペア 3	全ペア	解消率 (%)
1	0/0	0/2	0/1	0/3	100
2	0/55	1/63	0/33	1/151	99.3
3	0/38	4/29	0/26	4/93	95.7
独自	7/50	3/57	0/33	10/140	92.9

活性が一瞬高くなるときには釣れる」と判断し直し、結果としてユーザ 3 とユーザ 4 は同じルールに落ち着いた。

4.5.2 不都合の分類の妥当性

評価 1 の「不都合の分類の妥当性」について検証する。表 4 に、各ペアごとに（および全ペアで）、不都合の Type 別に、検出不都合数と最終的に修正/削除を行わずに残した不都合の数を示す（残存数/検出数で表す）。また、その不都合を解消した割合（1-残存数/検出数）も示す。また各自が独自で持っているルールに関しても同様に示す。ここで、いくつかの不都合と独自に持っているルールが、修正/削除が行われずに残っている。そこで、各不都合を残した理由のアンケートの結果を見ると、以下ようになった。

(Type2)

- 意見が違うため変更しなかった (1 件)。
- 曖昧で分からないため、自分では判断しなかった (1 件)。

(Type3)

- その他 (2 人とルールは一致しているが、グループの知識に追加するほど自信が持てなかった) (8 件)。

(独自に持っているルール)

- 曖昧で分からないため、自分では判断しなかった (16 件)。
- 意見が違うため変更しなかった (1 件)。

不都合の解消率はすべての Type において 95%以上であった。また、アンケート結果を見ると不都合を残した理由はすべてルールの内容に基づくものであり、「そもそもこれを不都合と呼ぶのはおかしい」という不都合の分類が原因となって残されているものはないことが分かる。これより、不都合の分類は妥当だったといえる。

また、お互いの考え方が合わず独自に持つルールがいくつか残されている点と、2 人とルールは一致しているがグループに公開するほど自信が持てなかったという不都合が残されている点から、グループに公開するための知識と各ユーザのための知識とを分けて構築できることを有効に利用していることが分かる。

表 5 質問の順番に関するアンケート結果

Table 5 The result of questionnaire about the order displayed flaws.

やりやすい	どちらかといえば やりやすい	普通	どちらかといえば やりにくい	やりにくい
3	1	1	0	0

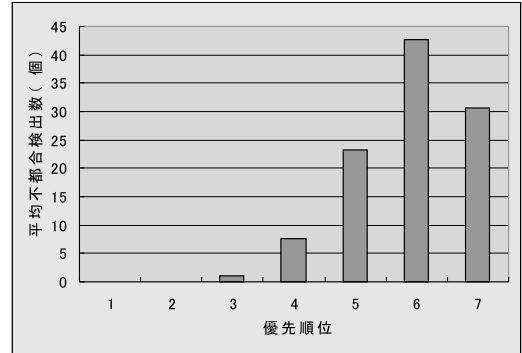


図 10 優先度ごとの平均不都合検出数  
Fig. 10 The number of detected flaws in each priority.

4.5.3 不都合の優先順位の主観的評価

本研究では、不都合の優先順位をどれほど致命的かどうかで定義したが、実際にその優先順位に従って不都合を提示することが、ユーザの主観的評価においてもどれほど使いやすかったのかを確かめる。実験後アンケートの「システムによる質問の順番は、知識を構築する際にやりやすかったか？」に対する結果を表 5 に示す。その結果、提示される不都合の順番にはおおむね不満がないことが分かる。また、優先順位ごとに不都合検出数の平均をグラフ表示したものを、図 10 に示す。図 10 より、優先順位が下がれば不都合検出数が上がることが分かる（優先度 7 は、もともと不都合ではないため、例外と見なす）。実験では、ユーザ側から見ると、最初に少数の致命的な不都合が提示され、それが終わると致命的とはいえないがお互いに修正しておいた方がよい不都合が、数が多くなる順序で列挙されたことになる。多数の不都合が列挙されればユーザにとって負担は増すが、不都合解消作業中にそれが多くなってきても、その致命的な度合いは低くなっていくことから、疲れれば作業を中止して完了してもやむをえないという安心感もあったと推測される。そのため、このような主観的評価に至ったと思われる。

4.5.4 グループとしての知識獲得を支援できているか否か

評価 3 の「グループとしての知識獲得を支援できているか否か」を評価するために、不都合検出機能を使用した場合と使用しなかった場合の実験結果を比較す

表 6 ルールの変更・削除・追加回数 (不都合検出あり)

Table 6 The number of rule operations for changing, deleting and adding (with flaw detection).

	ユーザ 1	ユーザ 2	ユーザ 3	ユーザ 4	ユーザ 5	ユーザ 6
変更	4	3	13	2	2	7
削除	3	7	11	11	4	0
追加	14	18	20	12	25	24

表 7 ルールの変更・削除・追加回数 (不都合検出なし)

Table 7 The number of rule operations for changing, deleting and adding (without flaw detection).

	ユーザ 7	ユーザ 8	ユーザ 9	ユーザ 10	ユーザ 11	ユーザ 12
変更	0	2	0	4	1	2
削除	1	0	20	5	1	5
追加	6	0	7	1	0	2

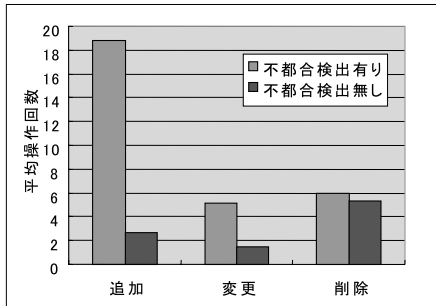


図 11 ルール操作回数の比較

Fig. 11 Comparison of the number of rule operations.

る。各ユーザごとの自分のルールを変更・削除・追加した回数に関して、不都合検出機能を使用した場合を表 6 に、不都合検出機能を使用しなかった場合を表 7 に示す。さらに、これらルール操作回数の全ユーザの平均を図 11 に示す。

図 11 より、不都合検出ありの場合が不都合検出なしの場合に比べて、ルールの追加・変更・削除のすべての操作において回数が上回っていることが分かる。t 検定 (片側) を行った結果、追加は  $p = 0.0001$ 、変更は  $p = 0.0462$ 、削除は  $p = 0.4281$  であったことから、ルールの変更と追加回数に対して有意差 ( $p < .05$ ) が見られた。また、ルールの削除回数に有意差が見られなかった原因は、表 7 のユーザ 9 の削除回数が 20 回と突出していることが大きく作用している。ユーザ 9 の削除回数が、非常に高い値となった理由を説明する。初期入力でユーザ 9 は、最初はルールと事例との区別が不明確なようであったため、冗長性のあるルール (整理されておらずたくさんの属性を使ったルール) を多く入力していた。ユーザ 10 にルールの冗長性を指摘され、冗長性のあるルールをすべて削除したため

表 8 新たな知識の獲得に関するアンケート結果

Table 8 The result of questionnaire about the acquisition of new knowledge.

多く得られた	ある程度得られた	ほとんど得られなかった	まったく得られなかった
2	3	1	0

表 9 知識の洗練に関するアンケート結果

Table 9 The result of questionnaire about the refinement of knowledge.

多く洗練された	ある程度洗練された	ほとんど洗練されなかった	まったく洗練されなかった
1	4	1	0

ある。このことを考慮すれば、削除を含めて、不都合検出機能を使うことによりルールの操作回数が増えることが分かる。つまり、ルールを獲得・洗練するきっかけをより多くユーザに与えていると解釈でき、グループとしての知識獲得を支援できていると考えられる。

また、グループ A の実験後アンケートの「システムの質問がきっかけとなって、新たな知識が得られたか?」と「システムの質問がきっかけとなって、自分の知識が洗練されたか?」に対する回答結果を表 8 と表 9 に示す。これらの表からも、システムを使ったグループは、おおむね新たな知識が得られたことと、自分の知識を洗練できたことが実感できたことが確認できる。

#### 4.6 会話の分析

以上の節では定量的な評価指標とアンケートの結果から、システムの有効性が検証できたが、ここでは不都合を解消するときのユーザの会話を分析することにより、より深い洞察を得ることとする。我々は実験を通じて経験的に、ユーザがルールを獲得・洗練していくときの会話は、以下の 5 つパターンに分けられることが分かった。

- パターン 1 お互いの経験から納得している場合
- パターン 2 その経験はないが、説明に納得している場合
- パターン 3 その経験はないが、相手を信頼して変更する場合
- パターン 4 議論がかみ合わず、相手に合わせようと変更する場合
- パターン 5 相手からの指摘なしに、自分で考え直した場合

パターン 1 は、相手がルールについて詳しく説明することなしで納得した場合や、お互いがそのルールに関する経験をあげることで、すぐに納得した場合である。パターン 2 は、相手がルールについて詳しく説明

表 10 議論のパターンに基づくルール操作回数  
Table 10 The number of rule operations due to the discussion.

パターン	ペア 1		ペア 2		ペア 3	
	ユーザ 1	ユーザ 2	ユーザ 3	ユーザ 4	ユーザ 5	ユーザ 6
1	15	19	19	13	23	25
2	2	5	11	2	3	2
3	0	1	6	0	0	0
4	0	0	0	0	8	0
5	1	3	7	7	4	3

することで自分が納得することができた場合である。パターン 3 は、相手がルールについて説明をしたが、その根拠を納得できていないにもかかわらず、相手を信頼してルールを獲得・洗練した場合である。パターン 4 は、相手とルールについて議論したが議論が噛み合わず、相手のルールに納得しているわけでもないが、その場を収集させるために相手に合わせてルールを獲得・洗練した場合である。パターン 5 は、自分のルールを見て自分のルールを変更・削除した場合や、相手に自分のルールの説明をしているときに思い直してルールを変更・削除した場合である。上記のパターンごとに、各ユーザのルール獲得・洗練時の会話を分類した。その結果を表 10 に示す。

ペアごとの人間関係は、ペア 1 が初対面同士（ただし、ユーザ 1 がユーザ 2 より上級者）、ペア 2 が友人関係かつ師弟関係（ユーザ 3 が弟子、ユーザ 4 が師匠）、ペア 3 が友人関係（両者の釣りレベルに差がない）である。ユーザのルール獲得・洗練パターンには、この人間関係が大きく影響すると実験の様子より考えられた。

釣りのレベルの差の影響と思われる点が、表 10 のパターン 2 である。ペア 1 と 2 では、ユーザ間にパターン 2 の値に差が見られる。ペア 3 は、それほど値に差はない。この値の差は、相手が自分より釣りが上級または下級であるという意識と、実際の知識量の違いから生じていると考えられる。

次に、釣りのレベルの差と師弟関係の影響が出ていると思われる点が、表 10 のパターン 3 である。師弟関係が成り立っているペア 2 のパターン 3 の値は 6 と 0 であり、他のペアでは 0 か 1 である。その様子が一番顕著に表れたシーンは、ユーザ 4 が、「自分にも理由は分からないけど、この条件ですごく釣れたときがあったから、自分のルールにはこのルールを入れておく」と言ったとき、ユーザ 3 が、「そんなことがあったなら、自分にも追加する」と言って追加した場面である。しかし、逆にユーザ 3 の説明できないこだわりをユーザ 4 が認めることはなかった。これは、師弟と

いう人間関係から生まれた現象であると考えられる。パターン 2 とパターン 3 から本手法による知識獲得では、確かな知識を持つユーザの意見に収束しやすい傾向があるといえる。

ペア 3 は、ペア 1 やペア 2 と比べると、釣りに対して確固たる知識をそこまでは持っていないペアであった。そのため釣りに対するこだわりが少ないことから、各ルールに対して、それを疑うということは少なく、お互いに納得して変更・追加する傾向があった。表 10 のペア 3 のパターン 1 の値（23, 25）が、他のペアのパターン 1 の値（15, 19）（19, 13）よりも大きくなっているところに、その傾向が表れている。ユーザ 6 は、自分のものも他人のものも、どんなルールに対しても何らかの理由付けを行い、それを口に出していることで、納得しようとするタイプであった。しかし、その理由付けがこじつけになっている場合があり、ユーザ 5 は困惑する場面が多く見られた。そのような場合、ユーザ 5 は相手に気を遣う性格であったため、相手に合わせてルールを獲得・修正していた。これが、表 10 のユーザ 5 のパターン 4 が 8 回も発生した理由である。本システムは、お互いがその場で知識を洗練・獲得していくものであったため、このように優しい性格をしているユーザや、ユーザ間に上下関係がある場合には、それが妨げになり自分専用の知識を思うように構築していけない問題がある。

パターン 1 は、自分だけではうまくまとめることができなかった知識を相手との議論により、完全な知識として明示化できたことを意味し、パターン 2, 3 は、自分にはまったくなかった考え方や経験を相手から説明され、自分の知識としたことを意味し、パターン 5 はシステムの手助けで、自分の意識の中では弱かった条件などを想起できたことを意味する。これらの点で、システムからの論点提示が知識の洗練化、獲得につながったといえる。しかし、パターン 4 は、ユーザの性格やグループの人間関係により発生しうる問題であり、今後はこのようなことが起きないように工夫が必要となる。

## 5. 関連研究

関連研究として、知識獲得に関するものと議論支援システムに関するものについて取り上げる。

### 5.1 知識獲得

知識獲得はエキスパートシステムの初期の頃からの課題である。知識を得るのに最も信頼できる方法は、インタビューである<sup>17)~19)</sup>。インタビューは何らかの質問戦略に基づいてなされるのが効率的と見なされ、

あるドメインのエキスパートシステムの開発経験を生かして質問戦略の一般化を試みたり<sup>17),20)</sup>、いくつかの観点に基づく質問戦略を提案してそれらを組み合わせたりしている<sup>21)</sup>。また、インタビューで獲得した知識の洗練化のきっかけを提示するような研究もある。Eshelmanらは、インタビューのみで行う知識獲得と実際にエキスパートシステムを試作して試用中に問題が生じたときに行う知識獲得とを組み合わせている<sup>22)</sup>。また、ユーザに知識の不足を気付かせるために、獲得した知識を獲得ときに利用した知識の表現方法とは違う方法で見せる手法<sup>23),24)</sup>が提案されている。

知識獲得を自動化する研究としては、事例から帰納学習<sup>4),5)</sup>するものがあげられる。代表的なシステム・手法として、KAISER<sup>7)</sup>、Lebowitzの研究<sup>8)</sup>、Towellらの研究<sup>9)</sup>がある。KAISERやLebowitzの研究は、帰納的学習の結果の演繹の説明付けを試みることにより、意味的・理論的裏付けをともなった知識を獲得している。Towellらの研究では、領域知識はそもそも不完全なものとし、事例から領域知識をよくしようとしている。しかし、上記のシステムや手法では、知識を組織のために1つだけ考えており、本手法のように知識を、事例に基づく知識、あるユーザの知識、さらに別のユーザの知識と分けて、これらの中で知識の矛盾や相違の検出を行うようなことはしていない。このため、2人のユーザがそれぞれの知識を持ち寄り、組織のための知識を獲得しようとしても、それらを人手で1つにまとめる必要があった。

また、不都合に注目すると、KAISERにおいても知識の矛盾や相違を不都合として検出している。KAISERで検出する不都合は、領域知識による不都合だけでなく、帰納学習(決定木)の結果に対しての、構造的な不都合も検出している。その意味で、本手法における不都合はKAISERのサブセットと見ることができる。しかし、KAISERでは事例に基づく知識とユーザの領域知識間の矛盾や相違のみを見ており、もう1人のユーザが増えたときに、誰の知識に誤りの可能性があるかなどまでは示唆していない。

グループでの知識獲得を支援する別の試みがある。AQUINAS<sup>29)</sup>というシステムでは、複数のユーザに属性と判断からなるRating gridと呼ばれる行列を埋めてもらう。その行列を用いて、ユーザ間の類似性・相違・重なりを見つけている。その結果により、ユーザ同士の話し合いを誘導している。これは、本手法と目的は同じで、2人に限らず3人以上の複数のユーザにおける知識獲得も行える。しかし、最初にユーザに埋めてもらうのは単一の属性からなるルールに相当し、

本手法のように複雑なルールを扱うことができない。また、GRAPE<sup>30)</sup>ではグループ内の複数のユーザがリアルタイムに議論し合って、知識ベースとなる分類木を作成できるようなインタフェースとチャット機能などを提案している。しかし、GRAPEは初めにユーザは、自分の考える完成されたルールを入力するわけではないので、各ユーザが持つ知識を最大限利用しているわけではない。これら2つのシステムと比較すると、本手法は初めにユーザが考える完成されたルールを入力するところから始まり、そのルール間の不都合を検出する。その意味で、条件部が複雑なルールを扱うことができ、なおかつ各ユーザが持つ知識を最大限生かして、知識獲得ができると思われる。

## 5.2 議論支援システム

提案手法は、矛盾や相違のない知識を構築することを目的としているが、知識を洗練化するための論点提示により、二次的な効果として2人のユーザ間の議論を支援していることにもなっている。ここでは、これまでグループウェアの分野でさかんに研究されてきた議論支援システムとの関連を述べる。これまでの議論支援システムは、(1) 会話の流れをモデル化しておき、発言に明示的なタグを付けることにより、計算機の支援を可能とするものと、(2) プレーンストーミングからアイディアの収束までの会議の流れを支援するものとに分けられると考える。

(1) の代表的なシステムとしては、Conversation Theory(会話理論)に基づくシステムであるThe CoordinatorやIBISモデルに基づくシステムであるgIBISなどがある。会話理論<sup>31)</sup>とは、会話中の発言を要求、提案、逆提案、約束、拒否などに分類し、状態間の遷移に上記のラベルを付けることで、会話を状態遷移図で表すようにしたものである。これに関連する研究としては、会話モデルを動的に変更するためのツールキットを提供したSTRUDEL<sup>32)</sup>や、1つの会話を提案、合意、実行、満足の4行程からなるループとして表現したAction Workflow<sup>33)</sup>などがある。IBISモデル<sup>34)</sup>とは、問題、案、意見とその間の関係(賛成/反対、質問、提案、一般化/特殊化)からなる議論をモデル化したものである。会話理論と同様に「私は誰その意見に反対である」というラベルを自分の発言に付ける。また、議論のモデル化としては議論を主張、根拠、論拠、反証、裏付けから構成されるとしたToulminのモデル<sup>35)</sup>が有名であり、最近ではこのモデルを環境教育における議論支援システムに応用した例がある<sup>36)</sup>。これらのシステムは、自分の発言に会話中での役割に関するラベルを明示的に付けること



で、議論の明確化や議論の中で結論が出ていないような部分の提示を行うことができる。しかし、議論対象の中身の整合性については考慮しておらず、一貫性を持った知識を構築するには不十分である。

(2) の代表的なシステムには、Cognoter<sup>37)</sup> や GrIPS<sup>38)</sup> がある。Cognoter は、ブレインストーミングのモデルに基づき、グループによるアイデア生成を支援するシステムである。グループメンバが思い付いたキーワードを画面に貼り付けて、それを全メンバで配置を変えたり、リンクを張ったりすることで、アイデアの構造化を行う。GrIPS は、KJ 法<sup>39)</sup> をベースにしたシステムで、Cognoter と同様に画面にキーワードを情報カードの形式で貼り付けるところから発想支援を行う。Keyword Associator というツールでアイデアを連想的にふくらませるキーワードを提示し、D-ABDUCTOR というツールで画面上的の情報カード間の関係を自動描画する支援を行っている。また、最近の研究例では、KJ 法の考え方をベースにしつつ、より上位レベルの議論支援をも試みるものがある。グループ思考支援環境 AIDE<sup>40)</sup> では、グループ思考を個人思考モードと意思疎通モード、協同思考モードから構成されるとし、ブレインストーミングのフェーズだけでなく、より広く議論支援を行おうとしている。これらのシステムでは、アイデアをキーワードや短い文章、さらには図など、非常に柔軟に表現することができる。しかし、アイデアの関係性を明確にすることは可能であるが、自動的に論理関係の矛盾や相違を検出することはできない。

## 6. 結 論

本研究では、事例から帰納学習（決定木）により獲得した知識と 2 人のユーザの経験的な知識間の不都合を検出し、それを基にユーザに論点提示を行うことで、ユーザが議論により個人の知識とグループの知識を獲得・洗練できる手法を提案した。この手法を実現するにあたって、不都合の定義を行い、不都合の提示の優先順位を整理し、優先順位を付けながらこれら不都合を検出するアルゴリズムを構築した。また、3 つの知識間の条件と判断の一致・部分一致の関係からユーザに提示する質問文を整理した。

そして、本手法を実装したシステムを用いて被験者実験を行い、不都合分類の妥当性の検証、不都合の優先順位に関するユーザの主観的評価、グループの知識獲得を支援できているかの検証を行った。その結果、不都合の解消率は 95% 以上となり、アンケートからも不都合の分類が原因となって残されている不都合はな

かったため、不都合の分類は妥当であると判断した。また、不都合の提示順序に関するアンケートでは、おおむね使いやすいとの回答を得た。最後に、不都合検出機能を使用した場合と使用しなかった場合の 2 通りを比較したところ、ルールの操作回数において不都合検出機能を使用した場合の方が高かった。このことは、システムがユーザが議論する機会をより多く与えたと考えられるため、グループとしての知識獲得を支援できているということが検証されたといえる。

今回の手法ではユーザを 2 人に限定している。そのため、組織において複数の人が参加して知識獲得を行うためには、最初に 2 人がシステムを用いて知識の構築を行い、構築された知識に対して、別の 2 人がシステムを用いて知識の構築を行うことになる。しかし、3 人以上のユーザが一度に集まることが可能な環境において、このように順番にシステムを利用していき、知識の構築を行うのは効率が悪い。今後は、3 人以上のユーザが同時に参加できるようにし、さらにその環境において議論の発散を防ぐため、3 人目以降の参加者に問題提起者やオブザーバのような役割を決めて、知識獲得のプロセスに加わるような改良が必要と考える。

## 参 考 文 献

- 1) 野村直之：ナレッジマネジメントツールの配備、実践動向と次世代技術、人工知能学会誌、Vol.16, Vol.1, pp.33-41 (2001).
- 2) 野中郁次郎、梅本勝博：知識管理から知識経営へ—ナレッジマネジメントの最新動向、人工知能学会誌、Vol.16, Vol.1, pp.4-14 (2001).
- 3) Davenport, T. and Prusak, L.: *Working Knowledge*, Harvard Business School Press (1998).
- 4) Mitchell, T.M., Utgoff, P.E. and Baserji, R.: Learning by Experimentation: Acquiring and refining problem-solving heuristics, *Machine Learning*, pp.163-190, Springer-Verlag (1984).
- 5) Quinlan, J.: Induction of Decision Trees, *Machine Learning*, Vol.1, pp.81-106 (1986).
- 6) Quinlan, J.: *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publishers 2929 Campus Drive, Suite 260 San Mateo, CA94403 (1993).
- 7) 辻野克彦、西田正吾：帰納的学習と演繹的説明づけに駆動された知識獲得システム：KAISER, 人工知能学会誌、Vol.7, Vol.1, pp.149-159 (1992).
- 8) Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol.10, pp.219-240 (1986).
- 9) Towell, G.G., Shavlik, J.W. and Noordewier,

- M.: Refinement of Approximate Domain Theories by Knowledge-based Neural Networks, *AAAI-90*, pp.861-866 (1990).
- 10) 神戸市立海釣り公園 .  
http://www.umiduri.com/
  - 11) 兵庫県立農林水産技術総合センター .  
http://agri.pref.hyogo.jp/nsiweb/web2/nougi/
  - 12) 気象庁・電子閲覧室 .  
http://www.data.kishou.go.jp/
  - 13) シマノメンバーホームページ .  
https://fishmember.shimano.co.jp/member/member.asp
  - 14) 大阪府立水産試験場 .  
http://www.pref.osaka.jp/osakana/
  - 15) 全国の潮汐スクリプト . http://www.ancs.net/jsaf-tokyo/index.html
  - 16) 神戸波止グレ釣約会 . http://members.at.infoseek.co.jp/hiji/
  - 17) 川口敦夫, 溝口理一郎, 角所 収: インタビューシステムのためのシェル, SIS, 人工知能学会誌, Vol.4, No.4, pp.411-420 (1989).
  - 18) Michie, D.: Current Developments in Expert Systems, *Applications of Expert Systems*, Quinlan, J.R. (Ed.), pp.137-156, Wokingham, UK: Addison-Wesley (1987).
  - 19) Michie, D.: Problems of Computer-aided Concept Formation, *Applications of Expert Systems*, Quinlan, J.R. (Ed.), Vol.2, pp.310-333, Wokingham, UK: Addison-Wesley (1989).
  - 20) 川口敦夫, 松山映二, 沼田 薫, 溝口理一郎, 野村康雄, 角所 収: インタビューに基づくデータベース論理設計支援エキスパートシステム, 人工知能学会誌, Vol.4, No.4, pp.421-430 (1989).
  - 21) 滝 寛和, 椿 和弘: 知識獲得のための知識表現「専門家モデル」, 人工知能学会誌, Vol.5, No.2, pp.203-212 (1989).
  - 22) Eshelman, L. and McDermotto, J.: MOLE: A Knowledge Acquisition Tool That Uses Its Head, *Proc. AAAI'86*, pp.950-955 (1986).
  - 23) Kahn, G.: Strategies for Knowledge Acquisition, *IEEE Trans. PAMI*, No.5, pp.511-522 (1985).
  - 24) Boose, J.: A Knowledge Acquisition Program for Expert Systems based on Personal Construct Psychology, *Journal of Man-Machine Studies*, Vol.23, pp.495-525 (1986).
  - 25) Compton, P. and Edwards, G.: Ripple Down Rules: Turning Knowledge Acquisition into Knowledge Maintenance, *Artificial Intelligence in Medicine*, Vol.4, pp.47-59 (1992).
  - 26) 和田卓也, 堀内 匡, 元田 浩, 鷲尾 隆: Ripple Down Rules 法における知識獲得の特性評価に基づくデフォルト知識の決定規範, 人工知能学会誌, Vol.15, No.1, pp.177-186 (2000).
  - 27) Compton, P., et al.: A trade-off between domain knowledge and problem-solving method, *11th Banff Knowledge Acquisition for Knowledge Base System Workshop*, Canada (1998).
  - 28) Beydoun, G. and Hoffmann, A.: Hierarchical Incremental Knowledge Acquisition, *12th Banff Knowledge Acquisition for Knowledge Base System Workshop*, Canada (1999).
  - 29) Boose, J.H. and Bradshaw, J.: Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge-Acquisition Workbench for Knowledge-based Systems, *International Journal of Man-Machine Studies*, Vol.26, No.1, pp.3-28 (1987).
  - 30) 國藤 進, 上田晴康, 岩内雅直, 大津健太: グループ問題解決のための知識獲得支援ツール「GRAPE」, 日経 AI 別冊, pp.144-153, 日経 BP 社 (1991).
  - 31) Winograd, T. and Flores, F.: *Understanding Computers and Cognition*, Addison-Wesley Publishing Company, Inc (1986).
  - 32) Shepherd, A., Mayer, N. and Kuchinsky, A.: Strudel—An Extensible Electronic Conversation Toolkit, *Proc. CSCW'90*, pp.93-104 (1990).
  - 33) Medina-Mora, R., et al.: The Action Workflow Approach to Workflow Management Technology, *Proc. CSCW'92*, pp.281-288 (1992).
  - 34) Conklin, J. and Begeman, M.L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *Proc. CSCW'88*, pp.140-152 (1988).
  - 35) Toulmin, S.E.: *The Uses of Argument*, Cambridge University Press (1985).
  - 36) 鮫島良太ほか: 環境教育のための議論モデルと議論支援システムの設計と構築, ヒューマンインタフェース学会研究会報告集, Vol.4, No.4, pp.11-16 (2002).
  - 37) Gregg, F. and Stefik, M.: Cognoter, Theory and Practice of a Collaborative Tool, *Proc. CSCW'86*, pp.7-15 (1986).
  - 38) 神田陽治ほか: グループ発想支援システム: GrIPS, 人工知能学会, Vol.8, No.5, pp.65-74 (1993).
  - 39) 川喜田二郎: KJ 法, 中央公論社 (1986).
  - 40) 西本一志ほか: マルチエージェントによるグループ思考支援, 電子情報通信学会論文誌, Vol.J81-D-I, No.5, pp.478-487 (1998).

(平成 16 年 12 月 17 日受付)

(平成 17 年 11 月 1 日採録)



竹中 寿啓

昭和 54 年生．平成 14 年 3 月大阪大学基礎工学部システム科学科卒業．平成 16 年 3 月同大学大学院基礎工学研究科システム人間系専攻修士課程修了．同年 4 月西日本電信電話（株）入社．知識獲得の研究に従事．



楠村 幸貴（学生会員）

昭和 54 年生．平成 13 年 3 月大阪大学基礎工学部システム工学科卒業．平成 15 年 3 月同大学大学院修士課程修了．同年 4 月同大学大学院博士課程入学．テキストマイニング，情報抽出の研究に従事．



土方 嘉徳（正会員）

昭和 48 年生．平成 8 年 3 月大阪大学基礎工学部システム工学科卒業．平成 10 年 3 月同大学大学院修士課程修了．同年 4 月日本アイ・ピー・エム（株）東京基礎研究所入所．平成 14 年 11 月より，大阪大学大学院基礎工学研究科システム創成専攻助手．平成 15 年 8 月電気学会「論文誌 C 発刊 30 周年記念」特集最優秀論文賞，平成 17 年 3 月インタラクシオン 2005 ベストペーパー賞，各受賞．知的 Web 技術，パーソナライゼーション，テキストマイニングの研究に従事．電子情報通信学会，IEEE ほか会員．博士（工学）．



西田 正吾（正会員）

昭和 27 年生．昭和 49 年 3 月東京大学工学部電子工学科卒業．昭和 51 年 3 月同大学大学院修士課程修了．同年 4 月三菱電機（株）入社．同社中央研究所システム基礎研究部研究員，グループマネージャーを経て，平成 7 年 4 月大阪大学基礎工学部システム工学科教授．現在，大阪大学大学院基礎工学研究科システム創成専攻教授．システム技術，ヒューマンインタフェース技術，メディア技術の研究に従事．昭和 59～60 年，MIT メディアラボ客員研究員．昭和 61 年度，平成 5 年度電気学会論文賞，平成 4 年度電気学会著作賞，平成 7 年度電気学会進歩賞，平成 13 年度ヒューマンインタフェース学会論文賞，平成 16 年度電気学会業績賞，各受賞．電気学会，IEEE ほか会員．IEEE Fellow．工学博士．