

論点提示によるグループ知識獲得手法

土 方 嘉 徳[†] 竹 中 寿 啓[†]
楠 村 幸 貴[†] 西 田 正 吾[†]

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る手法があった。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、システムは知識の持ち主を識別して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、事例から帰納学習することで知識を生成し、生成した知識とあるユーザの経験的な知識、さらに別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質問を生成する手法を提案する。専門領域としてグレ釣りを取り上げ、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。

Group Knowledge Acquisition Method by Presenting the Issues

YOSHINORI HIJIKATA,[†] TOSHIHIRO TAKENAKA,[†]
YUKITAKA KUSUMURA[†] and SHOGO NISHIDA[†]

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 suited to each user for improving the knowledge. This paper proposes a method which compares the knowledge generated by inductive learning from cases, one user's domain knowledge and another user's domain knowledge, and provide 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伴い、組織としての知識をいかに運用するかと言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¹⁾。ナレッジマネジメントの基本的な考え方としては、野中のSECIモデル²⁾が有名である。SECIモデルでは、(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も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挙げられる。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^{3),4)}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経験的な知識を手作業で入力してもらい、それらにより演繹的な

[†] 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka
University

説明付けを行う折衷的な方法論を採っている^{5)~7)}。

この手法の代表例として、KAISER⁵⁾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持って、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。つまりKAISERは、ユーザが一人で組織のための知識を構築することを想定しており、二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得なかった。そのため、二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、事例から帰納学習した知識と、あるユーザの領域知識、さらに別のユーザの領域知識の3者間で不都合を検出する。そして、それに基づき知識を改善するための質問をユーザ個別に生成し、ユーザ個人用の知識とグループ用の知識を分けて構築できる仕組みを提案する。目指すところは、組織に属する二人の人間に対して議論の論点を提示することで、暗黙知から形式知への変換を支援し、異なった内容の知識をもった個人が相互に作用することで誘発される形式知の向上を支援することにある。

本手法の概要を図1に示し、その使い方を示す。本手法は分類型の問題（詳細は後で述べる）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。二人のユーザは初めに各自の領域知識を、互いに独立に相談しないで入力する。そして、システムから提示された論点に従い議論を行い、各ユーザの知識を自分専用の知識(individual knowledge)として向上させていく。また、事例に基づく知識も、誤って学習されたものを修正・削除を行い、また足りない知識を追加していくことで、向上させていく。ただしこの際、事例に基づく知識は、組織の他のメンバー共有の資産(group knowledge)として残していく目的で変更を行う。つまり、多くの事例から得た客観性の高い知識をベースとして、それに両ユーザが自信を持って他のメンバーに薦められる内容を知識として残していく。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、初めに各自が領域知識を相談しないで入力するのは、各自の持つ知識を議論の種として最大限に提供してもらうためである。また、3

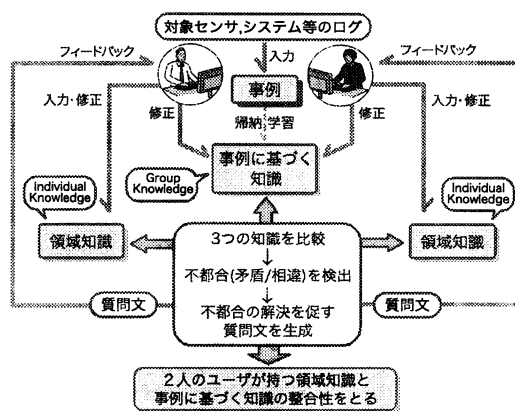


図1 システム概要
Fig. 1 System outline

つの知識によっては大量の不都合が検出されることも有り得る。この場合、ユーザへの質問を一度にかつ無秩序に表示してしまうと、ユーザを混乱させてしまう恐れがある。したがって、質問は不都合に付けられた優先順位に従い、並べ替えて提示される。

以上をまとめて、本手法の範囲を定義しておく。本手法は、2人のユーザの知識とそれらユーザが属するグループの知識間の不都合を検出して、2人のユーザに議論の論点を提示することで、各ユーザに知識の改善を促す仕組みを指す。また、その仕組みではグループの知識は、あらかじめ事例から帰納学習しておく。さらに本手法は、これら3つの知識間の不都合を自動検出するためのパターンと、そのパターンの提示優先順位、そのパターンに応じた論点提示のメッセージ、そして不都合を検出して優先順位に従いメッセージを提示するアルゴリズムから構成される。

本稿の構成は、2章で知識の記述形式について述べる。3章では不都合のパターンとそのパターンに対応するユーザへのメッセージの整理と、それらを提示する優先順位の整理を行う。また、そのパターンを検出して、優先順位に基づき検出されたパターンを提示するためのアルゴリズムを示す。さらに、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（関東で言うメジナ）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ビジネスにも広く利用可能と考えている。最後に、5章でまとめをのべる。

2. 知識の記述形式

本システムは分類型と呼ばれる問題を扱う。分類型の問題というのは、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問題のことである。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。

(1) 事例

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

「(属性 属性値) … (属性 属性値)(クラス)」
となる。

(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)」

(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。従って、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。

3. 不都合の整理と検出方法

本研究における不都合検出の基本的な考え方は、2つの領域知識のルールと事例に基づく知識のルールに対して、総当りで比較し、あらかじめ決めておいた不都合のパターンが存在するか否かを判断するというものである。不都合のパターンが存在した場合、不都合と共にその不都合を解消するための質問をユーザに提示する。

本章では、本システムにおける不都合を定義し、この定義に従い3種類のルールの条件と判断の全ての組み合わせに対して、不都合とその不都合を解消するためのユーザへの質問を整理する。また、検出した不都合をユーザに順序良く提示するために、不都合の優先順位を整理する。最後に上記で整理されたルールの組み合わせと優先順位に基づき、不都合を効率よく検出

するための手順を述べる。

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)

各知識からのルールを1つずつ比較したときに、条件が同じであるが判断が異なる組み合わせ

• Type2 (条件不足・過剰)

各知識からのルールを1つずつ比較したときに、条件が異なり、その条件に不足または過剰がある組み合わせ（さらに、判断が同じである場合を Type2-a とし、異なる場合を Type2-b とする。）

• Type3 (欠落)

総当りで比較した後、2つの知識に同じルールが存在するが、残りの1つの知識にそのルールが存在しない場合

3.2 ルールの条件節・判断節の組み合わせと不都合

2人のユーザをそれぞれユーザ A、ユーザ B と呼ぶ。また、それぞれの領域知識を領域知識 A、領域知識 B と呼ぶ。2つの領域知識の if then ルールと1つの事例に基づく知識の if then ルールが存在することから、ルールの条件と判断における組み合わせは以下の4つが存在する。

(1) 条件が3つとも一致しているパターン

(2) 条件が領域知識 A・B で一致しているパターン

(3) 条件が片方の領域知識と事例に基づく知識で一致しているパターン

(4) 条件が3つとも一致していないパターン

本研究では、これら4つのパターンに対して、不都合となるかどうかと不都合の場合にユーザへ提示するメッセージを整理した。表1に上記(2)のパターンの詳細を示す。この表では、各組み合わせの時に不都合かどうかとその時ユーザにどのような質問をするか(「…」で書かれた部分)を示している。また、Aは領域知識 A、Bは領域知識 B、木は決定木(事例に基づく知識)を示しており、○と△と□は、記号そのものに意味があるわけではなく、組み合わせを表すシンボルとして用いている。

条件不足・過剰のパターンには、以下の4つが存在する。

(1) 条件節が不足

(2) 条件節が過剰

(3) 条件節が不足 かつ 条件節が過剰

(4) 条件節の重なりなし

この4つの場合における2つのルール間のパターンについて整理すると表2のようになる。

表 2 条件不足と過剰についてのパターン

Table 2 Patterns of the lacks and excesses of the conditions

番号	条件	判断 then	例	ユーザへの質問
1	不足	同じ	if (A a1)(B b1)then ○に対して if (A a1)then ○	「どちらかの条件を追加や削除できないか？」
2	不足	異なる	if (A a1)then △	「ルールが過度に一般化されていないか？」 「ルールが過度に限定化されていないか？」
3	過剰	同じ	if (A a1)(B b1)(C c1)then ○	「どちらかの条件を追加や削除できないか？」
4	過剰	異なる	if (A a1)(B b1)(C c1)then △	「ルールが過度に一般化されていないか？」 「ルールが過度に限定化されていないか？」
5	不足かつ過剰	同じ	if (B b1)(C c1)then ○	「お互いに重要な条件を入れ忘れてないか？」 「判断に関係のない条件が含まれていないか？」
6	不足かつ過剰	異なる	if (B b1)(C c1)then △	「お互いに重要な条件を入れ忘れてないか？」 「判断に関係のない条件が含まれていないか？」
7	重なりなし	同じ	if (C c1)then ○	なし
8	重なりなし	異なる	if (C c1)then △	なし

表 1 条件が領域知識 A・B で一致しているパターン (A○ B:
○ 木:△)

Table 1 Patterns of flaw in which the conditions of the two user's domain knowledge are corresponding

番号	判断 then A B 木	不都合かどうかとユーザへの質問
2-1	○ ○ ○	Type2-a・3 の可能性 ◇ 決定木に条件不足・過剰があれば (Type2-a) - 条件不足・過剰に対する質問を出力 ◇ 総当たりチェック後、決定木に条件が一致するルールがなければ (Type3) - 「事例がない可能性あり」
2-2	○ ○ △	Type2-b・3 の可能性 ◇ 決定木に条件不足・過剰があれば (Type2-b) - 条件不足・過剰に対する質問を出力 ◇ 総当たりチェック後、決定木に条件が一致するルールがなければ (Type3) - 「事例がない可能性あり」
2-3	○ △ ○	Type1 「ユーザ A とユーザ B の知識で矛盾」 Type2-a・2-b の可能性 ◇ 決定木に条件不足・過剰があれば (Type2) - 条件不足・過剰に対する質問を出力
2-4	○ △ △	Type1 「ユーザ A とユーザ B の知識で矛盾」 Type2-a・2-b の可能性 ◇ 決定木に条件不足・過剰があれば (Type2) - 条件不足・過剰に対する質問を出力
2-5	○ △ □	Type1 「ユーザ A とユーザ B の知識で矛盾」 Type2-b の可能性 ◇ 決定木に条件不足・過剰があれば (Type2-b) - 条件不足・過剰に対する質問を出力

3.3 不都合の優先順位

検出された不都合には、以下の 3 つを基準とした優先順位を与える。

- (1) 条件は同じであるが、判断が異なるものを優先
 - (2) 条件が同じであるルールが多いものを優先
 - (3) 条件不足・過剰があるものは無いものよりも優先
- この 3 つの基準は (1)~(3) の順番で優先する。

以下にこの 3 つを基準とした理由を示す。(1) の場合、第 3 者が見たときにどちらを選べばよいか分からないからである。(2) の場合、誰もがその条件を重要と考えている可能性が高いからである。(3) の場合、より確かな提案 (何を追加・削除すれば良いかの提案)

表 3 不都合の優先順位

Table 3 Priority of flaws

優先順位	不都合の種類
1	3 つとも同じ条件のルールに対し、判断が異なるものが存在する場合
2	2 つが同じ条件で異なる判断の場合で、もう 1 つに判断が同じで条件が不足や過剰であるルールが存在する場合
3	2 つが同じ条件で異なる判断の場合で、もう 1 つに判断が同じで条件が不足や過剰であるルールが存在しない場合
4	2 つが同じルールで、もう 1 つに判断が同じで条件が不足・過剰であるルールが存在する場合
5	2 つが同じルールで、もう 1 つに判断が同じで条件が不足・過剰であるルールが存在しない場合
6	同じ条件であるルールは無いが、判断が同じで条件が不足・過剰であるルールが存在する場合
7	同じ条件であるルールも、条件が不足や過剰であるルールも存在しない場合

が出来るからである。

そして (1)~(3) の順番で優先する理由は、(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言える。(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である。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 3 に示す。なお、ここでは不都合ではないが、それぞれ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。

3.4 不都合の検出アルゴリズム

不都合の検出アルゴリズムでは、まず 2 つの領域知識と事例に基づく知識の if then ルールを総当たりで比較する。そのもとで、以下のように不都合を検出する。

- (1) 領域知識 A, B と決定木で条件が 3 つとも一致しているもの (3.2 節のパターン (1)) を探す。一致しているものがあれば、判断のチェックにより、

Type1 の不都合か否かを判別する。

- (2) 領域知識 A、B と決定木で条件が 2 つ一致しているもの (3.2 節のパターン (2)・(3)) を探す。一致しているものがあれば、判断のチェックにより、Type1 の不都合か Type3 の不都合かを判別する。そして、不足・過剰をチェックして、Type2 の不都合か否かを判別する。
- (3) 残りのパターン、つまり条件が一致していないパターン (3.2 節のパターン (4)) に対し、不足・過剰をチェックして、Type2 の不都合か否かを判別する。
- (4) 3.3 節の優先順位に基づいて、質問と共に検出された不都合を表示する。

3.5 不都合検出システム

システムの開発環境は、Windows 2000 上で、開発言語として 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 ID3⁴⁾ を使い、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 2 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。ルールは、判断と値をリストボックスから選択することで作成される。

「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、決定木表示画面と不都合検出画面 (図 2) である。ルール一覧画面では、システム、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。決定木表示画面では、システムがどのように事例からルールを抽出したかを木構造で見ることができる。不都合検出画面 (図 2) では、不都合の検出結果を見ることができる。

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

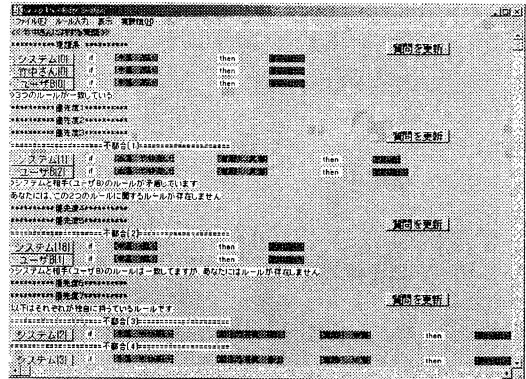


図 2 不都合検出画面

Fig. 2 Screenshot of flaw detection

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う、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行う。

4.2 実験の問題領域の設定

本研究では筆者らの一人が魚釣りの専門家であり、魚釣りのコミュニティに精通しているため、魚釣りの分野を対象として実験を行うこととする。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

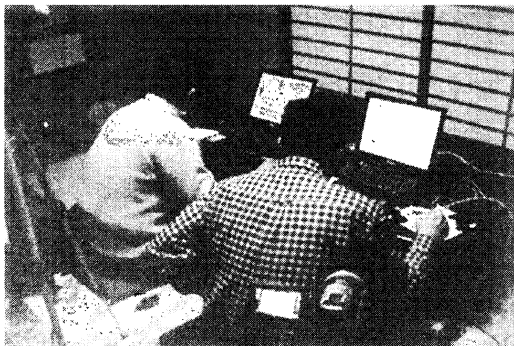


図 3 実験風景
Fig. 3 Experimental environment

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。

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、グレがどの程度釣れるか」という知識である。クラスは、「釣れない」「釣れる」の2段階に設定した。設定した属性及び属性値は、筆者のグレ釣りの経験から、水温や風速、潮の最大流速など11種類とした。これらのデータは、気象庁や神戸市立海釣り公園、兵庫県立農林水産技術総合センターなどの協力を得て獲得した。

釣行データは、筆者らの一人が運営する「神戸波止グレ考釣会⁸⁾」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。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ので、可能な限りきれいなデータを集めていると言える。学習の結果、19個のルールが生成された。

4.3 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名（20代から40代の男性11名と女性1名）を被験者として実験を行った。実験風景を図3に示す。このように、二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名で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。

実験では、練習課題を行ってもらった後、グレ釣りの課題を行ってもらった。まず始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の各被験者は下記のアンケートに回答した。

- (1) 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)

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、なぜそれら残したのかを各不都合ごとに以下のような選択式で尋ねている。

- (1) ルールの判断や条件の差は、どちらでも良い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

4.4 実験結果

4.4.1 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（及び全ペアで）、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、システムが行った不都合を残した理由のアンケートの結果を見ると、以下のようになった。

(Type2)

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

(Type3)

- その他（2人とも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

- 曖昧でわからないため、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）

表 4 不都合ごとの検出数・残存数・解消率

Table 4 The number of detected flaws, the number of left 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

不都合の解消率は全ての 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言える。

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 人とも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。

4.4.2 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、知識の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、概ね提示される不都合の順番には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 4 に示す。図 4 より、優先順位が下があ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、例外とみなす）。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言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も、その致命的な度合いは低くなってくることから、疲れれば作業を中止して完了しても止むを得ないと言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。

4.4.3 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、システムを使用した

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

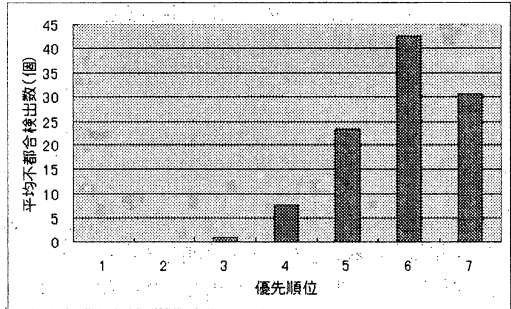


図 4 優先度ごとの平均不都合検出数

Fig. 4 The number of detected flaws in each priority

表 6 ルールの変更・削除・追加回数（システムあり）

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

	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（システムなし）

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

	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

場合と使用しなかった場合の実験結果を比較する。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、システムを使用した場合を表 6 に、システムを使用しなかった場合を表 7 に示す。さらに、システムを使用した場合と使用しなかった場合の各ルール操作の平均をグラフ表示したものを図 5 に示す。

図 5 より、システム有りの場合がシステム無しの場合に比べて、ルールの追加・変更・削除の全ての操作において回数が上回っていることがわかる。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

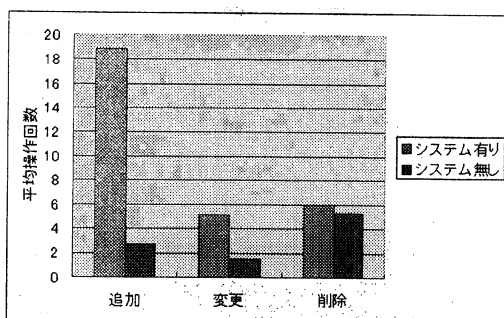


図5 ルール操作回数の比較

Fig. 5 Comparison of the number of rule operations

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は、初期入力でユーザ9は、最初は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全て削除したためである。このことを考慮すれば、削除を含めて、システム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。

5. ま と め

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を用いて、その3つの知識から不都合を検出することによってユーザに論点提示し、ユーザが議論することによって個人の知識とグループの知識を獲得・洗練する手法を提案した。この手法を実現するにあたって、3つの知識間の不都合のパターン、そのパターンに応じたユーザへの論点提示のメッセージ、それらの提示優先順位を整理し、不都合を検出して優先順位に従い論点提示するアルゴリズムを構築した。

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、概

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言える。

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ーナレッジマネジメントの最新動向ー, 人工知能学会誌, Vol. 16, Vol. 1, pp. 4-14 (2001).
- 3) Mitchell, T.M., U. P. and Baserji, R.: Learning by Experimentation: Acquiring and refining problem-solving heuristics, *Machine Learning*, Springer-Verlag, pp. 163-190 (1984).
- 4) Quinlan, J.: Induction of Decision Trees, *Machine Learning*, Vol. 1, pp. 81-106 (1986).
- 5) 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: K A I S E R, 人工知能学会誌, Vol. 7, Vol. 1, pp. 149-159 (1992).
- 6) Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol. 10, pp. 219-240 (1986).
- 7) Towell, G.G., S. J. and Noordewier, M.: Refinement of Approximate Domain Theories by Knowledge-based Neural Networks, *AAAI-90*, pp. 861-866 (1990).
- 8) 神戸波止グレ考釣会: <http://members.at.infoseek.co.jp/hiji/>.