複数の人間における概念相違検出のためのインタフェースの設計

近藤輝幸† 吉田哲也† 西田正吾†

複数の人間が参加して共同作業を行う場合には，異なる背景を持つ人間の間に存在する概念の相違が大きな問題となることが考えられる．本稿では，このような複数の人間における概念の相違について考察するとともに，同じシンボルを異なる意味に用いたり，異なるシンボルを同じ意味に用いたりするケースを対象に，ユーザの知識を表す決定木を用いて複数の人間における概念の相違を検出する方法について述べ，プロトタイプを作成し評価する．

Design of the Interfaces to Detect Conceptual Difference among Different People

TERUYUKI KONDO,† TETSUYA YOSHIDA† and SHOGO NISHIDA†

Conceptual difference among different people is a serious problem when people work in collabolation with others. In this paper, we study about conceptual difference among different people and describe a method of detecting conceptual difference by the decision trees constructed by users' knowledge in the cases that the same symbols are used in the different meaning and the cases that the different symbols are used in the same meaning. And a prototype sysytem is evaluated.

1. はじめに

大規模で複雑な問題に対処するには複数の人間が参加して協調問題解決を行うことが重要になるため，従来の知識情報処理の技術をさらに発展させてこの部分を支援することが求められている．また異分野間の協調を促進することの重要性も認識されつつあり，多くの研究も活発に行われてきている[1]．複数の人間の間の協調の支援方法としては，CSCW（Computer Supported Collaborative Work）に代表される様々な研究が行われてきている[2),3)]が，協調作業を支援するメディアとしての研究が多く，協調促進のために伝えるべき内容やその表現形式にまで踏み込んでなされている研究は比較的少ないと思われる．

本稿では複数の人間の協調に焦点を当て，複数の人間がグループとして共同作業を行う際のコミュニケーションを支援することを目指す．一般的に，ある物事に対する人間の持つ概念は各個人で相違があると考えられる．ここでは概念（あるいは視点）の相違が協調の障害になることに着目し，複数の人間が円滑に共同作業を行うためにその相違を取り除くことを目指す．これは，異なる背景知識を持つ人間がグループとして共同作業を行う場合に特に問題になると考えられる．

我々の提案するシステムは，ユーザから事例の形式で与えられたデータをもとにユーザの持つ概念の構造を具体的な決定木という構造により可視化し，そこでの差異をユーザとは別の視点から指摘する．本研究は，各ユーザの概念を決定木という構造によって具体的に表現することで協調促進を目指すところに特色がある．決定木のような階層構造によってユーザの概念を表現する枠組みは，意味ネットワーク[4)]や意思決定論[5),6)]の研究等でも利用されているが，複数の構造間に存在する概念の相違を扱う研究は少ないと思われる．ユーザはその結果を見て相互に話し合うことで個々の概念構造を変化させ，システムに提供するデータ自体も変更させていく．システムとユーザ，およびユーザ間のインタラクションを通じて徐々に概念の相違を明らかにしていき，お互いの視点や考え方を明確にしていくことでグループの協調を促進させることを目指す[7)]．概念の相違をユーザからインタラクティブに獲得していくという意味でグループ知識獲得支援ツールの1モ

† 大阪大学基礎工学部システム工学科
Department of Systems Engineering, Faculty of Engineering Science, Osaka University

ジュールとしても利用でき，また，ある物事に対する他人の概念を提示することから新たな発想を生み出すこともできると考えられるので，発想支援システムの1モジュールとして利用することも可能と考えられる[8)~11)]．

まず2章で概念の相違の定義をし，3章で提案する概念の相違を検出する方法とシステムの全体像，4章で概念の相違を検出する具体的な実現方法について述べる．そして，5章で構築したシステムの評価実験を行い，最後に現時点での問題点と今後考えられる展開について述べる．

2. 概念の相違の定義

本研究において概念の相違をどのように定義するかは，全体的な方針を定めるだけでなく，詳細な方法論にも影響があり，非常に重要である．人間がどのように物事を認識し，その概念を形成していくのかという問題については，哲学や心理学，認知科学など様々な分野で幅広く研究されている．しかし，これは非常に難しいテーマであるだけでなく，これ自体が本研究の主題ではないので，詳細な議論は上記の研究者に譲るものとし，ここでは**図1**に示す認識と表現のプロセスについて考察する．

人間がある事象に対する概念を形成しようとするとき，その事象が物理的に実体があろうと抽象的なものであろうと，その人の五感，もしくは六感を働かせて認識する．そしてその概念を自分の外へ持ち出すときには，何らかのシンボルを用いて表現する．事象が概念として認識されるときに，その概念がその事象の有するすべての情報を反映することは不可能で，情報の欠落が生じる．概念をシンボルとして表現するときも同様で，シンボルが概念を如実に反映することは有りえないかもしれない．しかし逆に考えれば，端的に概念を表しているものがシンボルであるということもできるし，実際のコミュニケーションはシンボルのやりとりで実現されている．そして，概念が異なればそれを反映しているシンボルも異なるだろうし，シンボルが同じであってもその使われ方や視点の相違などによっては異なる概念であることも考えられる．また，同じ事象であっても視点の相違により様々な概念が生じることも多く，異なる人間同士では特に顕著であると考えられる．本研究ではこのシンボルレベルでの概念の相違に注目し，以下の2つの場合を概念の相違と定義する．

- **同じシンボルを違う意味で用いている場合**
- **違うシンボルを同じ意味で用いている場合**

例として，具体的にA氏，B氏という2者を想定して考えてみる．ここでA氏は電気について，B氏は機械について造詣が深いとし，お互い同じモータの故障診断をしたとする．故障原因としてA氏は「電圧周波数異常」を，B氏は「回転数異常」をあげたとし，モータが故障したとき，「電圧周波数」と「回転数」は同じ変動を示すとする．一般的にはこの2つの違うシンボルの言葉は異なる概念だが，モータの故障診断という分野に限れば同じ意味で用いられていると考えてもよいことになる．逆に，同じシンボルの言葉であっても実際には違う意味で用いられる場合もあると考えられる．

また別の例をあげると，A氏が「熱い」と感じる温度とB氏が「熱い」と感じる温度はまったく同じということはありえず，ほとんどの場合，両方が「熱い」と感じる帯域には差異があると思われる．つまり，A氏は「熱い」と感じ，B氏は「ぬるい」と感じる部分があるのである．これより，A氏とB氏の「熱い」という概念には相違があると考えられるし，またA氏の「熱い」という概念とB氏の「ぬるい」という概念は同じ部分があるとも考えられる．

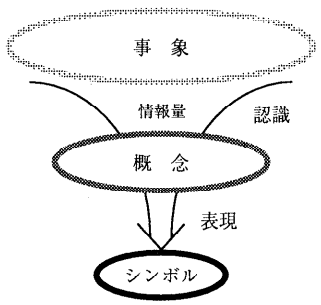


図1 認識と表現のプロセス
Fig. 1 Cognition and expression process.

3. システムの概要

図2に提案する概念相違検出システムの構成を示す．基本的な考え方としては，まず2人のユーザが自分自身の概念のシンボルを用いて，クラス・属性・属性値より構成される事例を知識として入力する．システムはその入力データより決定木を生成し，決定木の構造の違いからクラス・属性・属性値の各パラメータにおける概念の相違の検出を行う．前節で述べたように，概念の相違として2種類の定義があるので，全部で6種類の概念の相違があるということになる．検出された概念の相違は順位付けされてユーザに提示される．ユーザはその結果を見て話し合い，入力データを

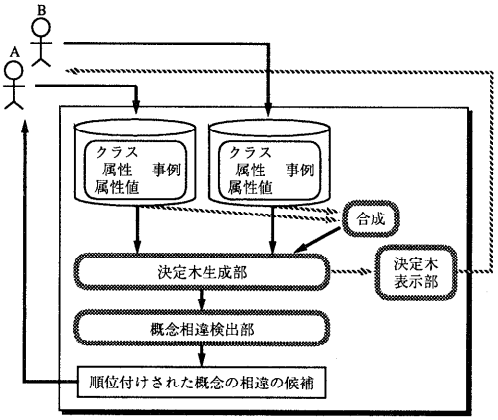


図2 システムの構成
Fig. 2 System architecture.

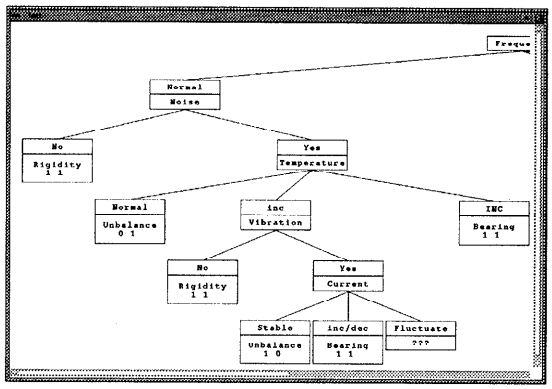


図3 決定木の表示例
Fig. 3 An example of displaying a decision tree.

修正する．この過程をインタラクティブに繰り返すことにより概念の相違を取り除いていく．また，本システムは2人のユーザが対象であるが，2人ずつの組合せを作ることにより，それ以上の人数でも適用可能である．

本システムで取り扱う知識は，分類型と呼ばれる種類の知識である．分類型の問題とは，あらかじめ定義された対象に関する特徴である属性とそれがとりうる値である属性値を入力して，そのような特徴を持つものが属するクラスを出力するものであるということができる．また事例とは，分類の対象を，用意した属性値のリストと所属すべきクラスの対で表したものである．事例に基づき，本システムはクラスを最も効率的に分離できる属性/属性値の組合せからなる分類知識を決定木の形式に生成する．決定木のノードにはそのノードで判定すべき属性が，リーフにはそこにたどり着いた場合の推定クラスが表されている．決定木生成アルゴリズムとしてはID3[12]を採用している．ID3は高速に作動するため，インタラクティブシステムには向いていると考えられる．またユーザは，決定木表示部により決定木を自由に見ることが可能である．決定木表示部の出力例を**図3**に示す．ユーザの知識構造が決定木として可視化されることで，ユーザが入力を訂正したり，さらなる知識が呼び起こされたりする効果が期待される．

決定木生成部への入力は，2人のユーザの知識とそれらを合成した知識である．知識の合成の手法は，合成知識として各ユーザの定義しているクラス・属性をすべて採用し，前提として同じシンボルを持つ属性は同じ属性値を持つこととしている．事例に関しては基本的には単純に足し合わせるが，たとえば，一方のユーザのみが持つ属性が存在する場合，もう一方のユーザの事例ではその属性がどのような属性値をとるのか分からないので，すべての属性値をとる可能性があると考え，事例を生成する．つまり合成知識とは，各ユーザの知識の和集合ということができる．知識の合成を行うことにより，ユーザ間に矛盾する事例が存在する場合，それらの事例は合成知識の決定木において矛盾した結果を生成するので，概念の相違の検出が可能となる．

4. 概念相違検出アルゴリズム

ここでは，図2における概念相違検出部の具体的なアルゴリズムについて述べる．それぞれのアルゴリズムは，概念の相違が存在する入力から決定木を生成した場合に，構造的にどのような差異が生じるかを考慮したヒューリスティックによるものである．前提条件として，各ユーザ自身の入力知識に内在する矛盾はないものとする．以下では，各ユーザの知識と合成知識をそれぞれA，B，A＋Bと呼ぶ．

4.1 同じクラスに違うシンボルを用いている場合

この場合，合成知識において，各属性について同一の属性値をとるにもかかわらず，異なるシンボルを持つ別々のクラスに属しているという矛盾した事例が存在することになる．これが決定木A＋Bにおいてどう反映されるかを考慮すると，概念の相違を含む2つのクラスが同じリーフに存在するということになる．さらに，一方のクラスはAの事例のみを持ち，他方のクラスはBの事例のみを持つ．ここで，重複するクラスが3つ以上あれば，各ユーザの知識自体に矛盾があるということなので，重複するクラスは多くても2つであることに注意されたい．そこで，上記のようなリーフを検出するという方針のもとに，次のような概

念相違検出アルゴリズムを考える．

検出アルゴリズム

Step1 決定木 A＋B から，クラスが 2 つ存在するリーフを探す．

Step2 一方のクラスは A の事例のみを持ち，もう一方のクラスは B の事例のみを持っていれば，その 2 つのペアは概念の相違の候補であるとする．

Step3 Step1～Step2 を繰り返し，候補にピックアップされた回数をカウントし，多い順に順位付けする．

4.2 違うクラスに同じシンボルを用いている場合

この場合，合成知識において，同じシンボルのクラスに属している A の事例と B の事例は，それぞれまったく異なる属性値の組合せによって構成されることになる．つまり，決定木 A＋B において，概念の相違を含むクラスは，A の事例のみを持つリーフと B の事例のみを持つリーフに分かれて存在する．そこで，以下のような概念相違検出アルゴリズムを考える．

検出アルゴリズム

Step1 A と B のそれぞれのクラスに，共通する同じシンボルのものがあるかどうか調べる．

Step2 見つかれば，決定木 A＋B から，そのクラスが割り当てられているリーフのうち，A（または B）の事例のみを持つものを探す．

Step3 そのようなリーフが見つかれば，さらに同じクラスで，B（または A）の事例のみを持つリーフを探す．

Step4 そのようなリーフがあれば，そのクラスは概念の相違の候補であるとする．

Step5 Step1～Step4 を繰り返し，候補にピックアップされた回数をカウントし，多い順に順位付けする．

4.3 同じ属性に違うシンボルを用いている場合

この場合，決定木 A と決定木 B それぞれについて，概念の相違を含む属性を持つノードの各属性値下のクラス分布がそれぞれ似たものになると考えられる．ここで，クラス分布の相違度は，以下のアルゴリズムで計算される．

クラス分布の相違度の計算アルゴリズム

合成知識のクラス数は n 個であり，決定木 A と決定木 B それぞれにおけるクラス分布を比較するノードの同じ属性値下の各クラスに属する事例は，A_1～A_n 個，B_1～B_n 個あるとする．

Step1 以下のような n 次元ベクトル $\overrightarrow{OA}$，$\overrightarrow{OB}$ を用意する．

$$\overrightarrow{OA} = (A_1, A_2, \cdots, A_n)$$

$$\overrightarrow{OB} = (B_1, B_2, \cdots, B_n)$$

Step2 各ベクトルを正規化し，ベクトル $\overrightarrow{OA}$，$\overrightarrow{OB}$ の差を計算する．

$$\frac{\overrightarrow{OA}}{\|\overrightarrow{OA}\|} \rightarrow \overrightarrow{OA}, \quad \frac{\overrightarrow{OB}}{\|\overrightarrow{OB}\|} \rightarrow \overrightarrow{OB}$$

$$\overrightarrow{AB} = \overrightarrow{OB} - \overrightarrow{OA}$$

Step3 ベクトル $\overrightarrow{AB}$ のノルム $\|\overrightarrow{AB}\|$ を，この属性値におけるクラス分布の相違度とする．

Step4 すべての属性値において Step1～Step4 を行い，その平均値をこの比較した属性のペアにおけるクラス分布の相違度とする．

つまり，クラス分布の相違度が小さいほど，その属性のペアは同じものであると考えられる．また順位付けは，クラスにおける概念相違検出アルゴリズムと同様に，検出頻度によっても行うこともできるが，クラス分布の相違度の大小もその判断基準として考えられる．そこで，以下のような概念相違検出アルゴリズムを考える．

検出アルゴリズム

Step1 A と B のそれぞれの属性に，同じ属性値を持つものがあるかどうか調べ，あればその属性をペアとする．

Step2 決定木 A と決定木 B それぞれから，Step1 でペアとなった属性を持つノードを探す．

Step3 あれば，そのノード下のクラス分布の相違度を計算する．

Step4 Step1～Step3 を繰り返す．

Step5 各ペアにおいて相違度の最小値を選び．その中から x 番目の値を閾値とし，それよりも大きいものは除く．

Step6 各ペアにおいて残ったものカウントし，多い順に出力する．

Step5 の x は最小候補数となり，ユーザの決定に委ねられる．x が小さいほど相違度重視で順位付けされ，大きいほど出現頻度重視で順位付けされる．

4.4 違う属性に同じシンボルを用いている場合

この場合は，概念の相違を含む属性を持つノード下のクラス分布が異なるものになると考えられる．つまり，同じ属性に違うシンボルを用いている場合の概念相違検出アルゴリズムにおいて，クラス分布の相違度が大きいものを選ぶように変更すればよい．以下に検出アルゴリズムを示す．

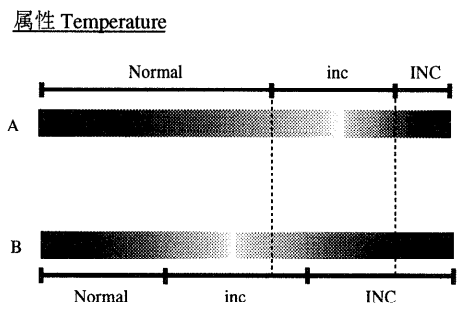


図4 属性値における概念の相違例
Fig. 4 A example of conceptual difference in value.

検出アルゴリズム

Step1 AとBのそれぞれの属性に，共通する同じシンボルのものがあるかどうか調べる．

Step2 あれば，決定木Aと決定木Bそれぞれから，その属性を持つノードを探す．

Step3 あれば，そのノード下のクラス分布の相違度を計算する．

Step4 Step1～Step3を繰り返す．

Step5 各ペアにおいて相違度の最大値を選び．その中から x 番目の値を閾値とし，それよりも小さいものは除く．

Step6 各ペアにおいて残ったものカウントし，多い順に出力する．

4.5 同じ属性値に違うシンボルを用いている場合

属性値における概念の相違はクラスや属性と違い，2章で述べた「熱い」，「ぬるい」の例のような程度問題となる．例として，属性 "Temperature" の属性値 "Normal/inc/INC" に概念の相違がある場合の関係を図4に示す．

この場合，合成知識において，概念の相違を含む属性値を持つ属性は，AとBで異なる属性値をとるにもかかわらず同じクラスに属する事例が存在するということになる．これは，決定木A＋Bにおいて，その属性で事例が分類されると，ある属性値下にはAの事例のみを持つあるクラスが存在し，別の属性値下にはBの事例のみを持つ同じクラスが存在するという形で反映される．しかし，様々な事例が入力されると，そのような典型的な分布はあまり存在せず，決定木のより上の層では特に現れにくい．そこで，ある属性での分類の前後において，あるクラスに属する事例が各属性値でどのような傾向で分類されるかに注目した．つまり，あるクラスに属する事例において，分類前のA，Bそれぞれの事例数（たとえばAは10個，Bは20個とする）は，一方の属性値で分類した後のA，Bそれぞれの事例数の変動比の差（Aが10個，Bが2個なら変動比の差は $100\% - 10\% = 90\%$）がある閾値より大きく，かつもう一方の属性値で分類した後のB，Aの事例数の変動比の差（Bが18個，Aが0個なら変動比の差は $90\% - 0\% = 90\%$）もある閾値より大きければ，その2つの属性値は同じものではないかという指摘ができる．そこで，以下のような概念相違検出アルゴリズムを考える．

検出アルゴリズム

Step1 AとBのそれぞれの属性に，共通する同じシンボルのものがあるかどうか調べる．

Step2 あれば，その属性の持つ属性値から2つずつのペアを作る．

Step3 決定木A＋Bにおいて，その属性を持つノードを探す．

Step4 Step2で作った各ペアにおいて，その属性での分類前後における事例数の変動比の差をそれぞれ計算する（上記参照）．

Step5 事例数の変動比の差がある閾値 t よりも大きければ，その属性値のペアは概念の相違の候補であるとする．

Step6 Step1～Step5を繰り返し，候補にピックアップされた回数をカウントし，多い順に順位付けする．

Step5の t は0%～100%の値をとり，小さすぎると精度が悪化するのである程度大きい方が良い．ただし，大きくすればするほど，概念の相違の候補として前述したような典型的な分布に近いものを選ぶため，大きすぎると検出できないこともある．

4.6 違う属性値に同じシンボルを用いている場合

この場合，合成知識において，概念の相違を含む属性値を持つ属性は，AとBで同じ属性値をとるにもかかわらず，違うクラスに属する事例が存在するということになる．言い換えれば，同じクラスに属する事例がその属性で分類されると，概念の相違を含む属性値においては，A，Bそれぞれの事例数の変動比が大きく異なるということになる．つまり，同じ属性値に違うシンボルを用いている場合での変動比の差の計算を1つの属性値のみで行えばよい．以下に，概念相違検出アルゴリズムを示す．

検出アルゴリズム

Step1 AとBのそれぞれの属性に，共通する同じシンボルのものがあるかどうか調べる．

Step2 あれば，決定木A＋Bにおいて，その属性を持つノードを探す．

Step3 各属性値において，その属性での分類前後に

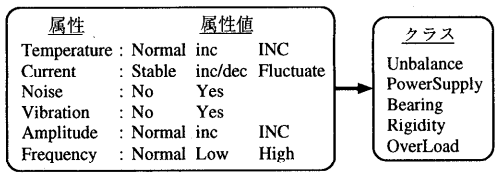


図5 入力データ
Fig. 5 Input data.

おける事例数の変動比の差をそれぞれ計算する．

Step4 事例数の変動比の差がある閾値 t よりも大きければ，その属性値は概念の相違の候補であるとする．

Step5 Step1～Step5を繰り返し，候補にピックアップされた回数をカウントし，多い順に順位付けする．

5. プロトタイプによる評価

前章で述べた概念相違検出アルゴリズムをコーディングして，プロトタイプシステムを構築し，評価実験を行った．開発環境はUNIXワークステーション上で，C言語で記述している．入力知識としては，**図5**に示すモータの故障診断例を用いている．このデータは同一のモータに関する知識をA，Bの2人から得たものであり，事例数はそれぞれ30個，事例を構成する属性は6種類でその属性値は2または3種類，クラスが5種類である．また，このデータに対して概念の相違を人為的に作成して，Bの入力データに変更を加えている．入力した概念の相違の一例を以下に示す．これ以降，“同じクラスに違うシンボルを用いている場合”は“**C1**”というように，概念の相違の各パターンを以下の例の見出しを用いて表す．

概念の相違例

C1 同じクラスに違うシンボルを用いている場合
A，B共通のクラス “PowerSupply”
→ Bのみ “Insulation” に変更

C2 違うクラスに同じシンボルを用いている場合
A，B共通のクラス “Unbalance”
→ Bの “Unbalance” に属する事例を変更

A1 同じ属性に違うシンボルを用いている場合
A，B共通の属性 “Noise”
→ Bのみ “Stench” に変更

A2 違う属性に同じシンボルを用いている場合
A，B共通の属性 “Vibration”
→ Bの各事例において “Vibration” がとる属性値を変更

表1 概念の相違が単独で存在する場合の結果
Table 1 The result of the case that each conceptual difference exists alone.

	試行回数	第1候補	第2候補	第3候補
C1	5	5	0	0
C2	5	5	0	0
A1	6	6	0	0
A2	6	2	2	2
V1	6	4	0	0
V2	6	4	0	0

属性に関しては，$x = 3$ で実行している．
属性値に関しては，$t = 50\%$ で実行している．

V1 同じ属性値に違うシンボルを用いている場合
属性 “Amplitude” のA，B共通の属性値 “inc”
→ Bの各事例において “inc” を “Normal” に変更

V2 違う属性値に同じシンボルを用いている場合
属性 “Amplitude” のAの属性値 “Normal”，Bの属性値 “inc”
→ Bの各事例において “inc” を “Normal” に変更

まず最初に，概念の相違の各パターンがそれぞれ単独で存在する場合について実験を行った．図5に示すデータの各パラメータのシンボルや事例の構成をランダムに変化させ，複数回の試行のうち，第3候補までに検出できた回数を集計した．その結果を**表1**に示す．クラスと属性に関しては第3候補までにすべて検出でき，属性値に関しても6回の試行で第1候補として4つ検出できた．第1候補で100%の検出ができているわけではないが，システムの側はあくまでも候補を提示するだけであるので，単一の概念の相違に対してはほぼ十分な検出能力を有することが確認された．

また，複数のパターンの概念の相違が同時に存在する場合の出力例を**図6**に示す．イタリックで示したものが概念の相違であり，右側の数値で示される検出頻度により順位付けされている．この入力の場合に関しては，第2候補までにすべて検出できていることが分かる．さらに，第1候補として検出されている**C1**，**C2**，**A1**，**V1**，**V2**の概念の相違を修正して，再度システムに入力した．この場合の**A2**の出力結果を**図7**に示す．これより，第2候補であった概念の相違が，検出された他の概念の相違を修正することで第1候補になっていることが分かる．

以上のプロトタイプを用いた評価実験から，概念相違検出アルゴリズムが，図2のシステムの構築へ向けてほぼ十分なパフォーマンスを持っていることが確認された．また，図2に示したシステムとユーザのインタラクションをインタラクティブに行い，検出された概念の相違を明確にし，また修正していくことで，よ

C1 *[1] A:PowerSupply B:Insulation (2)*
[1] A:Unbalance B:Rigidity (2)
[3] A:Unbalance B:Bearing (1)

C2 *[1] Unbalance (21)*
[2] Rigidity (12)
[3] Bearing (2)

A1 [1] A:Noise B:Vibration (3)
[2] A:Noise B:Stench (2)
[3] A:Vibration B:Stench (1)

A2 *[1] Vibration (4)*
[2] Current (3)
[3] Frequency (1)

V1 *[1] attr<Amplitude> A: inc B: Normal (3)*
[2] attr<Frequency> A: Normal B: Low (2)
[2] attr<Vibration> A: No B: Yes (2)
[4] attr<Current> A: inc/dec B: Stable (1)
[4] attr<Frequency> A: Normal B: High (1)

V2 *[1] attr<Amplitude> Normal (3)*
[1] attr<Amplitude> inc (3)
[3] attr<Vibration> No (2)
[3] attr<Vibration> Yes (2)
[3] attr<Frequency> Normal (2)
[3] attr<Frequency> Low (2)
[7] attr<Frequency> High (1)
[7] attr<Current> inc/dec (1)
[7] attr<Current> Stable (1)

図6 概念の相違が重複して存在する場合の結果
Fig. 6 The result of the case that each conceptual difference exists in the same time.

A1 *[1] A:Noise B:Stench (3)*
[2] A:Vibration B:Stench (1)
[2] A:Noise B:Vibration (1)

図7 検出された概念の相違を修正した後の結果
Fig. 7 The result after correcting detected conceptual differences.

り確らしい概念の相違は上位に現れ，ユーザ間に存在する概念の相違を解消していくことの可能性を確認できた．

6. おわりに

本稿では，ユーザの持つ知識を決定木の形で表し，その構造的差異から概念の相違を検出する手法を提案した．また，ヒューリスティックによる概念相違検出アルゴリズムをコーディングしてプロトタイプシステムを構築し，評価実験を行い，その結果を示した．これより，モータの故障診断例の入力に対して種々の概念の相違を与えた結果，ほぼうまく検出することが確認できた．現実の知識構造はとても複雑だが，今回提案したようなシンプルなモデルでもうまく利用すれば有効であり，今後はさらに多くのデータを適用して結果を確認し，さらなるシステムの改良を目指したい．また，ユーザ自身の知識にも矛盾がある場合についても考慮することは重要であり，そのような矛盾を検出する機能を含んだシステムの開発や，人工的なデータではなく，実際の被験者による実験および検証も今後の課題として考えていきたい．

謝辞 本研究は，一部文部省科研費（No.07243105, No.09450159）により行われたものである．

参考文献

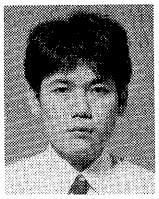
1) 久保田晃弘：コラボレーションとインターネット，人工知能学会誌，Vol.11, No.5, pp.689–693 (1996).
2) Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S. and Suchman, L.: Beyond the Chalkboard: Computer-Support for Collaboration and Problem Solving in Meeting, *Comm. ACM*, Vol.30, No.1, pp.32–47 (1987).
3) Winograd, T. (Ed.): Groupware: *The Next Wave or Just Another Advertising Slogan?*, (*Proc. IEEE COMPCON*) (1989).
4) Hayes, P.J.: On semantic nets, frames and associations, *Proc. 5th International Joint Conference on Artificial Intelligence* (*IJCAI-77*), pp.99–107, Cambridge, MA (1977).
5) Horvitz, E.J., Breese, J.S. and Henrion, M.: Decision theory in expert systems and artificial intelligence, *International Journal of Approximate Reasoning*, Vol.2, pp.247–302 (1988).
6) Oliver, R.M. and Smith, J.Q. (Eds.): *Influence Diagrams, Belief Nets and Decision Analysis*, Wiley, New York (1990).
7) 角 康之，小川竜太，堀 浩一，大須賀節雄，間瀬健二：思考空間の可視化によるコミュニケーション支援システム CSS，信学技報，TL95-6, pp.11–22 (1995).
8) Hori, K. and Ohsuga, S.: Computer Aided Thinking for Software Development, *Proc. 2nd. Pacific Rim International Conference on Artificial Intelligence '92* (*PRICAI92*), Seoul, pp.203–208 (1992).
9) Hori, K.: A System for Aiding Creative Concept Formation, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.24, No.6, pp.882–894 (1994).
10) 堀 浩一：発想支援システムの効果を議論するための一仮説，情報処理学会論文誌，Vol.35, No.10, pp.1998–2008 (1994).
11) 國藤 進：発想支援システムの研究開発動向とその課題，人工知能学会誌，Vol.8, No.5, pp.552–559 (1993).
12) Quinlan, J.: Induction of Decision Trees, *Machine Learning*, Vol.1, No.1, pp.81–106 (1986).

13) 近藤輝幸，才脇直樹，辻本浩章，西田正吾：複数の人間における概念相違検出の一手法，信学技報，HCS96-2, pp.7–12 (1996).

(平成9年 6 月30日受付)
(平成9年11月 5 日採録)

近藤 輝幸

昭和48年8月9日生．平成8年3月大阪大学基礎工学部システム工学科卒業．同年4月同大学大学院修士課程進学，現在に至る．ヒューマンインタフェース，ヒューマンインタラクションの研究に従事．



吉田 哲也

昭和43年8月27日生．平成3年3月東京大学工学部航空工学科卒業．平成4年10月より1年間エジンバラ大学大学院留学．平成6年8月より1年間カリフォルニア大学バークレー学校交換留学．平成9年3月東京大学大学院博士課程修了．同年4月大阪大学助手．知識処理技術，ヒューマンコミュニケーション技術の研究に従事．工学博士．

西田 正吾（正会員）

昭和27年1月5日生．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，同大学大学院基礎工学研究科システム人間系専攻教授．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．工学博士．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受賞．IEEEほか会員．