

RBRとCBRを統合したヘルプシステムの実現と評価

栗山 健^{1) 2)} 寺野 隆雄¹⁾

¹⁾ 筑波大学大学院経営システム科学専攻

²⁾ (株)学習研究社

規則に基づく推論と事例に基づく推論の統合化手法を提案し、その枠組みをヘルプ機能に適用することで、提案した手法の有用性について考察する。推論の枠組みは、CBRを基本とし、CBRで行き詰まった場合にはRBRを適用することで解を導出する。導出した解を実行し、うまく行った場合にはその解をコンパイル知識として学習する統合化手法を提案する。本枠組みは、事例の獲得を容易に行え、かつ適用や更新が容易であり、問題解決に複雑なルールを必要としない領域に有用である。次に、その枠組をヘルプシステムに適用した実験から、動的環境下での有用性について評価を行った。

INTEGRATING RBR INTO CBR DEVELOPMENT OF A CBR HELP SYSTEM

Ken Kuriyama^{1) 2)} Takao Terano¹⁾

¹⁾ Graduate School of Systems Management

University of Tsukuba, Tokyo

3-29-1 Otsuka, Bunkyo-ku, Tokyo, Japan

²⁾ Gakken Corporation

1-17-15 nakaikegami, ota-ku, Tokyo, Japan

This paper proposes an integration method of Rule-Based Reasoning into CBR, and discusses the effectiveness of the method by applying it to a help system. The key ideas of the method are that (i) CBR is used as the base of the problem solving aid, (ii) RBR is applied only when CBR meets a deadlock, (iii) the problem solve results by RBR are stored in a Case-Base as compiled knowledge. The proposed method is useful especially when (I) a large amount of cases are easy to acquire, (II) case adaptation or modification is easy, (III) complex rule bases are indispensable to solve given problems, and (IV) the problem domain is open in the sense of problem solving knowledge. This method is applied to a help system for naive users. The problem domain has the features discussed above. The experiments have been carried out in the application of mail operation help. Although the range of the experiments is small, the effectiveness of proposed method has been confirmed.

1. はじめに

本論文では、規則に基づく推論と事例に基づく推論の統合化手法を提案し、その枠組みをヘルプ機能に適用することで、提案した手法の有用性について考察する。

従来のエキスパートシステムの多くは、規則に基づく推論(RBR:Rule-Based Reasoning)を適用している。最近では、RBRに代わるあるいは補完する推論方式として事例に基づく推論(CBR:Case-Based Reasoning)が注目されている。また、両者(RBRとCBR)は互いに相補的であるという考えから両者を統合化する研究が行われている。

本論文では、CBRを推論の基本とし、CBRで行き詰まった場合にRBRを適用する統合化手法を提案する。本枠組を適用する場合には、典型的な事例を獲得することと、プリミティブなルールベースを作成するだけで済むため、システムの構築が容易である。また、CBRを主としているため初期状態が不完全であっても、利用を重ねることで事例が充実し、システムの完成度が高まっていく利点がある。

次に、その枠組をヘルプシステムに適用した実験から、動的環境下での有用性についての評価を行う。実験で作成したヘルプシステムは、UNIXマシン上でのメール操作を支援するものであり、利用者が計算機の操作を行っている時に、以降の操作が分からなくなってしまう行き詰まりや、間違った操作による誤りに陥ってしまった時にアドバイスをを行う。アドバイスは、利用者の途中までの操作から利用者が行おうとしている目的を類推し、現在の状態が行き詰まりなのかまたは、誤りなのかを判断し、解決策の提示を行う。

今回の実験は、メールの操作という限られた領域への適用であったが、提案した枠組の有用性は確認できた。

2章では、RBRとCBRの統合へ向けて先行研究事例の紹介と提案する統合化手法について述べる。3章では、提案した手法をヘルプシステムに適用し、評価を行う。

2. 規則に基づく推論と事例に基づく推論の統合

2.1 統合化への期待

RBRは、条件がすべて適合しなければルールが実行されない。そのため、ルールベースを構築する際には、起こりうるすべての問題を包含するよう注意する必要がある。万一問題解決に必要なルールがルールベースに無い場合には、解答不能に陥る可能性がある。一方CBRは、条件がすべて満足しなくてもある程度の類似性があれば起動される。そのため、システムの初期にはある程度不完全な事例ベースでも起動できる。しかし、一般化されたルールを個々の事例として持つのは無駄である。このように両者(RBRとCBR)は長所と短所をもっており、統合化することにより短所を補える可能性があることなどから、両者の統合の必要性は小林らにより指摘されている。[ICOT 1990a][ICOT 1991a]

2.2 統合化の研究事例

(1) 名前発音システム

Golding等は、一般知識を規則の形で持ち、例外知識を事例の形で持つシステムの研究を行っている。基本的アイデアは、まずルールにより近似的な答えを得て、もし非常に類似した例外事例がある場合は、そちらの結論を優先する。この枠組みを名前(姓)の発音に適用し、商業的システムと同程度の性能を示したと報告している。[Golding 1992]

(2) ALACALT

Nakatani等は、事例に基づいて規則(ルール)を動的に変更する統合システムALACALTを提案してい

る。四川料理のレシピを作成する CBR システム CHEF を基にしたシステムである。[Nakatani 1992]

(3) 帳票記入支援システム

安信等は、適用する規則があればルールベースを行い、類似度の高い事例がありそうならば CBR を行う、統合化の一方法を提案している。この枠組みを帳票記入支援システムに適用している。[Yasuno bu 1992]

2. 3 統合化の課題と解決案

従来の研究は、基本的な推論にはルールを利用し、ルールの補完として事例を利用する枠組みである。RBR を適用したシステムが既にある場合には、この枠組みは有効であるが、新たなシステムの構築を考える場合には、RBR が抱えている知識獲得の問題を解決できないと考えられる。

コンピュータ利用のように利用者があらゆる操作をする可能性が有る場合、熟達者の操作事例を獲得することは容易であるが、すべての利用パターンを予測しルール化しておくことは困難である。しかし、コンピュータのような人工的なシステムは原理からルールを導出する事は可能であり、プリミティブな操作をルールで記述する事はそれほど困難ではない。

そこで、CBR を主とした RBR との統合化システムの構築を考える。事例を基に帰納推論を行いルールを生成する研究が行われている[Saito 1992]が、本研究では、ルールを取り入れた事例を新たに生成し、コンパイル知識として扱う手法を提案する。また、ルールと事例のトレードオフ問題は寺野らに指摘されている[Terano 1992]が、この手法は、領域に独立なインデキシングを行う事で検索の高速化を実現できると考えられる。

統合化するにあたり次に示す考慮点がある。

① 誤った操作例の獲得をどう行うか

熟練者からは正しい知識を獲得できるが、誤った操作の例を獲得するのは難しい。熟練者は、必ずしも自分が初心者のときに陥った誤りとその解決方法を記憶しているわけではないからである。誤った操作の事例は持たずにシステムを構築し、システムを運用しながら誤った操作の事例を獲得する方法を提案する。具体的には、利用者が誤った時の操作と解決策（なんらかの形でシステムが導出する）を事例として獲得する方法である。

② 事例を計算機上でどう表現するか

1 事例に対応するオブジェクトを 1 つ生成し、あらかじめ設定してある特徴スロットに事例の特徴を記述する方法を採用する。

③ 事例間の類似度をどう測るか

類似度の計算は、両者（事例と事例）のスロット値を比較して行う。特に、操作系列の比較は、操作の順序に意味があるので、操作と順序とを考慮した比較になる。

④ 事例とルールのバランスをどうするか

統合化するにあたり、対象領域の中でも事例で持った方が良いサブ領域と、ルールで持ったほうが良いサブ領域があると考えられる。典型的な操作の手順を事例で持ち、基本的な操作をルールで持つようにする。例えば、開始から A, B, C, D を経て X に至る手順は事例で持ち、A の状態から B の状態に移るような基本操作はルールで持つ。

⑤ 推論の切り換えのタイミングをどこに設定するか

基本的な推論は CBR を適用し、CBR で行き詰まった場合に RBR の適用を考える。例えば、

システムの初期には誤った操作の事例が無いので、その場合の解決策の導出にはRBRを適用することになる。しかし、1度導出した解決策が事例として登録されれば、以降は事例が優先的に利用される。

3. ヘルプ機能への適用

本章では、最近のヘルプ機能の動向について分析したうえで、本研究におけるヘルプ機能の位置づけを明確にし、その構成を述べる。また、実行例を示すとともに評価を行う。

3.1 ヘルプ機能の動向

最近の計算機は、なんらかの形でヘルプ機能が提供されているが、その形態はさまざまである。

特徴のあるヘルプを分析してみたところ、①紙媒体のマニュアルを検索するときと同様の機能を計算機上に実現したもの、②マウスが指しているオブジェクトの内容を表示するもの、③説明文の文字列を更に詳しく説明するためにリンクを張ってあるもの、④エディタの状況に応じて表示する説明文を変更するものなどがあつた。

これらは、予測される状況に応じて表示する説明文を予め作成してある。そのため、ユーザの意図とは関係なく、計算機の状況に応じて固定された情報が提供される静的なヘルプ機能と呼べる。

本研究では、これに対し、利用者の意図をふまえ、的確なアドバイスを行う動的なヘルプ機能の実現を考え、その枠組みの研究を行う。

3.2 ヘルプ機能の構成と機能

本研究では、メール操作時における利用者の意図を類推し、次に行うべき操作の提示を行うヘルプ機能の構築を試みる。

3.2.1 ヘルプ機能の構成

ヘルプ機能は、利用者の意図を類推する類推モジュールと、解決案を導出する解決案導出モジュールから構成される。類推モジュールでは、利用者の操作履歴をキーにCBRを行い、熟練者の操作履歴から利用者の目的を類推し、利用者に目的を確認する。解決案導出モジュールは、利用者の操作履歴と熟練者の操作手順を比較しながら、解決案の導出を行う。

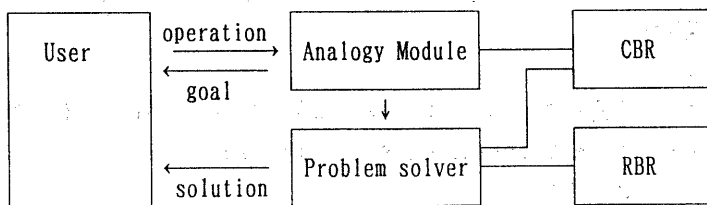


図 ヘルプ機能の構成

3.2.2 意図の類推

利用者は、ある目的を持ち、目的を遂行するために計算機の操作を続ける。(ここでは、目的を意図と定義する)。利用者が、ヘルプを呼び出す理由は、次の2とおりが考えられる。

・行き詰まり

目的に向けて操作を続けてきたが、途中で次の操作が分からなくなってしまった場合に行き詰まり

と呼ぶ。

・誤り

目的に向けて操作を続けてきたが、途中で誤った操作をしてしまった場合を誤りと呼ぶ。

利用者は正しい目的を持っていることを前提とし、利用者が、ヘルプを呼び出す直前まで行ってきた1連の操作手順をもとに、意図の類推を行う。類推方法は、利用者の操作手順と過去の熟練者の事例とを比較し、類似した熟練者の事例を検索する。熟練者の事例には、操作手順と、ゴール（意図=目的）が書かれているため、結果として意図を類推できる。

(1) 類似度の算出

熟練者の操作事例を完全系列、利用者の途中までの操作を部分系列と呼ぶとして、完全系列と部分系列を比較し類似度を算出する。

(ア) 完全系列（熟練者の事例）： 正しいコマンドの系列

```
operation := |-----|
goal      := x
solution  := nil
```

(イ) 部分系列（行き詰まり）： 途中で停止が発生した系列

```
operation := |-----|
goal      := x
solution  := |-----|
```

(ウ) 部分系列（誤り）： 途中で誤りが発生した系列

```
operation := |-----△
goal      := x
solution  := |-----|
           ▲
           ↑
```

誤りからの脱出が必要

(2) 類推の例

利用者の意図をどのように類推するのか、メールを送信する例を用いて示す。

次に示す事例が事例ベースに蓄えられているとする。この事例はメールを送信する場合の典型的な操作である。

事例1

```
operation := emacs/ESC-x rm/r/write letters/C-c C-c/C-x C-c/
goal      := reply-mail
```

利用者が emacs/ESC-x rm/r/まで操作したところでヘルプを呼び出したとする。ヘルプは利用者の途中までの操作 (emacs/ESC-x rm/r/) をキーに事例ベースを検索し、事例1を見つけ出し、利用者の意図 (goal) はreply-mailだろうと類推する。

3. 2. 3 アドバイス

ヘルプの類推結果と利用者の意図が合致した場合には、利用者が次に行うべき操作を提示する。

利用者の操作が行き詰まりの場合には、正しい事例を提示することで、利用者は以降の操作を行える。

誤りの場合には、正しい操作例を提示するだけでは、不完全であり、解決案を提示したことにはならない。誤りの場合には、誤りからの脱出方法を提示する必要がある。

大概らは、学習者が正しい理解に至る過程を次のように分析している。[Otsuki 1992]

- ①間違いがあることに気づく
- ②間違いの発生している場所を特定する
- ③自分の持っている知識の何がどう違っているのかを認識する
- ④正しい解法を見つける

計算機の利用者が上記の過程を辿るとすれば、正しい事例を提示し、かつ、誤りからの脱出方法を提示することでアドバイスの役割を果たすことになる。

誤りからの脱出方法の導出は、ルールの適用を考える。利用者が行った操作のうち、最後に行った正しい操作の位置をRETURN POINT、誤った操作も含む最後の操作の位置をCURRENT POINT と呼ぶことにする。即ち、利用者はRETURN POINTまでは正しい操作を行っていたが、途中から誤った操作を行い、現在はCURRENT POINT にいることになる。

誤りからの脱出はCURRENT POINT からRETURN POINTに戻るることとなる。RETURN POINTまで戻れば、以降は行き詰まりと同様である。

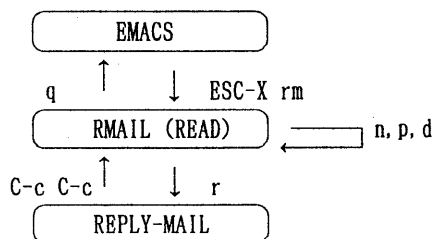
```
事例 2
operation := emacs/ESC-x rm/n/q/C-x C-c/
goal      := read-next-mail
rujido    := 1.9
```

```
my problem
operation := emacs/ESC-x ①rm/②r/
goal      := read-next-mail
```

① RETURN POINT

② CURRENT POINT

CURRENT POINTからRETURN POINTに戻る方法をルールから検出する。次に示した状態遷移図に対応するルールがあれば、解を求められる。



状態遷移図

解決案

```
new object
operation := emacs/ESC-x rm/r/C-c C-c/n/q/C-x C-c/
goal      := read-next-mail
```

解決案オブジェクト

この解決案を利用者が適用し、うまく行けば新しい事例として学習する。以降、同様の問題が起こった場合には、ルールを適用せずに事例を適用する。

3. 3 実行例

提案した枠組みを用いて、P S I - II上にヘルプシステムを実現した。開発したヘルプシステムでは、システム起動後熟練者の正しい操作例を入力して事例ベースを構築した。実験では、典型的な操作例を5個、また、ルールは10個記述してある。

searchコマンドを入力後、利用者(初心者)の操作履歴を入力する。すると、ヘルプシステムは事例ベースの中から類似した事例を検索し、類似度の高い順に事例のゴールが利用者の目的に合致するか否かを利用者に問い合わせる。利用者の目的がヘルプシステムのゴールと合致したと回答すると、ヘルプシステムは解決案を提示する。利用者はヘルプシステムが提示した解決案を実行し、目的が達成した場合にはその旨をヘルプシステムに伝える。ヘルプシステムは、提示した解決案を基に新しい事例を作成し、事例ベースに登録する。

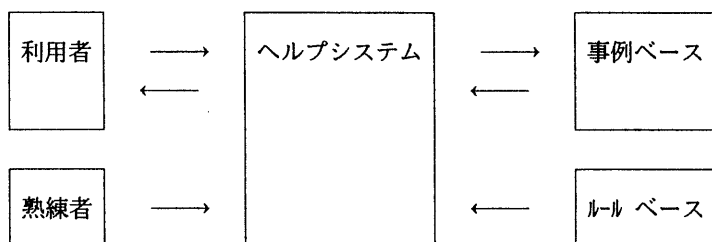


図 ヘルプシステムの概要

3. 4 評価

(1) 規則と事例による解決案の導出

利用者の状況が、行き詰まりなのか、誤りなのかをシステムが把握し、その解決案の導出に事例ベースとルールベースを適用している。ルールを適用し、正しい解を得られた場合には、そのルールを事例に取り込んだ形でコンパイル知識として学習する。

本枠組みは、問題解決機能と学習機能を持ち合わせたシステムと言える。

(2) システム構築の容易性

本枠組みは、熟練者の操作を典型的な事例として獲得している。また、事例の表現方法として操作の履歴をそのままスロットに記述しているため、確実に操作履歴を事例として表現することができる。なお、ルールは領域に依存した基本的な操作を記述している。

誤りからの脱出方法は、あらかじめ用意するのではなく、利用者が陥ったときに動的に生成しているため、初期の段階では事例を綿密に登録する必要がない。

本枠組みは、知識獲得が容易な事例ベースを基本としたルールベースとの統合化システムであるため、従来のエキスパートシステムの構築方法に比べシステムの構築が容易である。

(3) トレードオフ問題

今回の実験では、事例ベース中の全事例に対し類似度の算出を行っている。そのため、事例の件数の増加に伴い検索時間が増加している。CBRを主とした推論方式を展開するためには、有効なインデキシング処理が望まれ、現在ARTIM上に移植を行い評価を行っている。

4. おわりに

RBRとCBRを統合化する一手法について提案し、提案した枠組みをヘルプシステムに適用し、動的環境下での有用性を実証した。

本論文で述べた統合化手法は、体系化された知識を熟練者から獲得し、基本的な知識を予めシステムに持つことが特徴である。また、予測しえない誤りは、システムの運用を重ねることで獲得している。この枠組みは、エキスパートシステムの構築にむけ、短期間での知識獲得の可能性を追求できると考えられる。

また、本実験システムはプロトタイプ版であり、極限られた領域での実験に留まっている。今後は、広い領域での有用性についても実験を進めたい。

参考文献

- [Yasunobu 1992] 安信 千津子 他：ルールベース推論と事例ベース推論の統合化の一方法，人工知能学会誌，Vol. 7, No. 6 (1992)
- [Nakatani 1992a] Nakatani, Y., Israel, D.: An Architecture for Tuning Rules by Cases, Center for the Study of Language and Information CSLI-92-173 (1992)
- [Golding 1992] Golding, A. R.: Improving Rule-Based Systems through Case-Based Reasoning, AAAI-91 (1991)
- [ICOT 1990a] ICOT KSA/KAR サブワーキンググループ報告書：事例ベース推論の研究動向に関する報告書 (1990)
- [ICOT 1991a] ICOT KSA/KAR サブワーキンググループ報告書：CBR, EBL及び統合的問題解決に関する研究報告 (1991)
- [Otsuki 1992] 大槻 節呼, 竹内 章：学習者一人ひとりに適応した環境づくり，コンピュータ科学 Vol. 2, No. 5, 秀潤社 (1992)
- [Terano 1992] Terano, T. Nabeta, S.: What a CBR System Learns, Proc. JKAW92 (1992)
- [Saito 1992] 斎藤 和己 他：事例からのルール抽出，情報処理学会論文誌，Vol. 33, No. 5 (1992)