

1 Q-2

手続き型言語によるプロダクションシステム制御手法の
CAIシステムへの適用と評価

中田 圭子 佐々木 裕 福原 美三

NTT 情報通信網研究所

1. まえがき

従来、プロダクションシステムの利点として、ルールの記述・読解が容易であること、ルールの追加・削除等の保守が容易であることが挙げられてきた。しかし、実際には、ルールベースの保守というのは思いの外困難であり、大きな問題となっている [1]。

しかしながら、実用レベルのプロダクションシステムを実際に運用していくに当たっては、機能追加・変更等のユーザからの要求に対応することが必須である。我々の開発した知的 CAI システム CAIRNEY [2] においても、ユーザから様々な要求が挙げられ、その吸収のためにルールベースの大幅な更新を行った。これは大変困難な作業であったが、その主な原因はルールの可読性の悪さにあった。本稿ではまずその経過について報告する。

一方、我々の研究所では、ルールの可読性を向上させるため、制御知識と領域知識を分離し、手続き型言語を用いてプロダクションシステムを制御する手法を考案している [3]。この手法を用いて CAIRNEY ルールベースの更新を再現し、効果を検討することとした。本稿の後半ではその実験の概要と評価結果の一部を報告する。

2. CAIRNEY ルールベース更新の経験

CAIRNEY ルールベース更新の概要

知的 CAI システム CAIRNEY では、学習者の理解状況に応じて教材を選択するための教育戦略を、プロダクションルールを用いて記述している。

現在、CAIRNEY は、社内教育用 CAI として運用されており、ユーザから実際に機能追加・変更等の様々な要求が挙がってきた。そのうち、ルールの更新が必要なものは全部で 12 件であった。

これらの要求をすべて吸収し、同時に従来のルールを整理した結果、ルールベースの規模は 100kB から 70kB になった。また、全ルール 370 個のうち削除・修正の対象となったのは 148 個、追加されたルールは 61 個であった。

ルールベース更新の結果

ルールベース更新の結果を定量的に示すため、12 件の要求のうち、演習時の誤答対応の仕方に関する以下の 3 件の要求を吸収した場合のルール数の変化を測定した。誤答対応とは、学習者が演習問題に誤った解答をした場合に、フィードバックとしてヒントや正解の画面を選択し提示することである。誤答対応のためのルールは全部で 140 個である。

要求 1. 誤答対応専用の画面と学習用の画面を区別しないで選択してほしい

要求 2. 画面種別 (各画面に設定されている属性、画面選択の基準の 1 つ) を選択基準から外してほしい

要求 3. ○×式問題での誤答対応の仕方と選択式問題での誤答対応の仕方を同一のものにしてほしい

ルールベースの更新に伴って、削除されたルール数、新規に作成され追加されたルール数、修正されたルール数を表 1 に示す。

	削除数	追加数	修正数	計
要求 1	3	1	1	5
要求 2	3	1	1	5
要求 3	5	0	2	7
計	11	2	4	17

表 1: ルールベース更新によるルール数の変化

ルールベース更新の問題点

ユーザから挙がってきた 12 件の要求に従ったルールベースの更新は大変困難な作業であった。ルールベースを更新するためには、

1. 現状のルールの把握 (理解),
2. 変更仕様の決定,
3. 実際のルールの書き換え,

の 3 つのフェーズが必要であるが、ルール間の依存関係が把握しにくいと、1, 3 の両フェーズが非常に大変な作業となる。具体的には、以下のような問題点が生じる。

● 理解フェーズにおける問題点

本来、宣言的な知識の記述であるはずのルールの中に、制御知識が埋め込まれているため、個々のルールの意味が取りにくくなっている。また、ルールの全体的な流れを把握することも難しく、多くのルールを順次辿っていかなければならない。

● 書き換えフェーズにおける問題点

影響範囲の大きいルールを更新するときは、その正当性の確認もまた困難な作業である。制御知識を更新した場合、実際に書き換わる部分はわずかであっても、その正当性を確認するためには、結局ルールベースの大部分をチェックしなければならないことも稀ではない。

3. 制御知識と領域知識の分離

手続き型言語による制御知識の記述

プロダクションシステムのルールから制御知識を分離し、それを制御プランとして手続き型言語を用いて記述する手法が考案されている [3]。この手法は以下のような利点を持つことが確認されている。

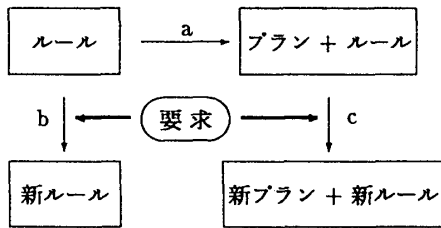


図 1: ルールベースの再更新実験

- ルールの繋がりが一目でわかる
プランは、ルールの制御知識を抽出し列挙したものであるから、ルール間の依存関係が把握しやすい。
- ルールの可読性が向上する
ルールから制御知識が削除されるため、ルールは純粋に領域知識のみを表現したものとなる。
- 使用しないルールの削除が不要
不要になったルールを実際にルールベースから削除しなくても、プランから呼び出さない限りそのルールは発火せず、システムの動作に影響を与えない。

CAIRNEY ルールベースの再更新

文献 [3] では、制御プランを用いて、CAIRNEY のルールベースを実験的に再構築し、その利点を検証している。ここでは、この手法により、前述したルールベース更新上の問題点がどの程度解消されるかを検証する。図 1 に CAIRNEY ルールベースの更新の経緯を示す。各矢印は以下を意味する。

- 矢印 a: ルールベースを制御プランを用いて再構築した作業 (文献 [3])
 矢印 b: 2 節で述べた 3 件の要求吸収のためのルールベースの更新作業
 矢印 c: プランを用いて再構築されたルールベースの更新作業 (2 節と同じ 3 件の要求吸収)

作業 a の結果、ルールベースの規模は 100kB から 94kB (プラン 20kB, ルール 74kB) に減少した。このうち、誤答対応を行う部分は、ルール 135 個、プラン 343 行となった。

今回行った実験は作業 c である。以下、その結果について述べ、2 節で述べた作業 b との比較を行う。

ルールベース再更新の結果

再更新に伴って、更新対象となったルール数を表 2 に、プランの更新行数を表 3 に示す。表 2 の削除数の () 内の数は、プランから参照されなくなったルール数である。

	削除数	追加数	修正数	計
要求 1	0(2)	1	0	1
要求 2	0(5)	1	0	1
要求 3	0(5)	0	0	0
計	0(12)	2	0	2

表 2: 再更新実験によるルール数の変化

以下に作業 b と作業 c との比較を示す。

- 表 1, 2 より、更新対象ルールの合計は 17 個から 2 個に減少した。表 3 より、プランの更新行数は 29 行である。

	削除行数	追加行数	修正行数	計
要求 1	5	1	0	6
要求 2	9	1	0	10
要求 3	13	0	0	13
計	27	2	0	29

表 3: 再更新実験によるプラン行数の変化

- 作業 b ではルールが削除されているが、作業 c ではルールは実際に削除されていない。プランの行削除のみによって同等の更新を行うことができています。
- 作業 b では、ルールの追加・削除を行うと同時に、そのルールと依存関係を持つルールをすべて修正しなければならない。作業 c では、プラン内の該当依存関係の追加・削除のみで依存関係を更新できる。

制御知識と領域知識分離の効果

制御知識と領域知識の分離によって得られた効果は以下の通りである。

- 理解フェーズにおける効果
プランを参照することにより、ルールの全体的な流れを把握することができる。ある特定のルールの影響範囲の見当をつけることも比較的容易である。また、プランは、ルールベース更新と共に必ず更新されるため、ドキュメントを完備するのと同等の効果がある。
- 書き換えフェーズにおける効果
制御の更新はプランで、領域知識の更新はルールで、各々独立に行える。ルールの削除や実行順序の変更等、制御知識の更新はプランのみで行えるため、更新対象となるルールの数を大幅に減らすことができる。ルールの修正は、領域知識自体を修正する時のみに限られる。制御の更新に要する作業量の定量的な評価は現在測定中であるが、更新箇所がルールベース内に散在している場合よりも、プランの形で 1 箇所にとままっている方が作業量は軽減されると考えられる。

4. むすび

制御知識と領域知識の分離がルールベース更新作業の軽減に効果をもたらすことを示した。知的 CAI システム CAIRNEY のルールベースの 2 回の更新実験を通じて感じられたことは、定量的には測られていないが、この手法は、ルールの理解、更新後の正当性の確認等の作業の軽減に様々な効果を持っているということである。この感触を裏付けるための更新作業の定量化と効果の測定法の開発は今後の課題である。

参考文献

- [1] E. Soloway, J. Bachant and K. Jensen, "Assessing the Maintainability of XCON-in-RIME: Coping with the Problem of a VERY Large Rule-base," *AAAI-87*, pp. 824-829, 1987.
- [2] 福原, 鈴木, 木山, 岡本, "教育支援エキスパートシステム", *NTT R&D*, Vol. 39, No. 3, pp. 421-428, 1990.
- [3] T. Ishida, Y. Sasaki and Y. Fukuhara, "Use of Procedural Programming Languages for Controlling Production Systems," *7th CAIA*, pp.71-75, 1991.