

ネットワーク対応型プロダクションシステムにおける 4F-9 協調メカニズムの試作例

藤田 茂 宇田川 則幸 菅沼 拓夫 福島 学 城戸 健
千葉工業大学

1. はじめに

近年、計算機の急速な普及にともない、様々な分野で利用されるようになってきており、処理の1つとして知識処理がおこなわれるようになった。知識処理システムとしては、複数の知識処理システムで協調して処理を行なう、分散知識処理システムが知識記述の容易さ、記述された知識の保守の面から有効である [1]。しかし、単独で動作するよう設計された知識処理システムを複数使用するだけでは知識処理システム同士で協調処理を行なうことはできず、例えば競合を起こさないよう予め知識を記述しなければならない等の問題がある。このため専門知識を処理するだけではなく、協調するためのメカニズムが必要である。本稿では、分散知識処理システムの一例として試作した、ネットワーク対応型プロダクションシステムにおける協調処理のためのメカニズムについて述べる。

2. ネットワーク対応型プロダクションシステム

複数の知識処理システムに対して知識記述を行なう際に、他のL.A.N.上の複数のマシン上で動く知識処理システム間の通信制御を記述することは、知識記述者の負担となり、記述された知識の保守性、再利用性が下がる。このため、知識処理システムを知識処理部とは独立に構築された通信支援機構（以下C.S.Sと略す） [2] をもちいて、プロダクションシステム間の通信を実現することにより (図1) [3]、知識記述における通信記述の負担を軽減した。

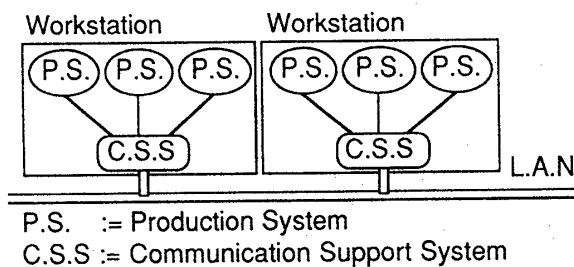


図1 C.S.Sによるプロダクションシステム間の通信

3. 協調のためのメカニズム

現在までに我々は2章で述べたプロダクションシステムを用いて、ネットワーク型演習支援システムにおけるアドバイス生成システムを作成した [4]。2章で述べたプロダクションシステムではC.S.Sに

よる知識記述における通信制御記述、負担の軽減が主であり、協調のためのメカニズムが十分ではなかった。このため、プロダクションシステム間の協調動作のための知識と、専門知識が混在し、記述された知識の保守性の低下、知識記述者に対する負担が問題点となった。

そこで、2章で述べたプロダクションシステムを拡張し、専門知識とは独立に協調のためのメカニズムおよび知識を導入し、この問題を解決する。このため知識をInteraction - Knowledge (協調の知識)、Domain - Knowledge (専門知識)に分類した。さらに、Interaction - Knowledgeを他のプロダクションシステムの要求を受ける知識と、他のプロダクションシステムへの問い合わせを行う知識に分類した。ここでは、前者をnegative - knowledge、後者をactive - knowledgeとする (図2)。

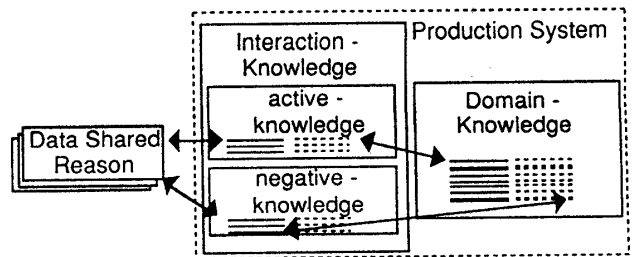


図2 プロダクションシステムにおける専門知識と協調のための知識

他のプロダクションシステムとの通信記述は一般的にKeywordによっておこなっているが、この場合すべてのKeywordを予め定義する必要があり、また状況変化に対応できなかった。これに対して本プロダクションシステムでは、協調のための知識を種類ごとに分類し、ボキャブラリと定義した。

ルールを分類を細かく定めて明確に他の知識を利用する様に記述する手法もあるが [5]、本プロダクションシステムでは知識を明確に利用するよりは、利用される側の知識が状況変化に対応できるよう、ボキャブラリを先述のように定義した。

本システムによる複数のプロダクションシステムによる協調処理の基本的な流れを図3に示す。このときプロダクションシステム間で利用されたボキャブラリは、イニシャライズ (プロダクションシステムの初期化)、ラン (ドメイン知識の推論を開始する)、キャンセル (ドメイン知識の推論を停止する)、

リクエスト（プロダクションのもつデータを要求する）の4種類である。基本的な流れを以下に述べる（図3）。

- 1) P.S.-1が推論を行なう
- 2) P.S.-1の推論結果を受けてP.S.-2が推論を行なう
- 3) P.S.-2の推論結果を受けてP.S.-1が再び推論を行なう
- 4) P.S.-1、P.S.-2に対して推論のためのデータの要求を行なう
- 5) P.S.-2は要求を受けて推論をはじめめる
- 6) P.S.-1はP.S.-2からのresultが得られないので、推論をP.S.-2からのデータ確認なしで進めることとし、P.S.-1はP.S.-2のDomain - Knowledgeの推論を停止する
- 7) P.S.-1は推論を終了する

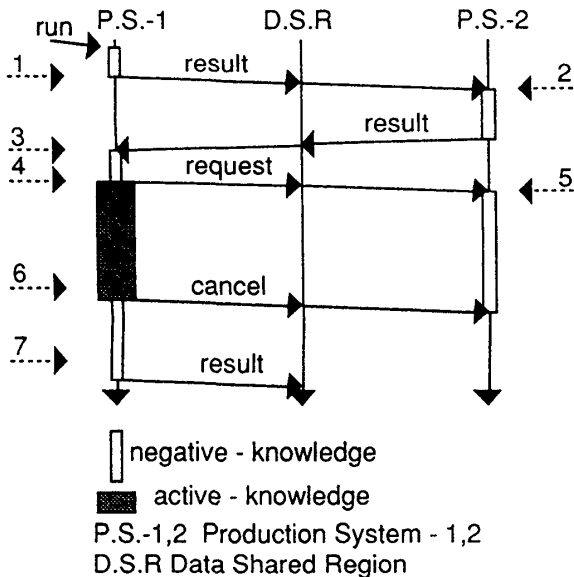


図3 協調処理のための通信

一方、ボキャブラリにはプロダクションシステム間で使われるもの、Interaction - KnowledgeとDomain - Knowledgeの間で使われるボキャブラリがある。基本的な流れを以下に述べる（図4）。

- 1) あるプロダクションシステムに対して外部からnegative - knowledgeのボキャブラリを使って推論の依頼がくる
- 2) negative - knowledgeのボキャブラリにより推論が行われ、最終的にこのプロダクションシステムはDomain - Knowledgeの推論を開始する
- 3) プロダクションシステムのDomain - Knowledgeから推論のためのデータが不足であるとして、active - knowledgeのボキャブラリを使用して外部へデータの要求を行なう
- 4) 要求の結果を得る
- 5) active - knowledgeのボキャブラリにより推論が行われ、Domain - Knowledgeの要求したデータが

Domain - Knowledgeへ渡される

- 6) 必要なデータが揃ったのでDomain - Knowledgeの推論が終了し、結果がnegative - knowledgeのボキャブラリを使用して外部へ伝えられる
- 7) negative - knowledgeは依頼された推論の終了を記述された知識に従って外部に書き込む

このようにボキャブラリを使用することによって知識記述に際して抽象的な表現によって知識の利用が図れている。

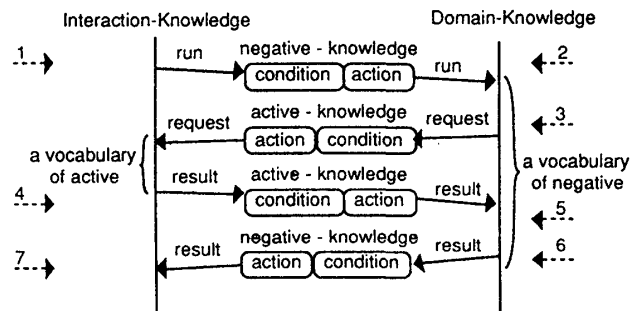


図4 Interactive - KnowledgeとDomain - Knowledge間の通信

4. まとめ

本稿では、ネットワーク対応型プロダクションシステムでの協調のためのメカニズムの試作例として、知識記述を協調のための知識とプロダクションシステムの処理知識に二分し、さらに協調のための知識を外部からの処理要求と、内部からの処理要求に分割し、それぞれに対応する知識をnegative - knowledge, active - knowledgeとした。これにより、従来開発してきたネットワーク型演習支援システムにおけるアドバイス生成機構[4]の知識記述に対して、本稿で述べたメカニズムを導入することにより、以前の知識記述と比較して、協調のための知識の記述がおこないやすくなった。

参考文献

- [1] 分散型問題解決における問題割当のための一通信方式：北村康彦、小林均、北橋忠宏、電子情報通信学会論文誌D, vol.71J-D, No.2, pp.439-447, 1988年2月
- [2] 菅沼拓夫、福島学、城戸健一：ネットワーク型演習支援システム構築のための通信支援機構の構成、第43回情報処理学会全国大会講演論文集、1991年10月
- [3] 藤田茂、宇田川則幸、福島学、城戸健一：ネットワーク型演習支援システムにおけるプロダクションシステムの構成、第43回情報処理学会全国大会講演論文集、1991年10月
- [4] 宇田川則幸、福島学、城戸健一：ネットワーク型演習支援システムのためのアドバイス機構の構成、電気関係学会東北支部連合大会講演論文集、1991年8月
- [5] Mark Klein: Supporting Conflict Resolution in Cooperative Design Systems, Proceedings of the 10th International Workshop on Distributed Artificial Intelligence, Chapter 25