

# Smalltalk-80による分散知識処理を支援する ボードシステムの開発

3B-3

福島 学・菅原 研次・大泉 充郎  
 ・富士ゼロックス(株) ・千葉工業大学  
 ・仙台応用情報学研究振興財団

## 1. はじめに

近年情報処理システムに対する要求の高度化に伴って大規模な知識処理システムを構築する必要性が増加している。大規模システムを構築するためには、対象とする問題をいくつかの副問題に分割し、解決していく協調問題解決が有効であるとされている。そこで、協調問題解決を実現するための知的分散処理システムを実現するための十分な機能を持った支援システムが必要になる。

本研究ではオブジェクト指向に基づく分散知識オブジェクトによる協調問題解決システムの支援環境の開発を目的としている。分散知識オブジェクトはLANの上の分散処理環境内で実現される。本稿では知識オブジェクト間で情報を共有し協調問題解決を行なうためのボードシステムについて述べる。

## 2. ボードシステムによる知識オブジェクト間通信方式

分散知識オブジェクトによる協調問題解決のためには、分割された副問題を処理する各分散知識オブジェクトと、それらの間で情報を共有するための情報交換を管理する機構が必要である。これにより各知識オブジェクトの処理結果の統合が行なわれ、全体として対象とする問題の解決が行なわれる。

知識オブジェクト間で情報の交換を行なう場合、従来のSmalltalk-80のメッセージセンディングによるオブジェクト間通信機能では不十分であり、さらに高度なプロトコルが必要である。

しかしながら、この通信プロトコルを知識オブジェクト内でサポートする方式では、知識オブジェクトの記述が複雑になる。かつ、その通信制御機能は本来的には問題解決の知識記述には無関係である。従って、本方式で知識オブジェクト間の情報交換をサポートする機構を知識オブジェクトとは独立に提供することにし、この通信機構をボードシステムと呼ぶことにする。

## 3. ボードシステムの概要

### 3. 1. ボードの概要

協調問題解決の場合、知識オブジェクト間で情報を共有することが必要である。Smalltalk-80の提供するメッセージセンディングだけでは、複数のオブジェクトの間で同一の情報を共有するためには必要な全てのオブジェクトに対してメッセージセンディングを行なわなければならない。

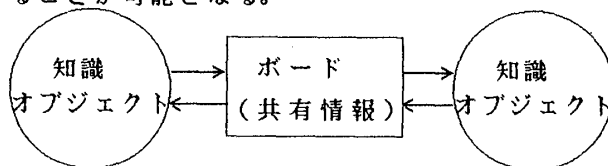
また、全ての知識オブジェクトに対して同一の

情報を送る場合、その制御が複雑になるばかりでなく、受けた知識オブジェクトにとっては不必要な情報である場合が多く発生する。

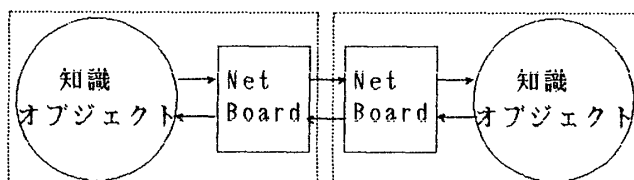
このため、ボードで特定のオブジェクトの間で共有する必要のある情報をブールすることにより、特定のボードを共有することにより必要な情報の効率的共有を実現する。ボードにはシステム内でのみ利用されるLocalBoardとネットワーク環境内で利用されるNetBoardの2種類のボードがある。

ボードへのアクセスが許可されている知識オブジェクトのリストは各ボードに記述される。

知識オブジェクトは協調問題解決に必要な知識オブジェクトとボードを共有することにより情報の共有を実現する。(図1) 知識オブジェクトから他の知識オブジェクトはすべてボードとしてとらえられるため、相手の知識オブジェクトがシステム内に存在するか、システム内に存在しないかを意識する必要がない。これにより知識オブジェクトに通信制御の記述を行なう必要がないため、容易に他の知識オブジェクトと情報を共有することが可能となる。



(a)LocalBoard



(b)NetBoard

図1 ボード

### 3. 2. ボードシステムの構造

NetBoardの様に他のシステムと通信を行い一貫性を保たなければならないボードの場合、各ボードが独自に通信を行なうことは困難である。そこで、ボードシステムを用いボードの管理を行なう。ボードシステムはボードを保管管理しており、NetBoardに記述されている情報がネットワーク環境内で一貫性を保つよう制御する。

ボードシステムは図2に示される様に、システム内の知識オブジェクトとのLocalInterfaceと他のシステムとのNetInterface、各インタフェース

とボードの管理を行なうInterface&BoardsManagerとBoardSetにより構成される。

### 3. 3. ボードシステムの動作

システム内の知識オブジェクトからボードシステムはLocalInterfaceを通して通信をおこない、他のシステムとはNetInterfaceを通して通信を行なう。Interface&BoardsManagerは通信してきた相手がボードへのアクセス権を調べ、ボードへアクセスを行なう。

NetBoardに対して変更を伴うアクセスが起こった場合、Interface&BoardsManagerは他のシステムにある同一のNetBoardの内容を更新するため、NetInterfaceを用いて他のシステムと通信を行う。NetBoardに対して同時に変更を伴うアクセスが起こった場合、情報についているタイムタグを比較し、最新の情報を優先する。これはLocalBoardに対しても同様であり、最新の情報を優先する。

ボードシステムはシステムが立ち上がった時点で起動され、NetBoardに記述されている他のシステムが利用可能な状態にあるかを調べ、ネットワーク環境の状態を保持する。これにより利用不可能なシステムに対する通信の発生を回避する。

この時点でNetBoardの更新を行う。他のシステムが起動され、起動メッセージの到達を受けてInterface&BoardsManagerは保持しているネットワーク環境の状態を更新する。

これにより、知識オブジェクトはネットワーク環境の状態を意識する事なく常に最新のネットワーク環境の中で処理を進めることが可能になる。(図3)

### 4. Smalltalk-80を用いた試作

試作システムは富士ゼロックス製の1161 A IWS上のSmalltalk-80を用いて実現した。現在知識オブジェクトとしてルール型知識オブジェクト(RKO)の試作を行い、これを用いて協調問題解決を行なっている。

RKOはSmalltalkのクラスとして提供され、それぞれクラスからのインスタンスレーションによって生成される。RKOは独自のワーキングメモリを持ち、推論の状態等を保持する。

ボードもSmalltalkのクラスとして提供され、NetBoardおよびLocalBoardはBoardクラスのサブクラスとして提供されている。ボードはインスタンスレーションにより生成され、ボードシステムにより管理保管される。

本システムではこれらのクラスを提供するだけでなく、知識ベースシステムの開発を支援するためのユーザインタフェースとしてルール用ユーザインタフェースとボード用ユーザインタフェースを提供する。(図4) これらのユーザインタフェースを利用することにより知識の記述や修正が容易に行える。

### 5. おわりに

本稿では協調問題解決のための分散知識オブジェクト間の通信を支援するボードシステムについて述べた。このボードシステムを用いる事により各々の知識オブジェクトに通信を制御するための記述を行なう必要がないため、知識の記述が容易になっている。また、ボードを用いることに

より、各々の知識オブジェクトが個有に持っている情報に影響を与える事なく知識オブジェクト間での情報の共有を実現している。また、このボードシステムを用いる事により、知識オブジェクトは協調問題解決に必要な知識オブジェクトがシステム内にあるかないかを意識する事なく処理を進める事が可能となっている。

現在協調問題解決の例として、車のシミュレーションを行い検証を進めている。車のシミュレーションでは、車、道、運転者のそれぞれ次の状態を他の推論結果と自分の情報を基に推論しながら最終目的であるゴールへ到達するという問題を解決する。

〔謝辞〕試作システムのインプリメントに協力を頂いた千葉工業大学電子工学科鈴木公一氏に感謝します。

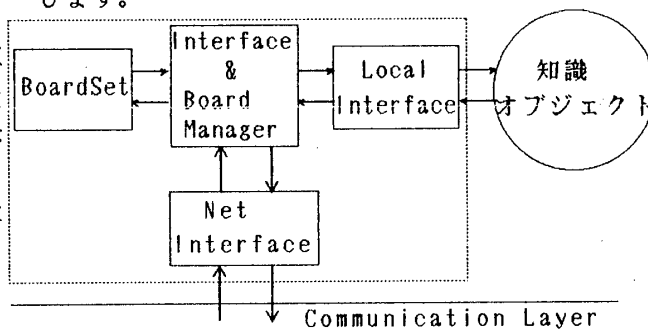


図2 ボードシステム

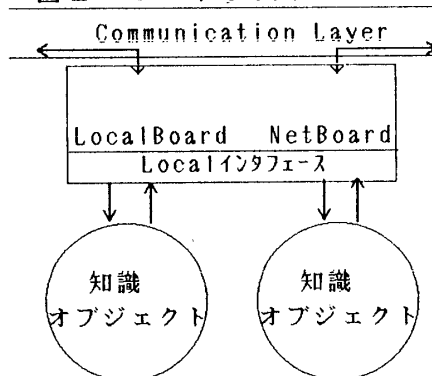


図3 ボードシステムと知識オブジェクト

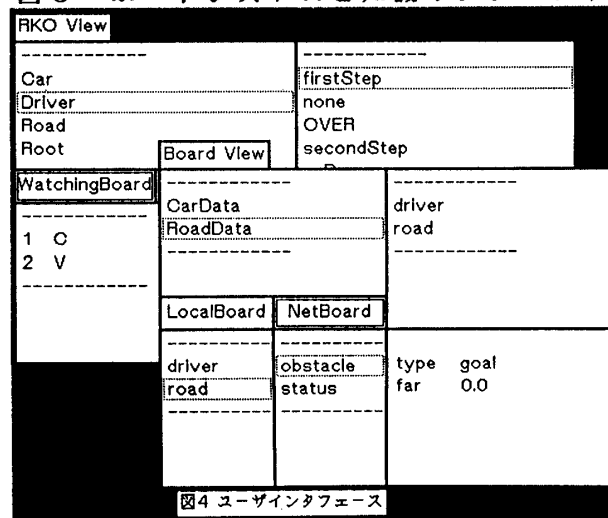


図4 ユーザインタフェース