

8E-3 自律分権協調における「スコープ」と動的グルーピング

佐藤義則¹ 布川博士² 武宮博³ 白鳥則郎¹ 野口正一⁴¹東北大学工学部 ²東北大学電気通信研究所 ³日立東北ソフトウェア ⁴東北大学応用情報学研究センター

1. はじめに

われわれは、自律分権協調¹⁾という概念をもとに研究を行っている。ここでいう「自律分権協調」とは

自律：計算主体が、外部からコントロールを受けることなしに、自らの行動を決定できること。

分権：計算主体の集合が活動する過程で、その活動を永続的・大域的に監視制御するものが存在しないこと。

協調：計算主体がその行動様式を、周囲の状況に合わせて、より望ましい方向に変えていくこと。

という3つの概念がわかり難く結び付けられた、分割不可能な単一概念のことを指す。自律分権協調の、特に協調という側面を考えた場合、他の計算主体（即ち外部環境）の情報を自分の内部状態と同レベルで扱い、これらをもとに適切に行動様式を変更していく必要がある。

本来協調に必要な情報のやり取りは、その時点で協調を行う必要のある計算主体間で「のみ」行えばよいものである。したがって、協調にとって最も自然な相互作用は「必要な相手と・必要なときに・必要なだけ」情報をやりとりすることである。

武宮ら²⁾が提案している「協調スコープ」の概念は、このような相互作用を計算モデルとして定式化したものである。

本論文では、この協調スコープを言語FrameWork上に導入し、動的グルーピングを行う際の有効な枠組として働くことを示す。

2. 協調スコープ

「協調スコープ」は、動的に協調可能な相手を見つけるための一手法である。

これまでにも、同じ目的を持った種々の手法が提案されてきた³⁾。しかしそれらの多くは、特定の協調行為を行なえるかどうかを、あるグループに入っているかどうかで記述している（「フィールド型の境界」⁴⁾）にとどまっている。これでは、グループに入っている計算主体は、グループ内で規定されている協調手法に従って（潜在的に）お互いに協調可能であるという「協

調可能性」しか記述できない。したがって、ある協調手法を使っている計算主体の一部が協調を行ない、その結果計算主体の協調構造が変化して行くような、より動的な協調を記述することはできない。このように、より一般的な協調を記述するためには、「どの計算主体と協調できるか」という協調可能性だけでなく、「どの計算主体と協調しなくてはいけないか」という「協調必要性」を記述できる必要がある。「協調スコープ」は、この「協調必要性」を計算モデルのレベルで記述可能としたものである。

協調スコープの概念図を図1に示す。

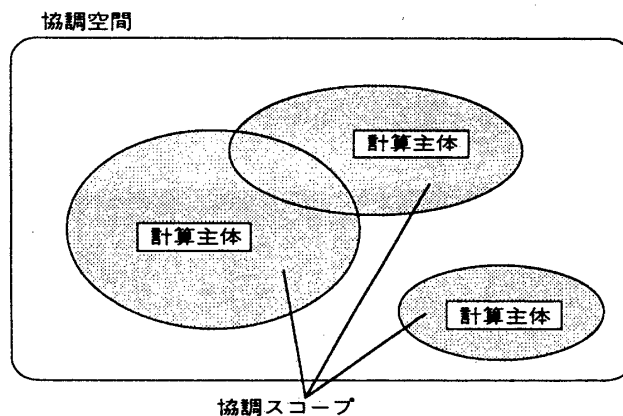


図1 協調スコープの概念図

協調空間内に入っていることが協調可能性を、協調スコープの重なりが協調必要性を示している。協調型言語に協調スコープを導入することにより、言語の基本機能として動的グルーピングを自然な形で導入することができる。

この手法の有効性を示すために、自律分散型言語FrameWorkの計算主体であるFaを用いて協調スコープを持った計算主体を実現する。

3. Fa(Functional Atom)

Faは自律分散型言語FrameWork³⁾の計算主体である。Faの特徴は、行動様式を記述したスクリプト（メソッドに相当）を動的に追加変更削除できることである。これにより、行動様式を動的に

Dynamic Grouping with Coordination Scope in Autonomous Decentralized Coordination

SATO Yoshinori¹, NUNOKAWA Hiroshi², TAKEMIYA Hiroshi³, SHIRATORI Norio¹, NOGUCHI Shoichi⁴

¹ Faculty of Engineering, Tohoku University

² Research Institute of Electrical Communication, Tohoku University

³ Hitachi Tohoku Software Co., Ltd.

⁴ Research Center for Applied Information Sciences, Tohoku University

変更することができる。

Faへのスクリプトの追加を'attach', 変更を'overwrite', 削除を'detach'という。

Faの概念図を図2に示す。

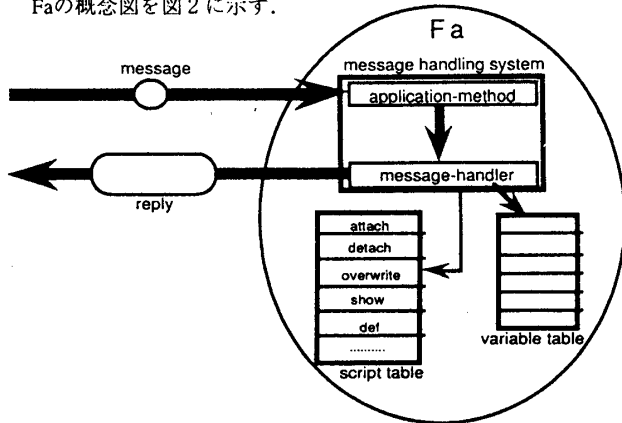


図2 Faの概念図

行動様式を動的に変更できるというFaの特徴を利用して、協調スコープを持った計算主体を実現する。

4. 協調スコープの実現

Faによる協調スコープの実現の概要を図3に示す。

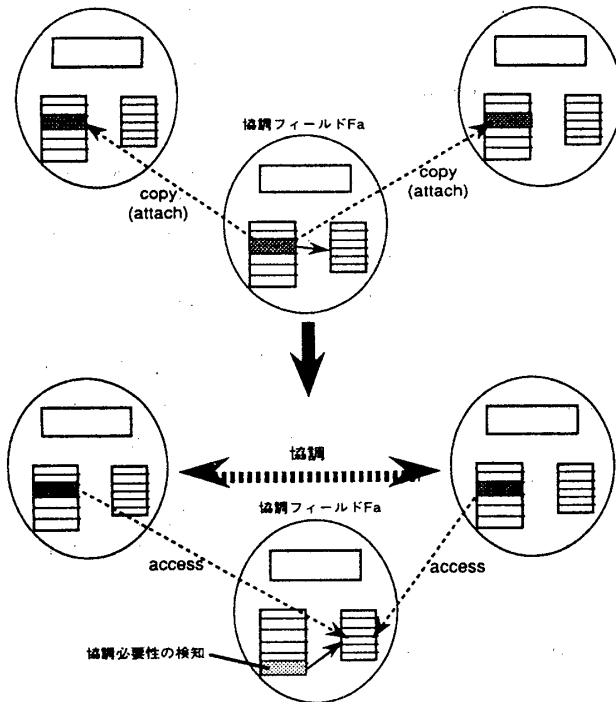


図3 Faによる協調スコープの実現の概念図

1つの協調空間を1つのFaを用いて表現する。これを「協調空間Fa」と呼ぶ。協調空間Faは、この協調空間にアクセスするためのスクリプトを保持している。

一般のFaがこの協調空間を利用して協調を行なう場合は、このアクセス用スクリプトを自分自身にattachする。

協調空間Faは他に、協調スコープの重なりを検知し適切な

処理を行なうためのスクリプトを保持している。何が「適切な処理」であるかは協調空間ごとに異なるため、協調空間Fa構成時に個々にスクリプトを構成する。

協調スコープを持ったFaの使用例として、協調ランダムウォークを取り上げる。これは、お互いに協調しながら衝突を避けつつ、非同期的にグリッド上をランダムに動く複数の計算主体をモデル化したものである(図4)。

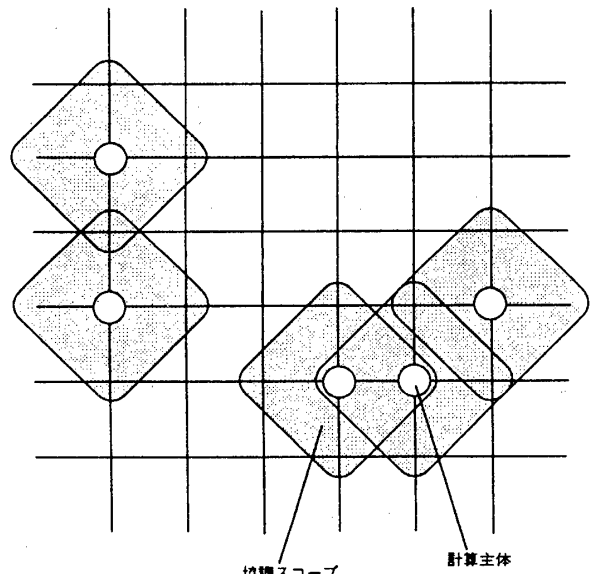


図4 協調ランダムウォーク

どの計算主体も残りの全ての計算主体と「協調可能性」を持っている。しかしながら、衝突しないようにするための「協調必要性」を持っている計算主体は、自分の近くにいるごく一部である。協調スコープにより、協調必要性を持った計算主体のみとの相互作用の記述が可能となった。

5. おわりに

自律分散型言語FrameWorkの計算主体であるFaを用いて、協調スコープを持つ計算主体を実現した、これにより「協調必要性」に基づいて協調を行ない、動的に協調対象とその構成が変化して行くような対象をも自然な形でモデル化できるようになった。

今後は、協調スコープを用いた計算主体に基づいて、不定型な集団協同活動をサポートするグループウェアの計算モデル化を行いたいと考えている。また、協調スコープの概念を拡張して、自己組織系、あるいは自己創出系の記述の可能性を探りたいと考えている。

参考文献

- 1) 矢野, 武宮, 布川, 野口: 自律的な協調処理を行う分権型計算モデル Kemari, 情報処理学会論文誌 Vol.33, No.12 (1992), pp.1476-1486.
- 2) 武宮, 矢野, 布川, 野口: 協調スコープを持つ協調型計算モデル, 情報研報, Vol.91-PRG-3(1991), pp.37-46.
- 3) 佐藤, 布川, 丹野, 野口: 自律分散メタプログラミング言語, 日本ソフトウェア科学会第8回大会論文集(1991), pp.69-71.