

# 分散 ATMS に基づくスケジュール管理システムの実現

## Implementing a scheduling system using Multi TMS

伊藤孝行<sup>†</sup> 新谷虎松<sup>†</sup>

Takayuki Ito<sup>†</sup> Toramatsu Shintani<sup>†</sup>

<sup>†</sup> 名古屋工業大学 知能情報システム学科

<sup>†</sup> Intelligence and Computer Science, Nagoya Institute of Technology

**Abstract:** we propose an intelligent scheduling system using MultiTMS[Multi Truth Maintenance System] based on Distributed ATMS. MultiSystem is a system that consists of many independant agents. MultiTMS has following features:

- (a) Agents have a global data and a local data. Agents have only to take care of global data to maintain a truth of all the system. It decrease the overhead of computation and communication.
- (b) Agents can make a choice of TMSs (ATMS,JTMS,etc.) by a problem given. It decrease the overhead of computeaion, also.
- (c) Agents are independant each other, the number of agents can increase and decrease without influence on all the system. We have implemented a schedule system to see how flexibly and effectively the MultiTMS can be used.

## 1 はじめに

人間の行動予定を記すスケジュールは多様であるため、多くスケジュールを管理・調整するのは困難である。しかし、ネットワークと計算機が接続された分散環境をうまく利用できれば、計算機にスケジュールをうまく管理・調整させることができる。

本論文では、ある仕事を与えられた複数のシステム（以下エージェントと呼ぶ）が協調することによって問題を解決していく分散システムを論じる。ここでは、分散 ATMS を用いて、多人数のスケジュールを知的に管理・調整できる知的スケジュールシステムをインプリメントし、知的化の有用性を示す。

なお、本システムは Macintosh 上の HyperCard2.1[Dan 90] 及び MacPROLOG4.5[Mac 92] で記述し、AppleEvent[AE 90] を用いて、スケジュールシステム同士の通信を行なった。AppleEvent は、アプリケーションの起動や終了、ファイルの印刷などの各種アプリケーションの間で共通する操作に当たるメッセージを集めたもののことをいう。AppleEvent のメッセージをあるアプリケーションに送ることによって、それに相当する動きをアプリケーションの外から実行させること

ができる。

## 2 マルチシステムに基づく分散 ATMS の実現

分散システム (Distributed System) は、一つのシステムを分散することによって生まれた複数のシステムが互いに協調しながら、全体としては仮想的な一つのシステムであるかのように振舞うシステムである。

ここで本論文では、自分自身で問題を解く能力を持つ独立した複数の扱える仕事の異なるエージェントが、協調する分散知識システムをマルチシステムと呼ぶ。

分散 ATMS[石田 92] は、ATMS(Assumption-based Truth Maintenance System)[De Kleer 86] の計算量の指数的爆発を避けるために、ATMS の仮説節点集合を局所的な一貫性を保持するエージェントに分配する。そして、各エージェントが局所的に矛盾を解消していくことによって、最終的に全体の域的一貫性を保持していくというものである。ここで既存の分散 ATMS の欠点を挙げる。

- 域的一貫性を解消するために、局所的なデータのあまり関わりのないデータまで計算することに

よって、計算量や通信量を増加させてしまう。

- 基本的に一つの目標のためにすべてのエージェントは同一の機構を持たねばならない。そのため、与えられた問題に不向きな機構を使わざるをえない場合がある。
- 全体で1つのシステムとしての一貫性を保たなければならず、そのエージェント数を変化させることは、全体に影響を及ぼし、システム全体の再構築を必要とする場合もある。つまり拡張性が乏しい。

本研究では、このような欠点を解決するためにマルチシステムに基づく分散 ATMS を提案する。このシステムを MultiTMS と呼ぶ。

MultiTMS は、複数の独立したエージェントによって構成され、各エージェントは、TMS (Truth Maintenance System) と知識ベースを持つ。各エージェントはメッセージを通信し合うことによって互いに協調作業をする。

各エージェントに特有のローカルなデータは、各々の TMS によって整合性を保持・調整される。すべてのエージェントで共有されるグローバルなデータもある。グローバルなデータは任意のひとつまたは複数のエージェントの持つ TMS によって整合性を保持・調整される。ここで、既存の分散 ATMS と比べた時の MultiTMS の長所を挙げると、

- 全体のすべてのデータの一貫性を保つのではなく、全体で共有するデータだけの一貫性を保てばよいので、計算量、通信量ともに減らすことができる。
- MultiTMS のエージェントの持つ TMS は、正当化に基づいた TMS (Justification-based TMS:JTMS)、仮説に基づいた TMS (Assumption-based TMS:ATMS) など、整合性を保持するためのシステムであればよい。つまり、与えられた問題の性質によって ATMS や JTMS を使い分けることができる。また、データの整合性が失われないエージェントであれば、整合性保持機構をもつ必要もない。これにより、与えられた問題の解決に最適な方法を選ぶことができるので、余分な計算を避けることができる。
- エージェントは独立しているため、エージェントの数を増やそうが減らそうが全体には影響しない。つまり、拡張性に富んでいる。

MultiTMS の特徴をまとめると、全体の一貫性は、全体で共有するデータによって必要最小限の一貫性を保つことにより実現している。整合性保持機構は与えられた問

題によって使い分けることができる。エージェントの独立性によって拡張性にも富んでいる。

そこで、本論文では上のような MultiTMS の特徴を生かした知的スケジュールシステムを実際にインプリメントした。そして、本システムでのエージェントを以下のように定義する。

『スケジューラとそのデータを用いて、その整合性を保持するための機構 ATMS を持つ。ATMS はスケジューラからのデータの受信、整合性の調整、送信をする。スケジューラはその結果をメッセージとして、他のエージェントに送信する。また、他のエージェントからのメッセージを受信し、スケジューラのもつデータとの整合性を ATMS によって調整する。』

本システムでは、上で定義したエージェント同士が協調することによって、個々のスケジュールと全体のスケジュールの整合性を保っていくシステムをいう。ここでいう協調とは、全体のスケジュールだけの整合性を保持できるメインエージェントと個々のスケジュールだけの整合性を保持できるローカルエージェントがメッセージによって通信し合い、全体としての整合性を保っていくことをいう。ここでのメッセージはスケジュールのデータである。

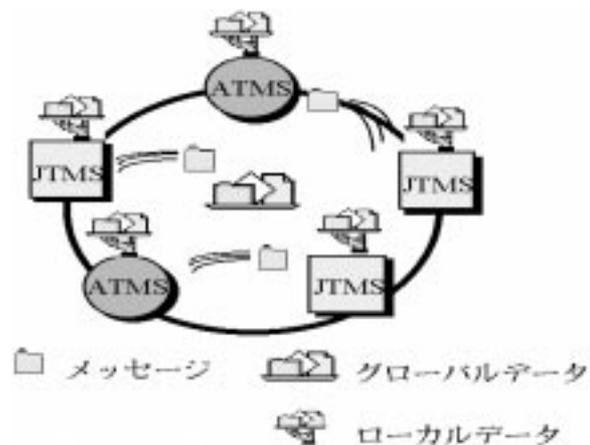


図 1: MultiTMS

### 3 スケジュール管理システムの構成

本システムは、スケジューラと整合性保持機構の2つを持つエージェントの集合によって MultiTMS のシステムを構成している(図2)。スケジューラはスケジュールの管理を行なう。整合性保持機構は簡単な推論システムと ATMS によってスケジュールの調整を行なう。エージェントは、メインエージェントと、ローカル

エージェントに分けられる。メインエージェントの持つスケジューラーと ATMS をそれぞれメインスケジューラー、メイン ATMS と呼び、ローカルエージェントの持つスケジューラーと ATMS をそれぞれローカルスケジューラー、ローカル ATMS と呼ぶ。

また、本システムの特徴として、スケジュールのデータは時間的に範囲を持つことができる。ユーザーのスケジュールは、ある程度決まっていれば良く、ある程度範囲をもたせて管理される。スケジュールのデータはその範囲内で調整される。つまり、グローバルなデータをその範囲内で調整したとき、その調整によって影響を受けるローカルなデータも範囲を持っているので、ローカルなデータの持つ範囲内で調整することが可能になる。

メインエージェントの持つメインスケジューラーは、すべてのローカルスケジューラーの管理や、本システム全体のスケジュールのデータの整合性の保持を以下のように行なう。

- [1] 各ローカルスケジューラーの持つスケジュールのデータの収集
- [2] メイン ATMS によるデータの整合性の調整
- [3] 整合性を調整されたデータの各ローカルスケジューラーへの分配

ローカルエージェントの持つローカルスケジューラーは、各ユーザーに実際に割り当てられるスケジューラーであり、次のような仕事を割り当てられる。

- [a] ユーザーのスケジュールの管理  
ユーザーのスケジュールは範囲をもたせて管理されているので、その範囲のなかで、システムは予定を調整する。そして、年単位、月単位、日単位で管理され、ユーザーまたは、メインスケジューラーの参照の要求に、迅速に回答できる。
- [b] メインスケジューラーとの通信  
メインスケジューラーの参照の要求があった場合、ローカルスケジューラーはメインスケジューラーに指定された範囲のスケジュールのデータを送信する。  
また、メインスケジューラーで処理されたデータを受信し、持っているスケジュールとの整合性をローカル ATMS によって調整する。
- [c] ユーザーとの対話  
ユーザーとの対話は、GUI ( Graphical User Interface ) を通して行なわれる。GUI の設計には、HyperCard を用いたため、視覚的に分かりやすくなっ

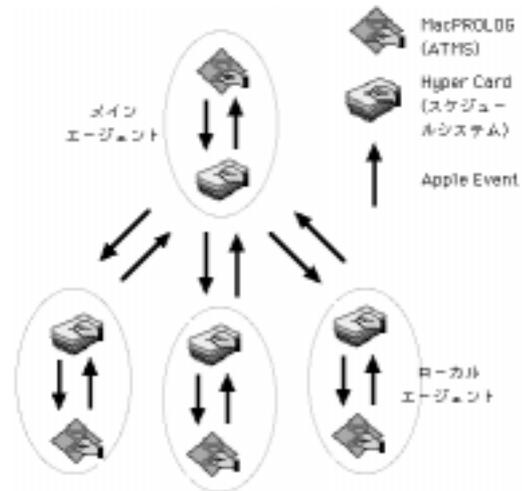


図 2: システム構成

た。さらに、ユーザーは操作のほとんどをマウスだけでできる。

## 4 スケジュール管理システムの動作

本論文でインプリメントしたシステムが目的とする動作は、ユーザー個々のスケジュールとユーザー全体のスケジュールを知的に調整することである。

各ユーザーには、ローカルエージェントが割り当てられ、そのローカルスケジューラーには、ユーザーのスケジュールデータが格納されている。もちろん、ユーザーは、スケジュールの参照、書き込み、取消し、変更などができる。

ここで、ユーザー全員に関係するスケジュールをたてるとしよう。例えば、会議の時間を決定するとする。全体のスケジュールを管理するのはメインスケジューラーである。

- [1] 会議時間の問い合わせ  
メインスケジューラーは会議の実行が可能な時間をローカルスケジューラーすべてに問い合わせる。ここで会議の実行が可能な時間とは、予定が全くない時間か、予定を他の時間にずらせば予定がなくなる時間である。
- [2] 個々のスケジュールの収集  
ローカルスケジューラーは、メインスケジューラーからの問い合わせに答え、範囲を持った時間のデータを、メインスケジューラーに渡す。
- [3] 全体スケジュールの調整

そのデータを受けとったメインスケジューラーは、メイン ATMS に各ユーザーの範囲を持った時間のデータを送る。送るデータは予定を他の時間にずらせば予定がなくなる時間であって、最初からできると分かっている予定の全くない時間のデータは、計算量の軽減のため ATMS には送らない。この時、ユーザーには優先度があり、会議の性質によって、優先度が変わってくる。つまり、会議に必ず出なければならないユーザー、どちらでも良いユーザー、などに分かれる。メイン ATMS では、渡されたデータの整合性を調整し、結果をメインスケジューラーに返す。

- [4] 会議時間の候補の割り出し  
メインスケジューラーの受けとった結果はこの時点では複数解、つまり、会議が実行可能な時間をいくつか持つ。その中から1つを決定するため、その複数解を、ローカルスケジューラーに送信する。
- [5] 個々のスケジュールとの照合  
複数解を受けとったローカルスケジューラーは、ローカル ATMS によってどの解が自らのスケジュールのデータと整合性が保たれるかを探す。整合性の保たれる解をすべて見つけ、それを再びメインスケジューラーに返す。
- [6] 会議時間の決定・送信  
メインスケジューラーでは、最も実行可能であるユーザーが多い会議時間を選び、その時間を会議時間と決定する。そして、決定された会議時間をローカルスケジューラーに送信する。



図 3: 実行結果 (スケジューラー)

- [7] 個々のスケジュールの調整  
決定された会議時間を受けとったローカルスケジ

ューラーは、もう一度ローカル ATMS によって、スケジュールの調整を行なう。

以上のように、本システムは動作し、全体のスケジュールの調整を行なう。実行結果を図3に示す。図3はスケジューラーである。

## 5 まとめ

本論文では、分散 ATMS である MultiTMS を用いた知的スケジュールシステムのインプリメントを通して、スケジュールシステムの知的化の有用性を示した。MultiTMS はグローバルなデータとローカルなデータを使い分けることにより、効率の良い処理ができる。これは、全体のスケジュールと個々のスケジュールをデータに持つスケジュールシステムを構築するには、非常に都合が良く、効率の良い知的なスケジュールシステムを構築できた。

今後の課題として、分散 ATMS のアルゴリズムの改良による高速化、並列化による高速化、を研究中である。また、並列化には実機である並列マシン AP-1000 を使った試作を検討中である。

## 参考文献

- [AE 90] AppleEvent : Macintosh System7.0 , Apple Computer , Inc.(1990)
- [Dan 90] Dan Winkler, Scot kamins : HYPERTALK 2.0 : BANTAM COMPUTER BOOKS (1990)
- [De Kleer 86] de Kleer , J : An Assumption based TMS , Artificial Intelligence , Vol28 , No.2(1986)
- [Mac 92a] MacPROLOG4.1 : MacPROLOG 4.1 Reference Manual , Logic Programming Associates Ltd.(1992)
- [石田 92] 石田, 桑原 : 分散人工知能 (1) 協調問題解決 , 人工知能学会誌 Vol.7 , No.6(1992)