

タスク結合によるタスク実行の効率化とタスクスケジューリング*

5M-8

伊藤成記[†] 渡辺裕之[‡] 渡邊豊英[†]名古屋大学工学部情報工学科[†]名古屋大学大学院工学研究科情報工学専攻[‡]

1 はじめに

計算機システムの利用用途は、単一タスク処理から複合的な処理へと拡大している。種々の関連するタスクを効率良く処理することが重要である。タスクを知的に、迅速に、効果的に実行するには、タスクの実行状態に応じて、また新規に発生するタスクの特性に合わせて、タスクの実行環境を整備し、タスクを実行・制御しなければならない。例えば、人間の賢い秘書は関連する要求を同時的、また、連続的に処理し、要求の内容に応じて適切な手順を計画して、要領良く柔軟に様々な仕事に対処している。

本稿では、次々に発生する複数の要求をその内容に応じて実行単位となるタスクを生成し、かつ関連するタスクを総合的、効率的に実行するための制御システムを報告する。特に、処理内容として類似する異なったタスク群をタスク結合機構の下に一緒に処理するための方法について言及する。

2 タスク実行制御の概略

種々様々な要求によって生成されるタスクは必ずしも総てが独立で、逐次的であることはない。相互に関連し合っている場合が少なくない。我々は知的に、かつ柔軟に効率良く要求を処理する枠組みとして、イベント・ハンドラとタスク・スケジューラから構成されたシステムを提案している。¹⁾ イベント・ハンドラはシステムに発生する要求に応じて適切なタスク集合を生成し、これらのタスクをタスク間の実行順序関係に基づいて整理化させる。一方、タスク・スケジューラは整理化されたタスクから現在の他のタスクの実行状態、及びそのタスクの実行条件を対応させて、実行可能なタスクから実行させる。イベント・ハンドラが要求に基づいて生成されたタスクを静的に順序付けるのに対し、タスク・スケジューラはシステムに存在する種々のタスクの状態に基づいて、順序付けられたタスクから実行順序が保持されるという条件の下で動的な視点より適切なタスクを実行する。

このような枠組みの下で、タスクの実行制御が実現される。タスクは処理の最小単位であり、これを実行するのがオブジェクトである。すなわち、タスクの実行器であるオブジェクトから捉えると、タスクはメッセージとみなすことができる。従って、タスクにはそのパラメータとして、(1)タ

スクが送信され、実行されるオブジェクト(実行オブジェクト)名、(2)タスクの処理の種類を表すメソッド名、(3)タスクが実行時に参照するオブジェクト(参照オブジェクト)名のリスト、(4)タスクの実行順序を表す先行タスクのリスト、(5)タスクを実行開始する時間を指示する実行開始時間が付帯している。また、それぞれを識別するための識別子を持つ。例えば、「資料の提出」という処理は、(1)資料の提出者、(2)資料を提出する、(3)資料と提出場所、(4)資料の作成、(5)〇日から×日の間、という属性が必要である。

3 タスクの類似性とタスク結合

i). タスクの類似性

複数のタスクが実行可能な状態にあるとき、それらには少なからず共通した処理手続きを含むタスクや、似かよった処理手続きから成るタスクが存在する。このように、共通の処理手続き、似かよった処理手続きが認められる二つ以上のタスクを類似関係にあるタスクと呼ぶ。類似関係のタスクを一つのタスクとして実行できれば、システム全体の処理効率は向上する。

タスク間の類似関係は個々のタスクに付帯するパラメータを照合・検査することにより判定される。この照合パラメータは実行オブジェクト名、メソッド名、及び参照オブジェクト名であり、実行オブジェクト名とメソッド名が一致し、かつ参照オブジェクトの一部が一致するタスクを類似関係があるとみなされる。参照オブジェクトはこの照合・検査のために、主参照オブジェクトと副参照オブジェクトに区分され、オブジェクトに定義されている個々のメソッドごとに分類される。主参照オブジェクトは、類似関係の照合に直接利用され、副参照オブジェクトは、類似関係の照合には利用されない。

ii). タスク結合

タスク結合は二つ以上のタスク間で類似関係が検出されたとき、一つのタスクとして結合する処理である。タスク結合によって一つにまとめられた結合タスクは、通常のタスクと同様に扱われ、その実行結果は結合される以前の個々のタスクを実行した結果と同じである。

結合タスクは類似関係の定義より、実行オブジェクト名、メソッド名、主参照オブジェクト・リストが一致しているので、新たに保持しなければならない情報は先行タスク・リスト、実行開始時間、副参照オブジェクト・リストである。図1に結合の様子を表した。

これらの新たに獲得する情報間の整合性処理には、実行開始時間の計算がある。結合タスクの実行開始時間は、結合さ

*Efficient Task Execution and Scheduling Based on the Union Method of Interrelated Tasks

[†]Department of Information Engineering, School of Engineering, Nagoya University

[‡]Department of Information Engineering, Graduate School of Engineering, Nagoya University

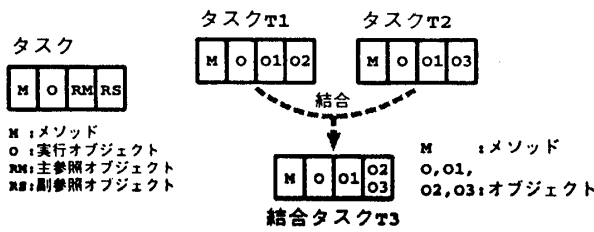


図 1: タスク結合

れるタスクの持つ実行開始時間の論理積によって求められる。すなわち、結合される総てのタスクの実行開始時間を満たすときのみ結合タスクの実行開始時間が満たされる。

タスクの順序関係は先行タスクによって表されており、先行タスクの状態を調べるには、その識別子からタスクを判別する。結合タスクは結合されるタスクの識別子を受け継ぎ、複数の識別子を持つことによって、先行タスクとして指定されているタスクが結合された場合にも、先行タスクが通常のタスクである場合と同様にタスクの状態を調べることができる。図 2 に具体的な結合処理状況を示す。

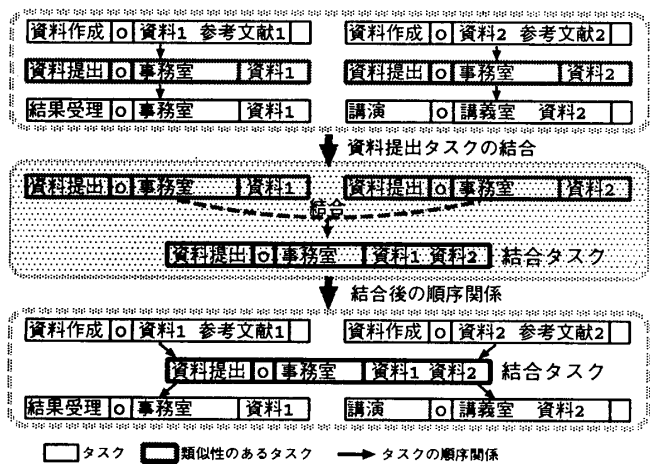


図 2: タスク結合と順序関係

結合タスクは、通常のタスクの実行と同様に、対応するオブジェクトのメソッドを起動することによって実行される。メソッドはその引数(参照オブジェクト)の数から結合タスクであるか否かを判別し、それに応じた方法でメソッドを実行する。

4 タスク結合によるスケジューリング

タスク・スケジューラはイベント・ハンドラで生成、整列化されたタスクからその実行開始条件が満たされたものを効

率的に実行させることが役割である。実行開始条件とは、先行タスクが終了していること、実行開始時間に到達していること、既実行タスクと競合しないことである。

先行タスクは総てが直列化されている訳でなく、非決定的にイベント・ハンドラで整列化されている。また、実行開始条件は実行開始時間が時刻印方式に関連するタスクに付与されていて、非決定的な整列タスク列と実行開始時間に基づいて、システム状態とリソース、プロセス間の状態によってタスク実行が制御される。

タスクの実行を制御するために、三つのバッファを用意している。それぞれのバッファは次のようなタスクを保持する。

- 1) Sleeping バッファ：実行開始時間を満たさないタスク
- 2) Waiting バッファ：実行開始時間を満たしたタスク
- 3) Running バッファ：実行中のタスク

図 3 にタスク・バッファ間の関係を示す。タスク結合は Waiting バッファで実行される。すなわち、Waiting バッファは実行開始時間が満たされたタスクを保持し、その下で結合タスクを生成し、残りの実行開始条件が満たされるまで一時的にタスクを保持する。

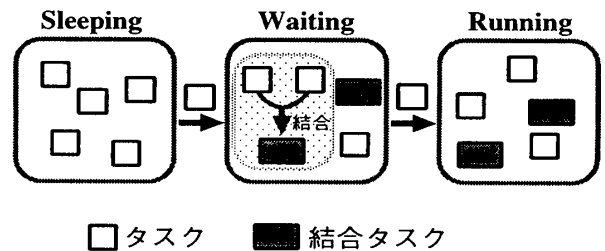


図 3: タスク・バッファ

5 おわりに

本稿では複数のタスクを処理しなければならない環境で、相互に関連するタスクを検出し、全体の実行効率を向上させる試みを述べた。そのためにタスク間に類似関係を定め、その類似関係によって結合タスクを生成する方法を示した。このような応用は人間の賢い秘書のように、複数の仕事を効率的に処理しなければならないオフィス環境の情報システムにとって重要な機構であり、秘書システムの実現に向けてシステムを開発する必要がある。

参考文献

- 1) H.WATANABE, T.WATANABE and N.SUGIE: "A Flexible Control Mechanism or Managing Interrelated/Interdependent Tasks Successively", Proc. of 18th COMPSAC, pp.78-83(1994).