(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4891225号
(P4891225)
(45)発行日　平成24年3月7日(2012.3.7)　(24)登録日　平成23年12月22日(2011.12.22)

(51)Int.Cl.　F I
G06F 9/48 (2006.01)　G06F 9/46 457

請求項の数 16 (全 15 頁)

(21)出願番号　特願2007-502997(P2007-502997)
(86)(22)出願日　平成17年3月8日(2005.3.8)
(65)公表番号　特表2007-528081(P2007-528081A)
(43)公表日　平成19年10月4日(2007.10.4)
(86)国際出願番号　PCT/US2005/007923
(87)国際公開番号　WO2005/086906
(87)国際公開日　平成17年9月22日(2005.9.22)
審査請求日　平成20年1月15日(2008.1.15)
(31)優先権主張番号　10/795,374
(32)優先日　平成16年3月8日(2004.3.8)
(33)優先権主張国　米国(US)

(73)特許権者　509123208
アビニシオ　テクノロジー　エルエルシー
アメリカ合衆国　０２４２１　マサチューセッツ州　レキシントン　スプリング　ストリート　２０１
(74)代理人　110000213
特許業務法人プロスペック特許事務所
(72)発明者　インチンゴロ　フランク
アメリカ合衆国　イリノイ州　６２０２５　エドワードヴィル　カントリー　クラブ　ビュー　１０８
(72)発明者　スタンフィル　クレイグ　ダブリュ
アメリカ合衆国　マサチューセッツ州　０１７７３　リンカーン　ハックルベリー　ヒル　ロード　４３

最終頁に続く

(54)【発明の名称】依存性グラフのパラメータのスコーピング

(57)【特許請求の範囲】
【請求項１】
プロセッサとデータ格納システムとを含むコンピューターシステムにおける、前記プロセッサによって実行される方法であって、
複数のタスクの実行順序を規定する依存性グラフに従って、複数の前記タスクの定義を取得するステップと、
前記タスクと関連付けられており、名前と割り当てられる値とを有する少なくとも１つのパラメータを含み、それぞれが異なるタスクスコープと関連付けられた、複数のコンテキストを保持するステップと、
第１の前記タスクで用いられる第１のパラメータに値をバインドするステップであって、複数の前記コンテキストのうちの第２の前記タスクと関係付けられる第１のコンテキストを前記依存性グラフに従って識別するステップおよび前記識別した前記第１のコンテキストから前記第１のパラメータに対する前記値を引き出すステップを含むステップと、
前記第２のタスクの実行中に前記第１のパラメータに割り当てるための値を決定するステップと、
前記第１のコンテキスト内の前記第１のパラメータに対する前記値を格納するステップと、
を含む方法。
【請求項２】
前記第１のコンテキストを識別するステップが、前記第１のタスクの前に実行する必要

があるタスクを識別するステップを含む請求項１の方法。
【請求項３】
前記第１のコンテキストを識別するステップが、前記第１のパラメータに値を割り当てるタスクを識別するステップを含む請求項２の方法。
【請求項４】
前記第１のパラメータに値を割り当てるタスクを識別するステップは、前記依存性グラフに従って最後に実行するタスクを識別するステップを含む請求項３の方法。
【請求項５】
プロセッサとデータ格納システムとを含むコンピューターシステムにおける、前記プロセッサに、以下のステップを実行させるための命令を含むコンピュータ可読媒体上に格納されるソフトウエアであって、
複数のタスクの実行順序を規定する依存性グラフに従って、複数の前記タスクの定義を取得するステップと、
前記タスクと関連付けられており、名前と割り当てられる値とを有する少なくとも１つのパラメータを含み、異なるタスクスコープとそれぞれが関係付けられた、複数のコンテキストを保持するステップと、
第１の前記タスクで用いられる第１のパラメータに値をバインドするステップであって、複数の前記コンテキストのうちの第２の前記タスクと関係付けられる第１のコンテキストを前記依存性グラフに従って識別するステップおよび前記識別した前記第１のコンテキストから第１の前記パラメータに対する値を引き出すステップを含むステップと、
前記第２のタスクの実行中に前記第１のパラメータに割り当てるための値を決定するステップと、
前記第１のコンテキスト内の前記第１のパラメータに対する前記値を格納するステップと、
を実行させるためのソフトウエア。
【請求項６】
前記第１のコンテキストを識別するステップが、前記第１のタスクの前に実行する必要があるタスクを識別するステップを含む請求項５のソフトウエア。
【請求項７】
前記第１のコンテキストを識別するステップが、前記第１のパラメータに値を割り当てるタスクを識別するステップを含む請求項６のソフトウエア。
【請求項８】
前記第１のパラメータに値を割り当てるタスクを識別するステップは、前記依存性グラフに従って最後に実行するタスクを識別するステップを含む請求項７のソフトウエア。
【請求項９】
プロセッサとデータ格納システムとを含むコンピューターシステムに実装された、タスク管理システムであって、
複数のタスクの実行順序を規定する依存性グラフに従って、複数の前記タスクを定義するよう構成されるタスクマネージャ定義モジュールと、
前記タスクと関連付けられており、名前と割り当てられる値とを有する少なくとも１つのパラメータを含み、それぞれが異なるタスクスコープと関係付けられた、複数のコンテキストを保持する格納空間を有するタスクマネージャ動的コンテキストモジュールと、
前記コンテキストへのアクセスを有するタスクマネージャーパラメーターバインドモジュールであって、
第１の前記タスク内で用いられる第１のパラメータに値をバインドするステップであって、複数の前記コンテキストのうちの第２の前記タスクと関係付けられる第１の前記パラメータコンテキストを前記依存性グラフに従って識別するステップおよび前記識別した前記第１のコンテキストから前記第１のパラメータに対する値を引き出すステップと、
前記第２のタスクの実行中に前記第１のパラメータに割り当てるための値を決定するステップと、

前記第１のコンテキスト内の前記第１のパラメータに対する前記値を格納するステップと、
を実行するタスクマネージャーパラメーターバインドモジュールと、
を含むシステム。
【請求項１０】
前記第１のコンテキストを識別するステップが、前記第１のタスクの前に実行する必要があるタスクを識別するステップを含む請求項９のシステム。
【請求項１１】
前記第１のコンテキストを識別するステップが、前記第１のパラメータに値を割り当てるタスクを識別するステップを含む請求項１０のシステム。
【請求項１２】
前記第１のパラメータに値を割り当てるタスクを識別するステップは、前記依存性グラフに従って最後に実行するタスクを識別するステップを含む請求項１１のシステム。
【請求項１３】
プロセッサとデータ格納システムとを含むコンピューターシステムに実装された、タスク管理システムであって、
複数のタスクの実行順序を規定する依存性グラフに従って、複数の前記タスクの定義を取得するための手段と、
前記タスクと関連付けられており、名前と割り当てられる値とを有する少なくとも１つのパラメータを含み、それぞれが異なるタスクスコープと関係付けられた、複数のコンテキストを保持するための手段と、
第１の前記タスク内で用いられる第１のパラメータに値をバインドするための手段であって、複数の前記コンテキストのうちの第２の前記タスクと関係付けられる第１のコンテキストを前記依存性グラフに従って識別するステップおよび前記識別した前記第１のコンテキストから前記第１のパラメータに対する値を引き出すステップを含む手段と、
前記第２のタスクの実行中に前記第１のパラメータに割り当てるための値を決定するための手段と、
前記第１のコンテキスト内の前記第１のパラメータに対する前記値を格納するための手段と、
を含むシステム。
【請求項１４】
前記第１のコンテキストを識別するステップが、前記第１タスクの前に実行する必要があるタスクを識別するステップを含む請求項１３のシステム。
【請求項１５】
前記第１のコンテキストを識別するステップが、前記第１のパラメータに値を割り当てるタスクを識別するステップを含む請求項１４のシステム。
【請求項１６】
前記第１のパラメータに値を割り当てるタスクを識別するステップは、前記依存性グラフに従って最後に実行するタスクを識別するステップを含む請求項１５のシステム。
【発明の詳細な説明】
【発明の詳細な説明】
【０００１】
発明の背景
本発明は、依存性グラフのパラメータのスコーピングに関する。
【０００２】
コンピュータのジョブ管理システムはプログラムを実行すべき順序に関する制約に従って、コンピュータープログラム、または他の処理の実行をシーケンス化するために用いられている。順序制約を規定する一手法は、依存性グラフを用いる。プログラムは、ジョブがシステムに提示された時点で規定される引数を受け入れることができる。このようなジョブ管理システムの一例には、異なるコンピュータまたは異なる種類のコンピュータ（例

えば、スーパーコンピュータ）上で異なるジョブを実行するよう規定でき、異なるジョブ間の依存性を、ジョブを実行する前に明示的に識別する分散型バッチシステムがある。ジョブ間の情報交信の一手法は、ファイルシステム内のファイルを介する等の共通データ格納を介して行う。このようなジョブ管理の別の例は、分散型パーソナルコンピュータ環境におけるジョブスケジュール、例えば、特定シーケンス内の異なるコンピュータ上で実行する必要があるスケジュールメンテナンスタスクに関する。

【０００３】

概要

一般的な局面では、本発明は、パラメータの値をバインドするための方法を特長とする。幾つかのタスクが、依存性グラフに従って定義される。多数のパラメータコンテキストが保持され、それぞれはタスクの異なるスコープと関係付けられている。第１のタスクに用いるパラメータは値とバインドされる。このバインドは、依存性グラフに従って第１のコンテキストを識別することと、識別したコンテキストからパラメータに対する値を引き出すことが含まれる。

【０００４】

この局面は、以下の特長の内の一つ以上を含むことができる：

【０００５】

パラメータに割り当てるための値は、第２のタスクを実行中に決定される。パラメータに対する値は、第２のタスクと関係付けられる第１のパラメータコンテキストに格納される。

【０００６】

依存性グラフは、タスクに対する実行順序を規定する。第１のコンテキストの識別には、第１のタスクの前に実行する必要があるタスクを識別することが含まれる。第１のコンテキストの識別には、第１のパラメータに値を割り当てるタスクを識別することと、依存性グラフに従って最後に実行するタスクを識別することも含まれる。

【０００７】

別の局面では、一般的に、本発明は、少なくとも幾つかのタスクが互いに順序付けられていない複数のタスクに対する順序制約の仕様を受け入れることを含むパラメーターアクセスをチェックするための方法を特長とする。このチェックには、一つ以上のパラメータのそれぞれに対して、これらパラメータにアクセスするタスクを識別することと、順序制約の仕様に従って、識別したタスクによるパラメータへのアクセスにおける潜在的なコンフリクトをチェックすることが含まれる。

【０００８】

本発明の局面には、以下の長所の内の一つ以上が含まれる：

【０００９】

タスク間で渡される明示的なパラメータを用いることにより、パラメータ値のコンフリクトおよび曖昧さをチェックできる。例えば、このようなチェックにより、タスク計画の繰り返される実行においてパラメータが同一値をとることが確実になる。

【００１０】

依存性グラフの手順に従うスコーピングは、ネスティングまたは計画およびサブ計画に従ってスコーピングすることが好ましいことがある。なぜなら、パラメータは、計画内のピアタスク間で渡すことができるからである。

【００１１】

多数のパラメータコンテキストの使用により、スコーピング規則に従ったコンフリクトまたは曖昧さを招くことのない多重使用の影響を受ける依存性グラフの異なる部分で、同一パラメータ名を用いることが可能になるので、名前コンフリクトを低減できる。これにより、異なる開発者が名前コンフリクトを招かずに、異なる名前領域を明示的に識別しなくても、計画の異なる部分を書くことが可能になる。

【００１２】

本発明の他の特長または利点は、以下の説明およびクレームから明らかである。

【００１３】
説明
概略
図１を参照すると、タスク管理システム１００は、計画のどのタスクを他のタスクが実行される前に完了しなければならないかを規定する依存性制約に従って、全体計画を作り上げるタスクの実行を制御する。有向の依存性グラフ１３２を用いて従属性制約が表され、各ノード１３４が計画の異なるタスクに対応し、あるノードから別のノードへの有向の弧１３６のそれぞれが、第１の「上流」ノードに対応するタスクは第２の「下流」ノードと対応するタスクが実行される前に完了しなければならないという順序制約を示す。更に一般的には、ノードと対応するタスクは、グラフ内の全ての上流タスクが完了した後だけ実行できる。依存性グラフはタスクの部分的な順序付けを確立する。従って、タスクは、グラフベースの依存性制約を含む制約を受けて同時に実行できる。また、タスクを異なるコンピュータ上で実行して、計画の全体実行における並列処理による速度向上を達成してもよい。
【００１４】
タスク管理システム１００には、計画仕様１２０を読み込むタスクマネージャ１１０が含まれる。計画仕様は、依存性グラフ１３２を定義するグラフ仕様１２２、およびそれぞれのタスクの特性を定義するタスク仕様１２４を含む。グラフ仕様１２２は、ノード１３４のリスト、およびそれぞれがグラフのソースノードおよび宛先ノードを識別する弧１３６のリストにより、依存性グラフを定義する。各ノードはタスク定義と関係付けられていて、これはタスク仕様１２４で規定され、タスクで実行される特定操作を規定する。
【００１５】
タスクは、タスクテンプレートを用いて定義でき、計画は計画テンプレートを用いて定義できる。タスクまたは計画のためのテンプレートは、シンボル的に表され、タスクが実行されるまで必ずしも値にバインドされるとは限らないパラメータへの参照を含むことができる。例えば、タスクテンプレートは、FILENAMEと命名されたパラメータへの参照を含むとともに、「ドル記号」構文の＄FILENAMEを用いて表現するそのパラメータへの参照を含むことができ、タスクの実行前または実行中に、タスクマネージャ１１０により特定の値にバインドされる。
【００１６】
幾つかのパラメータは、全体計画に対してグローバルであり、全体計画が最初に実行される時に値を割り当てられる。このようなパラメータは全体として、計画の正式なパラメータとすることができ、例えば、全体計画を呼び出すコマンド内で規定することができるか、または計画を呼び出すユーザからインタラクティブに引き出すことができる。
【００１７】
他のパラメータは、計画が最初に実行される時に規定される値を持っているとは限らない。このような動的パラメータは、一つ以上のタスクの実行中またはその実行の結果として、値を割り当てることができ、次いで、他のタスクのテンプレートにより参照される。例えば、図２では、パラメータＡは、タスク１によりある値に設定することができ、次いで、その値をタスク２が用いることができる。
【００１８】
タスク管理システム１００は、以下に説明に関連する幾つかの追加の特長をサポートする。第１の特長は、ノード１３４が単一タスクではなく、全体の「サブ計画」と関係付けることができることである。サブ計画は、部分的に順序付けしたタスクの下位レベルの計画である。サブ計画は、トップレベルの計画仕様と同一または類似の構造の計画仕様１２０を有する。
【００１９】
第２の特長は、タスクがその実行を通じて計画仕様１２０を修正できることである。詳細には、タスクは、グラフ１３２に追加のノード１３４および弧１３６を加え、タスクマネージャ１１０が提供するサービスを用いて追加ノードと関係付けられるタスクに対する

特性を定義することができる。例えば、第１タスクは、実行すべき未知数（ランタイム前には）のタスクを有する計画に対するテンプレートを用いてサブ計画を生成できる。テンプレート内のタスクの数は、動的パラメータを用いてランタイム時にバインドする。生成したサブ計画は、タスクマネージャ１１０により第１タスクの高次レベルの計画に組み込まれる。

【００２０】

バインド手法

タスクマネージャ１１０は、以下に説明するようにパラメーターバインドの幾つかの選択的な手法の内の一つを用いる。あるいは、タスクマネージャ１１０は、例えば、パラメータの宣言に従って異なるパラメータには異なる手法を用いる組合せバインド手法を用いる。幾つか（または全て）の手法が異なる一局面は、パラメータ定義の適用範囲にある。

【００２１】

手法１

図３を参照すると、第１バインド手法では、タスクマネージャ１１０は、全てのパラメータに対して単一のグローバルコンテキスト３１０を保持する。タスクマネージャ１１０は、任意のパラメータの単一コピーを保持する。すなわち、タスクマネージャ１１０は、幾つかの異なるパラメータ名３２２のそれぞれを、そのパラメータの現在値３２４と関係付けるデータ構造３２０を保持する（対応するパラメータに値が割り当てられていない場合、値３２４は空のことがある）。このデータ構造３２０を用いて、グローバルパラメータの値および実行する際にタスクが動的に割り当てる値を保持することができる。

【００２２】

図２および図３を参照すると、この第１手法の実施例では、タスク１は、割当てA=FOOにより示されるように値FOOをパラメータＡに割り当てる。タスクマネージャ１１０は、割り当てられた値FOOをタスク１から受け取り（タスク１からパラメータＡの値セル３２４への矢印により示される）、値セル３２４にその値を格納する。タスクマネージャ１１０は後で、タスク２に対するテンプレート内のパラメータＡへの参照をバインドするために、値FOOをタスク２に提供する。本実施例では、タスク２は、割当てB=$A/BAZを含み、パラメータＢには値FOO/BAZが割り当てられることになる。タスクマネージャ１１０は、Ｂに対して割り当てられた値を受け取り、グローバルコンテキスト３１０に格納する。グラフ１３２に従って、タスク２は、タスク１が完了した後のみ実行される。従って、パラメータＡへのFOOの割り当ては、タスク２の実行の前に実行されていることが保証される。

【００２３】

一般的に、計画のマルチタスクのグループは、同時に実行することができる。特定の依存性グラフでは、互いに順序が付けられていない二つ以上のタスクは、同時に実行できるか、または従属性グラフ内でそれらの間に直接パスがない場合、予測できないシーケンスで実行される可能性がある。図２では、タスク２および３が、このような順序不定のタスクである。図２の実施例では、タスク３がパラメータＡにBARの値を割り当てる。タスク３をタスク２の前に実行する場合、パラメータＢには、タスク２でBAR/BAZの値が割り当てられる一方で、タスク３をタスク２の後で実行する場合、パラメータＢには、FOO/BAZの値が割り当てられる。

【００２４】

幾つかのアプリケーションでは、Ｂに割り当てられる値の不確定性は好ましいことではない。このような状況を防ぐ一手法は、依存性グラフの静的なチェックを実行して、このような不確定性を示す計画の使用を禁じることである。チェックは、グラフによって順序付けられていないタスクグループの識別を含む。特定パラメータを、グループの内の一つのタスクが用いて、そのグループの別のタスクが割り当てる場合、曖昧さが発生する可能性がある。同様に、そのようなグループの内の二つのタスクが、特定のパラメータに値を割り当てる場合にも、コンフリクトが存在する。一般に、多数のタスクが特定パラメータにアクセスする任意の順序不定タスクのグループでは、どのタスクもそのパラメータに値

を割り当てることができない。この静的チェックは、タスク２とタスク３との間の、パラメータＡの参照に関するコンフリクトを識別する。タスク管理システム１００は、静的チェックにより識別されるコンフリクトを扱う（例えば、エラーハンドリングメカニズムを用いて）。本実施例で注意すべきは、タスク２およびタスク５もまた、予測できない順序で実行できるが、両者はパラメータＡの値を割り当てるのではなく、使用するのであり、従って、コンフリクトまたは曖昧さを表すことはない。

【００２５】

依存性グラフの静的チェックは、計画実行の前に、タスクマネージャ１１０が実行するか、または代替として、計画仕様の妥当性評価をする別のモジュールが、処理前に実行してもよい。グラフがタスクの実行により修正される状況では、静的チェックを再度実行するか、または強化してグラフ修正にあたる。

【００２６】

サブ計画を含む計画では、本手法の２つの変形を用いることができる。第１の変形では、タスクマネージャ１１０は、サブ計画の全てのレベルについての全パラメータに対して単一のコンテキストを保持する。サブ計画をもつノードを含む計画の静的チェックでは、そのサブ計画の任意のタスクによるパラメータ割り当ては、そのパラメータの値を用いる高位レベルの計画内の、任意の順序不定（そのサブ計画をもつノードに関する）タスクとコンフリクトする。

【００２７】

サブ計画と関わる第２の変形では、タスクマネージャ１１０は、各サブ計画のパラメータに対して別々のコンテキストを保持し、サブ計画が完了した時に、サブ計画から次に高位レベルの計画にどのパラメータがエクスポートされるかを、サブ計画の仕様が明示的に識別する。そのような変形では、静的チェックが、サブ計画のエクスポートされたパラメータと、サブ計画実行と順序付けられていない次に高位レベルの計画内のタスクのそのパラメータへのアクセスとの間のコンフリクトと関わる。

【００２８】

本手法の別のバージョンでは、タスクマネージャ１１０は、任意の特定のグローバル動的パラメータへの値の繰り返し割り当てを禁止する。

【００２９】

手法２

図４Ａを参照すると、第２手法で、タスクマネージャ１１０は、全体として計画に設定されるパラメータに対するグローバルコンテキスト３１０、および依存性グラフの異なるノードとそれぞれが関係付けられる多数の動的コンテキスト４１０を保持する。パラメータを割り当てる各タスクは、そのパラメータの新規「インスタンス」を生成し、割り当てられた値を（タスクマネージャ１１０を介して）そのタスクと対応するノードと関係付けられる動的コンテキスト４１０内に格納する。従って、タスクマネージャ１１０は、パラメータと関係付けられる多数の値を格納でき、それぞれの値は、パラメータの異なるインスタンスと関係付けられる。任意の特定パラメータに対しては（例えば、Ａ）、タスクマネージャ１１０は、パラメータが割り当てられている各タスクと関係付けられたパラメータの別々の値を保持する（各タスクは別々のノードおよび別々の動的コンテキスト４１０と対応しているからである）。

【００３０】

図４Ｂおよび図４Ｃを参照すると、パラメータの各インスタンスおよびその関係する動的コンテキスト４１０は、どのタスクがそのインスタンスに割り当てられた値を「見る」かを決定するタスクのスコープを有する。各動的コンテキスト４１０は、依存性グラフ１３２に従ってパターン４２０に付随する動的コンテキスト４１０間の関連性に基づいて、タスクの異なるスコープと関係付けられる。例えば、タスク１でA=FOOを割り当てられたパラメータＡのインスタンス（動的コンテキスト４１０に格納されている）は、タスク３でA=BARを割り当てられているパラメータＡのインスタンスと関係付けられるタスクスコープ４３２とは異なるタスクスコープ４３０を有する。タスク２でB=$A/BAZを割り当てら

れているパラメータＢのインスタンスと関係付けられるタスクスコープ４３４は、タスク４を含むが、タスク５は含まない。本手法では、タスクスコープまたはパラメータのインスタンス（および関係するその動的コンテキスト４１０）の「スコープ」は、パラメータのインスタンスに値を割り当てられたタスクおよびそのタスクの後「下流」に順序付けられたタスクのみを含む。タスクがパラメータを参照する場合、タスクマネージャ１１０は、そのスコープがタスクを含むパラメータのインスタンスの値をその参照にバインドする。

【００３１】

タスク１が値FOOをパラメータＡに割り当てる場合、タスクマネージャ１１０は、タスク１と関係付けられる動的コンテキスト４１０に値を格納する。同様に、タスク３が値BARをパラメータＡに割り当てる場合、タスクマネージャ１１０は、タスク３と関係付けられる動的コンテキスト４１０に値を格納する。タスク２は、タスク１で値A=FOOを割り当てられたパラメータＡのインスタンスのスコープ内にあるが、タスク３でA=BARを割り当てられたパラメータＡのインスタンスのスコープ内にはない。従って、本スコーピング手法のもとでは、タスク２のＢの値に関する不確定性または曖昧さがない。タスクマネージャ１１０は、タスク２でのパラメータＡへの参照を、タスク１に対する動的コンテキスト４１０からの値とバインドする。タスク２は、値FOO/BAZをＢに割り当て、タスク２と関係付けられる動的コンテキスト４１０内に格納するためにタスクマネージャ１１０に提供する。

【００３２】

一般に、実行中は、タスクマネージャ１１０は、パラメータに値を割り当てるタスクで終了するグラフを通る上流パス（すなわち、単一弧を通る直接パス、または既に実行されたタスクを通る間接的なパス）を考慮することにより、タスク内のパラメータへの参照に値をバインドする。本実施例では、タスク２でパラメータＡをバインドする場合、タスクマネージャ１１０は、パラメータがセットされているタスク１まで上流パスをたどる。

【００３３】

図２の実施例では、タスク４は、タスク１のパラメータＡのインスタンス、およびタスク３のそれの両方のスコープ内にある可能性をもつ。すなわち、タスク４からタスク１（タスク２を通じて間接的に）およびタスク３（直接的に）までの上流パスがあり、それぞれ値をＡに割り当てている。しかしながら、本実施例では、タスク３はタスク１の後に実行するという制約があるので曖昧さはない。多数の上流のパラメータ割り当てがあり、他の全ての後に順序付けられている一つのタスクがある場合、曖昧さはなく、最後のタスクの動的コンテキスト４１０のパラメータのインスタンスに割り当てられた値が用いられる。

【００３４】

第１の手法と同様に、潜在的な曖昧さは、依存性グラフから静的にチェックできる。詳細には、グラフ内のノード（タスク）でのパラメータへの各参照に対して、パラメータへの値を割り当てることができる上流タスクは、最初に識別される。そのようなタスクがない場合、パラメータは、定義されていない値をもつことになるか、または計画に対するグローバルコンテキストから提供される。正確に一つの上流タスクがある場合、曖昧さはない。パラメータに値を割り当てる多数の上流タスクがある場合、これらのタスク（すなわち、依存性グラフにより他のものの後に実行する制約がある）の内の唯一つが他のものの後に順序付けられない限り、曖昧さが存在する。

【００３５】

実装手法

図５を参照すると、タスク仕様５００（図１のタスク仕様１２４に含まれている）は、宣言部５１０および命令部５２０を含む。宣言部５１０は、対応するタスクの実行を通じて割り当てられる動的パラメータを識別する（例えば、一つ、または複数の宣言命令文を用いて）。図５に示すタスク仕様は、図２のタスク１と対応する。パラメータＡは、宣言「動的Ａ」５１２として示すように、タスクにより動的に割り当てられるとして宣言する

。宣言部５１０はまた、利用可能なパラメータのスコープに従って、タスクマネージャ１１０がバインドするよう要求されるパラメータへの参照も識別する。

【００３６】

命令部５２０は、タスクが実行する操作のためのコンピュータ命令（例えば、プログラム言語または他の手続き命令文）、および命令「A=FOO」５２２等の動的パラメータに値を割り当てるための命令を含む。例えば、このようなコンピュータ命令は、ユーザが規定するか、またはコンピュータ処理により自動的に生成できる。

【００３７】

タスクマネージャ１１０は、計画に対するタスク仕様の宣言部を処理して、対応するタスクの実行中に、どのパラメータを割り当てるか、または参照するか、またはその両方をすべきか決定できる。タスクマネージャ１１０は、タスク仕様の命令部を追加して処理することにより、または代替として、動的パラメータ割り当てまたは参照に対する別の種類の宣言文を用いることにより、割り当てまたは参照を区別できる。例えば、図２に示す実施例では、タスクマネージャ１１０は、タスク仕様を構文解析して、タスク２がパラメータＡを参照する一方、タスク３がＡに値を割り当てることを決定できる。これにより、タスクマネージャ１１０は、タスク２での参照およびタスク３での割り当てがコンフリクトする第１手法（図３参照）に対する静的チェックのような静的チェックを実行できる。

【００３８】

宣言部が、参照されるべきパラメータを識別できる場合、タスクマネージャ１１０は、パラメータ環境と命令部５２０の命令を実行する前にバインドした値とをアセンブルできる。タスクの環境をアセンブルすると、タスクマネージャ１１０は、タスク仕様の命令部の命令を実行する処理を呼び出す。次いで、その処理は、それに対して作成された環境からのパラメータ値にアクセスする。実行中は、動的パラメータに対する割り当て命令処理の結果、タスク実行の出力である動的パラメータ割り当て部５３０に（パラメータ、値）の対が記録される。例えば、対（A, FOO）５３２が割り当て部５３０に記録されることになる。タスクマネージャ１１０は、命令部５２０の命令実行の後、割り当て部５３０を受け取り、その内容を用いてパラメータ値を更新する。

【００３９】

本手法の一特定の実装では、命令部５２０には、「ksh」等のシェルスクリプト言語の命令文が含まれる。命令を解釈するシェルスクリプトを呼び出す前に、タスクマネージャ１１０は、宣言部５１０を用いてシェル命令を処理するのに用いるパラメータ値の環境をアセンブルする。この実装では、割り当て命令文は、スクリプトのテキスト出力への出力命令文の形式をとる。すなわち、割り当て「A=FOO」を用いるのではなく、スクリプト命令は「PRINT A=FOO」等の命令文を含み、その結果「A=FOO」の行が出力される。スクリプトのテキスト出力は、動的パラメータ割り当て部５３０を形成する。タスクマネージャ１１０は、テキスト出力を受け取り、適切な動的コンテキスト４１０に値を格納する割り当て命令文を解釈する。

【００４０】

別の実装では、命令部５２０の命令を解釈する処理は、タスクマネージャ１１０と通信し、タスクマネージャ１１０は、動的パラメータの値を格納、またはアクセスするリクエストのサービスを行う。パラメータ参照を値にバインドして命令を実行する必要がある場合、処理は、そのパラメータに対する適切な動的コンテキスト４１０を決定し、その動的コンテキスト４１０の値を戻すタスクマネージャ１１０にリクエストを送る。同様に、パラメータを値に割り当てる場合、処理は、適切な動的コンテキスト４１０の値を格納するリクエストを送る。

【００４１】

代替の実装では、タスク仕様５００の宣言部５１０は、実行中にタスクが必要とするパラメータを識別せず、タスクの命令が実行されている間、パラメータ値をリクエストするのに適した交信メカニズムを利用することができない。この実装では、そのタスクマネージャ１１０は、スコープがタスクを含む全てのパラメーターインスタンスの完全なリスト

をアセンブルする。バインドが曖昧さを招く可能性があるパラメータに対しては（例えば、多数の上流の順序不定タスクによりパラメータに値が割り当てられたため）、パラメータ値がambiguous-Value（曖昧な値）等のインジケータで置換されて、パラメータがタスク内で参照されたというエラーが発生する。

【００４２】

宣言部５１０が、動的に割り当てられるか、および／または各タスク内で参照されるパラメータの宣言を含まない場合、依存性グラフの静的チェックが不可能になることもある。しかしながら、タスクマネージャ１１０は、計画実行中に発生するかもしれない幾つかの潜在的コンフリクトを依然として識別できる。例えば、単一のグローバルコンテキスト（図３参照）をもつスコーピング手法を用いると、タスクマネージャ１１０が計画の実行中に、多数の順序不定タスクが一つのパラメータに値を割り当てたことを検出した場合、タスクマネージャ１１０は、潜在的なコンフリクトを識別できる。次いで、タスク管理システム１００は、タスクマネージャが識別したコンフリクトを扱うことができる。

【００４３】

実施例

図６Ａを参照すると、動的に割り当てたパラメータを用いる実施例が、幾つかのデータファイル６４１～６４３の処理に関わっていて、そのそれぞれは全体計画の異なるタスク６３１～６３３により処理される。データファイル６４１～６４３のファイル名は計画を最初に実行する時には未知である。正確には、ファイル名はテキストファイル６２２にリストアップされ、その名称は全体計画に対してグローバルパラメータLIST_FILEとして提供される。

【００４４】

計画は、LIST_FILEのファイル６２２を開いて、その内容を読み出すのを受け持つ第１タスク６２０を含む。このタスク６２０は、ファイルを開いて内容を読み取り、リストにされた別々のデータファイルの数を決定する。本実施例では、リストファイル６２２は３つのファイル名、FNAME1.DAT、FNAME2.DAT、およびFNAME3.DATを有する。タスクはリストにされたファイル６４１～６４３を、例えば、データネットワークにわたるファイル転送プロトコルを用いて検索する。タスク１は、幾つかの動的パラメータ、NUM_INPUTおよびDATA_FILE_1～DATA_FILE_3に値を割り当て、適切なコンテキスト６１０にそれを記録するタスクマネージャ１１０に、この割り当て情報を渡す（例えば、使用するスコーピング手法に依存するグローバルコンテキストまたはタスク１と関係付けられる動的コンテキスト）。

【００４５】

本実施例のこの第１バージョンでは、開始時に規定した計画は、入力ファイルを処理するための、タスク２～４（６３１～６３３）の３つのインスタンスのタスクを有する。各タスクは、タスク１が割り当てた動的パラメータを参照することにより、その対応する入力データファイルの名称を検索する。

【００４６】

本実施例の変形は、タスクが計画仕様を修正することができるという利点がある。計画は、開始時には、単一タスク、GET NAMESタスク１（６２０）だけを有する。GET NAMESタスクが、処理するデータファイルが３つあるということを決定した後、タスクは、タスクマネージャ１１０のサービスを利用して、３つの処理タスク６３１～６３３を作成し、それらを自身に接続する（タスク１）。タスク１が完了した後、タスクマネージャ１１０は、上記のように新規に作成したタスク６３１～６３３を実行する。

【００４７】

代替として、本実施例の別の変形では、タスクマネージャ１１０は、修正可能な幾つかのタスクを有する計画テンプレートを用いることができる。図６Ｂを参照すると、計画は、開始時には、二つのタスク、GET NAMESタスク１（６２０）、およびPROCESSタスク２（６５０）を有する。GET NAMESタスクが、処理するデータファイルが３つあるということを決定した後、PROCESSタスクは、修正可能な幾つかのタスクを有する計画テンプレート

を用いてサブ計画を作成する。PROCESSタスクは、動的パラメータNUM_INPUTを参照して、３つの（順序付けられていない）処理タスク６３１～６３３をもつサブ計画を作成する。タスクマネージャ１１０は、上記のように、新規に作成したサブ計画内のタスク６３１～６３３を実行する。

【００４８】

上記のパラメータースコーピング手法は、コンピュータ上で実行するためのソフトウエアを用いて実装できる。例えば、ソフトウエアは、一台以上のプログラムされるか、またはプログラム可能な（分散型、クライアント／サーバ、またはグリッド型等の各種のアーキテクチャからなってもよい）コンピューターシステム上で実行される一つ以上のコンピュータープログラム内に手順を形成する。コンピューターシステムそれぞれは、少なくとも一つのプロセッサ、少なくとも一つのデータ格納システム（揮発性または不揮発性メモリおよび／または格納素子を含む）、少なくとも一つの入力装置またはポート、および少なくとも一つの出力装置またはポートを含む。ソフトウエアは、例えば、計算グラフの設計および構成と関連する他のサービスを提供する長いプログラムの一つ以上のモジュールを形成する。

【００４９】

ソフトウエアは、汎用または特殊用途のプログラマブルコンピュータにより可読なＣＤ－ＲＯＭ等の媒体上に提供するか、またはそれを実行するコンピュータにネットワークを越えて配信する（伝搬信号で符号化する）。全ての機能は、特殊用途のコンピュータ上で、またはコプロセッサ等の特殊用途ハードウエアを用いて実行できる。ソフトウエアは、ソフトウエアが規定する計算の異なる部分を異なるコンピュータで実行する分散形式で実装してもよい。かかるコンピュータープログラムはそれぞれ、汎用または特殊用途プログラマブルコンピュータにより可読な格納媒体または装置（例えば、固体メモリまたは媒体、または磁気式または光式媒体）に格納するかまたはダウンロードし、その格納媒体、または装置をコンピューターシステムが読み取って、本明細書で説明した手順を実行する場合に、コンピュータを構成、および動作させるのが好ましい。発明性のある本システムは、コンピュータープログラムにより構成したコンピュータ可読格納媒体として実装されると考えてもよく、そのように構成した格納媒体は、コンピューターシステムに規定の、かつ所定の方法で本明細書で説明した機能を実行するよう動作させる。

【００５０】

言うまでもなく、上記説明は、説明を意図したものであり、添付クレームの適用範囲により定義される本発明の適用範囲を制限する意図はない。他の実施の形態は、以下のクレームの適用範囲内にある。

【図面の簡単な説明】

【００５１】

【図１】タスク管理システムである。

【図２】依存性グラフである。

【図３】タスクマネージャ内の単一グローバルコンテキストを含む図である。

【図４Ａ】タスクマネージャ内の多数のタスク特定の動的コンテキストを含む図である。

【図４Ｂ】図２の依存性グラフに従って、動的コンテキスト間の関連性を示す図である。

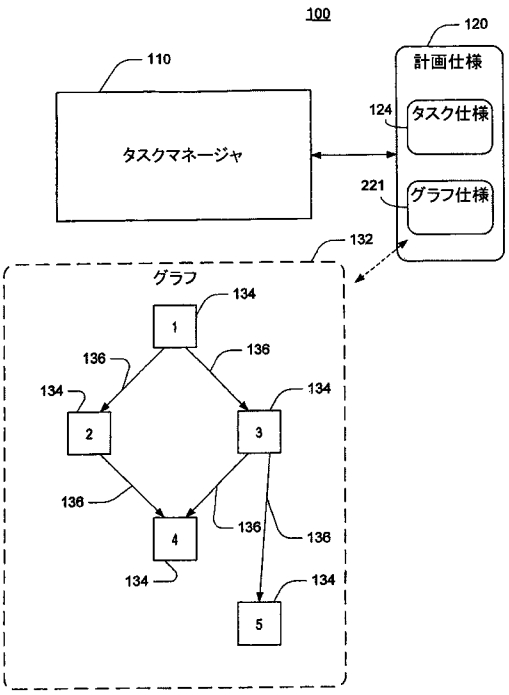
【図４Ｃ】図２の依存性グラフ内のタスクのスコープを示す図である。

【図５】タスク仕様である。

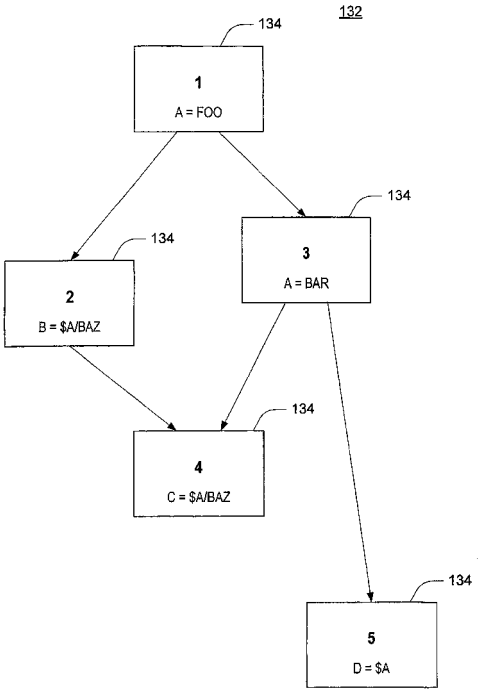
【図６Ａ】動的パラメータを用いるアプリケーションの実施例である。

【図６Ｂ】動的パラメータを用いるアプリケーションの実施例である。

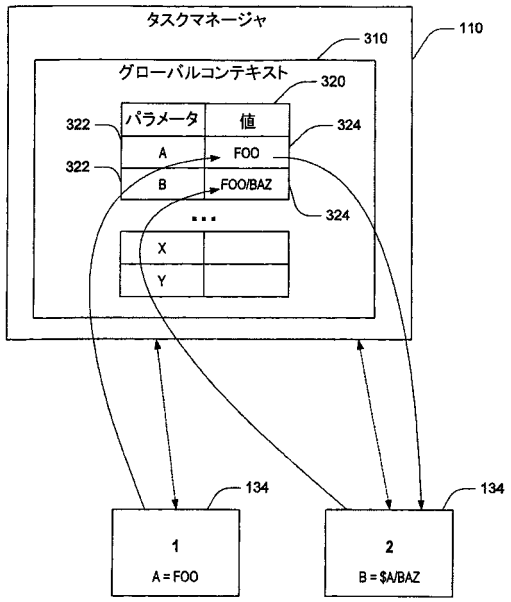
【図１】


100
120
計画仕様
110
タスクマネージャ
124
タスク仕様
221
グラフ仕様
132
グラフ
134
1
136
2
3
4
5

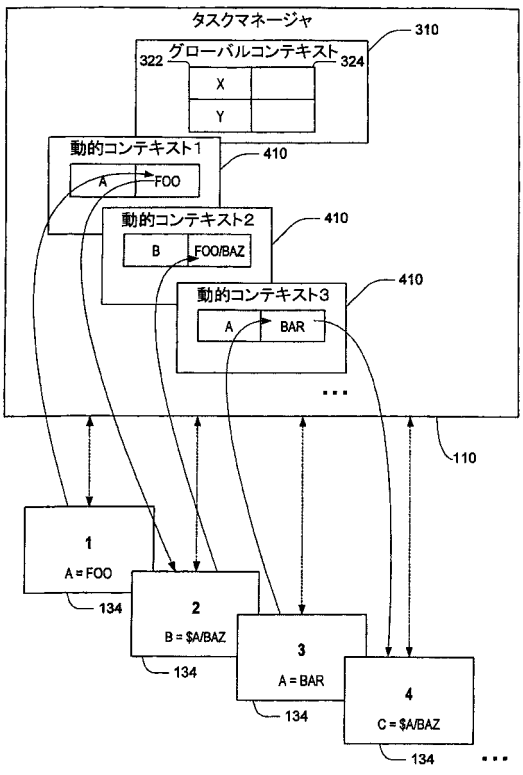
【図２】


132
134
1
A = FOO
2
B = $A/BAZ
3
A = BAR
4
C = $A/BAZ
5
D = $A

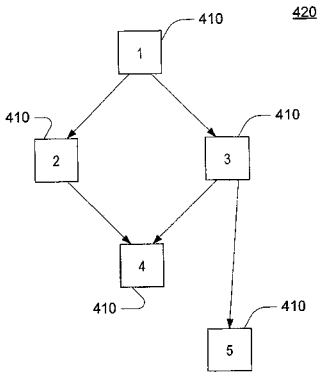
【図３】


タスクマネージャ
110
310
グローバルコンテキスト
320
322
パラメータ
値
324
A
FOO
B
FOO/BAZ
X
Y
134
1
A = FOO
2
B = $A/BAZ

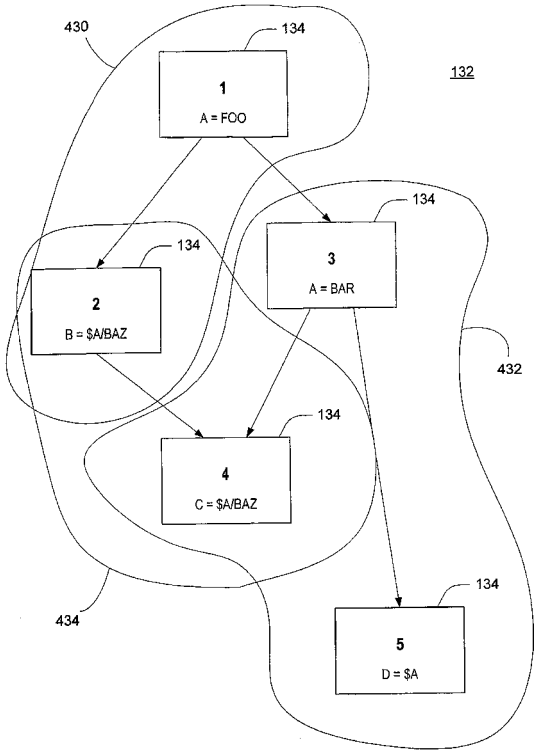
【図４Ａ】


タスクマネージャ
310
グローバルコンテキスト
322
324
X
Y
動的コンテキスト1
410
A
FOO
動的コンテキスト2
B
FOO/BAZ
動的コンテキスト3
A
BAR
110
1
A = FOO
134
2
B = $A/BAZ
3
A = BAR
4
C = $A/BAZ

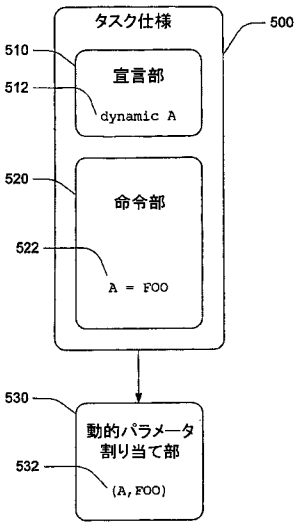
【図４Ｂ】



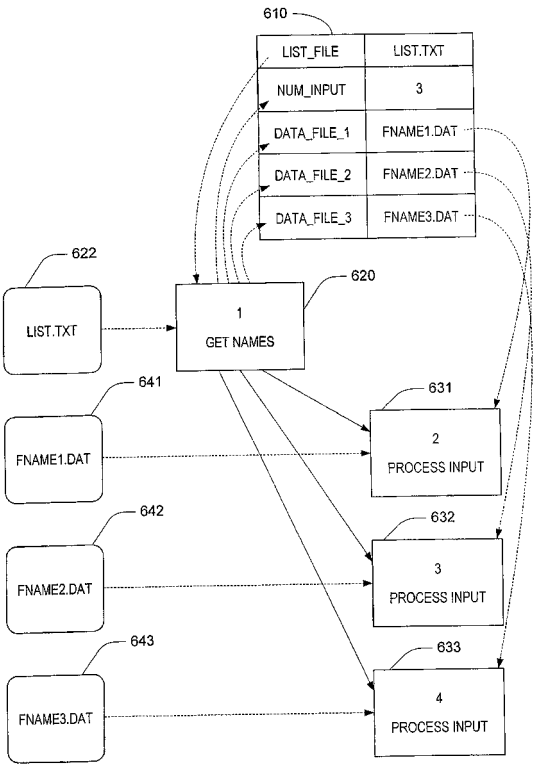
【図４Ｃ】



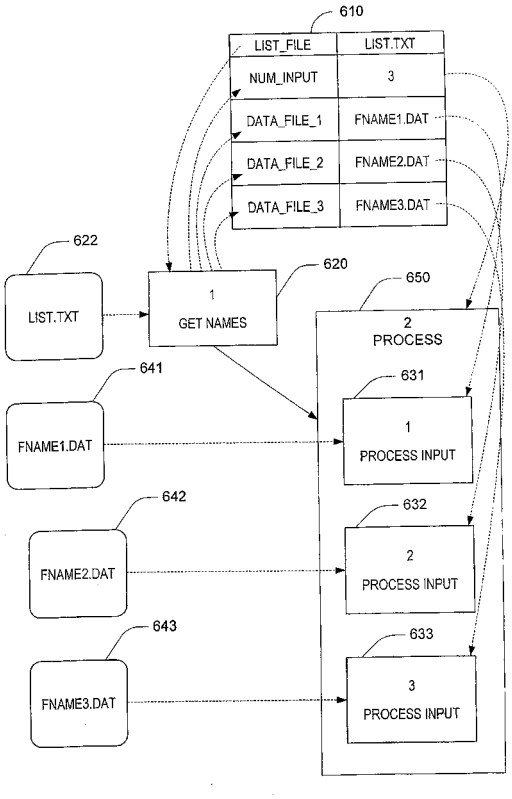
【図５】



【図６Ａ】



【図６Ｂ】


610
LIST_FILE
LIST.TXT
NUM_INPUT
3
DATA_FILE_1
FNAME1.DAT
DATA_FILE_2
FNAME2.DAT
DATA_FILE_3
FNAME3.DAT
622
LIST.TXT
1
GET NAMES
620
650
2
PROCESS
641
FNAME1.DAT
631
1
PROCESS INPUT
642
FNAME2.DAT
632
2
PROCESS INPUT
643
FNAME3.DAT
633
3
PROCESS INPUT

フロントページの続き

審査官　川崎　優

(56)参考文献　大塚ほか，タスク並列スクリプト言語ＭｅｇａＳｃｒｉｐｔによるタスク動作モデル記述，情報処理学会研究報告，日本，社団法人情報処理学会　Information Processing Society of Japan，２００３年　８月，第2003巻，第83号，P.113-118
西里ほか，タスク並列スクリプト言語ＭｅｇａＳｃｒｉｐｔのランタイムシステムの設計と実装，情報処理学会研究報告，日本，社団法人情報処理学会　Information Processing Society of Japan，２００３年　８月，第2003巻，第83号，P.119-124

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　9/44,46-54