(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5886712号
(P5886712)

(45)発行日　平成28年3月16日(2016.3.16)　(24)登録日　平成28年2月19日(2016.2.19)

| (51)Int.Cl. | | F I | | |
|---|---|---|---|---|
| G06F 11/34 | (2006.01) | G06F | 11/34 | S |
| G06F 11/32 | (2006.01) | G06F | 11/32 | B |
| G06F 11/28 | (2006.01) | G06F | 11/28 | 310A |

請求項の数 19　外国語出願　(全 53 頁)

(21)出願番号　特願2012-180251 (P2012-180251)
(22)出願日　平成24年8月15日 (2012.8.15)
(65)公開番号　特開2013-50950 (P2013-50950A)
(43)公開日　平成25年3月14日 (2013.3.14)
審査請求日　平成27年8月14日 (2015.8.14)
(31)優先権主張番号　13/211,143
(32)優先日　平成23年8月16日 (2011.8.16)
(33)優先権主張国　米国 (US)

早期審査対象出願

(73)特許権者　512212896
シーエー，インク
ＣＡ，Ｉｎｃ．
アメリカ合衆国　１１７４９　ニューヨーク州　アイランディア　ワン　シーエープラザ
Ｏｎｅ　ＣＡ　Ｐｌａｚａ　Ｉｓｌａｎｄｉａ，　Ｎｅｗ　Ｙｏｒｋ　１１７４９
Ｕ．Ｓ．Ａ．
(74)代理人　110000110
特許業務法人快友国際特許事務所
(72)発明者　ガグリアルディ，マルコ
アメリカ合衆国　１１７４９　ニューヨーク州　アイランディア　ワン　シーエープラザ　シーエー，インク内
最終頁に続く

(54)【発明の名称】分散環境におけるトランザクション別に区別されたメトリックの効率的収集

(57)【特許請求の範囲】
【請求項１】
コンピュータに実装される方法であって、
アプリケーションに対するツリーデータ構造にアクセスすること；
前記ツリーデータ構造は、第１ブランチと第２ブランチを含む複数のブランチを有しており、
前記第１ブランチは、前記アプリケーションの第１トランザクションを表すものであって、前記第１トランザクションの複数のコンポーネントの呼び出しの開始と終了を表す複数のノードを含んでおり、
前記第１トランザクションの前記複数のコンポーネントは、第１コンポーネントを含んでおり、
前記第１ブランチの前記複数のノードは、前記第１トランザクションにおける前記第１コンポーネントの呼び出しの開始または終了を表すノードであって第１収集部にリンクされている第１ノードを含んでおり、
前記第２ブランチは、前記アプリケーションの第２トランザクションを表すものであって、前記第２トランザクションの複数のコンポーネントの呼び出しの開始と終了を表す複数のノードを含んでおり、
前記第２トランザクションの前記複数のコンポーネントは、前記第１コンポーネントを含んでおり、
前記第２ブランチの前記複数のノードは、前記第２トランザクションにおける前記第

１コンポーネントの呼び出しの開始または終了を表すノードであって第２収集部にリンクされている第２ノードを含んでおり、
前記第１トランザクション内の前記第１コンポーネントの呼び出しの開始と終了を検出することを含んでおり、アプリケーションの呼び出された計測された複数のコンポーネントのシーケンスを検出するためにアプリケーションをトレースすること；
前記第１トランザクションにおける前記第１コンポーネントの前記呼び出しの開始と終了の検出結果に基づいて、前記第１トランザクションにおける前記第１コンポーネントのメトリックを収集すること；
前記アプリケーションの呼び出された計測された複数のコンポーネントのシーケンスが前記第１ブランチの前記複数のノードと一致するか否かを判定すること；
前記アプリケーションの呼び出された計測された複数のコンポーネントのシーケンスが前記第１ブランチの前記複数のノードと一致するか否かの判定に応答して、前記第１収集部の識別とともに、前記第１トランザクションにおける前記第１コンポーネントのメトリックをマネージャに報告すること；
を有する。
【請求項２】
前記メトリックは前記第１ブランチの識別子とともにマネージャに報告され、これにより前記メトリックが前記第１トランザクションとリンクされる、請求項１の方法。
【請求項３】
前記アプリケーションの呼び出された計測された複数のコンポーネントのシーケンスが前記第１ブランチの前記複数のノードと一致するか否かを判定することは、前記第１トランザクションにおける前記第１コンポーネントの呼び出しが、前記第１ブランチに関連付けられている特定のエントリポイントを介して行われたか否かを判定することを含む、請求項１の方法。
【請求項４】
前記第１収集部は、前記トレースすること、前記収集すること、前記判定すること、及び、前記報告すること、を実行するエージェントの一部である、請求項１の方法。
【請求項５】
前記メトリックは、前記第１トランザクションにおける前記第１コンポーネントのエラー回数を含む、請求項１の方法。
【請求項６】
前記メトリックは、前記第１トランザクションにおける前記第１コンポーネントの呼び出し回数を含む、請求項１の方法。
【請求項７】
前記メトリックは、前記第１トランザクションにおける前記第１コンポーネントの呼び出しの平均応答時間を含む、請求項１の方法。
【請求項８】
前記メトリックは、前記第１トランザクションと前記第２トランザクションにおける前記第１コンポーネントの同時呼び出しの回数を提供する、請求項１の方法。
【請求項９】
前記アプリケーションの呼び出された計測された複数のコンポーネントの第２シーケンスを検出するために前記アプリケーションをさらにトレースすること；
前記さらにトレースすることは、前記第２トランザクションにおける前記第１コンポーネントの呼び出しの開始と終了を検出することを含み；
前記第２トランザクションにおける前記第１コンポーネントの前記呼び出しの開始と終了の検出結果に基づいて、前記第２トランザクションにおける前記第１コンポーネントのメトリックを収集すること；
前記アプリケーションの呼び出された計測された複数のコンポーネントの第２シーケンスが前記第２ブランチの前記複数のノードと一致するか否かを判定すること；
前記アプリケーションの呼び出された計測された複数のコンポーネントの第２シーケン

スが前記第２ブランチの前記複数のノードと一致するか否かの判定に応答して、前記第２収集部の識別子とともに、前記第２トランザクションにおける前記第１コンポーネントのメトリックをマネージャに報告すること；
を有する、請求項１の方法。
【請求項１０】
前記第１収集部のコンテキストは、前記第１トランザクションにリンクされており、
前記第２収集部のコンテキストは、前記第２トランザクションにリンクされている、
請求項９の方法。
【請求項１１】
前記第１ブランチの前記第１ノードは前記第２収集部にリンクされておらず、それによって前記第１トランザクションにおける前記コンポーネントに対するメトリックは前記第２収集部のコンテキストにリンクされておらず、
前記第２ブランチの前記第２ノードは前記第１収集部にリンクされておらず、それによって前記第２トランザクションにおける前記コンポーネントに対するメトリックは前記第１収集部のコンテキストにリンクされていない、
請求項１の方法。
【請求項１２】
前記第１ブランチの前記第１ノードは、前記第１トランザクションにおける前記第１コンポーネントの２回目の呼び出しの開始または終了を表す、請求項１の方法。
【請求項１３】
前記報告に基づいて、複数の辺によって接続された頂点を有する有効グラフを含むユーザインタフェースを提供すること；前記複数の辺は、前記第１トランザクションを表す第１辺部分と、前記第２トランザクションを表す第２辺部分を含む一つの辺を含んでおり、
前記第１辺部分は、前記第２辺部分から、視覚的に区別される、
請求項１の方法。
【請求項１４】
前記有効グラフは、前記第１ブランチの識別子に基づいて前記第１辺部分が描かれており、前記第２ブランチの識別子に基づいて前記第２辺部分が描かれている、請求項１３の方法。
【請求項１５】
前記第１辺部分は、前記第１トランザクションの前記メトリックによって修飾されており、前記第２辺部分は、前記第２トランザクションの前記メトリックによって修飾されている、請求項１３の方法。
【請求項１６】
前記第１辺部分は、前記第１トランザクションに起因する前記第１コンポーネントの呼び出し回数に対する前記第２トランザクションに起因する前記第１コンポーネントの呼び出し回数を示すように、前記第２辺部分から視覚的に区別される、請求項１３の方法。
【請求項１７】
前記第１辺部分は、太さによって前記第２辺部分から視覚的に区別される、請求項１６の方法。
【請求項１８】
前記第１ブランチは、前記第１トランザクションにおける前記第１コンポーネントの繰り返し呼び出しを表すノードを含んでいる、請求項１の方法。
【請求項１９】
コンピュータに、
アプリケーションの夫々のトランザクションをトレースすること；
前記トレースすることは、第１トランザクションにおける前記アプリケーションの呼び出された計測された複数のコンポーネントのシーケンス、及び、第２トランザクションにおける呼び出された計測された複数のコンポーネントのシーケンスを検出し、前記第１トランザクションにおける呼び出された計測された複数のコンポーネントの前記シーケンス

は、コンポーネントの第１インスタンスを含み、前記第２トランザクションにおける呼び出された計測された複数のコンポーネントの前記シーケンスは、前記コンポーネントの第２インスタンスを含んでおり、

前記コンポーネントの前記第１インスタンスが前記第１トランザクションのコンテキスト内で検出されたか否かを判定すること；前記「前記コンポーネントの前記第１インスタンスが前記第１トランザクションのコンテキスト内で検出されたか否かを判定すること」は、前記第１トランザクションにおける呼び出されたコンポーネントのシーケンスと、前記第１トランザクションを表すデータ構造の中の複数のノードのシーケンスとが一致するか否かを判定することを含み、前記「前記第１トランザクションを表すデータ構造の中の複数のノードのシーケンス」は、複数のコンポーネントの夫々に対して、呼び出しの開始と終了を示す複数のノードを含んでおり、

前記コンポーネントの前記第２インスタンスが前記第２トランザクションのコンテキスト内で検出されたか否かを判定すること；前記「前記コンポーネントの前記第２インスタンスが前記第２トランザクションのコンテキスト内で検出されたか否かを判定すること」は、前記第２トランザクションにおける呼び出されたコンポーネントのシーケンスと、前記第２トランザクションを表すデータ構造の中の複数のノードのシーケンスとが一致するか否かを判定することを含み、前記「前記第２トランザクションを表すデータ構造の中の複数のノードのシーケンス」は、複数のコンポーネントの夫々に対して、呼び出しの開始と終了を示す複数のノードを含んでおり、

第１収集部を使って前記第１トランザクションにおける前記コンポーネントの第１メトリックを収集すること；

第２収集部を使って前記第２トランザクションにおける前記コンポーネントの第２メトリックを収集すること；

前記コンポーネントの前記第１インスタンスが前記第１トランザクションのコンテキスト内で検出されたことに応答して、前記第１トランザクションのコンテキストの中で前記第１メトリックをマネージャに報告すること；

前記コンポーネントの前記第２インスタンスが前記第２トランザクションのコンテキスト内で検出されたことに応答して、前記第２トランザクションのコンテキストの中で前記第２メトリックをマネージャに報告すること；

を備えており、

前記第１トランザクションの前記コンテキストは、前記第２トランザクションの前記コンテキストとは異なっており、

「前記第１トランザクションのコンテキストの中で前記第１メトリックをマネージャに報告すること」は、前記第１収集部の識別子を報告することを含んでおり、

「前記第２トランザクションのコンテキストの中で前記第２メトリックをマネージャに報告すること」は、前記第２収集部の識別子を報告すること、

を実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

コンピューティング環境においてソフトウェアを監視する技術を提供する。

【背景技術】

【０００２】

インターネットの存在が増長するにつれて、イントラネット、エクストラネットなどの他のコンピュータネットワークと同様に、電子商取引、教育、他の分野において多くの新しいアプリケーションをもたらした。組織は、ビジネスやその他目的を行うのにそのようなアプリケーションへの依存を深めており、期待通りの成果を確実に得るために相当の資源を投入している。このような状況のもと、様々なアプリケーションの管理技術が開発されている。

【０００３】

１つのアプローチとして、アプリケーションで呼び出される個々のソフトウェアコンポーネントに関するアプリケーションの実行データを集めることにより、アプリケーションの基礎構造を監視することが挙げられる。このアプローチでは、監視されるシステムに元来、存在するエージェントを用いることができる。例えば、ソフトウェアによる計測を用いることによって、呼び出される個々のコンポーネントを特定するためや、個々のコンポーネントの実行時間のような実行データを得るために、スレッド又はプロセスがトレースされ得る。コンピュータプログラムが実行するステップの詳細なレコード、又はトレースを得ることをトレーシングという。トレースのタイプの１つにスタックトレースがある。トレースはデバッグにおける補助の役目を果たし得る。

【０００４】

典型的には、静的及び動的なデータを含むトランザクショントレースデータは、エージェントからマネージャへと伝えられる。しかしながら、既存のアプローチは効率的ではなく、相当なオーバヘッドコストが嵩む。

【発明の概要】

【０００５】

本発明は、上記課題及び他の課題を解決するモニタリングソフトウエアの技術を提供する。

【０００６】

一実施形態において、少なくとも１つのアプリケーションを監視するための方法を実行するため少なくとも１つのプロセッサをプログラミングするために、コンピュータで読み取り可能なソフトウェアを有するプロセッサで読み取り可能な１以上の有体の記憶装置が具現化されて提供される。その方法は：（ａ）複数のブランチを有するツリーデータ構造を提供することを含み、その複数のブランチのそれぞれは、少なくとも１つのアプリケーションの複数のトランザクションのそれぞれを表すとともに、各トランザクションの複数のコンポーネントの開始と終了ポイントを表すノード群を有し、その複数のブランチのうちの１つのブランチは、複数のトランザクションのうちの１つのトランザクションを表し、第１収集部にリンクされて、１つのコンポーネントに対する少なくとも１つのノードを有し、その１つのブランチ内の１つのコンポーネントに対するその少なくとも１つのノードは、その１つのトランザクション内のその１つのコンポーネントの１つの呼び出しの開始及び終了の少なくとも１つを表す。

【０００７】

その方法は、さらに、（ｂ）呼び出された１つのシーケンスを検知するために少なくとも１つのアプリケーションをトレースすること、そのトレースは、１つのトランザクション内の１つのコンポーネントの１つの呼び出しの開始と終了の少なくとも１つを検出することを含む、（ｃ）トレースすることが１つのトランザクション内で、１つのコンポーネントの１つの呼び出しの開始及び終了の少なくとも１つを検出するとき、第１収集部のコンテキスト内の１つのコンポーネントの少なくとも１つのメトリックを収集するために第１収集部を使用すること、（ｄ）呼び出されたコンポーネントの、１つのシーケンスが１つのブランチと一致していることを判定するために、トレースの結果をツリーデータ構造と比較すること、及び、（ｅ）呼び出されたコンポーネントの、１つのシーケンスが１つのブランチと一致していることを判定することに応答して、マネージャに、第１収集部のコンテキスト内の１つのコンポーネントのための少なくとも１つのメトリックを報告すること、を含む。

【０００８】

別の実施形態において、少なくとも１つのアプリケーションを管理するための方法を実行するため少なくとも１つのプロセッサをプログラミングするために、コンピュータで読み取り可能なソフトウェアを有するプロセッサで読み取り可能な１以上の有体の記憶装置が具現化されて提供される。その方法は、１以上の報告が、複数のトランザクションの少なくとも１つのトランザクション及び関連付けられた第１収集部の識別のための、少なくとも１つのメトリック、並びに複数のトランザクションの少なくとも別のトランザクショ

ン及び第２収集部の関連付けられた識別のための、少なくとも１つのメトリックを含む場合、少なくとも１つのアプリケーションにおける複すのトランザクションをトレースする少なくとも１つのエージェントから１以上の報告を受信することを含み、第１収集部は、少なくとも１つのトランザクション内のコンポーネント呼び出しに応答した少なくとも１つのトランザクションのための少なくとも１つのメトリックを収集するために使用され、及び第２収集部は、少なくとも別のトランザクション内のコンポーネント呼び出しに応答した少なくとも別のトランザクションのための少なくとも１つのメトリックを収集するために使用される。

【０００９】

この方法は、１以上の報告に基づくユーザインタフェースを提供することをさらに含み、そのユーザインタフェースは、少なくとも１つのトランザクションを表す少なくとも第１辺部分及び少なくとも別のトランザクションを表す第２辺部分を有する辺を含む辺によって接続される頂点を有する有効グラフを有し、その第１辺部分は第２辺部分から視覚的に区別される。

【００１０】

別の実施形態において、少なくとも１つのアプリケーションを監視するための方法を実行するため少なくとも１つのプロセッサをプログラミングするために、コンピュータで読み取り可能なソフトウェアを有するプロセッサで読み取り可能な１以上の有体の記憶装置が具現化されて提供される。その方法は：（ａ）少なくとも１つのアプリケーションの複数のトランザクションをトレースすること、トレースすることは、少なくとも１つのトランザクション内のコンポーネントの１つのインスタンス及び少なくとも別のトランザクション内のコンポーネントの別のインスタンスを検出すること、（ｂ）コンポーネントの１つのインスタンスは、少なくとも１つのトランザクションのコンテキスト内で検出され、それとは別に、コンポーネントの別のインスタンスは、少なくとも別のトランザクションのコンテキスト内で検出されたこと、を判定すること、（ｃ）コンポーネントの１つのインスタンスが少なくとも１つのトランザクションのコンテキスト内で検出されたことを判定することに応答して、マネージャに、少なくとも１つのトランザクションのコンテキスト内のコンポーネントの１つのインスタンスの少なくとも１つのメトリックを報告すること、及び（ｄ）１つのコンポーネントの別のインスタンスが少なくとも別のトランザクションのコンテキスト内で検出されたことを判定することに応答して、少なくとも１つのトランザクションのコンテキストが少なくとも別のトランザクションのコンテキストから区別される場合、マネージャに、少なくとも別のトランザクションのコンテキスト内のコンポーネントの別のインスタンスの少なくとも１つのメトリックを報告すること、を含む。

【００１１】

対応するプロセッサに実装された方法は、プロセッサで読み取り可能な１以上の有体の記憶装置のコンピュータに実装された手順を実行することを提供する。

【００１２】

対応するシステムは、プロセッサで読み取り可能な１以上の有体の記憶装置、及び、プロセッサで読み取り可能な１以上の有体の記憶装置を読み取るための１以上のプロセッサ、を含むものを提供する。

【００１３】

対応するコンピュータ又はプロセッサ読み取り可能な有体の記憶装置は、実行時に、本明細書で提供される方法手順を実行する命令がエンコードされたプロセッサで読み取り可能なものが提供される。

【図面の簡単な説明】

【００１４】

【図１Ａ】アプリケーションの複数のインスタンスが異なるサーバ上で実行し、サーバのエージェントがマネージャに報告するシステムの例を示す。

【図１Ｂ】アプリケーションの複数のインスタンスが異なるサーバ上で実行し、サーバのエージェントが中間収集手段（intermediate collectors）を経由してマネージャに報告

するシステムの例を示す。
【図２Ａ】トランザクショントレースを開始するためのプロセス［処理］の一実施形態を説明するフローチャートである。
【図２Ｂ】トランザクショントレースを終了するためのプロセス［処理］の一実施形態を説明するフローチャートである。
【図３】図１Ａ又は１Ｂのネットワークのコンピューティングデバイスを示す。
【図４】１以上のアプリケーションの操作を説明するのに使用される階層を示す。
【図５Ａ】図４の報告及び見積の業務トランザクション内で呼び出されるコンポーネントのシーケンスの例における依存関係を示す。
【図５Ｂ】図５Ａの依存関係の代わりとなる表示で、よりコンパクトなものを示す。
【図６】図６（Ａ）～図６（Ｃ）は、図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｄ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｅ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｆ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｇ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｈ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図６Ｉ】図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。
【図７Ａ１】図６Ａから６Ｉのトランザクショントレースに基づき提供されるエージェント１及びエージェント２のツリーデータ構造の例を示す。
【図７Ａ２】図７Ａ１のツリーデータ構造の代替的及び等価的な表示を示す。
【図７Ｂ】新しいブランチの形式をした図７Ａ１のエージェント１のツリーデータ構造への更新を示す。
【図７Ｃ１】図７Ａ１の、エージェント１及びエージェント２のツリーデータ構造を結合した、マネージャのツリーデータ構造を示す。
【図７Ｃ２】図７Ａ１のエージェント１のツリーデータ構造における最終ノードと図７Ｃ１のマネージャのツリーデータ構造の最終ノードの間の対応を示す。
【図７Ｄ】図７Ｂにおけるエージェント１のツリーデータ構造への更新と一致する、新しいブランチの形式をした図７Ｃ１のマネージャのツリーデータ構造への更新を示す。
【図８Ａ１】図７Ａ１のツリーデータ構造におけるサブシステム１のためのブランチ及びコンポーネント呼び出しのレコードを示す。
【図８Ａ２】図７Ａ１のツリーデータ構造におけるサブシステム２のためのブランチ及びコンポーネント呼び出しのレコードを示す。
【図８Ａ３】図７Ａ１のツリーデータ構造におけるマネージャのためのブランチ及びコンポーネント呼び出しのレコードを示す。
【図８Ｂ１】図７Ａ１のツリーデータ構造におけるサブシステム１の異なるノード／コンポーネントのための静的データへの参照のレコードを示す。
【図８Ｂ２】図７Ａ１のツリーデータ構造におけるサブシステム２の異なるノード／コンポーネントのための静的データへの参照のレコードを示す。
【図８Ｂ３】図７Ｂにおけるエージェント１－新しい－ブランチのための図８Ｂ１のレコードへの更新を示す。
【図８Ｂ４】図７Ｃ１のツリーデータ構造におけるマネージャの異なるノード／コンポーネントのための静的データへの参照のレコードを示す。
【図８Ｂ５】図７Ｄにおけるマネージャ－新しい－ブランチ７のための図８Ｂ４のレコー

ドへの更新を示す。
【図８Ｃ】図７Ａ１のツリーデータ構造のサブシステム１の異なるノード／コンポーネントのためのトレース詳細からの動的データのレコードを示す。
【図８Ｄ】様々なコンポーネントに関連付けられた静的データのレコードを示す。
【図９】少なくとも１つのアプリケーションのために図７Ａ１にあるようなエージェントがツリーデータ構造を保守するプロセス［処理］の例を示す。
【図１０Ａ】エージェントから受信するように、例えば図７Ａ１にあるように、動的データの報告及びツリーデータ構造のブランチの識別子に基づいて、マネージャがユーザインタフェースを提供するプロセス［処理］の例を示す。
【図１０Ｂ】１以上のエージェントから受信した更新に基づいて、図７Ａ１から７Ｃ１にあるように、マネージャがツリーデータ構造を更新するプロセス［処理］の例を示す。
【図１１】図１１（Ａ）、（Ｂ）は、静的及び動的データを用いて注記を加えた、図６Ａのトランザクショントレースを示す。
【図１２Ａ】それぞれのトランザクションのそれぞれのブランチにおける１つのコンポーネントのためにノードにリンクされた収集部を有する図７Ａ１のツリーデータ構造を示す。
【図１２Ｂ】異なるそれぞれのトランザクションの異なるそれぞれのブランチにおける同じコンポーネントの複数の発生のためにノードにリンクされた収集部を有する図７Ａ１のツリーデータ構造を示す。
【図１２Ｃ】それぞれのトランザクションの、同じそれぞれのブランチにおける同じコンポーネントの複数の発生のためにノードにリンクされた収集部を有する図７Ａ１のツリーデータ構造を示す。
【図１３Ａ】図１２Ａのツリーデータ構造のための収集部１２００及び１２０２への参照のレコードを示す。
【図１３Ｂ】図１２Ｂのツリーデータ構造のための収集部１２０４への参照のレコードを示す。
【図１３Ｃ】図１２Ｃのツリーデータ構造のための収集部１２０６への参照のレコードを示す。
【図１４Ａ】図１３Ａのツリーデータ構造に基づくユーザインタフェースの例を示す。
【図１４Ｂ】図１３Ｂのツリーデータ構造に基づくユーザインタフェースの例を示す。
【図１４Ｃ】図１３Ｃのツリーデータ構造に基づくユーザインタフェースの例を示す。
【図１５Ａ】図５Ｂ及び図１４Ａに一致するユーザインタフェースの例を示す。
【図１５Ｂ】図１５Ａに代わるユーザインタフェースの例を示す。
【図１５Ｃ】ユーザインタフェースの別の例を示す。
【図１６Ａ】少なくとも１つのアプリケーションのために、エージェントがトランザクション別のメトリックを取得するプロセス［処理］の例を示す。
【図１６Ｂ】図１６Ａのプロセス［処理］に対応して、エージェントからのトランザクション別のメトリックの報告に基づいてマネージャがユーザインタフェースを提供するプロセス［処理］の例を示す。
【発明を実施するための形態】
【００１５】
本発明は、エージェントからマネージャへ、静的及び動的なデータを含むトランザクショントレースデータを効率的に伝える監視ソフトに関する技術を提供する。効率を改善してオーバーヘッドコストを削減するため、エージェント及びマネージャによって保守管理されるツリーデータ構造は、ソフトウェアの呼び出されたコンポーネントのシーケンスを記述する。各コンポーネントの開始と終了は、ツリーデータ構造のブランチ内のノードによって表される。トランザクションを識別するため、エージェントは、例えばブランチの最終ノードの識別子など、ブランチ固有の識別子を伝える。これにより、呼び出されたコンポーネントのシーケンスがエージェントからマネージャへより効率的に報告される。さらに、静的データは、１以上のノード又はコンポーネントにインデックス付けされ、及び

、エージェント及び／又はマネージャによってアクセスされる。静的データは、通常、ソフトウェアに付与されたバージョンのために固定され、及び固定された、又は実行依存しないデータとしても考えられる。静的データは、例えば、コンポーネントに関連付けられたクラス名又はメソッド名、メソッド呼び出しのシーケンス、トレースされたクラスファイルが展開されたアーカイブファイル（ＪＡＶＡ（登録商標）アーカイブファイル又はＪＡＲファイル又はウェブアーカイブファイル又は．ＷＡＲファイルなど）の名称、テキスト文字列、コンポーネントの型（例えば、サーブレット、ＥＪＢ）、サーブレット又はソケットのためのポート番号、ＵＲＬ、ホスト名、及びローカル又はリモートインタフェース名を含んでよい。これらは、ソフトウェアのトレースから入手可能である全種類の情報である。静的データのインデックス付けは、エージェントからマネージャへ繰り返し伝える必要性、及びエージェント及び／又はマネージャが静的データを繰り返し取得する必要性を回避する。

【００１６】

動的データはトレースから取得される。動的データは、コンポーネントの開始及び終了時刻、及びパラメータ値が監視されたメソッドに渡された、又は監視されたメソッドによって渡されたパラメータ値などの他の動的データを含む。動的データは、また１以上のノード又はコンポーネントにインデックス付けられる。動的データは、コンポーネントの開始及び／又は終了ノードにインデックス付けられる。このインデックス付けを介して、動的データは、エージェントからマネージャに効率的に報告される。

【００１７】

トランザクションがトレースされると、エージェントは、ツリーデータ構造内のブランチに一致するものを特定する。もし一致するものがなければ、エージェントは、ツリーデータ構造を更新し、マネージャに更新を報告する。その結果、エージェント及びマネージャは、ツリーデータ構造の同期したバージョンを保守し得る。さらに、マネージャは、ツリーデータ構造の異なる部分が異なるエージェントに関連付けられている場合、複数のエージェントからの報告に基づいてツリーデータ構造を保守管理する。エージェントが同じソフトウェアの異なるインスタンスを監視するとき、マネージャは、また１つのエージェントから別のエージェントへ受信される更新情報を渡す。このように、エージェントのツリーデータ構造が同期されるように新しいトランザクションがエージェント間で迅速に伝達する。

【００１８】

図１Ａは、アプリケーションの複数のインスタンスが異なるサーバ上で実行し、サーバのエージェントがマネージャに報告するシステム１００の例を示す。例示の管理対象のコンピューティングデバイス１０３、１０５及び１０９は、アプリケーションサーバ又は必要な機能を実現するためのコードを実行するプロセッサを有するコンピューティングデバイスの他の種類を含む。管理対象のコンピューティングデバイスは、互いに離れて位置することができ、又は同じ場所に位置することもできる。管理対象のコンピューティングデバイスは、この例ではネットワーク１０７を介してマネージャコンピュータ１１１と通信する。マネージャコンピュータ１１１は、管理対象のコンピューティングデバイスにローカル、又は管理対象のコンピューティングデバイスからリモートである。管理対象のコンピューティングデバイス１０３及び１０５は、またネットワーク１０２を介して、例示のウェブブラウザ１０１などのクライアントのコンピューティングデバイスと通信する。ウェブブラウザ１０１は、例えば、インターネットサービスプロバイダを介してネットワーク１０２にアクセスする。さらに、一例として、管理対象のコンピューティングデバイス１０３は、ウェブブラウザからのリクエストに応答するために必要な情報を取得するため、例えばウェブサービス呼び出し側のＥＪＢクライアントを経由して管理対象のコンピューティングデバイス１０９を呼び出す。管理対象のコンピューティングデバイス１０３は、ウェブブラウザからのリクエストに応答するために必要な情報を取得するため、例えばメインフレーム、データベース又は他の未計測のコンピューティングデバイスなどのバックエンドシステム１０８もまた呼び出す。性能メトリックの全範囲は、計測の使用により

管理対象のコンピューティングデバイスから取得される一方で、限定された情報は、管理対象のコンピューティングデバイスから呼び出して使用されるメソッドから、未計測のサブシステムに着目して得られる。管理対象のコンピューティングデバイスは、フロントエンドのサブシステムであると考えられる。ネットワーク１０２及び１０７は、同じ、重複した、又は、異なるものであると考えられ、例えば、インターネット、他の広域ネットワーク及び／又はローカルエリアネットワークを含む。点線は、通信経路を示す。

【００１９】

例えば、ウェブベースの電子商取引アプリケーションなどの企業のアプリケーションを実行している会社は、負荷分散のために１つの場所で複数のアプリケーションサーバを使用する。例えば、ウェブブラウザ１０１からのようなユーザからのリクエストは、ネットワーク１０２を介して受信され、任意の管理対象のコンピューティングデバイス１０３及び１０５に送られる。管理対象のコンピューティングデバイス１０３、１０５及び１０９上で実行するエージェントソフトウェアは、エージェントＡ１（１０４）、エージェントＡ２（１０６）及びエージェントＡ３（１１０）によってそれぞれ表され、それぞれの管理対象のコンピューティングデバイス上で実行されている、アプリケーション、ミドルウェア又はその他のソフトウェアから、情報を収集する。例えば、そのような情報は、計測を用いることによって得ることができ、その一例はバイトコードの計測である。しかしながら、集められたデータは他の方法でも得ることができる。エージェントは、監視するコンピューティングデバイスに元来存在してデータの取得ポイントを提供する。エージェントは、マネージャ１２４と通信してデータをまとめ最適化する。１つの実施形態では、管理対象のコンピューティングデバイス１０３、１０５において同じアプリケーションの異なる複数のインスタンスが実行され、管理対象のコンピューティングデバイス１０９において別のアプリケーションが実行される。

【００２０】

マネージャ１１１は、エージェントから受信したデータに基づく情報を表示するため、例えばモニタなどのユーザインタフェース１１３と通信するワークステーションのような分離したコンピューティングデバイス上に提供され得る。マネージャは、またエージェントから受信したデータを格納するためデータベース１１２にアクセスする。例えば、大きな組織は、セントラルネットワークオペレーションセンタを運用する。そこでは、１以上のマネージャが、地理的に異なる場所に分散している複数のエージェントからデータを取得する。説明すると、ウェブベースの電子商取引企業では、顧客の注文を受ける地理的に異なる場所にあるサーバからエージェントのデータを取得することがある。支払いを処理するサーバ、倉庫で在庫を調べたり、受注を受けたりするサーバなどである。マネージャ１１１及びユーザインタフェースディスプレィ１１３は、企業の本社の場所で提供され得る。必ずしも、ウェブベース又は小売、若しくはその他の販売に関する必要はなく、他のアプリケーションにおいて同様にシステムを管理するためにエージェントとマネージャを利用する。例えば、銀行では、小切手の処理やクレジットの口座用にアプリケーションを使用することがある。また、上述した複数コンピュータのデバイスアレンジに加えて、１以上のエージェントによって単一のコンピュータデバイスが同様に監視されることがある。

【００２１】

監視を実行するソフトウェアを計測するのに、様々なアプローチが知られている。例えば、最初に述べたように、トレーシングはソフトウェアの実行を追跡するために用いることができる。トレーシングの例が、”Transaction Tracer”と題する米国特許第７，８７０，４３１号（２０１１年１月１１日発行）に記載されている。その内容は参照により本明細書に組み込まれる。その中で述べられているアプローチにおいては、監視すべきアプリケーションのオブジェクトコード又はバイトコードが計測され、例えばプローブにより変更される。アプリケーションのジョブ又は他のロジックを変更することなくアプリケーションについての特定の情報をプローブが測定する。一旦、プローブがアプリケーションのバイトコードにインストールされると、管理されたアプリケーションと称され、また、

そのアプリケーションが実行されるコンピューティングデバイスは、管理されたコンピューティングデバイスと称される。エージェントソフトウェアは、プローブからの情報を受信し、その情報を、例えばマネージャ１１１において、別のプロセスに伝達することがある。また、情報が異常状況を示すか否かを判定するなど、情報をローカルで処理する。エージェントは、このようにプローブから受信した情報を収集し要約する。指示ファイルによって定義されるように、プローブは、情報を収集する。例えば、プローブからの情報は、トランザクション又は他の実行フローの開始や停止の回数、又はトランザクション／実行フロー内の個々のコンポーネントの開始や停止の回数を示す場合がある。この情報は、それが範囲内にあるかどうかを判定するために予め決められた基準と比較される。もし情報が範囲内にない場合には、エージェントは適切なトラブルシューティングが実行できるようにこの事実をマネージャに報告する。エージェントは、関連付けられているローカルの管理対象のコンピューティングデバイス上でソフトウェアが実行中であることを通常認識している。

【００２２】

プローブは、ＣＯＲＢＡメソッドタイマ、リモートメソッドインボケーション（ＲＭＩ）メソッドタイマ、スレッドカウンタ、ネットワークバンド幅、ＪＤＢＣ更新及びクエリタイマ、サーブレットタイマ、Java（登録商標）サーバページズ（ＪＳＰ）タイマ、システムログ、ファイルシステム入出力バンド幅メータ、使用可能及び使用済メモリ、並びにＥＢＪ（エンタープライズJava（登録商標）ビーンズ）タイマを含むメトリックの標準セットを報告する。メトリックは、特定のアプリケーションのアクティビティの計測値である。

【００２３】

エージェントは、アプリケーションによってアクセスされるリソースを識別するトランザクションに関する情報を報告する。１つのアプローチでは、トランザクションについて報告する場合における「呼び出された」という語はリソースを指す。このリソースは、消費者が親のコンポーネントであるところのリソース（又はサブリソース）である。例えば、トランザクションで呼び出される最初のコンポーネントがサーブレットＡであると仮定する。消費者のサーブレットＡ（下記参照）の下には、ＥＪＢと称されるサブリソースがある。消費者とリソースは、ツリーのような形でエージェントよって報告される。トランザクションのデータは、またツリーに従って格納される。例えば、もしサーブレット（例えばサーブレットＡ）が、ネットワークのソケット（例えば、ソケットＣ）の消費者であり、かつＥＪＢ（例えば、ＥＪＢ　Ｂ）の消費者でもあるとすれば、次にはＪＤＢＣ（例えば、ＪＤＢＣ　Ｄ）の消費者であり、ツリーは以下のように見える。

Servlet A （サーブレットＡ）
Data for Servlet A （サーブレットＡのデータ）
Called EJB B （呼び出されたＥＪＢ　Ｂ）
Data for EJB B （ＥＪＢ　Ｂのデータ）
Called JDBC D （呼び出されたＪＤＢＣ　Ｄ）
Data for JDBC D （ＪＤＢＣ　Ｄのデータ）
Called Socket C （呼び出されたソケットＣ）
Data for Socket C （ソケットＣのデータ）

【００２４】

一実施形態では、上記ツリーは、ブレイムスタックと称されるスタックにエージェントによって格納される。トランザクションが開始すると、トランザクションはスタックへプッシュされる。トランザクションが完了すると、トランザクションはスタックからポップされる。一実施形態では、スタック上の各トランザクションは、次に続く情報、トランザクションの型、トランザクションのためにシステムで使用される名称、パラメータのハッシュマップ又は辞書、トランザクションがスタックへプッシュされたときのタイムスタンプ及びサブエレメント、が格納されている。サブエレメントは、注目すべきトランザクション内から開始されている他のコンポーネント（例えば、メソッド、プロセス、プロシー

ジャ、関数、スレッド、命令セットなど）のためのブレイムスタックのエントリである。上記の例のようにツリーを使用すると、サーブレットＡのためのブレイムスタックのエントリは２つのサブエレメントを有する。第１サブエレメントは、ＥＪＢ　Ｂへのエントリで、第２サブエレメントは、ソケットスペースＣへのエントリである。サブエレメントは特定のトランザクションのためのエントリの一部であるにもかかわらず、サブエレメントはまた独自のブレイムスタックのエントリを有する。トランザクション／ブランチへのエントリポイントの例は、ＵＲＬである。上記のツリーに示されるように、ＥＪＢ　Ｂはサーブレットのサブエレメントであり、また独自のエントリを有する。トランザクションに対する一番上（最初）のエントリ（例えばサーブレットＡ）は、ルートコンポーネントと称される。スタック上の各エントリはオブジェクトである。

【００２５】

図１Ｂは、アプリケーションの複数のインスタンスが異なるサーバ上で実行し、サーバのエージェントが中間マネージャを経由してマネージャに報告するシステムの例を示す。この例では、付加的な管理対象のコンピューティングデバイス１１６及び１１８がエージェントＡ４　１１７及びエージェントＡ５　１１９と共にそれぞれ提供されている。さらに、中間、又は低レベルのマネージャコンピューティングデバイス１２０（マネージャＡ）及び１２１（マネージャＢ）は、エージェントＡ４及びエージェントＡ５からデータを受け取るものが、それぞれ提供される。中間マネージャは、次に、この場合、高レベルのマネージャがネットワーク１２２を介してデータをマネージャ１１１に報告する。ネットワーク１０２、１０７及び１２２は、同じ、重複又は異なるものであると考えられる。

【００２６】

図２Ａは、トランザクショントレースを開始するための処理の一実施形態を説明するフローチャートである。ステップは適切なエージェントにより実行される。ステップ１３０ではトランザクションを開始する。一実施形態では、プロセス［処理］がメソッド（例えば、“ loadTracer ”メソッドの呼び出し）の開始によってトリガされる。ステップ１３２において、エージェントは所望のパラメータ情報を取得する。一実施形態では、ユーザは、どのパラメータ情報が構成ファイル又はＵＩを介して取得されるかを設定することができる。取得されたパラメータは、ブレイムスタックへプッシュされるオブジェクトの一部であり、ハッシュマップ又は辞書に格納されている。他の実施形態では、パラメータの識別は、予め設定されている。格納されるパラメータには様々なものがある。一実施形態では、使用されるパラメータの実際の一覧表は、監視されるアプリケーションに依存している。以下の表は、取得され得るいくつかのパラメータの例を示す。

【００２７】

【表１】

| パラメータ | 表示 | 値 |
|---|---|---|
| UserID | サーブレット、ＪＳＰ | httpサーブレットのリクエストを呼び出すエンドユーザのユーザＩＤ |
| URL | サーブレット、ＪＳＰ | サーブレット又はＪＳＰを通過するＵＲＬで、クエリ文字列が含まれないもの |
| URL Query | サーブレット、ＪＳＰ | httpリクエストにおいてクエリ変数を特定するＵＲＬの部分（区切り記号「？」に続くテキスト） |
| Dynamic SQL | 動的ＪＤＢＣ文 | 一般化された形式における、又は現在の呼び出しからのすべての所定のパラメータを有する動的なＳＱＬ文 |
| Method | ブレイムドメソッドタイマ（サーブレット、ＪＳＰ文及びＪＤＢＣ文を除くすべて） | トレースされたメソッドの名称。トレースされたメソッドが、同じコンポーネント内の他のメソッドを直接呼び出した場合、「一番外側で」最初に検出されたメソッドだけが取り込まれる。 |
| Callable SQL | 呼び出し可能なＪＤＢＣ文 | 一般化された形式における、又は現在の呼び出しからのすべての所定のパラメータを有する呼び出し可能なＳＱＬ文 |
| Prepared SQL | 用意されたＪＤＢＣ文 | 一般化された形式における、又は現在の呼び出しからのすべての所定のパラメータを有する用意されたＳＱＬ文 |
| Object | すべての非静的メソッド | トレースされたコンポーネントのこのオブジェクトのtoString()で、いくつかの文字が上限で切り捨てられたもの |
| Class Name | すべて | トレースされたコンポーネントのクラスの完全修飾名称 |
| Param_*n* | *WithParams* カスタムトレーサを有するすべてのオブジェクト | コンポーネントのトレースされたメソッドに渡されたｎ番目のパラメータのtoString() |
| Primary Key | エンティティビーンズ | エンティティビーンズのプロパティキーのtoString()で、いくつかの文字が上限で切り捨てられたもの |

【００２８】

パラメータは、クエリ、クッキー、ＰＯＳＴ、ＵＲＬ及びセッションの型の名称／値の組を含む。

【００２９】

ステップ１３４では、システムは現在の時刻を示すタイムスタンプを取得する。ステップ１３６ではスタックエントリが作成される。ステップ１３８において、スタックエントリはブレイムスタックへプッシュされる。一実施形態では、タイムスタンプがステップ１３８の一部として付加される。トランザクションが開始されるときにプロセス［処理］が実行される。同様のプロセス［処理］が、トランザクションのサブコンポーネントが開始するときに実行される（例えば、ＥＪＢ　ＢはサーブレットＡのサブコンポーネントであ

る－上述したツリーを参照のこと）。

【００３０】

図２Ｂは、トランザクショントレースを終了するためのプロセス［処理］の一実施形態を説明するフローチャートである。トランザクションが終了するときにエージェントによりプロセス［処理］が実行される。ステップ１４０において、プロセス［処理］がトランザクション（例えばメソッド）の終了（例えば、“finishTrace”メソッドの呼び出し）によってトリガされる。ステップ１４２では、システムは現在の時刻を取得する。ステップ１４４では、スタックエントリが削除される。ステップ１４６において、トランザクションの実行時間は、ステップ１４２からのタイムスタンプをスタックエントリに格納されているタイムスタンプと比較することによって算出される。ステップ１４８では、トレースのためのフィルタが適用される。例えば、フィルタは１秒の閾値期間が入る。したがって、ステップ１４８は、ステップ１４６から算出された時間幅が１秒よりも大きいか否かを決定することを含む。閾値を超えない場合（ステップ１５０）、トランザクションのデータは破棄される。一実施形態では、スタックエントリの全体が破棄される。別の実施形態では、パラメータとタイムスタンプだけが破棄される。他の実施形態では、データの様々なサブセットが破棄される。いくつかの実施形態で、閾値の時間幅を超えていない場合には、エージェントにより、データは図１Ａ又は１Ｂのシステム内の他のコンポーネントに送信されない。時間幅が閾値を超える場合（ステップ１５０）、ステップ１６０においてエージェントがコンポーネントデータを組み立てる。コンポーネントデータは、報告されるトランザクションに関するデータである。一実施形態では、コンポーネントデータは、トランザクションの名称、トランザクションの型、トランザクションの開始時刻、トランザクションの時間幅、パラメータのハッシュマップ、及びサブエレメント（エレメントの帰納的なリスト）のすべてを含む。その他の情報もまたコンポーネントデータの一部である。ステップ１６２において、エージェントは、マネージャ１１１にＴＣＰ／ＩＰプロトコルによりコンポーネントデータを送信することによってコンポーネントデータを報告する。

【００３１】

図２Ｂは、トランザクションが終了すると何が起こるかを表している。しかしながら、サブコンポーネントが終了すると、実行されるステップは、タイムスタンプを取得すること、サブコンポーネントのためのスタックエントリを削除すること及び完了したサブエレメントを以前のスタックエントリに加えることを含む。一実施形態では、フィルタ及び判断ロジックは、特定のサブコンポーネントというよりも、トランザクションの開始及び終了に適用される。

【００３２】

一実施形態では、トランザクショントレーサがオフになっている場合、システムは依然としてブレイムスタックを使用するが、しかし、パラメータは格納されことなく、コンポーネントデータは作成されないことに注意されたい。いくつかの実施形態では、トレーシング技術をオフにすることによってシステムはトレーシングを開始しない。トレーシングは上述したように、ユーザがリクエストした後にだけ開始する。

【００３３】

図３は、図１Ａ又は１Ｂのネットワークのコンピューティングデバイスを示す。コンピューティングデバイス３００は、図１Ａ又は１Ｂに関連して説明したように、ウェブブラウザ、アプリケーションサーバ、マネージャ及び／又はユーザインタフェースで使用されるシステムを簡略化して表したものである。コンピューティングデバイス３００は、ハードディスク又はポータブルメディアのような記憶装置３１０、他のコンピューティングデバイスと通信するためのネットワークインタフェース３２０、ソフトウェアの命令を実行するためのプロセッサ３３０、例えば、記憶装置３１０からロードされた後にソフトウェアの命令を格納するためのＲＡＭのような作業メモリ３４０、及び１以上のビデオモニタのようなユーザインタフェースディスプレィ３５０を含むものである。ユーザインタフェースは１以上のモニタを提供する。記憶装置３１０は、本明細書で説明した機能を提供す

るための方法を実行するのにプロセッサ３３０をプログラミングするために具現化されているプロセッサ読み取り可能なコードを有する、プロセッサ又はコンピュータで読み取り可能な有体であり一時的でない記憶装置と考えることができる。ユーザインタフェースディスプレィ３５０は、１以上のエージェントから受信したデータに基づいて、人間のオペレータに情報を提供する。ユーザインタフェースディスプレィ３５０は、グラフィカル又は表形式のような既知の任意の表示方式を使用する。画面上の表示に加えて、プリンタからのハードコピーなどの出力も提供する。

【００３４】

記憶装置３１０がアプリケーションサーバ、マネージャ及び／又はユーザインタフェースのようなコンピューティングデバイス３００の一部である場合、データベースは記憶装置３１０に含まれる。記憶装置３１０は、１以上のエージェントから受信したデータを格納し、本明細書で説明したようにユーザインタフェースを提供するためにデータを取得するためにアクセスされる、１以上の記憶装置を表す。記憶装置３１０は、データストアを表す。

【００３５】

また、本明細書で説明する機能は、ハードウェア、ソフトウェア又はハードウェアとソフトウェアの両方の組み合わせを使用して実装されてもよい。ソフトウェアについては、１以上のプロセッサをプログラミングするために具現化されているプロセッサで読み取り可能なコードを有する、プロセッサで読み取り可能な１以上の一時的でない有体の記憶装置が使用される。プロセッサ読み取り可能な一時的でない有体の記憶装置は、揮発性及び不揮発性メディア、リムーバブル及び非リムーバブルメディアなどのコンピュータで読み取り可能な媒体を含む。例えば、コンピュータにより読み取り可能な一時的でない有体の媒体には、コンピュータにより読み取り可能な命令、データ構造やプログラムモジュール又は他のデータなどの情報を記憶するために、任意の方法や技術で実装された、揮発性、不揮発性、リムーバブルや非リムーバブルメディアが含まれ得る。コンピュータにより読み取り可能な一時的でない有体の媒体の例としては、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリ、又は他のメモリ技術、ＣＤ－ＲＯＭ、ディジタルバーサタイルディスク（ＤＶＤ）、又は他の光学ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置、他の磁気記憶装置、又は所望の情報を格納したり、コンピュータによってアクセスしたりすることに用いられる他の媒体などがある。他の実施形態においては、一部又はすべてのソフトウェアは、カスタムＩＣ、ゲートアレイ、ＦＰＧＡ、ＰＬＤや特殊用途向けプロセッサなど、専用のハードウェアに置き換えることができる。一実施形態では、１以上の実施形態を実装するソフトウェア（記憶装置に格納されている）は、１以上のプロセッサをプログラムするために用いられる。１以上のプロセッサは、コンピュータにより読み取り可能な１以上の有体の媒体／記憶装置、周辺機器及び／又は通信インタフェースと通信することができる。

【００３６】

図４は、１以上のアプリケーションの操作を示す際に用いられる階層を示す。異なるレベルの階層が必要な組織構造に基づいて定義される。例えば、階層は、人間が理解し易い用語を含み、その用語は、クライアントの、監視されるアプリケーションとの相互関係の理解を容易にするものである。階層は、相互関係が営利目的の業務の領域にあるか否か、例えば、電子商取引のトランザクションか、教育組織又は政府組織のためにあるかなど、アプリケーションとの相互関係の類型を包含する。さらに、１以上の階層は、各ノードが記述名を持つ１以上の階層の異なるレベルにおけるノードを含む。階層は、人間のオペレータにいっそう理解され易い仕方でアプリケーションを実行するやり方についての情報を、体系化する方法を提供する抽象的な構成であると考えられる。

【００３７】

階層の最上位は、「ドメイン」と名付けられたドメインレベル４００である。階層の次のレベルは、業務サービスレベル４０２である。業務サービスの例として、ウェブサイトを用いた株式の取引に関するものがある。このように、「取引」は、階層の業務サービス

レベルにおけるノードの名称とすることができる。階層のその次のレベルは、業務トランザクションレベルである。業務サービスは、いくつかの業務トランザクションで構成されている。例えば、取引のために、業務トランザクションは、報告４０４（例えば、株式又は口座に関する報告を表示する）及び見積４０６（例えば、株価の見積を取得する）を含む。さらに、業務トランザクションは、１以上の業務トランザクションコンポーネントに関連付けられる。１つのアプローチでは、業務トランザクションは、唯一の識別コンポーネントを有する。業務トランザクションコンポーネントは、サーブレット又はＥＪＢなどのサーバによって認識可能かつ測定可能なアプリケーションのコンポーネントの型になり得る。１つのアプローチでは、アプリケーションのコンポーネントの１つは、業務トランザクションのための識別トランザクションコンポーネントである、業務トランザクションコンポーネントとして設定される。

【００３８】

業務トランザクションコンポーネントは、業務トランザクションに対する識別トランザクションであるトランザクションのための識別トランザクションコンポーネントである。トランザクションは、クライアントに応答するレスポンスを提供するために、クライアントからのリクエストに応答して呼び出されるソフトウェアコンポーネントのシーケンスを表す。例えば、業務トランザクションコンポーネントは、エージェントによって報告されたコンポーネントデータが一連のルールに一致することをもって識別される。この定義は、例えば、特定のＵＲＬのホスト名称、ＵＲＬのパラメータ、ＨＴＴＰポストパラメータ、クッキー及び／又はセッションマネージャのパラメータなどを含む。加えて、又はその代わりに、この定義は、特定のＵＲＬのホスト名称で開始するトランザクションを必要とする場合がある。エージェント又はマネージャは、例えば、業務トランザクションコンポーネントが業務トランザクション内に存在していることを判定するために、コンポーネントデータを一連のルールと比較する。業務トランザクションコンポーネントが検出される場合、関連付けられた業務トランザクションは特定の型のものである。例えば、業務トランザクションコンポーネント４０８が検出される場合、関連付けられた業務トランザクションは、報告４０４である。業務トランザクションコンポーネント４１０が検出される場合、関連付けられた業務トランザクションは見積４０６である。

【００３９】

図５Ａは、図４の報告及び見積の業務トランザクション内で呼び出されるコンポーネントの一例のシーケンスおける依存関係を示す。コンポーネントは、フロー経路内のブロックとして示される。同じコンポーネントが２回以上現れる。また、コンポーネントは異なるサブシステム内、即ちサブシステム１（点線の上方のコンポーネント）又はサブシステム２（点線の下方のコンポーネント）内で実行する。

【００４０】

コンポーネント指向のプログラミングモデルは、コンポーネントに関係する構成要素からアプリケーションや他のプログラムをプログラマに組ませるのに役に立つ。各コンポーネントは、ソフトウェアの全体的機能に適合するよう特定の機能を実行する。さらに、コンポーネントは、コンポーネントのシーケンスがプログラム内で呼び出されるように他のコンポーネントを呼び出し、同様に、再帰呼び出しでは自分自身を呼び出す。コンポーネント指向のプログラミングモデルの一例は、Ｊ２ＥＥであるが、Java（登録商標）サーバページズ、エンタープライズJava（登録商標）ビーンズ（ＥＪＢ）、サーブレット及びJava（登録商標）データベースコネクティビティ（ＪＤＢＣ）のコンポーネントといったコンポーネントを用いることができる。ＪＤＢＣは、クライントがどのようにデータベースにアクセスするかを定義するJava（登録商標）プログラミング言語のためのアプリケーションプログラミングインタフェース（ＡＰＩ）である。それは、データベース内のデータを照会や更新する方法を提供する。しかしながら、.NETのような他のコンポーネント指向のプログラミングモデルを使用することもできる。また、プログラミングモデルは、オブジェクト指向である必要はない。

【００４１】

この例では、前述の報告及び見積の業務トランザクションの詳細を提供する。１つの可能な実装では、業務トランザクションの各コンポーネントは、１以上のクラスメソッドの組を含む。例えば、サーブレットはJava（登録商標）クラスである。リクエストを受信して対応する応答を生成するのは、オブジェクトである。クラスメソッドの組は、class.methodの表記により表される。例えば、報告は、所望の報告に関するユーザの入力を受信するために、ユーザインタフェース（ＵＩ）上の報告画面を表示するコンポーネントＣ１（５０２）を含む。Ｃ１に対するクラスメソッドの組のフォーマットの例は、ServletA1.DisplayReportScreenである。Ｃ１は、ルート５００の下にある。したがって、エージェントがＣ１の呼び出されたことを検出する度に、現在のトランザクションが報告の一部であり、そのコンポーネントデータが報告に関連付けられることになる。

【００４２】

Ｃ１は、リクエストされた報告に関係するＣ２（５０４）を呼び出す。Ｃ２は、リクエストされた報告のユーザ入力を処理するServletA2.RequestedReportなどのクラスメソッドの組を含む。この処理は、リクエストのフォーマットを調べることを含み、例えば、フォーマットが有効である場合、報告リクエストを受信する、サブシステム２内のコンポーネントＣ５（５０８）を呼び出すことを含む。例えば、この呼び出しは、クロスプロセス、クロススレッドトランザクション又はクロスサブシステム呼び出しである。フォーマットが無効である場合、制御フローは、例えば、Ｃ１に戻り、エラーメッセージを表示するためにＣ３、を呼び出す。

【００４３】

Ｃ５に対するクラスメソッドの組のフォーマットの例は、ServletA3.ReceiveReportRequestである。Ｃ５は、例えば、報告リクエストの種類に基づいて、データベース１にアクセスするためにＣ６（５１０）、及び／又は、データベース２にアクセスするためにＣ７（５１２）、を呼び出す。例えば、Ｃ６及びＣ７は、１以上のＳＱＬ文を呼び出すＪＤＢＣドライバをそれぞれ含む。制御フローは、次にＣ５に戻り、その次にＣ２に、そしてその次にＣ１に戻る。その後、Ｃ１は、データベースから検索されたデータに基づいてリクエストされた報告の表示などの表示を提供することに関連しているＣ３（５０６）を呼び出す。制御フローは、次にＣ１に戻る。

【００４４】

Ｃ１は、例えば、報告データを異なる形式（異なる期間に亘る、等々）で表示するためのユーザコマンド（例えば、再表示）に基づくなどして、表示を調整するためにＣ３をさらに何回か呼び出すことがある。

【００４５】

また、ルート５００の下において、コンポーネントＣ４（５１４）は、所望の見積に関するユーザの入力を受信するためにユーザインタフェース（ＵＩ）上に見積画面を表示するものを提供する。Ｃ１は、リクエストされた報告に関するＣ２（５０４）を呼び出す。Ｃ２は、例えば、リクエストのフォーマットを調べることによってユーザ入力を処理し、フォーマットが有効である場合、例えば、サブシステム１にローカルであるデータソースから、リクエストされた見積を取得することを処理する。フォーマットが無効である場合、制御フローはＣ４に戻り、Ｃ４は、例えば、エラーメッセージを表示するために、Ｃ３を呼び出す。

【００４６】

制御フローは、次にＣ４に戻る。Ｃ４は、データソースから検索されたデータに基づいてリクエストされた見積の画面などの画面を提供することに関連するＣ３（５１８）を呼び出す。Ｃ４は、例えば、見積データを異なる形式（例えば、別の移動平均を伴い、異なる期間に亘る、等々）で表示するためのユーザコマンド（例えば、再表示）に基づくなどして、表示を調整するためにＣ３をさらに何回か呼び出すことがある。

【００４７】

コンポーネントは、非同期、マルチスレッド又はマルチプロセスのモードで実行を開始する別のコンポーネントを呼び出した後、実行を継続できることに注意されたい。又は、

この呼び出されたコンポーネントが同期、シングルスレッド又はシングルプロセスのモードで実行を終了するまで、コンポーネントを一時的に休止できる。休止しているコンポーネントは、待機期間にあると考えられ、一方、実行されているコンポーネントは、アクティブで、実行モードであると考えられる。また、コンポーネントは、トランザクションの期間中、２回以上呼び出すことが可能である。

【００４８】

図５Ｂは、図５Ａの依存関係の代わりとなる表示で、よりコンパクトなものを示す。ノード５０５はノード５０４及び５１６を結合し、そしてノード５０７はノード５０６及び５１８を結合する。

【００４９】

図６（Ａ）から（Ｃ）、及び、図６Ｄから図６Ｉは、図５Ａのトランザクション内で呼び出されたコンポーネントの異なるシーケンスに対するトランザクショントレースを示す。水平方向は時間を表し、一方、垂直方向は呼び出しスタックの深さや位置を表す。また、呼び出しスタックと称されるトランザクショントレースは、１以上のプログラム、プロセス又はスレッドの実行中に、呼び出される又は起動される計測されたコンポーネントを識別する。計測されたコンポーネントのトレースデータは、アプリケーションを理解したりデバッグしたりするために依存データとともに使用される。トランザクショントレースは、トレース又はトランザクションの全部若しくは一部とすることができ、それぞれのエージェントを有する１以上のコンピューティングデバイスに及ぶ。特に、異なる別々のトランザクショントレースは、別々のスレッドが別々のトランザクショントレースに分離されるように、各エージェントに対して提供される。トランザクショントレースは、ユーザインタフェース上のグラフ表示によって提供される。

【００５０】

図６（Ａ）のトランザクショントレースは、図５Ａのブロック５０２及び５０４に対応している。グラフ部分６００はＣ１を表し、グラフ部分６０２はＣ２を表す。Ｃ１は、ｔ０で実行を開始して、ｔ７で終了又は停止する。Ｃ１によって呼び出されるＣ２は、ｔ１で実行を開始してｔ６で終了する。

【００５１】

図６（Ｂ）のトランザクショントレースは、図６Ａのトランザクショントレースと時間軸を合わせており、図５Ａのブロック５０８及び５１０に対応している。グラフ部分６１０はＣ５を表し、グラフ部分６１２はＣ２を表す。Ｃ５は、ｔ２で実行を開始してｔ５で終了する。Ｃ５によって呼び出されるＣ６は、ｔ２で実行を開始してｔ４で終了する。

【００５２】

図６（Ｃ）のトランザクショントレースは、図６（Ａ）のトランザクショントレースと時間軸を合わせており、図５Ａのブロック５０８及び５１２に対応している。グラフ部分６２０はＣ５を表し、グラフ部分６２２はＣ７を表す。Ｃ５は、ｔ２で実行を開始してｔ５で終了する。Ｃ５によって呼び出されるＣ７は、ｔ２で実行を開始してｔ４で終了する。図６（Ｃ）のトランザクショントレースは、例えば、データベース２がデータベース１の代わりに呼び出された場合、図６（Ｂ）のトランザクショントレースに代えることができる。タイムポイントｔ２－ｔ５は、図６（Ｂ）と必ずしも同じである必要はない。また、タイムポイントｔ０、ｔ１、ｔ２等々は、一般的に等分の時間増加分を表す必要はない。

【００５３】

図６Ｄのトランザクショントレースは、図５Ａのブロック５０２及び５０４に対応している。グラフ部分６３０はＣ１を表し、グラフ部分６３２はＣ２を表す。Ｃ１は、ｔ０で実行を開始してｔ３で終了する。Ｃ１によって呼び出されるＣ２は、ｔ１で実行を開始してｔ２で終了する。このトランザクショントレースは、Ｃ１がＣ２を呼び出し、Ｃ２がユーザリクエストのフォーマットが無効であると判定したため制御フローがＣ１に直接戻るというと場合を表す。

【００５４】

図６Ｅのトランザクショントレースは、図５Ａのブロック５０２及び５０６に対応している。グラフ部分６４０はＣ１を表し、グラフ部分６４２はＣ３を表す。Ｃ１は、ｔ０で実行を開始してｔ３で終了する。Ｃ１によって呼び出されるＣ３は、ｔ１で実行を開始してｔ２で終了する。このトランザクショントレースは、Ｃ１がＣ３を呼び出し、Ｃ３が報告を表示又は再表示する場合を表す。

【００５５】

図６Ｆのトランザクショントレースは、図５Ａのブロック５０２及び５０６に対応している。グラフ部分６５０はＣ１を表し、グラフ部分６５２及び６５４はＣ３の別々の呼び出しを表す。Ｃ１は、ｔ０で実行を開始してｔ５で実行を終了する。Ｃ３は、Ｃ１によって最初に呼び出されたとき、ｔ１で実行を開始してｔ２で終了する。Ｃ３は、Ｃ１によって２回目に呼び出されたとき、ｔ３で実行を開始してｔ４で終了する。このトランザクショントレースは、Ｃ１がＣ３を、報告を表示するために最初に呼び出す場合、及び報告を再表示するために２回目に呼び出す場合を表す。

【００５６】

図６Ｇのトランザクショントレースは、図５Ａのブロック５０２、５０４及び５０６に対応している。グラフ部分６６０はＣ１を表し、グラフ部分６６２はＣ３を表し、及びグラフ部分６６４はＣ２を表す。Ｃ１は、ｔ０で実行を開始してｔ５で終了する。Ｃ３は、Ｃ１によって呼び出されたとき、ｔ１で実行を開始してｔ２で終了する。Ｃ２は、Ｃ１によって呼び出されたとき、ｔ３で実行を開始してｔ４で終了する。このトランザクショントレースは、Ｃ１が報告を表示するためにＣ３を呼び出し、ユーザが報告に対する別のリクエストをするが、そのリクエストは無効のフォーマットであり、それゆえ、制御フローがＣ２からＣ１に直接戻る場合を表す。

【００５７】

図６Ｈのトランザクショントレースは、図５Ａのブロック５１４及び５１６に対応している。グラフ部分６７０はＣ４を表し、グラフ部分６７２はＣ２を表す。Ｃ４は、ｔ０で実行を開始してｔ３で終了する。Ｃ１によって呼び出されるＣ２は、ｔ１で実行を開始してｔ２で終了する。このトランザクショントレースは、Ｃ４が見積に対するユーザリクエストでＣ２を呼び出す場合を表す。

【００５８】

図６Ｉのトランザクショントレースは、図５Ａのブロック５１４及び５１８に応答している。グラフ部分６８０はＣ４を表し、グラフ部分６８２はＣ３を表す。Ｃ４は、ｔ０で実行を開始してｔ３で終了する。Ｃ１によって呼び出されるＣ３は、ｔ１で実行を開始してｔ２で終了する。このトランザクショントレースは、Ｃ４がＣ３を呼び出し、Ｃ３が見積を表示する場合を表す。

【００５９】

図７Ａ１は、図６から６Ｉのトランザクショントレースに基づき提供されるエージェント１及びエージェント２のツリーデータ構造の例を示す。ツリーデータ構造は、有効グラフ、又は、ノード及び矢印若しくはノードを接続する辺を含む分散ツリーによって表される。ツリーを通じたそれぞれの異なる経路が、ツリーのブランチを表す。各個別のブランチは、少なくとも１つのアプリケーションの呼び出された複数のコンポーネントのそれぞれのトランザクション又はシーケンスを表す。また、各ノードは、コンポーネントの実行の開始又は終了を表す。各ノードは、また固有の識別子を含む。そして、ブランチの最終ノードの識別子は、そのブランチ固有の識別子（例えば、サブシステム／エージェント内で固有である）としての役割を果たす。即ち、ブランチ内の最終ノードの識別子が与えられると、ブランチ内の各先行ノードを最初のノードまで、即ち、そのブランチのルートノードまで遡ることができる。ツリーのブランチは、複数のサブシステムに亘って延長する、コンポーネントのシーケンス又はトランザクションもまた表し得る。例えば、点線よりも上方のブランチ部分は、サブシステム１で実行するコンポーネントのノードを含み、点線より下方のブランチ部分は、サブシステム２で実行するコンポーネントのノードを含む。複数のブランチは、少なくとも部分的に重複し、共通のノードを有する。通常、少なく

ともルートノードは、複数のブランチに共通である。

【００６０】

アプリケーション又は他のソフトウェアを監視するエージェントは、関連付けられたツリーデータ構造を保守する。例えば、サブシステム１のエージェント１（ａｇｔ１）は、ルートノード７００から始まるツリーデータ構造を保守し、サブシステム２のエージェント２（ａｇｔ２）は、ルートノード７４２から始まるツリーデータ構造を保守する。マネージャは、１以上のエージェントのツリーデータ構造に基づくツリーデータ構造を保守管理する。例えば、マネージャは、エージェント１及びエージェント２のツリーデータ構造を結合する、図７Ｃ１のツリーデータ構造を保守管理する。

【００６１】

ルートノード７００は、サブシステム１におけるすべてのブランチの開始ノードである。第１ブランチ（エージェント１－ブランチ１、トランザクションのエージェント１－Ｔ１を表す）は、ノード７０２、７０４、７０６及び７０８を含む。第２ブランチ（エージェント１－ブランチ２、トランザクションのエージェント１－Ｔ２を表す）は、ノード７０２、７１０、７１２及び７１４を含む。第３ブランチ（エージェント１－ブランチ３、トランザクションのエージェント１－Ｔ３を表す）は、ノード７０２、７１０、７１２、７１６、７１８及び７２０を含む。第４ブランチ（エージェント１－ブランチ４、トランザクションのエージェント１－Ｔ４を表す）は、ノード７０２、７１０、７１２、７２２、７２４及び７２６を含む。第５ブランチ（エージェント１－ブランチ５、トランザクションのエージェント１－Ｔ５を表す）は、ノード７２８、７３０、７３２及び７３４を含む。第６ブランチ（エージェント１－ブランチ６、トランザクションのエージェント１－Ｔ６を表す）は、ノード７２８、７３６、７３８及び７４０を含む。

【００６２】

ルートノード７４２は、サブシステム２におけるすべてのブランチの開始ノードである。第１ブランチ（エージェント２－ブランチ１、トランザクションのエージェント２－Ｔ１を表す）は、ノード７４４、７４６、７４８及び７５０を含む。第２ブランチ（エージェント２－ブランチ２、トランザクションのエージェント２－Ｔ２を表す）は、ノード７４４、７５２、７５４及び７５６を含む。

【００６３】

各ノードの識別子は、例えば識別子内の値の数に基づいて、ブランチ内におけるノードのシーケンスの位置を示す。例えば、ノード７０２は、識別子「０：０」を有する。この識別子は、ルートノード（識別子「０」を有する）の後の、ブランチの第２ノードであることを示す、コロンで区切られた２つの値を有する。第２、第３及び第４ブランチでは、ノード７０２、７１０及び７１２（第２、第３及び第４ノード）は共通である。第２ブランチでは、最終ノード７１４は、識別子０：０：１：０：０を有する。第３ブランチでは、最終ノード７１６は、識別子０：０：１：０：１を有する。第４ブランチでは、最終ノード７２２は、識別子０：０：１：０：２を有する。様々な他のノードの識別方式／コード言語が同様に使用されてもよい。

【００６４】

ノードの識別子は独立して割り当てられているため、異なるサブシステム内で潜在的に繰り返され得る。しかしながら、サブシステムの識別子（例えば、エージェントの識別子）とノードの識別子の組み合わせは固有のものである。

【００６５】

ツリーデータ構造は、様々な方法で提供され得る。１つのアプローチでは、サブシステムのエージェントは、付加的なトランザクションがトレースされるに従って時間の経過とともにツリーデータ構造を構築する。各トランザクショントレースは、例えば、呼び出されたコンポーネントのシーケンスは、ツリーのブランチと比較され、一致するものが存在するか否かが判定される。一致するものが存在する場合、そのトランザクションは、既にツリーによって表されている。しかしながら、一致するものが存在しない場合、トランザクションは、ツリーによって表されておらず、ツリーデータ構造は、新しいトランザクシ

ョンを表すために更新される。更新は、部分的に、重複する又は重複しない新しいブランチを既存のブランチに付加することを含む。新しいブランチは、新しいトランザクションの呼び出されたコンポーネントの開始と終了を表す付加的なノードを付加することによって提供される。付加的なノードは、既にツリーデータ構造内に存在している呼び出されたコンポーネントの開始及び終了の別のインスタンスを表す。例えば、エージェント１－ブランチ３、ノード７１０及び７１２は、Ｃ３の１つのインスタンスの開始と終了を表し、ノード７１６及び７１８は、Ｃ３の別のインスタンスの開始と終了を表す。

【００６６】

複数のサブシステムに亘って伸びるツリーのブランチの例は、サブシステム１のエージェント１－ブランチ１とサブシステム２のエージェント２－ブランチ１又はエージェント２－ブランチ２とを結合して、図７Ｃ１に示されている。例えば、サブシステム２のブランチ１内のノード７４２は、サブシステム１のブランチ１内のノード７０４に続き、そしてブランチ１内のノード７０６に戻る。又は、サブシステム２のブランチ２内のノード７４２は、サブシステム１のブランチ１内のノード７０４に続き、そしてブランチ１内のノード７０６に戻る。いずれの場合も、少なくとも１つのコンポーネント、例えば、第１サブシステム内のＣ２は、少なくとも１つのコンポーネント、例えば、第２サブシステム内のＣ５を呼び出す。

【００６７】

サーバ及びマネージャの各エージェントは、互いに対応する独立したツリーデータ構造を保守する。理想的には、ツリーデータ構造は、それらが少なくとも１つのアプリケーション又は他のソフトウェアのトランザクションの同じセットを表すように、少なくとも部分的には同期がとられる。述べたように、エージェントが新しいトランザクションを検出すると、ツリーデータ構造を更新し、マネージャに更新を報告する。マネージャが次にツリーデータ構造を更新する。さらに、少なくとも１つのアプリケーションの他のインスタンスを監視する他のエージェントが存在し、そして、エージェントがそれらのそれぞれのツリーデータ構造を更新するために更新を受信することも同様に望ましい。１つのアプローチでは、新しいトランザクションを検出するエージェントが他のエージェントに直接にその更新を提供する。別のアプローチでは、エージェントはマネージャに更新を報告し、マネージャは他のエージェントに更新を中継する。このアプローチは、他のエージェントがマネージャに報告して、それらに更新が伝わるのをマネージャが知るため、効率的である。更新はどのようなフォーマットでも提供可能である。エージェントからマネージャに送られてくる更新は、動的データと共に、あるいは個別に伝えられる。

【００６８】

対応するツリーデータ構造をエージェント及びマネージャに保守管理させることによって、高い効率が得られる。例えば、トランザクション、及び、トランザクションのコンポーネントに関連付けられた静的データは、ツリーのノードにインデックス付けされ、それによって、単にツリー内のブランチを特定することでマネージャによるその使用が可能になる。静的データは、エージェントによってマネージャに繰り返し伝えられる必要はない。静的データは、一般的に、アプリケーション又は他の監視されたソフトウェアのバージョンに対するものであるため変化しない。したがって、与えられたトランザクション又はコンポーネントの複数の呼び出しに対して同じ静的データが関連付けられる。対照的に、コンポーネントの開始及び終了時刻のような動的データ、及びメソッドに渡されるパラメータ値のような他の動的データは、固定されず、それゆえ、与えられたコンポーネントの各呼び出し、及び、トレースされた各トランザクションごとに変化する。エージェントによって収集される動的データは、エージェントからマネージャへ報告される。しかし、効率は、動的データが適用される呼び出されたコンポーネントを容易に識別するために動的データをノードへインデックス付けすることによってもまだなお得られる。これらの効率を得るために使用される様々なデータ構造が図８Ａ１から８Ｃに関連して説明される。

【００６９】

図７Ａ２は、図７Ａ１のツリーデータ構造の代替的かつ等価的な表示を示す。ここで、

ノード７１１及び７１７は、ノード７１０と同じであり、ノード７１３及び７１９は、同じそれぞれのノードの識別子を有するノード７１２と同じである。この表示では、エージェント１－ブランチ２は、ノード７１０、７１２及び７１４を含み、エージェント１－ブランチ３は、ノード７１１、７１３、７１５、７１８及び７２０を含み、エージェント１－ブランチ４は、ノード７１７、７１９、７２１、７２４及び７２６を含む。この表示では、ノード７１４が、エージェント１－ブランチ３及びエージェント１－ブランチ４の一部ではないことを明確にしている。

【００７０】

図７Ｂは、新しいブランチの形式を有する、図７Ａ１のエージェント１のツリーデータ構造の更新を示す。明確にするため、サブシステム２は示されてない。トランザクションの「エージェント１－Ｔ新しい」を表す、「エージェント１－新しい－ブランチ」は、図９のプロセス［処理］に関連して後述するように、エージェント１のツリーデータ構造の更新により付加された新しいブランチである。「エージェント１－新しい－ブランチ」は、識別子「０：０」を有する既存のノード７０２（開始Ｃ１）、識別子「０：０：２」を有するノード７６０（開始Ｃ８）、識別子「０：０：２：０」を有するノード７６２（終了Ｃ８）、及び、識別子「０：０：２：０：０」を有するノード７６４（終了Ｃ１）を含む。これは、エージェント１によって新たに検出された経路である。

【００７１】

図７Ｃ１は、図７Ａ１の、エージェント１及びエージェント２のツリーデータ構造を結合した、マネージャのツリーデータ構造を示す。述べたように、マネージャのツリーデータ構造は、複数のエージェント及びアプリケーション又は他のソフトウェアに及ぶデータ構造を提供するために、異なるエージェントのツリーデータ構造を結合する。この場合、マネージャのツリーデータ構造の第１の部分は、第１エージェントのツリーデータ構造に対応しており、マネージャのツリーデータ構造の第２の部分は、第２エージェントのツリーデータ構造に対応している。破線のノード（ノード７４４、７４６、７４８、７５０、７５２、７５４及び７５６）は、エージェント２のツリーデータ構造からのノードに対応し、点線のノード（ノード７９０及び７９２）は、破線のノードに基づいて付加される。マネージャのツリーデータ構造の実線のノードは、エージェント１のツリーデータ構造に対応する。Ｃ２はＣ５を呼び出すことがわかっているため、ノード７０４はノード７４４を指し示す。Ｃ５はＣ２に戻ることがわかっているため、ノード７５０は、ノード７４４、７４６、７４８及び７５０のシーケンスが続いている場合には、ノード７０６を指し示し、そして、ノード７４４、７５２、７５４及び７５６のシーケンスが続いている場合には、ノード７５６がノード７９０を指し示すところに、付加的なノード７９０及び７９２が付加される。ノード７９０及び７９２は、エージェント１の観点から、それぞれノード７０６及び７０８と同じである。

【００７２】

このように、マネージャ（ｍｇｒ）のツリーデータ構造は、これらのブランチ：マネージャ－ブランチ１、マネージャ－ブランチ２、マネージャ－ブランチ３（エージェント１－ブランチ２と同じ）、マネージャ－ブランチ４（エージェント１－ブランチ３と同じ）、マネージャ－ブランチ５（エージェント１－ブランチ４と同じ）、マネージャ－ブランチ６（エージェント１－ブランチ５と同じ）、及びマネージャ－ブランチ７（エージェント１－ブランチ６と同じ）、を含む。トランザクションが複数のサブシステムに関与するので、マネージャ－ブランチ１は、クロスサブシステムのトランザクション内で呼び出されたコンポーネントのシーケンスを表す。マネージャ１－ブランチ１は、トランザクションエージェント２－Ｔ１（ノード７４４、７４６、７４８及び７５０）によって続けられ、トランザクションエージェント１－Ｔ１の残り（ノード７０６及び７０８）によって続けられる、例えばトランザクションエージェント１－Ｔ１の一部（ノード７０２及び７０４）のような、複数のトランザクションを結合するトランザクションマネージャ－Ｔ１を表す。トランザクションエージェント１－Ｔ１は、サブシステム１からのもので、エージェント２－Ｔ１は、サブシステム２からのものであることを想起されたい。マネージャ－

ブランチ２は、トランザクションマネージャ－Ｔ２を表し、それは、トランザクションエージェント１－Ｔ１の一部（ノード７０２及び７０４）を結合したものであり、その後にトランザクションエージェント２－Ｔ１（ノード７４４、７５２、７５４及び７５６）が続き、その後にトランザクションエージェント１－Ｔ１の残り（ノード７９０及び７９２）が続く。マネージャ－ブランチ３は、トランザクションのエージェント１－Ｔ２（ノード７０２、７１０、７１２及び７１４）と同じであるトランザクションマネージャ－Ｔ３を表す。マネージャ－ブランチ４は、トランザクションのエージェント１－Ｔ３（ノード７０２、７１０、７１２、７１６、７１８及び７２０）と同じであるトランザクションマネージャ－Ｔ４を表す。マネージャ－ブランチ５は、トランザクションのエージェント１－Ｔ４（ノード７０２、７１０、７１２、７２２、７２４及び７２６）と同じであるトランザクションマネージャ－Ｔ５を表す。マネージャ－ブランチ６は、トランザクションのエージェント１－Ｔ５（ノード７２８、７３０、７３２及び７３４）と同じであるトランザクションマネージャ－Ｔ６を表す。マネージャ－ブランチ７は、トランザクションのエージェント１－Ｔ６（ノード７２８、７３６、７３８及び７４０）と同じであるトランザクションマネージャ－Ｔ７を表す。

【００７３】

図７Ｃ１のノードの識別子は、ノード７４４（０：０：０：０）、ノード７４６（０：０：０：０：０）、ノード７４８（０：０：０：０：０：０）、ノード７５０（０：０：０：０：０：０：０）、ノード７０６（０：０：０：０：０：０：０：０）、ノード７０８（０：０：０：０：０：０：０：０：０）、ノード７５２（０：０：０：０：１）、ノード７５４（０：０：０：０：１：０）、ノード７５６（０：０：０：０：１：０：０）、ノード７９０（０：０：０：０：１：０：０：０）及びノード７９２（０：０：０：０：１：０：０：０：０）、を除いて、図７Ａ１と同じである。これらはマネージャの識別子である。マネージャ－ブランチ１、マネージャ－ブランチ２、マネージャ－ブランチ３、マネージャ－ブランチ４、マネージャ－ブランチ５、マネージャ－ブランチ６及びマネージャ－ブランチ７の識別子は、エージェント１から見て、それぞれノード７０８、７９２、７１４、７２０、７２６、７３４及び７４０の識別子である。

【００７４】

マネージャのツリーデータ構造が、異なる複数のエージェントによる複数のツリーデータ構造を結合するとき、マネージャのトランザクションは、複数のエージェントの複数のトランザクションを結合する。マネージャトランザクションのエージェントトランザクションに対する一対多の対応の例として、マネージャ－Ｔ１は、エージェント１－Ｔ１及びエージェント２－Ｔ１を結合する。図８Ａ１から８Ａ３を参照されたい。この場合、マネージャのトランザクションのユーザインタフェースディスプレィは、複数のエージェントトランザクションに基づくものである。

【００７５】

或いは、マネージャのツリーデータ構造は、異なる複数のエージェントによる複数のツリーデータ構造を結合する必要はないが、マネージャは、本質的に各エージェントのツリーデータ構造の複製である、各エージェントに対する個別のツリーデータ構造を保守管理する。この場合、マネージャのトランザクションは、エージェントのトランザクションと同じである。マネージャトランザクションのエージェントトランザクションに対する一対一の対応の例として、マネージャ－Ｔ３は、エージェント１－Ｔ２と同じである。この場合、マネージャのトランザクションのユーザインタフェースディスプレィは、単一のエージェントトランザクションに基づくものである。

【００７６】

又は、マネージャは、異なる複数のエージェントの複数の個別のツリーデータ構造、及び、異なる複数のエージェントによる複数のツリーデータ構造を結合した１つのツリーデータ構造の双方を保守管理し得る。個別のツリーデータ構造は、図９のステップ９０４にあるように、ブランチの一照合のために使用され、一方、異なる複数のエージェントによる複数のツリーデータ構造を結合した１つのツリーデータ構造は、例えば、図１０Ａのス

テップ１００８及び１０１０にあるようにユーザインタフェースを提供するために使用され得る。

【００７７】

図７Ｃ２は、図７Ａ１のエージェント１のツリーデータ構造における最終ノードと図７Ｃ１のマネージャのツリーデータ構造の最終ノードの間の対応を示す。述べたように、ツリーデータ構造内のブランチの最終ノードの識別子は、ブランチを一意的に識別するために使用される。場合によっては、同一の最終ノードの識別子がエージェント及びマネージャのツリーデータ構造で使用される。他の場合、例えばマネージャが異なる複数のエージェントによる複数のツリーデータ構造を結合する場合、複数の異なる最終ノードの識別子がエージェント及びマネージャの夫々のツリーデータ構造で使用される。マネージャは、複数の最終ノードの識別子間の対応レコードを保守管理し得る。例えば、０：０：０：０：０の、エージェント１の最終ノードの識別子は、識別子０：０：０：０：０：０：０：０：０及び０：０：０：０：１：０：０：０：０を有する、マネージャの２つの最終ノード（ノード７０８及び７９２）に対応する。エージェント１の残りの最終ノードの識別子（図７Ａ１のノード７１４、７２０、７２６、７３４及び７４０の識別子を参照）はマネージャのものと同じである。また、０：０：０：０：０の、エージェント２の最終ノードの識別子は、識別子０：０：０：０：０：０：０：及び０：０：０：０：１：０：０を有する、マネージャの２つの最終ノードに対応する。この例では、考慮すべきエージェント２の残りの最終ノードの識別子は存在しない。ノード番号は、理解の助けのために提供されているのであり、必ずしも対応レコードの一部であるわけではない。

【００７８】

したがって、マネージャが第１ノードのシーケンスのエージェント１からの最終ノードの識別子、及び、第２ノードのシーケンスのエージェント２からの最終ノードの識別子を受信するとき、マネージャは、１以上のこれらの最終ノードの識別子に基づいて、そのツリーデータ構造にアクセスすることができる。また、アクセスは、エージェント１及び／又はエージェント２の最終ノードの識別子、及び／又は、マネージャの対応する最終ノードの識別子に基づく。

【００７９】

図７Ｄは、図７Ｂにおけるエージェント１のツリーデータ構造の更新と整合する、新しいブランチの形式をした図７Ｃ１のマネージャのツリーデータ構造の更新を示す。更新は、ノード７６０、７６２及び７６４を含む新しいブランチ、「マネージャ－新しい－ブランチ」であり、それは、エージェント１のツリーデータ構造についての「エージェント１－新しい－ブランチ」の更新と整合する。

【００８０】

図８Ａ１は、図７Ａ１のツリーデータ構造におけるサブシステム１のためのブランチ及びコンポーネント呼び出しのレコードを示す。各ブランチは、最終ノードの識別子によって識別される。例えば、「０：０：０：０：０」は、図７Ａ１のノード７０８を識別し、それゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ１、及び、これに対応するトランザクションのエージェント１－Ｔ１もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ１（ノード７０２）、開始Ｃ２（ノード７０４）、終了Ｃ２（ノード７０６）、及び終了Ｃ１（ノード７０８）である。

【００８１】

「０：０：１：０：０」は、図７Ａ１のノード７１４を識別し、それゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ２及びトランザクションのエージェント１－Ｔ２もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ１（ノード７０２）、開始Ｃ３（ノード７１０）、終了Ｃ３（ノード７１２）、及び終了Ｃ１（ノード７１４）である。

【００８２】

「０：０：１：０：１：０：０」は、図７Ａ１のノード７２０を識別し、それゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ３及びトランザクションの

エージェント１－Ｔ３もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ１（ノード７０２）、開始Ｃ３（ノード７１０）、終了Ｃ３（ノード７１２）、開始Ｃ３（ノード７１６）、終了Ｃ３（ノード７１６）、及び終了Ｃ１（ノード７２０）である。

【００８３】

「０：０：１：０：２：０：０」は、図７Ａ１のノード７２６を識別し、れゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ４及びトランザクションのエージェント１－Ｔ４もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ１（ノード７０２）、開始Ｃ３（ノード７１０）、終了Ｃ３（ノード７１２）、開始Ｃ２（ノード７２２）、終了Ｃ２（ノード７２４）、及び終了Ｃ１（ノード７２６）である。

【００８４】

「０：１：０：０：０」は、図７Ａ１のノード７３４を識別し、それゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ５及びトランザクションのエージェント１－Ｔ５もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ４（ノード７２８）、開始Ｃ２（ノード７３０）、終了Ｃ２（ノード７３２）、及び終了Ｃ４（ノード７３４）である。

【００８５】

「０：１：１：０：０」は、図７Ａ１のノード７４０を識別し、それゆえ、両方ともサブシステム１に存在する、エージェント１－ブランチ６及びトランザクションのエージェント１－Ｔ６もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ５（ノード７４４）、開始Ｃ７（ノード７５２）、終了Ｃ７（ノード７５４）、及び終了Ｃ５（ノード７５６）である。

【００８６】

図８Ａ２は、図７Ａ１のツリーデータ構造におけるサブシステム２のためのブランチ及びコンポーネント呼び出しのレコードを示す。

【００８７】

「０：０：０：０：０」は、図７Ａ１のノード７５０を識別し、それゆえ、両方ともサブシステム２に存在する、エージェント２－ブランチ１及びトランザクションのエージェント２－Ｔ１もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ５（ノード７４４）、開始Ｃ６（ノード７４６）、終了Ｃ６（ノード７４８）、及び終了Ｃ５（ノード７５０）である。

【００８８】

「０：０：１：０：０」は、図７Ａ１のノード７５６を識別し、それゆえ、両方ともサブシステム２に存在する、エージェント２－ブランチ２及びトランザクションのエージェント２－Ｔ２もまた識別する。このブランチに対応するコンポーネント呼び出しは、開始Ｃ５（ノード７４４）、開始Ｃ６（ノード７４６）、終了Ｃ６（ノード７４８）、及び終了Ｃ５（ノード７５０）である。

【００８９】

図８Ｂ１は、図７Ａ１のツリーデータ構造におけるサブシステム１の異なるノード／コンポーネントのための静的データへの参照のレコードを示す。述べたように、静的データの様々な種類は、コンポーネントとそれに関連付けられたノードから参照され得る。例えば、ノード「０：０」は、コンポーネントＣ１に関連付けられ、static#data#C1（例えば、methodC1、classC1 及び JARC1など）に参照付けられる。参照される静的データの異なるレコードは、後述するように、図８Ｄに示される。クラス名は、１以上の、親又はスーパークラスの名称も含み得る。１つのアプローチでは、１以上のノードから静的データを参照し得る。別のアプローチでは、コンポーネントの開始を表すノード（ただし、コンポーネントの終了を表すノードではない）は、静的データを参照し得る。他のアプローチも可能である。目的は、例えば、トランザクショントレースのようなユーザインタフェースに注記を加えるために、与えられたコンポーネント又はノードに関連付けられた静的デー

タにマネージャがアクセスできるようにすることである。図８Ａ１及び８Ｂ１のレコードは、エージェントによって及び／又はエージェントが報告するマネージャによってツリーデータ構造の一部として提供されている。

【００９０】

レコードは、ブランチ又はブランチの一部を形成するノードをグループ化することができる。例えば、最初の５つのエントリ（「０：０」から「０：０：０：０：０」）は、エージェント１－ブランチ１に対するもので、最後のエントリ（「０：０：０：０：０」）は、ブランチの識別子である。エントリ０：０：１、０：０：１：０及び０：０：１：０：０は、エージェント１－ブランチ１にはないエージェント１－ブランチ２にあるノードに対するものである。

【００９１】

ノードは、静的データの１以上の種類に直接的に、又は静的データの１以上の種類を参照するための識別子を参照し得る。このように、静的データの識別子は、静的データの１以上の種類を繰り返すことなく、レコードにおいて効率的に繰り返される。

【００９２】

静的データは、静的データが参照される１以上のコンポーネントの計測を含む、ソフトウェアの計測から取得される。

【００９３】

トランザクションの静的データは、殆どが計測から取得される。しかしながら、原則として、それは他のソースから取得され、及びマッシュアップされ、又は必要に応じて他の静的データと結合される。例えば、与えられたコード部分が静的に常に与えられたアプリケーションに関連すること、又は静的に常に優先度が低くなることは、他のソースから検出される。この情報は、トレースの動作を決定するために使用される。

【００９４】

静的データは、ソフトウェアをトレースすることから入手可能なすべての種類の情報を含む。また、静的データは、ソフトウェアが構築される方法のために、特定のコンポーネントが、限定された数の１以上の親のコンポーネントによってのみ呼び出され得ること、及び／又は、特定のコンポーネントが、限定された数の１以上の子のコンポーネントのみを呼び出し得ること、を示すことができる。例えば、静的データは、Ｃ２がＣ１又はＣ４によってのみ呼び出され、及びＣ２がＣ５のみを呼び出すことを示す場合があってよい。静的データは、特定のコンポーネンントを呼び出した１以上の親コンポーネントに基づいて、その特定のコンポーネントが限られた数の１以上の子コンポーネントを呼び出し得ることを示すことができる。ツリーデータ構造の観点から言えば、例えば、特定のノードは、例えば特定のコンテキストにおいて、その特定のノードに到達した方法に基づいた１つの子のノードを有するのみである。この情報は、トランザクションのコンテキストに応じて、トランザクションデータを区別するのと同様に照合ステップ９０４において有益である。

【００９５】

別の例として、サーブレットは、ＳＱＬ文を使用してデータベースの様々なメソッドを呼び出す。しかしながら、サーブレットは、常に任意にメソッドを呼び出すわけではない。サーブレットは、何かが前もって発生した場合にあるＳＱＬを呼び出し、又は別の何かが前もって発生した場合に別のＳＱＬを呼び出す。これは、業務ロジック（ビジネスロジック］に従って関連性のあるＳＱＬのパーティションを提供する。例えば、ウェブサイト上で本を購入するためのトランザクションの場合、データベースロジックの特定の部分が使用され、一方、ウェブサイト上で帽子を購入するトランザクションの場合、データベースロジックの別の部分が使用される。どちらの場合も、サーブレットは、データベース呼び出しを行うために、同じソケットを使用する。しかし、ツリーデータ構造を使用すると、データが特定のトランザクションのコンテキスト内に集められる。このデータは、トランザクショントレース、及び、応答時間などを生じさせるメトリックを含むことができ、同様に、トランザクションのために取得された他のメトリックも含むことができる。

【００９６】
静的データは、ソフトウェア及び／又は計測から繰り返し検索される必要がないようにエージェントによってキャッシュされる。
【００９７】
図８Ｂ２は、図７Ａ１のツリーデータ構造におけるサブシステム２の異なるノード／コンポーネントのための静的データへの参照のレコードを示す。これらのレコードは、サブシステム２のエージェントによってツリーデータ構造の一部として提供され、関連付けられたマネージャに報告される。これは、例えば、サブシステム１のエージェントが報告するのと同じマネージャである。複数のエージェントが１つの共通のマネージャに報告する。レコードにおける一例としては、ノード「０：０」は、コンポーネントＣ５に関連付けられており、static#data#C5を参照する。
【００９８】
図８Ｂ３は、図７Ｂにおける「エージェント１－新しい－ブランチ」のための図８Ｂ１のレコードの更新を示す。ノード７６０、７６２及び７６４は、識別子０：０：２、０：０：２：０及び０：０：２：０：０をそれぞれ有し、static#data#C8、static#data#C8及びstatic#data#C1に、それぞれインデックス付けされている。
【００９９】
図８Ｂ４は、図７Ｃ１のツリーデータ構造におけるマネージャの異なるノード／コンポーネントのための静的データへの参照のレコードを示す。述べたように、各ノードは静的データに関連付けられている。
【０１００】
図８Ｂ５は、図７Ｄにおける「マネージャ－新しい－ブランチ７」のための図８Ｂ４のレコードの更新を示す。ノード７６０、７６２及び７６４は、識別子０：０：２、０：０：２：０及び０：０：２：０：０をそれぞれ有し、static#data#C8、static#data#C8及びstatic#data#C1に、それぞれインデックス付けされている。更新は、この例では、共通のノードの識別子のため、図８Ｂ３と同じである。他の場合では、例えば、異なるノードの識別子のために、更新は異なる。
【０１０１】
図８Ｃは、図７Ａ１のツリーデータ構造のサブシステム１の異なるノード／コンポーネントのためのトレース詳細からの動的データのレコードを示す。レコードは、サブシステム１のエージェントによってツリーデータ構造の一部として提供され、関連付けられたマネージャに報告される。動的データは、少なくとも１つのアプリケーション又は他の監視されたソフトウェアのインスタンスをトレースすることによって、エージェントにより取得される。動的データは、コンポーネントの開始及び終了時刻を示すことができる。他の動的データは、コンポーネント間の呼び出しで渡されたパラメータを含んでよい。例えば、図５Ａにおいて、Ｃ１は、報告の種類又は報告日の範囲に関連するリクエストされた報告に関連付けられた１以上のパラメータを有する、Ｃ２を呼び出す。制御フローがＣ１に戻るとき、Ｃ２は、１以上の関連するパラメータをＣ１に渡す。各サブシステムは、関連付けられたエージェントを介して動的データを取得してマネージャに報告する。レコードは、サブシステム１のエージェントによって、及び、エージェントが報告するマネージャによって、ツリーデータ構造の一部として提供される。
【０１０２】
動的データは、Ｃ１に関連付けられた、Ｃ１のための開始時刻（ｔ１）及び呼び出し中にＣ１に渡されるパラメータ１のような他の関連付けられた動的データ（dynamic#data#1）、を含むノード「０：０」に対するエントリを含む。ノード「０：０：０」に対するエントリは、Ｃ２に関連付けられており、Ｃ２のための開始時刻（ｔ２）及び呼び出し中にＣ２に渡されるパラメータ２のような他の関連付けられた動的データ（dynamic#data#2）を含む。ノード「０：０：０：０」に対するエントリは、Ｃ２に関連付けられており、例えばＣ２によって呼び出されたコンポーネントからＣ２へのプログラムフローの戻りなどの、戻り中にＣ２に渡されるパラメータ３のような、Ｃ２のための終了時刻（ｔ３）、及

び他の関連付けられた動的データ（dynamic#data#3）を含む。ノード「０：０：０：０：０」に対するエントリは、Ｃ１に関連付けられており、例えばＣ１によって呼び出されたコンポーネントからＣ１へのプログラムフローの戻りのなどの、戻り中にＣ１に渡されるパラメータ４のような、Ｃ１のための終了時刻（ｔ４）及び他の関連付けられた動的データ（dynamic#data#4）を含む。

【０１０３】

図８Ｄは、様々なコンポーネントに関連付けられた静的データのレコードを示す。本明細書で説明したように各レコードは、静的データの様々な種類を含む。静的データのレコードは、static#data#C1、static#data#C2、static#data#C3、static#data#C4、static#data#C5、static#data#C6、static#data#C7、及びstatic#data#C8を含む。静的データのレコードは、エージェント及びマネージャによって保守管理される。

【０１０４】

図９は、少なくとも１つのアプリケーションのために図７Ａ１にあるようなエージェントがツリーデータ構造を保守管理するプロセス［処理］の例を示す。ステップ９００は、コンポーネントの開始及び停止ポイントなどによって、少なくとも１つのアプリケーションの呼び出されたコンポーネントのシーケンスを表すブランチを有する、ツリーデータ構造を保守することを含む。ステップ９０２は、トランザクション、例えばトランザクションのインスタンスが複数のインスタンスに経時的に呼び出される期間中、少なくとも１つのアプリケーションの呼び出されたコンポーネントのシーケンスを識別することを含む。例えば、これはトランザクションをトレースすることを含む。着目している特定のトランザクションは、対象トランザクションと称される。

【０１０５】

ステップ９０８は、トランザクションの期間中、例えば呼び出されたコンポーネントの開始及び終了時刻などの、呼び出されたコンポーネントのシーケンスのためのメトリックのような動的データを取得すること、を含む。この動的データは、トランザクショントレースから取得される。判断ステップ９０４において、ツリーデータ構造内のブランチに一致するブランチが存在するか否かについての判定が行われる。例えば、トランザクショントレースが以下の呼び出されたコンポーネントのシーケンス、開始Ｃ１、開始Ｃ２、終了Ｃ２、終了Ｃ１になると仮定する。このシーケンスは、例えば、一致するブランチが発見されるまで、図７Ａ１のツリーデータ構造内の各ブランチと順々に比較される。１つのアプローチでは、比較は、最初のブランチから始めて一つのブランチごとに行われる。別のアプローチでは、トランザクショントレースの開始及び終了ポイントの数に対応するノード数を有する複数のブランチが最初に比較される。それ以外のアプローチもまた可能である。この例では、「エージェント１－ブランチ１」が一致するブランチである。ステップ９０６は、動的データ及び一致するブランチ（例えばエージェント１－ブランチ１、又はノード０：０：０：０：０）の識別子をマネージャに報告することを含む。動的データは、例えば、各時刻がブランチのいずれかのノードに対応し、報告された時刻の順番がブランチ内のノードの順番に対応する場合、呼び出されたコンポーネントの開始及び終了時刻のリストとして報告される。時刻は、例えば、エージェントの時計に基づいたタイムスタンプになる。

【０１０６】

一致するブランチは、トランザクションの呼び出されたコンポーネントのシーケンスの開始及び終了ポイントの数と同じ数のノードを有するブランチであり、そのブランチ内のノードのシーケンスは、トランザクションの呼び出されたコンポーネントのシーケンスの開始及び終了ポイントと一致する。ツリーのルートノードは、この照合では考慮する必要はない。場合によって、ブランチが、トランザクションの呼び出されたコンポーネントのシーケンスの開始及び終了ポイントが一致するノードのシーケンスを有するが、付加的なノードも同様に有し得る。この場合、トランザクションの呼び出されたコンポーネントのシーケンスに対する開始及び終了ポイントの部分的な一致があり、判断ステップ９０４は、偽［ノー］になる。この場合、対象トランザクションのトレースは、ツリーデータ構造

のブランチとして正確には表されておらず、ツリーデータ構造のブランチと共通の広がりを持つ呼び出されたコンポーネントのシーケンスの開始及び終了ポイントの新しいシーケンスを提供する。この判定に応答して、この判断ステップ９１０は、呼び出されたコンポーネントのシーケンスを表すとともに呼び出されたコンポーネントのシーケンスと共通の広がりを持つブランチを有するツリーデータ構造を更新することを含む。例えば、これは、図７Ｂの「エージェント１－新しい－ブランチ」である。共通の広がりを持つブランチは、シーケンスと同じ開始及び終了ポイントを有する。

【０１０７】

ステップ９１２で、更新は、新しいブランチで、トランザクショントレースにおける１以上の呼び出されたコンポーネントの開始及び終了ポイントを表すノードを提供することを含む。例えば、図７Ｂでは、エージェント１－新しい－ブランチは、新たに付加されたノード７６０、７６２及び７６４を含む。新しいブランチは、１以上の既存のブランチと部分的に重複する。例えば、図７Ｂにおいて、ノード７０２は、「エージェント１－ブランチ１」、「エージェント１－ブランチ２」及び「エージェント１－新しい－ブランチ」で存在（重複）しており、「エージェント１－新しい－ブランチ」が、「エージェント１－ブランチ１」及び「エージェント１－ブランチ２」と重複している。

【０１０８】

このように新しいトランザクションの呼び出されたコンポーネントのシーケンスは、既存のブランチ（例えば、「エージェント１－ブランチ１」及び「エージェント１－ブランチ２」）の少なくとも１つと重複する部分（ノード７０２）、及び、どの既存のブランチとも重複しない部分（ノード７６０、７６２及び７６４を含むブランチ部分）を有するブランチ（例えばエージェント－新しい－ブランチ）によって、ツリーデータ構造内で表される。新しいノード（ノード７６０、７６２及び７６４）は重複しない部分において現れるが、重複する部分では現れない。

【０１０９】

図９において、ステップ９１４は、ツリーデータ構造の更新が１以上の呼び出されたコンポーネントに関連付けられた静的データをノードにインデックス付けすることを含むことを示している。コンポーネントの静的データは、コンポーネントの計測からエージェントによってアクセスされ、図８Ｂ３に関連して説明したようにインデックス付けされる。

【０１１０】

ステップ９１６は、ツリーデータ構造の更新がエージェントからマネージャへ報告されることを含む。更新は、対象トランザクションのインスタンスの１以上の呼び出されたコンポーネントの開始及び終了ポイントを識別し、関連付けられた静的データにインデックス付けされる。この報告は、図８Ａ１又は８Ａ２で説明したようなブランチ定義、及び図８Ｂ１又は８Ｂ２で説明したような静的データへの参照という形式で提供される。

【０１１１】

新トランザクションに基づいてツリーデータ構造を更新した後、判断ステップ９０４において、トランザクショントレースの呼び出されたコンポーネントのシーケンスが更新されたツリーデータ構造と再度比較され、真［イエス］になる。ステップ９０６は、動的データ及び一致するブランチの識別子をエージェントからマネージャに報告することを含む。この報告は、例えば、図８Ｃのレコードの形式で提供される。この報告を受信すると、マネージャは、エージェントのツリーデータ構造に同期するようにそのツリーデータ構造を更新する。したがって、データを通信経路に送信するために必要なバンド幅の量及び／又はそのようなデータを通信及び格納するために必要なメモリの量のようなオーバーヘッドコストを削減しながら、エージェントは、効率的にトランザクションをマネージャに報告することができる。

【０１１２】

図１０Ａは、エージェントから受信するように、例えば図７Ａ１にあるように、動的データのレコード及びツリーデータ構造のブランチの識別子に基づいて、マネージャがユーザインタフェースを提供するプロセス［処理］の例を示す。ステップ１０００は、コンポ

ーネントの開始及び停止ポイントなどによって、少なくとも１つのアプリケーションの呼び出されたコンポーネントのシーケンスを表すブランチを有する、マネージャのツリーデータ構造を保守することを含む。ステップ１００２は、動的データ及び一致するブランチの識別子の報告をエージェントから受信することを含む。ステップ１００４は、識別子に基づいて、呼び出されたコンポーネントのシーケンスを識別することを含む。これは、「エージェント１－ブランチ１」が、最終ノードの識別子が「０：０：０：０：０」であるブランチによって識別されることを判定し、図８Ａ１にあるようにレコードにアクセスして、このブランチが、開始Ｃ１、開始２、終了２及び終了Ｃ１のコンポーネントシーケンスを含むことを判定することを伴う。

【０１１３】

或いは、ステップ１００４は、エージェント１の最終ノード０：０：０：０：０がマネージャの最終ノード０：０：０：０：０：０：０：０：０に対応することを判定するために図７Ｃ２にあるようにレコードにアクセスすること、並びに「マネージャ－ブランチ１」が、マネージャの最終ノード０：０：０：０：０：０：０：０：０によって識別され、図８Ａ３

にあるようにレコードにアクセスして、このブランチが開始Ｃ１、開始Ｃ２、開始Ｃ５、開始Ｃ６、終了Ｃ６、終了Ｃ５、終了Ｃ２、及び終了Ｃ１のコンポーネントのシーケンスを含むことを判定することを含む。

【０１１４】

ステップ１００６は、識別子に基づいて、トランザクションの呼び出されたコンポーネントに関連付けられた静的データを調べることを含む。これは、例えば、ノード／ブランチの識別子「０：０：０：０：０」及びブランチの各ノードにインデックス付けされているstatic#data#C1を識別するために、図８Ｂ１にあるようなレコードにアクセスすることを伴う。或いは、これは、図８Ｂ４にあるようにレコードにアクセスして、例えば、ノード／ブランチの識別子「０：０：０：０：０：０：０：０：０」及びブランチの各ノードにインデックス付けされているstatic#data#C1を識別することを伴う。

【０１１５】

ステップ１００８は、トランザクションの呼び出されたコンポーネントのシーケンスのトランザクショントレースを有するユーザインタフェース（ＵＩ）を提供することを含む。ブランチがブランチの各コンポーネントの開始及び終了を識別するので、トランザクショントレースは識別されたブランチから直接、提供される。ユーザインタフェース上に提供されたトランザクショントレースの例が図６から６Ｉ、１１Ａ、及び、１１Ｂに示されている。ステップ１０１０は、例えば、図１１（Ａ）及び（Ｂ）に示されるように、静的及び／又は動的データに基づいてトランザクショントレースに注記を加えることを含む。これは、静的及び／又は動的データをユーザインタフェース上に表示することを含む。別の例として、図１４Ａから１４Ｃに関連して説明する、ＵＩが提供される。

【０１１６】

図１０Ｂは、１以上のエージェントから受信した更新に基づいて、図７Ａ１から７Ｃ１にあるように、マネージャがツリーデータ構造を更新するプロセスの例を示す。ステップ１０２０は、例えばコンポーネントの開始及び停止ポイントなどによって、少なくとも１つのアプリケーションの呼び出されたコンポーネントのシーケンスを表すブランチを有する、ツリーデータ構造を保守することを含む。ステップ１０２２は、例えば第１サブシステムの第１エージェント及び第２サブシステムの第２エージェントなどの１以上のエージェントからツリーデータ構造の更新を受信することを含む。ステップ１０２４は、１つのエージェントから別のエージェントに更新を伝えることを含む。エージェントが同じソフトウェアの異なる複数のインスタンスを監視する場合、マネージャは、１つのエージェントから別のエージェントに受信された更新を渡す又は中継する。このように、エージェントのツリーデータ構造が同期をとられるように新しいトランザクションがエージェント間で迅速に伝達する。ステップ１０２６は、更新から呼び出されたコンポーネントのシーケ

ンスを表すブランチによってマネージャのツリーデータ構造を更新することを含む。例えば、これは、図７Ｄの「マネージャー新しい－ブランチ」を付加することを含む。更新は、例えば、図８Ｂ３のレコードに基づいて、マネージャのツリーデータ構造のレコードを更新することを伴う。

【０１１７】

ステップ１０２８では、更新は、トランザクションの１以上の呼び出されたコンポーネントの開始及び終了ポイントを表すノードを提供することを含む。例えば、これは、図７Ｄにおけるマネージャー新しい－ブランチのノード７６０、７６２及び７６４を付加することを含む。ステップ１０３０では、例えば図８Ｂ５のレコードに関連して説明したように、更新は、トランザクションの呼び出されたコンポーネントに関連付けされた静的データを、ノードにインデックス付けすることを含む。マネージャのツリーデータ構造への更新は、図７Ｄの「マネージャー新しい－ブランチ」の例においては、エージェントのツリーデータ構造（例えば、ノード７６０、７６２及び７６４）のノードのいくつかを含むが、エージェントのツリーデータ構造のノード（例えば、ノード７０２）の他のものは含まないことに注意されたい。

【０１１８】

図１１（Ａ）は、静的及び動的データを用いて注記を加えた、図６（Ａ）のトランザクショントレースを示す。トランザクショントレースは、トランザクション／実行フローの全体像を提供する。ここで、注記がＣ１のためのグラフ領域６００において、及びＣ２のためのグラフ領域６０２において提供されている。注記「methodC1|classC1|JARC1|dynamic#data#1」は、動的データによって続けられる３種類の静的データを含み、データの各部分が垂直バーによって区切られている。しかしながら、他のフォーマットも可能である。例えば、注記は、マウスオーバー又はホバーボックス、ツールチップ、あるいは、情報にアクセスするための右クリックによる、ポップアップウィンドウ、別のウィンドウ又は表示画面など、トランザクショントレースのグラフ領域の外側、例えば上や横、に備えられる。動的データは、その外観、色、フォント、位置等々によって静的データとは別に区別される。

【０１１９】

図１１（Ｂ）は、静的及び動的データを用いて注記を加えた、図６（Ａ）のトランザクショントレースを示す。注記は、Ｃ５のためのグラフ領域６１０において、及び、Ｃ６のためのグラフ領域６１２において提供される。図１１（Ａ）及び（Ｂ）のトランザクショントレースは、サブシステムに亘って延長するトランザクションの動作をユーザにより一層の理解をもたらすために、同じユーザインタフェース上に同時に表示される。Ｃ１及びＣ２は、サブシステム１に存在し、Ｃ５及びＣ６は、サブシステム２に存在することを想起されたい。複数のサブシステムのクロックが、適切に同期がとられている場合は、複数のサブシステムの複数のトランザクショントレースは、共通のタイムライン参照を用いて表示される。同期が保証されない場合は、複数のサブシステムの複数のトランザクショントレースは、別々のタイムライン参照を用いて表示される。マネージャは、Ｃ２がＣ５を呼び出すときに提供する相関識別子に基づいて、ユーザインタフェースにおいて２つのトランザクショントレースを関連付けることを決定することができる。トレースが関連付けられる必要があることを示すためにツリーデータ構造を用いてトランザクショントレースを報告するとき、エージェントはマネージャに相関識別子を提供する。詳細情報については、２００７年６月２１日公開の”Correlating Cross Process And Cross Thread Execution Flows In An Application Manager”と題する、ＵＳ２００７／０１４３３２３を参照のこと。その内容は参照により本明細書に組み込まれる。

【０１２０】

例えば、Ｃ２が「エージェント１－Ｔ１」のトランザクションで呼び出される場合、Ｃ２がＣ５を呼び出すときにそれは「エージェント１－Ｔ１」の識別子を含む。エージェント１は、「エージェント１－Ｔ１」のトランザクションに関してマネージャに報告する場合、「エージェント１－Ｔ１」の識別子を含む。同様に、エージェント２は、「エージェ

ント２－Ｔ１」のトランザクションに関してマネージャに報告するとき、「エージェント１－Ｔ１」の識別子、及び、「エージェント２－Ｔ１」の識別子を含む。マネージャは、その後、「エージェント１－Ｔ１」の識別子及び「エージェント２－Ｔ１」の識別子によるトランザクション／トランザクショントレースが関連付けられていることを知る。

【０１２１】

別の例では、ユーザインタフェースは、例えば、ノード及びノード間の辺を表示することなどによって、図７Ａ１から７Ｄのツリーデータ構造を、直接、提供する。ステータス及び動的データはノード内又はノードの横に表示される。

【０１２２】

図１２Ａは、それぞれのトランザクションのそれぞれのブランチにおける１つのコンポーネントに対する１つのノードにリンクされた収集部を有する図７Ａ１のツリーデータ構造を示す。１以上の収集部は、ツリーデータ構造のノードに関連付けられている。１以上の収集部は、１つのノードに関連付けられ、１以上のノードは１つの収集部に関連付けられている。１つの収集部に関連付けられている１以上のノードは、１以上のコンポーネントを表す。

【０１２３】

１つのアプローチでは、収集部は、ノードによって表されるコンポーネントの１以上のメトリックを収集する、エージェントコードのソフトウェアプロセスである。エージェントは、管理されるアプリケーションでメソッドのような計測されたコンポーネントに接続される組<エージェント　メトリック、収集部>の基本データ構造を使用することができる。メトリックは、例えば、コンポーネントが呼び出されて、コンポーネントの計測コードがトリガされるときに収集される。例えば、メトリックは、ノードによって表されるコンポーネントのインスタンスが呼び出される回数である呼び出し回数、ノードによって表されるコンポーネントのインスタンスの応答時間である応答時間、複数の呼び出しに亘る応答時間の平均、エラーメッセージがノードによって表されるコンポーネントに関連付けられているか否かを示すエラーメトリック、又は本明細書で説明されるものを含む他の任意のメトリックを含み得る。収集部をツリー内のノードにリンクすることにより、ノードによって表されるコンポーネント呼び出しは、収集部のコンテキストにリンクされる。同様に、収集部は、ブランチによって一意的に表される各トランザクション、及び、そのブランチのコンテキストに対して収集部の一対一のリンク設定がある場合には、そのノードにのみリンクされる。

【０１２４】

エージェントの有用な業務目的は、トランザクションの種類（トランザクションの分離可能性）によって分けられる固有のメトリックを提供することである。例を挙げると、例えば「書籍購入」トランザクション及び「ＣＤ購入」トランザクションのような与えられた顧客アプリケーション上の識別された各トランザクションのために、与えられたバックエンドに対する（又はバックエンド上で呼び出された特定のＳＱＬ文に対する）応答時間を報告したいと、する。この目的に向けたワンステップは、「トランザクション」メトリックセット、即ち、例えば、メトリックの値が、顧客のアプリケーションで呼び出された特定のトランザクションによって区別されるメトリック、を効率的に提供する能力である。

【０１２５】

ツリーデータ構造は、この目的のために使用される。述べたように、ツリーは、トランザクションのポイント又はノードの、分岐するシーケンスを介してトランザクションを説明する。シーケンスの辺は、トランザクションセグメントと称される。トランザクション構造は、それが検出されるコードの寿命と同等のライフサイクルを有する。トランザクション構造は、修正されたトレーサでアプリケーションを計測するエージェントによって最初に検出される。トランザクション構造は、次に他のエージェント及びマネージャと共有される。トランザクション構造は、またマネージャのデータベースに恒久的に格納される。

【０１２６】
エージェントは、トランザクション構造における各着目ポイントを１以上の収集部のセットで修飾する。各収集部は可能なメトリックの複数のセットに関連付けられている。エージェントは、そのメトリックに対する「数値」を収集し、エージェントの構造上の理由により、可能な各トランザクション経路の１つに関連付けられている特定の値を判定することが可能である。収集部のある種のものは、２以上のトランザクション構造のエレメント又はノードに関連付けられる。例えば、同時呼び出しの収集部は、収集されるメトリックに関連しているすべてのトランザクション構造のエレメントに関連付けられている。
【０１２７】
また、エージェントは、他のエージェントによって検出されたトランザクション構造に関するマネージャからの更新を受信する。これにより、例えば、クロスＪＶＭトランザクションの場合に、効率的に報告するエージェントを有することが可能となる。
【０１２８】
この例では、収集部１２００は、「エージェント１－Ｔ１」のトランザクションのノード７０６にのみリンクされており、収集部１２０２は、「エージェント－Ｔ４」のトランザクションのノード７２４にのみリンクされている。ノード７０６及び７２４は、両方ともＣ２の呼び出しの終了、例えばＣ２の停止時間を表すが、「エージェント１－Ｔ１」（「エージェント１－ブランチ１」によって表される）及び「エージェント１－Ｔ４」（「エージェント１－ブランチ４」によって表される）の各トランザクションの異なる複数のコンテキストにおけるものである。監視されたアプリケーションの性能をより一層理解できるようにするために、１以上のトランザクションのコンテキストに従って与えられたコンポーネントに対してメトリックを区別することは有益である。これは、例えば、コンポーネントに関する性能［パフォーマンス］の問題は、特定のトランザクションのコンテキストにおいては発生するが、別のコンテキストでは発生しないことを示している。問題のある特定のコンテキストは、さらに調査される。収集部は、例えばオペレータの経験に基づき、例えば、問題のある、コンポーネント及び／又はトランザクションを検出する自動化された分析に基づいて、設定される。
【０１２９】
収集部により、メトリックが取得され、ツリーにおける１以上の選択されたノードに基づいて選択的に報告される。コンポーネントが計測されることで、コンポーネントのすべての発生に関して、メトリックを取得し及び報告することが可能である。しかしながら、これは不必要なオーバーヘッドコストにつながる。必要に応じた場合にのみメトリックを収集し及び報告することが効率的である。トランザクション別（transaction-segregated）のメトリックを収集し報告することもまた有益である。例えば、ノード７０６で収集部１２００によって取得されたメトリックは、「エージェント１－Ｔ１」のコンテキストでは呼び出しＣ２のためのものである。或いは、収集部がノード７０６、７２４及び７３２にリンクされた場合、取得されたメトリックは、Ｃ２が呼び出されるすべてのトランザクションよりも少ないサブセットに特定されるわけではない。収集部１２００及び本明細書の収集部の他の表記は、１以上の種類のメトリックを収集する実体を表すものである。
【０１３０】
述べたように、別のアプローチでは、１つの収集部は複数のトランザクションにリンクされ、収集部によって取得された少なくとも１つのメトリックが複数のトランザクションのコンポーネントに関連付けられる。一般的に、様々な以下のようなバリエーションが可能である。（１）収集部の少なくとも１つのメトリックは、１つのトランザクションにおけるコンポーネントのインスタンスに関連付けられる（例えば、収集部１２００は、図１２Ａの「エージェント１－Ｔ１」におけるノード７０６のＣ２のインスタンスにリンクされる）。（２）収集部の少なくとも１つのメトリックは、１つのトランザクションにおける１つのコンポーネントの複数のインスタンスに関連付けられる（例えば、収集部１２０６は、図１２Ｃの「エージェント１－Ｔ３」におけるノード７１２及び７１８のＣ３のインスタンスにリンクされる）。（３）収集部の少なくとも１つのメトリックは、１つのト

ランザクションにおいて、１つのコンポーネントの、１つのコンポーネントのインスタンス、及び他のコンポーネントの、１つのコンポーネントのインスタンスに関連付けられる（例えば、収集部１２００は、図１２Ａの「エージェント１－Ｔ１」におけるノード７０６のＣ２のインスタンスにリンクされており、収集部１２００によって修正されたものもまた「エージェント１－Ｔ１」におけるノード７０８のＣ１のインスタンスにリンクされる）。並びに、（４）収集部の少なくとも１つのメトリックは、１つのトランザクションにおいて１つのコンポーネントの、１つのコンポーネントのインスタンスに、及び他のトランザクションにおいて他のコンポーネントの、１つのコンポーネントのインスタンスに関連付けられる（例えば、収集部１２００は、図１２Ａの「エージェント１－Ｔ１」におけるノード７０６のＣ２のインスタンスにリンクされており、収集部１２００によって修正されたものもまた「エージェント１－Ｔ２」におけるノード７１４のＣ１のインスタンスにリンクされる）。（４）の場合、少なくとも１つのメトリックは、例えば１つのトランザクション及び別のトランザクションなど、複数のトランザクション対するものである。

【０１３１】

図１２Ｂは、収集部を伴う図７Ａ１のツリーデータ構造を示すものであり、収集部は、異なる複数のトランザクションの異なる複数のブランチにおける同じコンポーネントの複数の実体のノード群にリンクされている。収集部１２０４は、ツリー内で一対多の関係でノード、即ち、異なる複数のトランザクションに存在するノード７０６及び７２４にリンクされている。この場合、収集部１２０４によって取得されるメトリックは、Ｃ２の異なる複数のインスタンス及び複数のトランザクションに亘って集約される。メトリックは、収集部１２０４のコンテキスト内で収集され、一つの特定のトランザクションのコンテキスト内では収集されない。メトリックは、「エージェント１－Ｔ１」及び「エージェント１－Ｔ４」を含む複数のトランザクションによる一つのグループのコンテキスト内で収集される。例えば、コンポーネントの１以上のインスタンスが複数のトランザクションを含む一つのグループ内で呼び出されたことを知ることが望ましいときには、有益であるが、各呼び出しを行ったトランザクションを区別する必要はない。

【０１３２】

図１２Ｃは、収集部を伴う図７Ａ１のツリーデータ構造を示すものであり、ここでの収集部は、複数のトランザクションの、同じ一つのブランチにおける同じコンポーネントの複数の実体のノード群にリンクされている。ここで、収集部１２０６は、ツリー内で一対多の関係でノード群にリンクされている、即ち、収集部１２０６は、「エージェント１－Ｔ３」のトランザクションにおける、同一のトランザクション内に存在し、及び同じコンポーネントＣ３の異なるインスタンスを表す、ノード７１２及び７１８にリンクされている。この場合、収集部１２０６によって取得されるメトリックは、「エージェント１－Ｔ３」のトランザクションにおけるＣ３の異なる複数のインスタンスに亘って集約される。メトリックは、一つの固有のトランザクションのコンテキストにおいて収集される。これは、コンポーネントの１以上のインスタンスがトランザクション内で呼び出されたことを知ることが望ましいときには、有益であるが、異なる複数のインスタンスのメトリック群を区別する必要はない。

【０１３３】

図７Ａ１に関連して説明したように、「エージェント１－Ｔ２」は、ノード７０２、７１０、７１２及び７１４のシーケンスを含み、「エージェント１－Ｔ３」は、ノード７０２、７１０、７１２、７１６、７１８及び７２０のシーケンスを含み、「エージェント１－Ｔ４」は、ノード７０２、７１０、７１２、７２２、７２４及び７２６のシーケンスを含むことを想起されたい。１以上のメトリックが、ノード７１２に対応するコンポーネント呼び出しに対して取得されるとき、３つの異なるノードのシーケンスのいずれもがノード７１２に続くため、トランザクションは、まだ一意的に定義されないことに注意されたい。この場合、複数のメトリックが収集され、次いで、そのトランザクションが特定のものではない場合、そのメトリックは破棄されマネージャには報告しないということが決定

される。これは、１つのアプローチでは、収集部１２０６がノード７１２にリンクされ、ノード７１２が、特定の一つのトランザクション、即ち、「エージェント１－Ｔ２」、「エージェント１－Ｔ３」又は「エージェント１－Ｔ４」の一部である場合にだけである。

【０１３４】

対照的に、１以上のメトリックがノード７１８に対応するコンポーネント呼び出しのために取得されるとき、トランザクション（エージェント１－Ｔ３）は、唯一のノードのシーケンスがノード７１８に続くことから、一意的に定義される。この場合、複数のメトリックは収集され、それらのメトリックがマネージャに報告されるという決定がなされる。一例として、エージェントは、例えばメトリックのような情報をマネージャに定期的に、例えばほんの短い間、報告する。通常、トランザクションは、終了するか、又はそうでなければ、次の報告時間以前に、一意的に識別し得るポイントに進む。エージェントは、未特定のトランザクションのメトリックを収集すると、そのトランザクションが特定されて、
メトリックを破棄するか、マネージャに対して報告するか、あるいは、その他の行動をとるか、が判定されるときまでそのメトリックを保持する。

【０１３５】

他のアプローチでは、メトリックを報告するか否かについての判定は、トランザクションが完了して識別された後に行われる。トランザクション別のメトリックが提供される、エージェント及びマネージャによる処理にそれぞれ関連している、図１６Ａ及び１６Ｂを参照されたい。

【０１３６】

図１２Ａから１２Ｃに示されるように、異なる種類の収集部は、組み合わせて使用してもよい。

【０１３７】

図１３Ａは、図１２Ａのツリーデータ構造における、収集部１２００及び１２０２についての参照のレコードを示す。エージェント及びマネージャは、例えば、レコードを保持することができる。レコードの第１エントリは、収集部１２００が、コンポーネントＣ２及び「エージェント１－Ｔ１」のトランザクションを表し、０：０：０：０の識別子を有するノードにリンクされていることを示す。レコードの第２エントリは、収集部１２０２が、コンポーネントＣ２及び「エージェント１－Ｔ４」のトランザクションを表し、０：０：１：０：２：０の識別子を有するノードにリンクされていることを示す。

【０１３８】

図１３Ｂは、図１２Ｂのツリーデータ構造における、収集部１２０４についての参照のレコードを示す。このレコードの第１エントリは、収集部１２０４が、コンポーネントＣ２及び「エージェント１－Ｔ１」のトランザクションを表し、０：０：０：０の識別子を有するノードにリンクされていることを示す。レコードの第２エントリは、収集部１２０４が、コンポーネントＣ２及び「エージェント１－Ｔ４」のトランザクションを表し、０：０：１：０：２：０の識別子を有するノードにもまたリンクされていることを示す。

【０１３９】

図１３Ｃは、図１２Ｃのツリーデータ構造における、収集部１２０６についての参照のレコードを示す。レコードの第１エントリは、収集部１２０６が、コンポーネントＣ３及び「エージェント１－Ｔ３」のトランザクションを表し、０：０：１：０の識別子を有するノードにリンクされていることを示す。レコードの第２エントリは、収集部１２０６が、コンポーネントＣ３及び「エージェント１－Ｔ３」のトランザクションを表し、０：０：１：０：１：０の識別子を有するノードにもまたリンクされていることを示す。

【０１４０】

図１４Ａは、図１３Ａのツリーデータ構造に基づくユーザインタフェースの例を示す。様々な種類のユーザインタフェース（ＵＩ）の表示は、１以上のエージェントからマネージャによって受信される、メトリック、ブランチの識別子及び収集部の識別子を含んでいる情報に基づいて提供される。１つの可能なアプローチにおいて、ＵＩの表示１４００は

、ツリーデータ構造及びそのノードを含む。線の色、パターン、幅又は塗りつぶし色若しくはパターンなどの視覚的特徴は、着目する１以上のトランザクションのノード群を識別し、着目していない１以上のトランザクションのノード群からこれらを区別するために使用される。同様に、視覚的な特徴は、着目しているコンポーネントを識別し、着目していないコンポーネントからこれらを区別するために使用される。

【０１４１】

例えば、太長破線は、「エージェント１－Ｔ１」を識別するために、ノード７０２、７０４、７０６及び７０８に使用され、太短破線は、「エージェント１－Ｔ４」を識別するために、ノード７０２、７１０、７１２、７２２、７２４及び７２６に使用される。領域１４０２は、収集部１２００によって収集されたデータに基づいて、ノード７０６によって表されたコンポーネントのインスタンスについてのメトリックの例を提供する。領域１４０４は、収集部１２０２によって収集されたデータに基づいて、ノード７２４によって表されたコンポーネントのインスタンスについてのメトリックの例を提供する。メトリックは、例えば、「エージェント１－Ｔ１」のトランザクションのコンテキストにおける、あるいは、それとは別に「エージェント１－Ｔ４」のコンテストにおける、エラー、平均応答時間、及び、呼び出し回数を含むことができる。

【０１４２】

図１４Ｂは、図１３Ｂのツリーデータ構造に基づくユーザインタフェースの例を示す。ＵＩの表示１４１０において、ＵＩの表示１４００にあるように、太長破線は、「エージェント１－Ｔ１」を識別するために、ノード７０２、７０４、７０６及び７０８に使用され、太短破線は、「エージェント１－Ｔ４」を識別するために、ノード７０２、７１０、７１２、７２２、７２４及び７２６に使用される。領域１４１２は、収集部１２００によって収集されたデータに基づいて、ノード７０６及び７２４によって表されたコンポーネントのインスタンスについてのメトリックの例を提供する。メトリックは、例えば、「エージェント１－Ｔ１」及び「エージェント１－Ｔ４」を含むトランザクションのグループのコンテキストにおける、エラー、平均応答時間及び呼び出し回数を含むことができる。

【０１４３】

図１４Ｃは、図１３Ｃのツリーデータ構造に基づくユーザインタフェースの例を示す。ＵＩの表示１４２０において、太長破線は、「エージェント１－Ｔ３」を識別するために、ノード７０２、７１０、７１２、７１６、７１８及び７２０に使用される。領域１４２２は、収集部１２０６によって収集されたデータに基づいて、ノード７１２及び７１８によって表されたコンポーネントのインスタンスについてのメトリックを提供する。メトリックは、例えば、「エージェント１－Ｔ３」のトランザクションのコンテキストにおける、エラー、平均応答時間、及び、Ｃ３の複数の実体の呼び出し回数を含むことができる。

【０１４４】

同じようなＵＩが他のサブシステムに提供される。ＵＩは、また、図７Ｃ１及び７Ｄに関連して説明したように、複数のサブシステムからの複数のノードを結合することができる。

【０１４５】

図１５Ａは、図５Ｂ及び１４Ａに対応するユーザインタフェースの例を示す。ＵＩ１５００は、ノード／頂点、及びノードを接続する矢印／辺を有するツリーデータ構造を含む有効グラフである。各ノードは呼び出されているコンポーネントを表し、それは、１つのノードがコンポーネントの実行又は呼び出しの、開始又は終了を表す図１４Ａから１４ＣのＵＩとは対照的である。各ノードは、１回以上開始及び停止するコンポーネントを表す。

【０１４６】

ノードは、ルートノード１５０１、並びに１つの経路において、Ｃ１のノード１５０２、Ｃ２のノード１５０４、又はＣ３のノード１５０６、Ｃ３のノード１５０６、Ｃ５のノード１５０８、Ｃ６のノード１５１０及びＣ７のノード１５１２を含む。別の経路は、同様に、Ｃ２又はＣ３を呼び出すＣ４のノードを含む。矢印／辺１５２４は、ノード１５０

２及び１５０４を接続しており、複数の辺部分を含む。１つの辺部分１５２０は、Ｃ２が「エージェント１－Ｔ１」のコンテキスにおいてＣ１によって呼び出されることを示しており、別の辺部分１５２２は、Ｃ２が「エージェント１－Ｔ４」のコンテキスにおいてＣ１によって呼び出されることを示している。さらに、辺部分１５２０及び１５２２は、それぞれの関連付けられたメトリックに基づいて、色、パターン又は太さなど様々な視覚的に異なる特徴を有する。各辺部分は、１以上の収集部に関連付けられることができ、その１以上の収集部によってメトリックが収集される。

【０１４７】

提供された例では、辺部分１５２０及び１５２２は、例えば、１つのトランザクション（エージェント１－Ｔ４）に起因する同じコンポーネント（Ｃ２）の呼び出しの回数に対する、別のトランザクション（エージェント１－Ｔ４）に起因する同じコンポーネントの呼び出しの回数に基づいて、太さによって互いに視覚的に区別される。この場合、辺部分１５２０は、辺部分１５２２よりも太く、例えば幅広である。例えば、辺部分１５２２の２倍の太さである辺部分１５２０は、Ｃ１が「エージェント１－Ｔ４」においてよりも「エージェント１－Ｔ１」で２倍呼び出されたことを示している。

【０１４８】

別の例では、辺部分１５２０及び１５２２の相対的な太さは、「エージェント１－Ｔ４」に起因するＣ２のエラーの数に対する、「エージェント１－Ｔ４」に起因するＣ２のエラーの数に基づくものである。さらに別の例では、辺部分１５２０及び１５２２の相対的な太さは、「エージェント１－Ｔ４」に起因するＣ２平均応答時間に対する、「エージェント１－Ｔ４」に起因するＣ２の平均応答時間に基づくものである。

【０１４９】

別の例では、赤などの暖色は、比較的高い呼び出し回数を示すために使用される一方で、青などの寒色は、比較的低い呼び出し回数を示す。別のアプローチで、赤はエラーの比較的高い数又は比較的高い平均応答時間を示す一方で、青はエラーの比較的低い数又は比較的低い平均応答時間を示す。他の多くの選択が可能である。

【０１５０】

さらに、表示領域１５２４及び１５２６は、それぞれ、辺部分１５２０及び１５２２に関連付けられているメトリックを提供する。各辺部分１５２０及び１５２２は、このように少なくとも１つのメトリックで修飾されている。表示領域１５２４は、「エージェント１－Ｔ１」の平均応答時間に対して設定された警告の「注意」状態を示す。平均応答時間が、例えば、閾値を超えるときにこの状態に設定される。一般的に、警告は、管理対象のコンピューティングデバイスの全体的な性能に対して、及び、管理対象のコンピューティングデバイスによる、例えば、別の管理対象のコンピューティングデバイス又は未計測のバックエンドデバイスの呼び出しに対して設定される。これらの警告は、ユーザによって作成されて設定されてもよい。警告は、また業務トランザクションに対して定義されてもよい。警告が定義される場合、それはいくつかの状態、例えば、正常（緑）、注意（黄色）又は危険（赤）のうちの１つによって表される。警告は、収集部によって取得されるトランザクション別のメトリックに対しても設定することができ、これによって、その警告は、１以上の特定のトランザクションに固有のものとなる。これは、ユーザがシステムを理解したり診断したりするのを助ける有益な情報を提供する。

【０１５１】

ＵＩは、表示領域１５２４又は１５２６内の複数のメトリックの１つのテキスト表記をユーザがクリックする、又は選択することを可能にし、選択されたメトリックのエッジ部分の幅又は他の視覚的特徴を変化させることができる。あるいは、プルダウンメニューなどの別のＵＩデバイスは、ユーザがＵＩを必要に応じて構成することを許容する。ＵＩは、また、着目する期間内にないデータを除去するなど、１以上の特定の基準を満たしていないデータを除去することができる。ＵＩは、ユーザの選択に応じて、１以上の報告されたエージェント／サブシステムに基づくデータを表示することができる。

【０１５２】

ＵＩにおける他のノード間の経路も同様に、使用可能なトランザクション別のメトリックに基づいて強調される。ＵＩは、また、ユーザに図１１Ａ及び１１Ｂにあるようなトランザクショントレースを表示することを可能にする。例えば、ＵＩは、図１５Ａ又は１５ＢのＵＩを提供するのに使われた個別の複数のトランザクションのリストを提供することができ、その複数のトランザクションのうちの一つを選択させて対応するトランザクショントレースを表示することができる。

【０１５３】

コンポーネントの性能メトリックをトランザクション又はトランザクションのグループによって区別する能力により、オペレータは、アプリケーションの性能をもっと容易に理解したり、コンポーネントに関連して、問題を診断したりすることができる。例えば、トランザクション別のメトリックを使用しないアプローチは、コンポーネントＣ１が異常に高い平均応答時間を有することを示すにすぎない。これと対照的に、トランザクション別のメトリックに基づいて、ＵＩは、「エージェント１－Ｔ１」及び「エージェント１－Ｔ４」の複数のトランザクションの夫々に対して平均応答時間を示すことができ、これによって、その複数のトランザクションのうちの特定の一つが、応答時間が遅いことの原因となっていることを判定することがおそらく可能となる。

【０１５４】

別の例として、電子商取引のウェブサイトの管理対象のコンピューティングデバイスが、顧客が商品アイテムの買い物をすることを可能にすること、及び、顧客の支払いの処理を可能にすること、を含むトランザクションを実行すると仮定する。あるいは、管理対象のコンピューティングデバイスが、顧客に様々な時間帯に買い物をすることを可能にすることを含むトランザクションを実行すると仮定する。トランザクション別のメトリックを提供する能力は、これら２つのトランザクションが別々に分析されることを可能にする。本明細書で提供されるアプローチは、収集されるデータの種類及びそれを収集するための処理を最適化する。

【０１５５】

図１５Ｂは、図１５Ａに代わるユーザインタフェースの例を示す。ＵＩ１５３０は、この例では、「エージェント１－Ｔ１」（アイコン１５３２によって示される）及び「エージェント１－Ｔ４」（アイコン１５３４によって示される）の、着目するトランザクションに焦点を合わせている。Ｃ１のノード１５３６は、トランザクション内の第１コンポーネントを表し、いずれもがＣ１によって呼び出される、Ｃ３のノード１５３７又はＣ２のノード１５３８は、トランザクション内の第２コンポーネントを表す。辺１５４５は、図１５Ａに関連して説明したように、それらの相対的な幅などによって互いに視覚的に区別される、辺部分１５４２及び１５４４を含む。メトリック表示領域１５４０及び１５４６は、それぞれ、辺部分１５４２及び１５４４にリンクされる。

【０１５６】

図１５Ｃは、ユーザインタフェースの別の例である。このＵＩ１５６０では、提供される詳細レベルは、デバイス上で実行するアプリケーションのソフトウェアコンポーネントのレベルというよりも、むしろ管理対象のコンピューティングデバイスのレベルである。管理対象のコンピューティングデバイスは、ノード１５７４及び１５７６によって示される、例えば、図１Ａのサーバ１０３及び１０９に対応する、アプリケーションサーバである。ＵＩ１５６０は、この例では、トランザクション１（Ｔ１）（アイコン１５６２によって示される）及びトランザクション２（Ｔ２）（アイコン１５６４によって示される）と一般的に称される、着目するトランザクションに焦点を合わせている。アプリケーションサーバ１５７６は、辺１５７５の、辺部分１５６８及び１５７０によってそれぞれ示されているように、Ｔ１及びＴ２に関連して、アプリケーションサーバ１５７４に呼び出される。辺部分１５６８及び１５７０は、前述したように、それらの相対的な幅などによって互いに視覚的に区別される。メトリック表示領域１５６６及び１５７２は、それぞれ辺部分１５６８及び１５７０にリンクされている。もちろん、ＵＩは、ノードの間に２以上の矢印、及び３以上のノードを表示する。実際には、複雑なＵＩは、数日間かかって取得

されるメトリックに基づいて、数百又は数千ものノードを伴って得られる。そのようなＵＩは、ユーザが傾向を視覚的に見つけることを可能にする。例えば、ＵＩが、比較的に応答の遅いトランザクションを示すのに赤などの暖色を使用する場合、ユーザは、赤い辺を見つけ、赤い辺を有するエリアを拡大するようにＵＩを調節して、問題をさらに調査することができる。同様に、ＵＩは、呼び出し回数が相対的に高いエリアをユーザが見つけることを可能にする。

【０１５７】

自動化された報告は、例えばリストの形式で提供される。そのリストは、問題が存在することをメトリックが示す、コンポーネント及び／又は管理対象のコンピューティングデバイスを、それらに関連付けられているトランザクションとともに識別し得る。

【０１５８】

図１６Ａは、少なくとも１つのアプリケーションについて、エージェントがトランザクション別のメトリックを取得するプロセス［処理］の一例を示す。ステップ１６００は、例えば、コンポーネントの開始及び停止ポイントなど、少なくとも１つのアプリケーションの、呼び出されたコンポーネントのシーケンスを表す複数のブランチを含むツリーデータ構造を保守するエージェントを含む。エージェントは、また、複数のブランチの１以上のノードにリンクされた１以上の収集部を保守する。ステップ１６０２は、トランザクションの期間中、少なくとも１つのアプリケーションの、呼び出されたコンポーネントのシーケンスを識別すること、及び、呼び出されたコンポーネントのメトリックを取得することを含む。例えば、これはトランザクションをトレースすることを含む。ステップ１６０４は、ツリーデータ構造内のブランチと一致するブランチを識別することを含む。ステップ１６０６は、１以上の収集部にリンクされている１以上のノードのトランザクション別のメトリックを識別することを含む。ステップ１６０８は、エージェントからマネージャに、トランザクション別のメトリック、一致したブランチの識別子、及び、１以上の収集部の識別子を報告することを含む。述べたように、メトリックは、例えば数分ごとに定期的に報告され、その中で、多くのトランザクションのメトリックが典型的に取得され報告される。ステップ１６１０では、エージェントは、ステップ１６０６でトランザクション別に識別されないメトリックを破棄してマネージャに報告しない、とすることができる。必要に応じて、エージェントは、トランザクション別でないメトリックを報告してもよい。１つのアプローチでは、トランザクション別ではないとき、メトリックの縮減されたセットが報告される。

【０１５９】

図１６Ｂは、図１６Ａのプロセス［処理］に対応しており、エージェントからのトランザクション別のメトリックの報告に基づいてマネージャがユーザインタフェースを提供するプロセス［処理］の一例を示す。ステップ１６２０は、少なくとも１つのアプリケーションの呼び出されたコンポーネントのシーケンスを表すブランチを有するマネージャのツリーデータ構造を保守することを含む。ステップ１６２２は、トランザクション別のメトリック、一致するブランチの識別子、及び、１以上の収集部の識別子、の１以上の報告を受信することを含む。ステップ１６２４は、一致するブランチの識別子に基づいて、トランザクションの呼び出されたコンポーネントのシーケンスを識別することを含む。ステップ１６２６は、１以上の収集部の識別子に基づいて、トランザクション別のメトリックにアクセスすることを含む。ステップ１６２８は、図１４Ａから１５Ｂに示されるように、トランザクション及び関連付けられたトランザクション別のメトリックを示すユーザインタフェース（ＵＩ）を提供することを含む。例えば、トランザクション別のメトリックは、領域１４０２、１４０４、１４１２、１４２２、１５２４、１５２６、１５４０及び１５４６において提供される。ＵＩは、またトランザクション別のメトリックに基づいた、辺部分（例えば、図１５Ａ及び１５Ｂの１５２０、１５２２、１５４２及び１５４４）の視覚的特徴を設定する。

【０１６０】

別の実施形態は、アプリケーションを監視するためにコンピュータに実装される方法を

含む。その方法は、複数のブランチを有するツリーデータ構造を保守すること、各ブランチは、それぞれのトランザクションの期間中、アプリケーションにおいて呼び出されたコンポーネントの各シーケンスを表す；対象トランザクションの期間中、エージェントを用いて、そのアプリケーションにおいて呼び出されたコンポーネントのシーケンス、及び、その対象トランザクションの呼び出された複数のコンポーネントの夫々の開始及び終了ポイントの時刻を識別すること；その複数のブランチのうちの一つを、対象トランザクションの呼び出されたコンポーネントのシーケンスに対して一致するブランチとして識別すること； 識別することに対応して、開始及び終了ポイントの時刻と、一致するブランチの識別子をエージェントからマネージャに報告すること、を含む。

【０１６１】

一例では、一致するブランチは、対象トランザクションの呼び出されたコンポーネントの開始及び終了ポイントを表すノードを有し；対象トランザクションの呼び出されたコンポーネントの終了ポイントを表すノードの１つが、一致するブランチの最終ノードである；及び、一致するブランチの識別子は、一致するブランチの最終ノードの識別子を有し、一致するブランチの最終ノードの識別子は、その一致するブランチを一意的に識別する。

【０１６２】

一例では、一致するブランチは、対象トランザクションの呼び出されたコンポーネントの開始及び終了ポイントを表すノードを有する；及び、対象トランザクションの呼び出されたコンポーネントに関連付けられた静的データは、対象トランザクションの呼び出されたコンポーネントの開始及び終了ポイントを表すノードにインデックス付けされている。

【０１６３】

いくつかの例では、静的データは、少なくとも１つのクラス及びメソッドの名称を有する；静的データは、トレースされたクラスが展開されるアーカイブファイルの名称を有する；又は、静的データは、少なくとも１つのテキスト文字列、コンポーネントの種類及びポート番号を有する。

【０１６４】

一例では、対象トランザクションの期間中、エージェントは、開始及び終了ポイントの時刻以外の動的データを取得し、開始及び終了ポイントの時刻以外の動的データは、対象トランザクション以前に知られることはない；及び、報告は、開始及び終了ポイントの時刻以外の動的データをマネージャに報告することを含み、動的データは、対象トランザクションの呼び出されたコンポーネントの開始及び終了ポイントを表す一致ブランチ内のノードにインデックス付けされ、動的データはメソッドに渡されたパラメータの値を含む。

【０１６５】

一例では、ツリーデータ構造は、それぞれのトランザクションの期間中、アプリケーションにおいて呼び出されたコンポーネントのそれぞれのシーケンスをトレースすることによって取得される。

【０１６６】

一例では、別の対象トランザクションの期間中、アプリケーションにおいて呼び出されたコンポーネントの他のシーケンスを識別するためにエージェントを使用すること；呼び出されたコンポーネントの上記別のシーケンスがツリーデータ構造内に表されているか否かを判定すること；呼び出されたコンポーネントの上記別のシーケンスがツリーデータ構造内に表されていない場合に対象トランザクションの呼び出されたコンポーネントの上記別のシーケンスを表すようにツリーデータ構造を更新すること；更新することは：（ｉ）別の対象トランザクションの１以上の呼び出されたコンポーネントの開始及び終了ポイントを表すノードを提供すること、及び、（ｉｉ）別の対象トランザクションの１以上の呼び出されたコンポーネントに関連付けられた静的データをノードにインデックス付けすること。

【０１６７】

一例では、エージェントは、ツリーデータ構造の更新をマネージャに報告し、その報告に応答して、マネージャが対応するツリーデータ構造を更新する。

【０１６８】
別の実施形態は、コンピュータシステムを含み、コンピュータシステムが：ソフトウェア命令を格納する記憶装置；ソフトウェア命令が記憶装置から作業メモリにロードされる、作業メモリ；及び、ソフトウェア命令を実行するためのプロセッサを有し、そのソフトウエア命令は、実行時に：複数のブランチを有するツリーデータ構造を保守し、各ブランチは、それぞれのトランザクションの期間中、アプリケーションにおいて呼び出されたコンポーネントのそれぞれのシーケンスを表す；対象トランザクションの期間中、エージェントを使って、アプリケーションにおいて呼び出されたコンポーネントのシーケンス、及び、対象トランザクションの呼び出された複数のコンポーネントの各々の開始と終了ポイントの時刻を識別する；複数のブランチのうちの１つを、対象トランザクションの呼び出されたコンポーネントのシーケンスに一致するブランチとして識別し；その識別に応答して、エージェントからマネージャへ、一致するブランチの識別子と開始及び終了ポイントの時刻を報告する。

【０１６９】
別の実施形態は、アプリケーションを管理する方法であってコンピュータに実装される方法を含む。その方法は：マネージャのツリーデータ構造を保守すること、マネージャのツリーデータ構造は、複数のブランチを有し、各ブランチは、それぞれのトランザクションの期間中、アプリケーションにおいて呼び出されたコンポーネントのそれぞれのシーケンスを表し、呼び出されたコンポーネントの各それぞれのシーケンスは、各それぞれのブランチにおいて、呼び出されたコンポーネントのそれぞれのシーケンスの呼び出されたコンポーネントの開始及び終了ポイントを表すノードのそれぞれのシーケンスによって表される；マネージャにおいて、アプリケーションの１つのインスタンスの呼び出されたコンポーネントのそれぞれのシーケンスのうちの１つを検出するエージェントから受信すること：（ｉ）それぞれのブランチの１つの識別子は、呼び出されたコンポーネントのそれぞれのシーケンスのうちの１つを表し、及び（ｉｉ）呼び出されたコンポーネントのそれぞれのシーケンスの１つの、動的データである、動的データは、呼び出されたコンポーネントのそれぞれのシーケンスの１つの、呼び出されたコンポーネントの開始及び終了ポイントの時刻を有する；識別に応答して、呼び出されたコンポーネントのそれぞれのシーケンスの１つに関連付けられた静的データを調べるためにマネージャのツリーデータ構造を使用すること；及び、トランザクショントレースを示すユーザインタフェースを提供すること、トランザクショントレースは、呼び出されたコンポーネントのそれぞれのシーケンスの１つの、呼び出されたコンポーネントの開始及び終了ポイントの時刻を示し、及びトランザクショントレースは、静的データに基づいて註を加えられる、を有する。

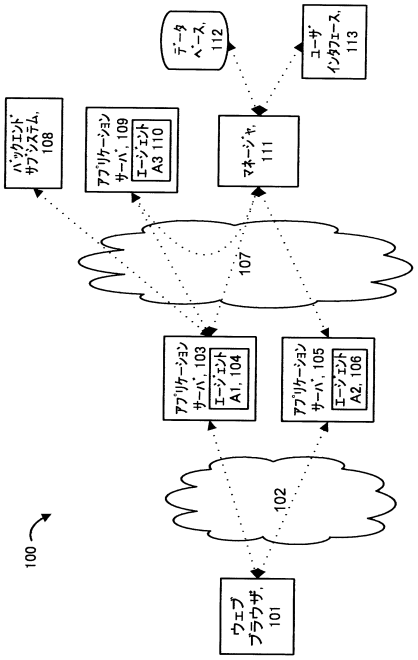
【０１７０】
一例では、エージェントは、少なくとも部分的には、マネージャのツリーデータ構造に対応するツリーデータ構造を保守し；実行されるその方法は：エージェントから、エージェントのツリーデータ構造への更新についての情報を受信すること、エージェントのツリーデータ構造は、少なくとも部分的には、マネージャのツリーデータ構造に対応する；及び、更新についての情報に応答して、マネージャのツリーデータ構造を更新することをさらに含む。

【０１７１】
一例では、少なくとも１つの他のエージェントは、アプリケーションの他のインスタンスのために呼び出されたコンポーネントのそれぞれのシーケンスを検出し；他のエージェントは、ツリーデータ構造を保守し、他のエージェントのツリーデータ構造は、少なくとも部分的には、エージェントのツリーデータ構造に応答する；及び、実行されるその方法は：ツリーデータ構造のさらなるインスタンスを更新するときに使用するために他のエージェントへの更新についての情報を通信すること、をさらに有する。

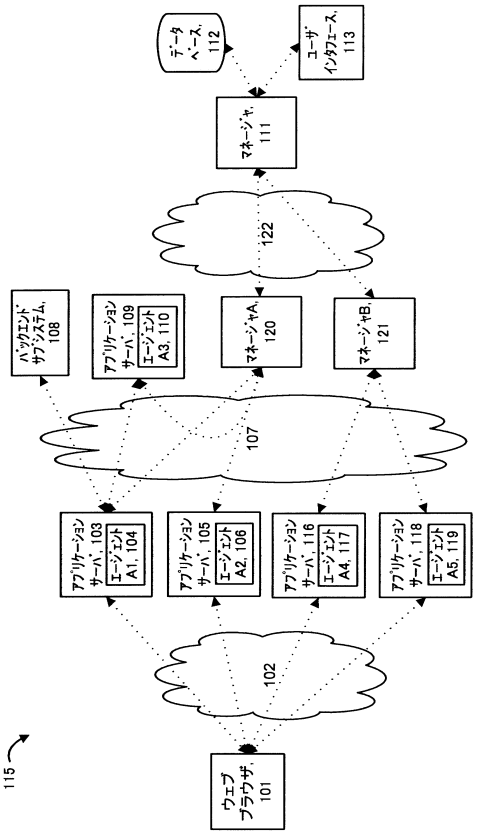
【０１７２】
一例では、更新についての情報は、更新に関連付けられた１以上の呼び出されたコンポーネントの開始及び終了ポイントを表すノードを識別する。

【０１７３】
一例では、更新は、静的データを更新に関連付けられた１以上の呼び出されたコンポーネントの開始及び終了ポイントを表すノードにインデックス付けすることを有する。
【０１７４】
以上、本発明の具体例を詳細に説明したが、これらは例示に過ぎず、本発明の範囲を限定するものではない。上記した技術には、以上に例示した具体例を様々に変形、変更したものが含まれる。上記した実施形態は、本発明の原理をベストに説明するために選定されたものであり、その実用に際しては、本発明が最適にその有用性を発揮するように、その特定用途に適するように当業者が様々に変形し得る。発明の範囲は、ここに添付した特許請求の範囲によって定められることを意図している。

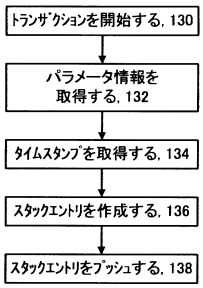
【図１Ａ】



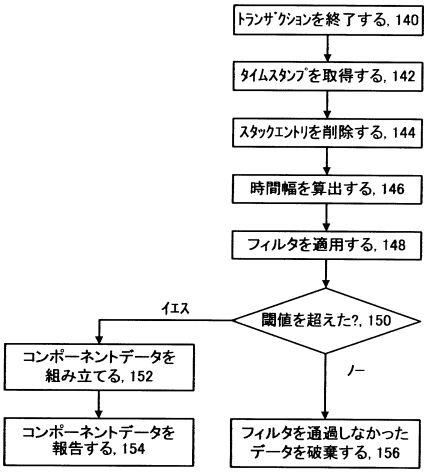

【図１Ｂ】



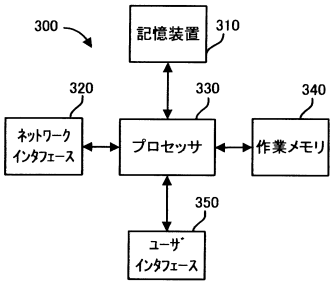
【図２Ａ】



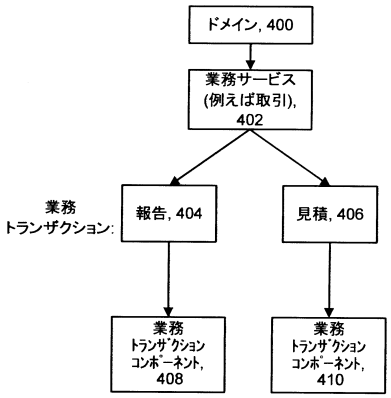

【図２Ｂ】



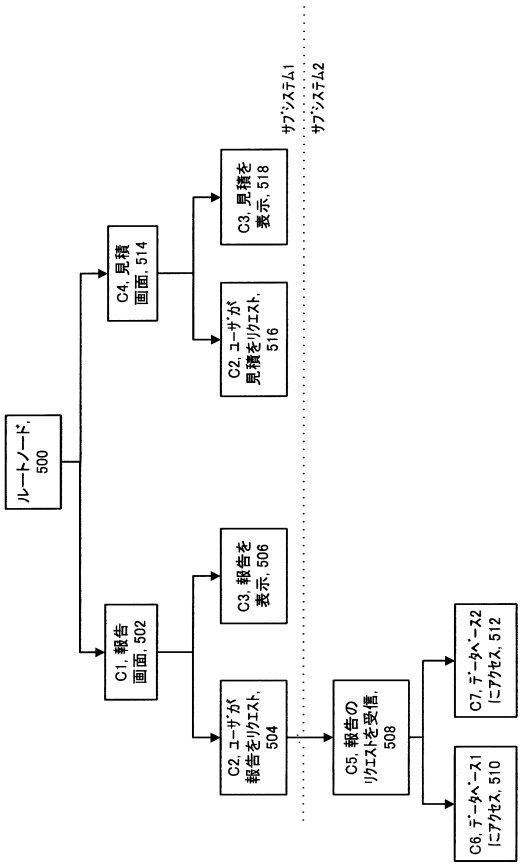

【図３】



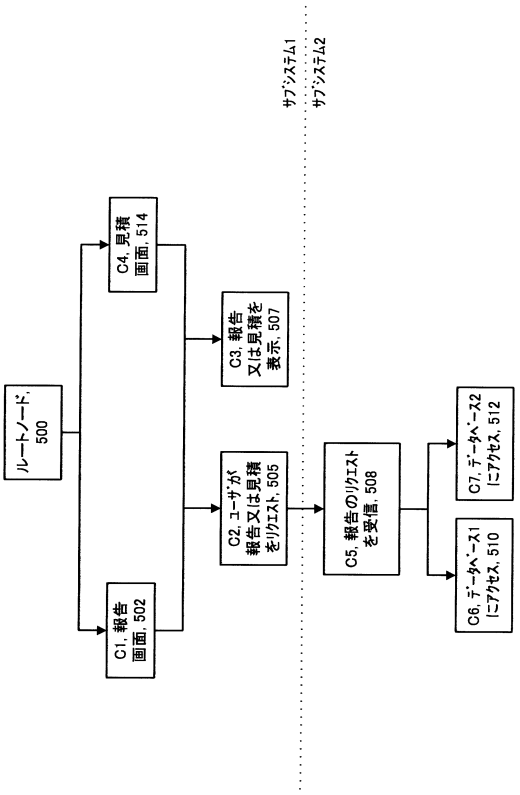

【図４】



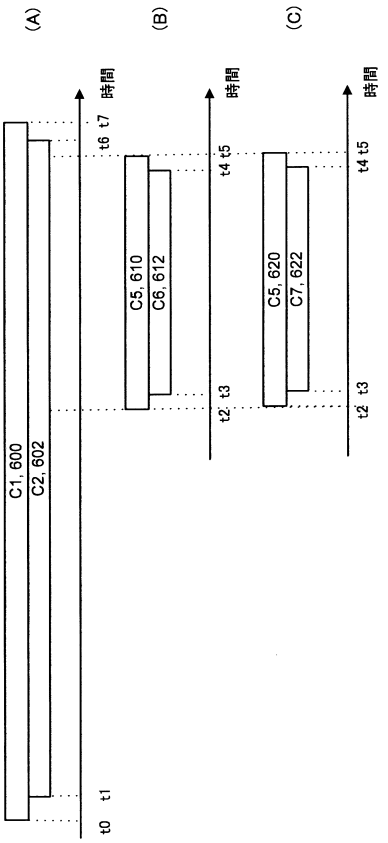

【図５Ａ】



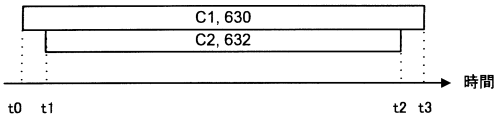

【図５Ｂ】



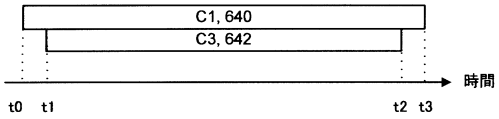
【図６】



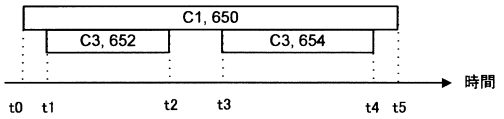

【図６Ｄ】



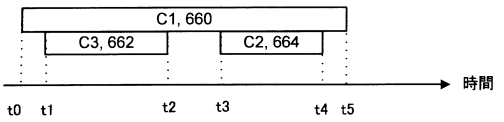

【図６Ｅ】



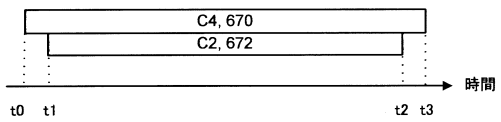

【図６Ｆ】



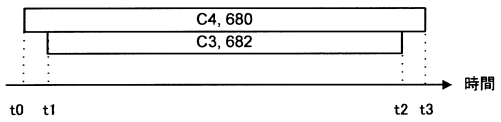

【図６Ｇ】



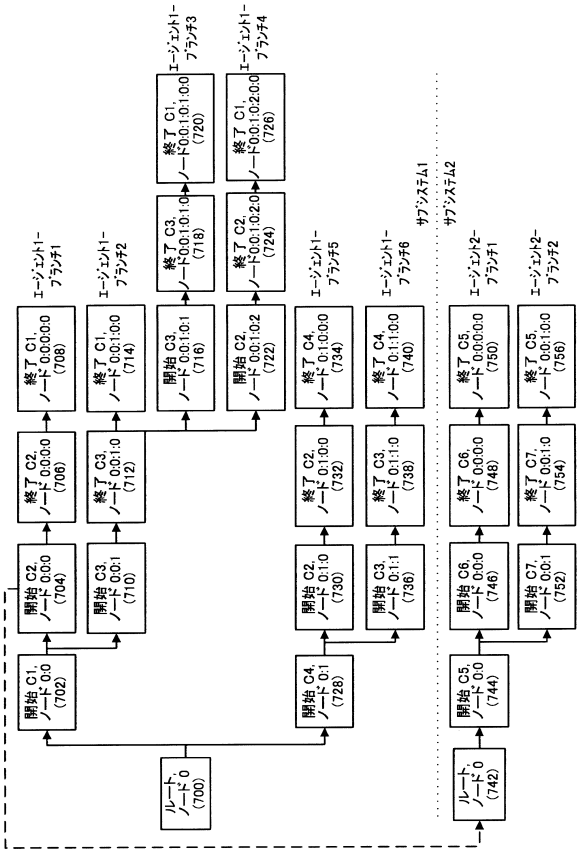

【図６Ｈ】



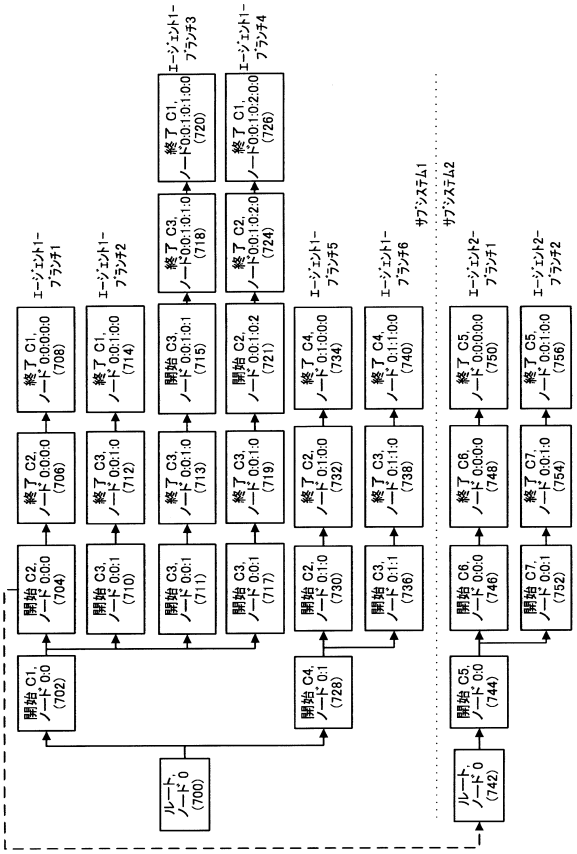

【図６Ｉ】



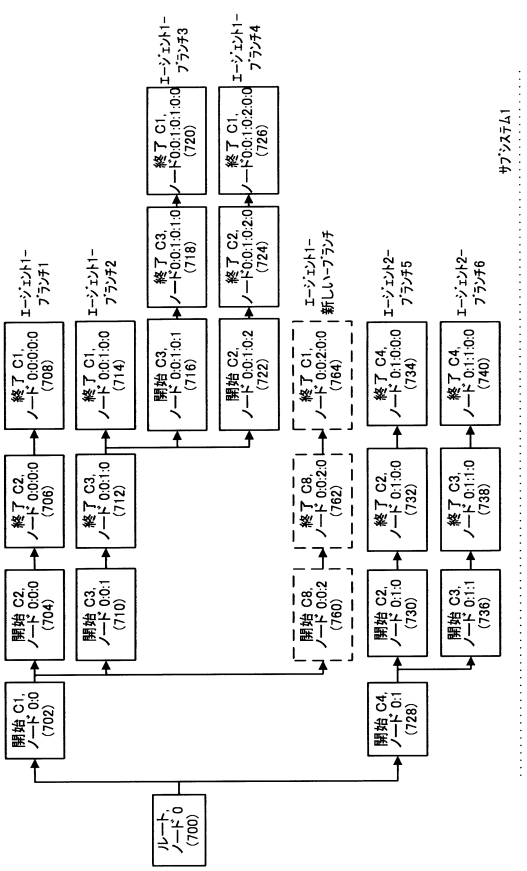

【図７Ａ１】



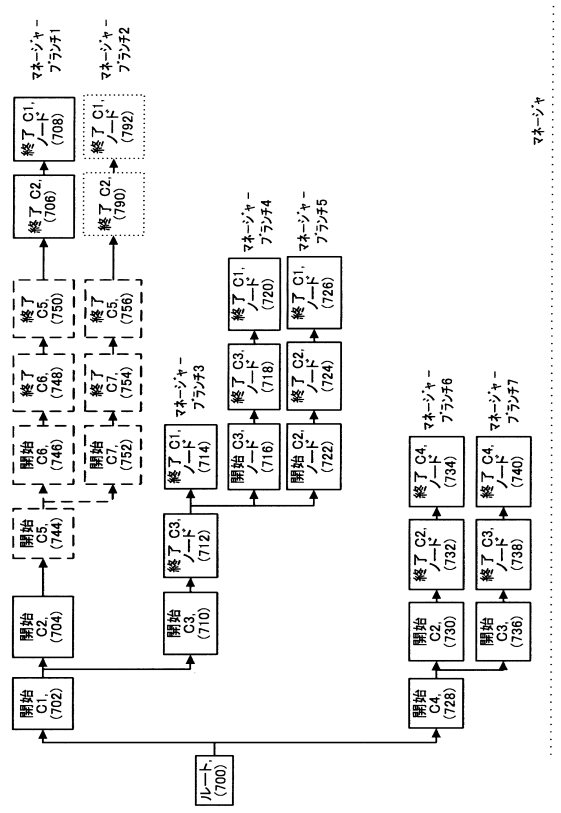
【図７Ａ２】




【図７Ｂ】




【図７Ｃ１】

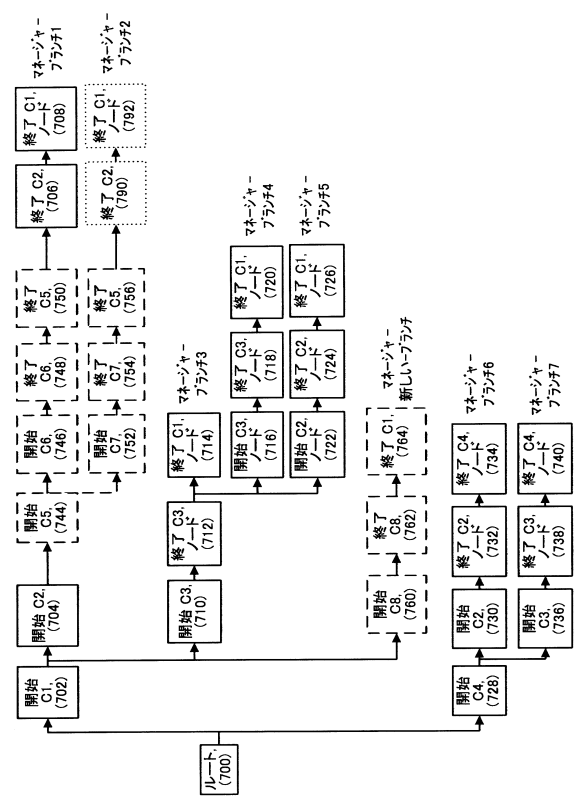



【図７Ｃ２】

エージェント－マネージャ 最終ノードの対応

| ノード: | マネージャ ブランチの 最終ノード: | エージェント1 ブランチの 最終ノード: | エージェント2 ブランチの 最終ノード: |
|---|---|---|---|
| 708 | 0:0:0:0:0:0:0:0:0 | 0:0:0:0:0 | |
| 792 | 0:0:0:0:1:0:0:0:0 | 0:0:0:0:0 | |
| 750 | 0:0:0:0:0:0:0 | | 0:0:0:0:0 |
| 756 | 0:0:0:0:1:0:0 | | 0:0:0:0:0 |
| 714 | 0:0:1:0:0 | 0:0:1:0:0 | |
| 720 | 0:0:1:0:1:0:0 | 0:0:1:0:1:0:0 | |
| 726 | 0:0:1:0:2:0:0 | 0:0:1:0:2:0:0 | |
| 734 | 0:1:0:0:0 | 0:1:0:0:0 | |
| 740 | 0:1:1:0:0 | 0:1:1:0:0 | |

【図７Ｄ】




【図８Ａ１】

ブランチ定義－サブシステム1

| ブランチの最終ノード: | ブランチ: | トランザクション: | コンポーネント呼び出し: |
|---|---|---|---|
| 0:0:0:0:0 | エージェント1-ブランチ1 | エージェント1-T1 | 開始 C1, 開始 C2, 終了 C2, 終了 C1 |
| 0:0:1:0:0 | エージェント1-ブランチ2 | エージェント1-T2 | 開始 C1, 開始 C3, 終了 C3, 終了 C1 |
| 0:0:1:0:1:0:0 | エージェント1-ブランチ3 | エージェント1-T3 | 開始 C1, 開始 C3, 終了 C3, 開始 C3, 終了 C3, 終了 C1 |
| 0:0:1:0:2:0:0 | エージェント1-ブランチ4 | エージェント1-T4 | 開始 C1, 開始 C3, 終了 C3, 開始 C2, 終了 C2, 終了 C1 |
| 0:1:0:0:0 | エージェント1-ブランチ5 | エージェント1-T5 | 開始 C4, 開始 C2, 終了 C2, 終了 C4 |
| 0:1:1:0:0 | エージェント1-ブランチ6 | エージェント1-T6 | 開始 C4, 開始 C3, 終了 C3, 終了 C4 |

【図８Ａ２】

ブランチ定義－サブシステム2

| ブランチの最終ノード: | ブランチ: | トランザクション: | コンポーネント呼び出し: |
|---|---|---|---|
| 0:0:0:0:0 | エージェント2-ブランチ1 | エージェント2-T1 | 開始 C5, 開始 C6, 終了 C6, 終了 C5 |
| 0:0:1:0:0 | エージェント2-ブランチ2 | エージェント2-T2 | 開始 C5, 開始 C7, 終了 C7, 終了 C5 |

【図８Ａ３】

ブランチ定義－マネージャ

| ブランチの最終ノード: | ブランチ: | トランザクション: | コンポーネント呼び出し: |
|---|---|---|---|
| 0:0:0:0:0:0:0:0:0 | マネージャ－ブランチ1 | マネージャ－T1 | 開始 C1, 開始 C2, 開始 C5, 開始 C6, 終了 C6, 終了 C5, 終了 C2, 終了 C1 |
| 0:0:0:0:1:0:0:0:0 | マネージャ－ブランチ2 | マネージャ－T2 | 開始 C1, 開始 C2, 開始 C5, 開始 C7, 終了 C7, 終了 C5, 終了 C2, 終了 C1 |
| 0:0:1:0:0 | マネージャ－ブランチ3 | マネージャ－T3 | 開始 C1, 開始 C3, 終了 C3, 終了 C1 |
| 0:0:1:0:1:0:0 | マネージャ－ブランチ4 | マネージャ－T4 | 開始 C1, 開始 C3, 終了 C3, 開始 C3, 終了 C3, 終了 C1 |
| 0:0:1:0:2:0:0 | マネージャ－ブランチ5 | マネージャ－T5 | 開始 C1, 開始 C3, 終了 C3, 開始 C2, 終了 C2, 終了 C1 |
| 0:1:0:0:0 | マネージャ－ブランチ6 | マネージャ－T6 | 開始 C4, 開始 C2, 終了 C2, 終了 C4 |
| 0:1:1:0:0 | マネージャ－ブランチ7 | マネージャ－T7 | 開始 C4, 開始 C3, 終了 C3, 終了 C4 |

【図８Ｂ１】

静的データの参照−サブシステム1

| ノード: | 静的データ: |
|---|---|
| 0:0 | static_data_C1 |
| 0:0:0 | static_data_C2 |
| 0:0:0:0 | static_data_C2 |
| 0:0:0:0:0 | static_data_C1 |
| | |
| 0:0:1 | static_data_C3 |
| 0:0:1:0 | static_data_C3 |
| 0:0:1:0:0 | static_data_C1 |
| | |
| 0:0:1:0:1 | static_data_C3 |
| 0:0:1:0:1:0 | static_data_C3 |
| 0:0:1:0:1:0:0 | static_data_C1 |
| | |
| 0:0:1:0:2 | static_data_C2 |
| 0:0:1:0:2:0 | static_data_C2 |
| 0:0:1:0:2:0:0 | static_data_C1 |
| | |
| 0:1 | static_data_C4 |
| 0:1:0 | static_data_C2 |
| 0:1:0:0 | static_data_C2 |
| 0:1:0:0:0 | static_data_C4 |
| | |
| 0:1:1 | static_data_C3 |
| 0:1:1:0 | static_data_C3 |
| 0:1:1:0:0 | static_data_C4 |

【図８Ｂ２】

静的データの参照−サブシステム2

| ノード: | 静的データ: |
|---|---|
| 0:0 | static_data_C5 |
| 0:0:0 | static_data_C6 |
| 0:0:0:0 | static_data_C6 |
| 0:0:0:0:0 | static_data_C5 |
| | |
| 0:0:1 | static_data_C7 |
| 0:0:1:0 | static_data_C7 |
| 0:0:1:0:0 | static_data_C5 |

【図８Ｂ３】

静的データの参照−更新のためのサブシステム1:

| | |
|---|---|
| 0:0:2 | static_data_C8 |
| 0:0:2:0 | static_data_C8 |
| 0:0:2:0:0 | static_data_C1 |

【図８Ｂ４】

静的データの参照−マネージャ

| ノード: | 静的データ: | ノード: | 静的データ: |
|---|---|---|---|
| 0:0 | static_data_C1 | 0:0:1:0:1 | static_data_C3 |
| 0:0:0 | static_data_C2 | 0:0:1:0:1:0 | static_data_C3 |
| 0:0:0:0 | static_data_C5 | 0:0:1:0:1:0:0 | static_data_C1 |
| 0:0:0:0:0 | static_data_C6 | | |
| 0:0:0:0:0:0 | static_data_C6 | 0:0:1:0:2 | static_data_C2 |
| 0:0:0:0:0:0:0 | static_data_C5 | 0:0:1:0:2:0 | static_data_C2 |
| 0:0:0:0:0:0:0:0 | static_data_C2 | 0:0:1:0:2:0:0 | static_data_C1 |
| 0:0:0:0:0:0:0:0:0 | static_data_C1 | | |
| | | 0:1 | static_data_C4 |
| 0:0:0:0:1 | static_data_C7 | 0:1:0 | static_data_C2 |
| 0:0:0:0:1:0 | static_data_C7 | 0:1:0:0 | static_data_C2 |
| 0:0:0:0:1:0:0 | static_data_C5 | 0:1:0:0:0 | static_data_C4 |
| 0:0:0:0:1:0:0:0 | static_data_C2 | | |
| 0:0:0:0:1:0:0:0:0 | static_data_C1 | 0:1:1 | static_data_C3 |
| | | 0:1:1:0 | static_data_C3 |
| 0:0:1 | static_data_C3 | 0:1:1:0:0 | static_data_C4 |
| 0:0:1:0 | static_data_C3 | | |
| 0:0:1:0:0 | static_data_C1 | | |

【図８Ｂ５】

静的データの参照−更新のためのマネージャ:

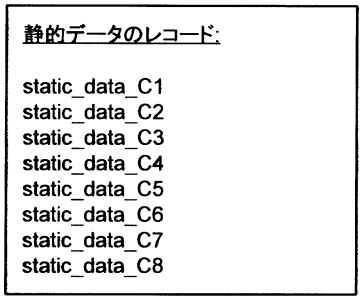
| ノード: | 静的データ: |
|---|---|
| 0:0:2 | static_data_C8 |
| 0:0:2:0 | static_data_C8 |
| 0:0:2:0:0 | static_data_C1 |

【図８Ｃ】

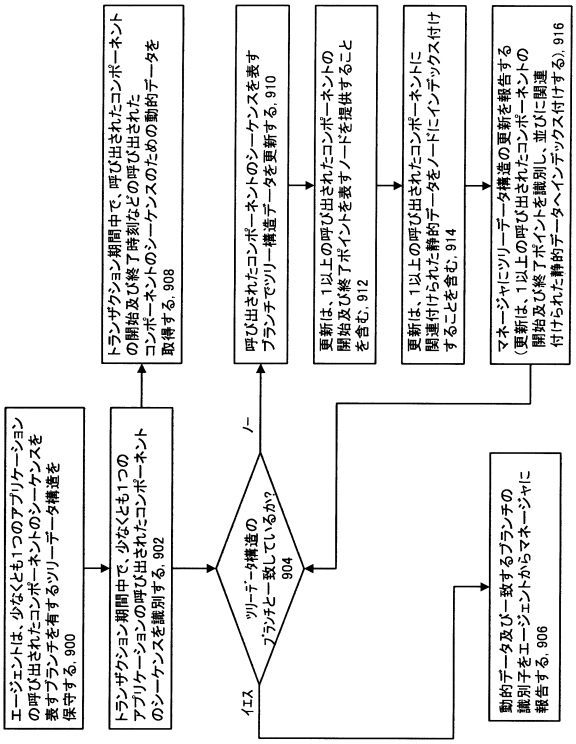
例示のトランザクション−サブシステム1のための動的データ

| ノード: | コンポーネント: | 開始／終了: | 時間: | 他の動的データ: |
|---|---|---|---|---|
| 0:0 | C1 | 開始 | t1 | dynamic_data_1 (例: 呼出し中で渡したパラメータ1) |
| 0:0:0 | C2 | 開始 | t2 | dynamic_data_2 (例: 呼出し中で渡したパラメータ2) |
| 0:0:0:0 | C2 | 終了 | t3 | dynamic_data_3 (例: 戻り中で渡したパラメータ3) |
| 0:0:0:0:0 | C1 | 終了 | t4 | dynamic_data_4 (例: 戻り中で渡したパラメータ4) |

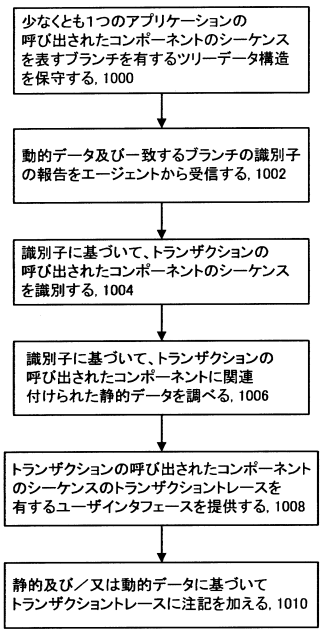
【図８Ｄ】



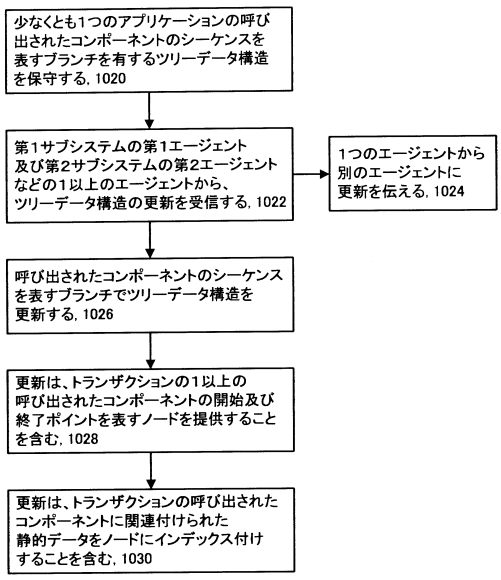

【図９】



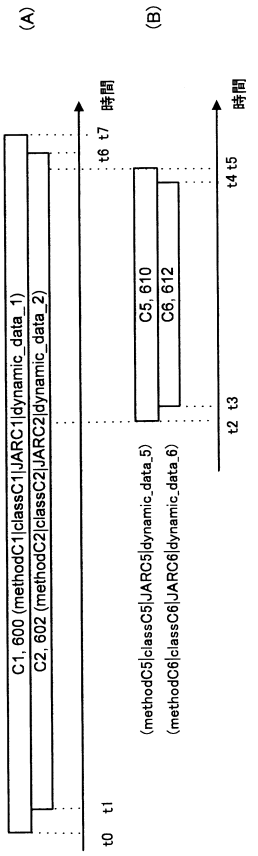

【図１０Ａ】



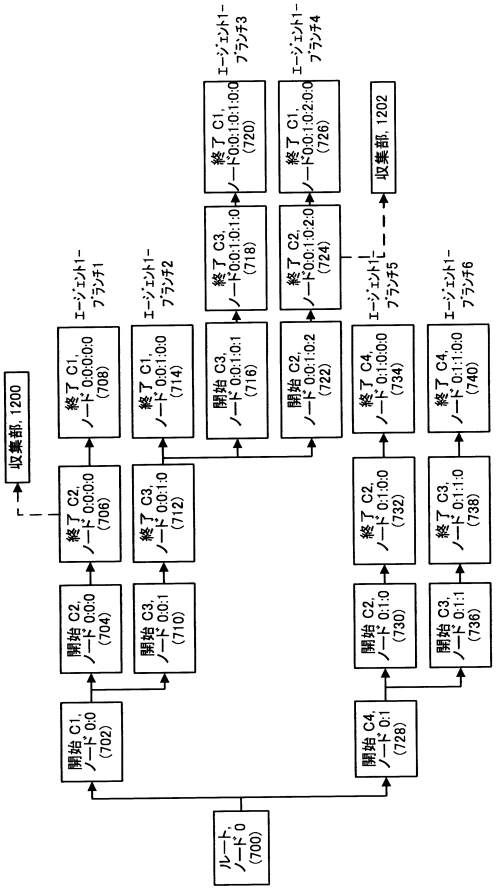

【図１０Ｂ】



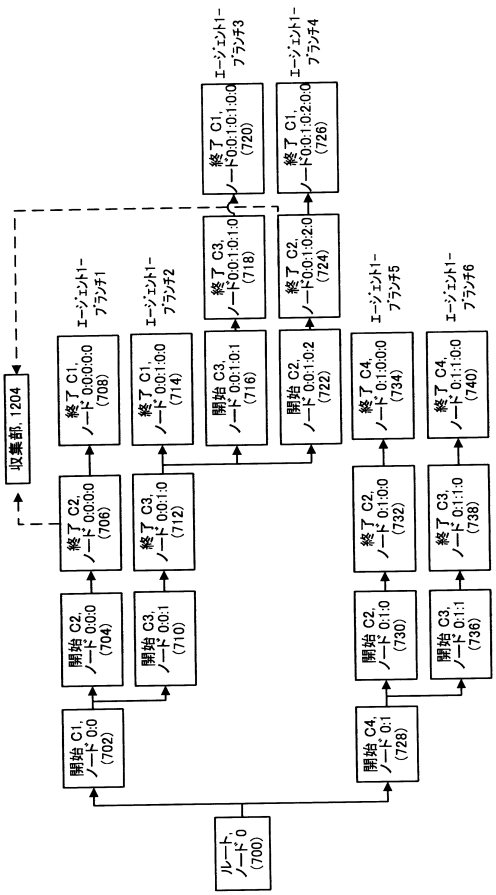
【図１１】



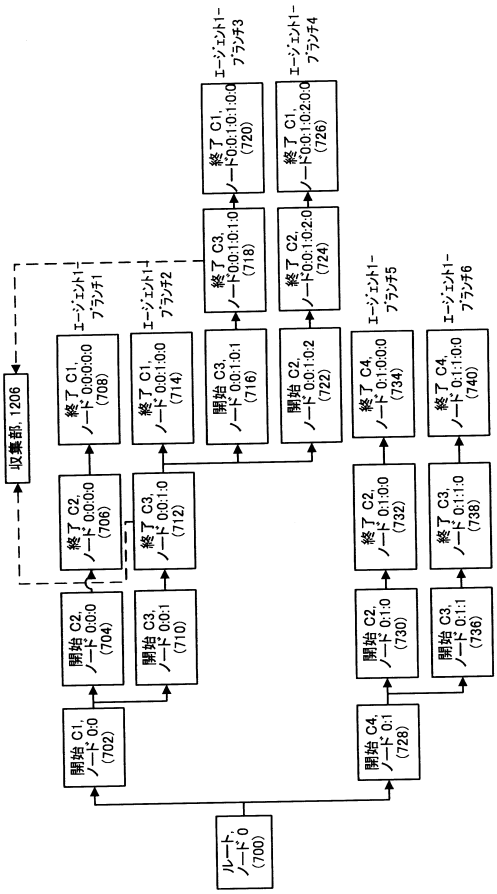

【図１２Ａ】




【図１２Ｂ】




【図１２Ｃ】



【図１３Ａ】

| 収集部: | ノード: | コンポーネント: | トランザクション: |
|---|---|---|---|
| 1200 | 0:0:0:0 | C2 | エージェント1-T1 |
| 1202 | 0:0:1:0:2:0 | C2 | エージェント1-T4 |

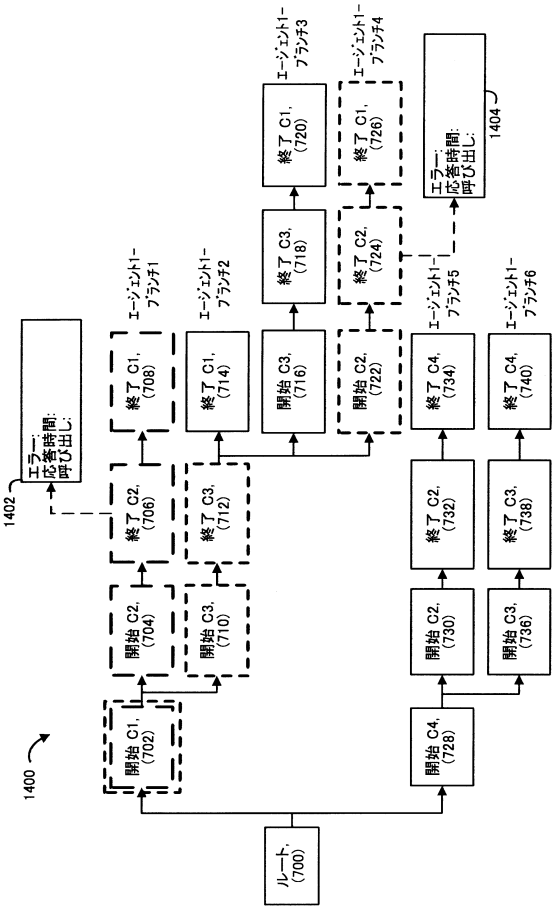
【図１３Ｂ】

| 収集部: | ノード: | コンポーネント: | トランザクション: |
|---|---|---|---|
| 1204 | 0:0:0:0 | C2 | エージェント1-T1 |
| 1204 | 0:0:1:0:2:0 | C2 | エージェント1-T4 |

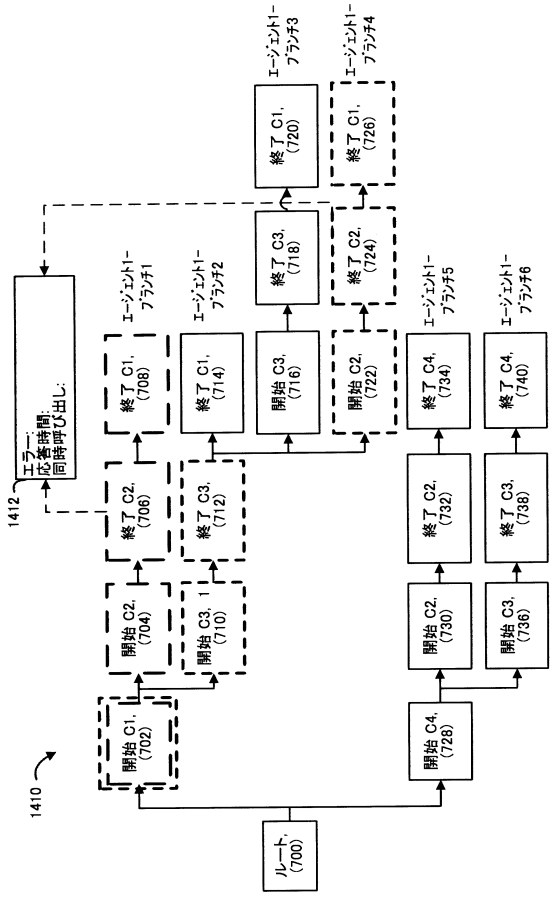
【図１３Ｃ】

| 収集部: | ノード: | コンポーネント: | トランザクション: |
|---|---|---|---|
| 1206 | 0:0:1:0 | C3 | エージェント1-T3 |
| 1206 | 0:0:1:0:1:0 | C3 | エージェント1-T3 |

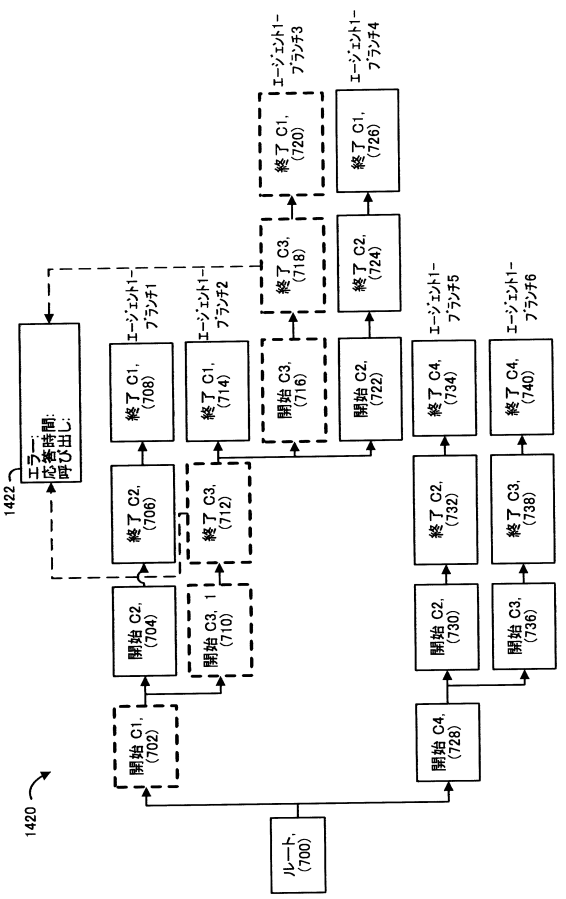
【図１４Ａ】



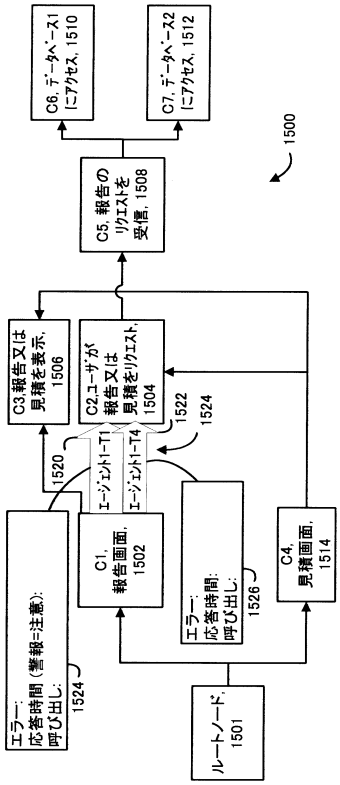

【図１４Ｂ】



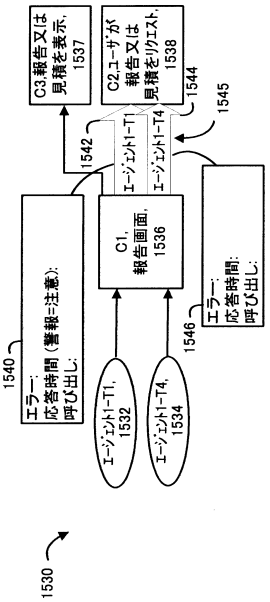

【図１４Ｃ】



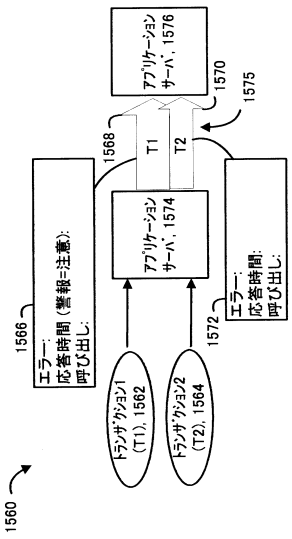
【図１５Ａ】



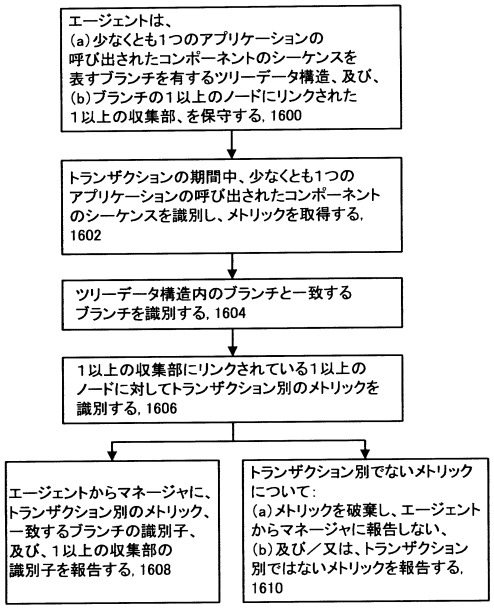

【図１５Ｂ】



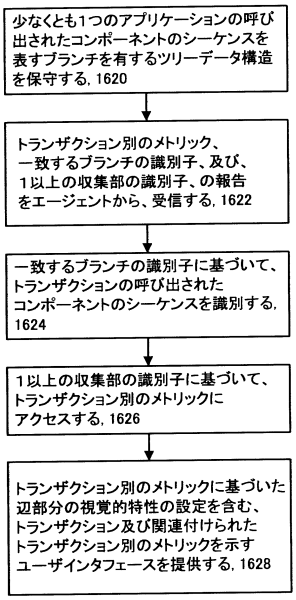

【図１５Ｃ】




【図１６Ａ】



【図１６Ｂ】


少なくとも1つのアプリケーションの呼び出されたコンポーネントのシーケンスを表すブランチを有するツリーデータ構造を保守する, 1620
トランザクション別のメトリック、一致するブランチの識別子、及び、1以上の収集部の識別子、の報告をエージェントから、受信する, 1622
一致するブランチの識別子に基づいて、トランザクションの呼び出されたコンポーネントのシーケンスを識別する, 1624
1以上の収集部の識別子に基づいて、トランザクション別のメトリックにアクセスする, 1626
トランザクション別のメトリックに基づいた辺部分の視覚的特性の設定を含む、トランザクション及び関連付けられたトランザクション別のメトリックを示すユーザインタフェースを提供する, 1628

フロントページの続き

(72)発明者　ウマンスキー，ウラジミール
アメリカ合衆国　１１７４９　ニューヨーク州　アイランディア　ワン　シーエー　プラザ　シーエー，インク内

審査官　多胡　滋

(56)参考文献　特開２０１０－１１７７５７（ＪＰ，Ａ）
特開２００６－１５７３１３（ＪＰ，Ａ）
特開２００５－３０２０５７（ＪＰ，Ａ）
特開２００１－０６０１６３（ＪＰ，Ａ）
米国特許第０７２９３２５９（ＵＳ，Ｂ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１１／２８
Ｇ０６Ｆ　１１／３２
Ｇ０６Ｆ　１１／３４