

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2015-194886
(P2015-194886A)

(43) 公開日 平成27年11月5日(2015.11.5)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/52 (2006.01)	G06F 9/46 472A	5B042
G06F 11/30 (2006.01)	G06F 11/30 305G	
G06F 13/00 (2006.01)	G06F 13/00 358F	

審査請求 未請求 請求項の数 7 O L (全 40 頁)

(21) 出願番号 特願2014-72194 (P2014-72194)
(22) 出願日 平成26年3月31日 (2014.3.31)

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1番1号
(74) 代理人 100074099
弁理士 大菅 義之
(74) 代理人 100133570
弁理士 ▲徳▼永 民雄
(72) 発明者 小田嶋 成幸
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
(72) 発明者 岡林 美和
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
Fターム(参考) 5B042 GA12 JJ04 JJ14 JJ30 KK15
KK20 MC09 MC35

(54) 【発明の名称】 分散データ処理装置、分散データ処理方法および分散データ処理プログラム

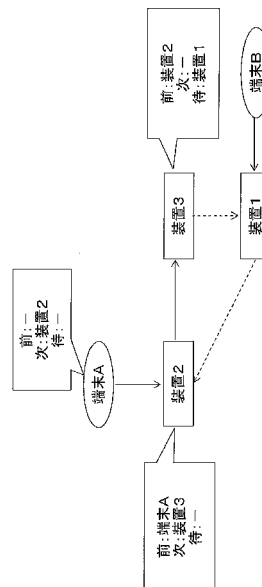
(57) 【要約】 (修正有)

【課題】 端末と被制御装置が通信できないことがある場合でも、デッドロックを検出する。

【解決手段】 各々が複数の被制御装置の一つに対応する複数のノードが、第1の端末が制御権を有する第1のノードから第1の端末が制御権を有する第2のノードに向かって伸びる第1のリンクと、第2の端末が制御権を有する第3のノードから第2の端末が制御権を付与されることを待っている第4のノードに向かって伸びる第2のリンクとを用いて結合されることで構成される資源循環グラフにおいて、第1のリンクおよび第2のリンクに沿ってノード間を転送されるように、複数の被制御装置の間を転送されるデッドロック検出メッセージを受信するネットワーク受信部と、デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定するデッドロック判定部とを含む。

【選択図】 図12

実施形態における資源循環グラフの例を示す図



【特許請求の範囲】**【請求項 1】**

複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御される複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つである分散データ処理装置であって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第 1 の端末が制御権を有する第 1 の被制御装置に対応する第 1 のノードから前記第 1 の端末が制御権を有する第 2 の被制御装置に対応する第 2 のノードに向かって伸びる第 1 のリンクと、第 2 の端末が制御権を有する第 3 の被制御装置に対応する第 3 のノードから前記第 2 の端末が制御権を付与されることを待っている第 4 の被制御装置に対応する第 4 のノードに向かって伸びる第 2 のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第 1 のリンクおよび前記第 2 のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信するネットワーク受信部と、前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定するデッドロック判定部と、を含む分散データ処理装置。

10

【請求項 2】

前記デッドロック判定部は、前記デッドロック検出メッセージが複数回、前記複数の被制御装置のうちの一つを訪問した場合に、デッドロックが発生していると判定する、請求項 1 に記載の分散データ処理装置。

20

【請求項 3】

さらに、

自身が前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の別の一つを前ノード、自身が前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを次ノード、自身が前記第 2 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 2 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを待ノードとして、前記前ノード、前記次ノード、および前記待ノードに関する情報を記憶する記憶部を含み、

30

前記資源循環グラフの前記ノードの各々は、前記記憶部（図 8、210）に格納されている前記前ノード、前記次ノード、および前記待ノードに関する情報が付与されている、請求項 1 または 2 に記載の分散データ処理装置。

【請求項 4】

さらに、

前記デッドロック判定部が、前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送るデッドロック解消部を含む、請求項 1 乃至 3 のいずれか一項に記載の分散データ処理装置。

40

【請求項 5】

前記デッドロック検出メッセージは、前記資源循環グラフに含まれる前記複数の被制御装置の優先度に関する情報を含み、

さらに、

前記デッドロック判定部が、前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記デッドロック検出メッセージに含まれる前記優先度に基づいて、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送るデッドロック解消部を含む、請求項 1 乃至 3 のいずれか一項に記載の分散データ処理装置。

【請求項 6】

50

コンピュータとして機能する複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御されるコンピュータとして機能する複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つで実行される分散データ処理方法であって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第1の端末が制御権を有する第1の被制御装置に対応する第1のノードから前記第1の端末が制御権を有する第2の被制御装置に対応する第2のノードに向かって伸びる第1のリンクと、第2の端末が制御権を有する第3の被制御装置に対応する第3のノードから前記第2の端末が制御権を付与されることを待っている第4の被制御装置に対応する第4のノードに向かって伸びる第2のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第1のリンクおよび前記第2のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信することと、

前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定することと、
を含む分散データ処理方法。

【請求項7】

コンピュータとして機能する複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御されるコンピュータとして機能する複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つで実行される分散データ処理プログラムであって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第1の端末が制御権を有する第1の被制御装置に対応する第1のノードから前記第1の端末が制御権を有する第2の被制御装置に対応する第2のノードに向かって伸びる第1のリンクと、第2の端末が制御権を有する第3の被制御装置に対応する第3のノードから前記第2の端末が制御権を付与されることを待っている第4の被制御装置に対応する第4のノードに向かって伸びる第2のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第1のリンクおよび前記第2のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信し、

前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定する、
処理をコンピュータに実行させることを特徴とする分散データ処理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、分散データ処理装置、分散データ処理方法および分散データ処理プログラムに関するものである。

【背景技術】

【0002】

ネットワーク機器の普及とモバイル端末の普及により、モバイル端末からネットワーク制御可能な装置を制御する技術が注目されている。例えば、ネットワークにより電球等のデバイスのスイッチのオン、オフが制御可能な電子基板や、ネットワーク制御可能なプロジェクト等が普及しつつある。

【0003】

そしてこれらの機器は、一般に「分散システム」と呼ばれる、モバイル端末からネットワーク制御可能な装置を制御し得るシステムで利用され得る。分散システムでは一般に、被制御装置は状態を持ち、モバイル端末から発せられるメッセージにより、被制御装置の持つ状態を変更する構成を取る。また、複数端末により意図しない状態の変更を防ぐため、被制御装置は制御元のモバイル端末に関する情報を管理し、更にモバイル端末からの制御権要求に基づき、モバイル端末への被制御装置の制御権の発行、または制御権待ちの通知を行う。これらの条件により、モバイル端末は制御先のデバイスを常時監視する必要が

10

20

30

40

50

なく、モバイル端末のネットワーク負荷を低減することが可能である。また、システム上に多くのモバイル端末、被制御装置が存在することが想定されるため、全ての制御、非制御関係を持つことは非現実的であり、モバイル端末、被制御装置共に、自身が制御する、又は制御されるモバイル端末、被制御装置の情報のみを保有する形態を取り得る。

【 0 0 0 4 】

そのような分散システムにおいて、データベースアクセスなどを行う複数のリソースマネージャーにまたがって発生するトランザクション（グローバルトランザクション）の間でのデッドロックを検出可能とする分散システムが知られている（たとえば、特許文献1）。このような分散システムでは、たとえば、トランザクションを一括して管理するトランザクションマネージャーが、複数のトランザクションの間で、複数のリソースマネージャーにまたがって発生する、ソフトウェア資源の占有に関する待ち関係におけるデッドロックを検出して良い。

10

【 0 0 0 5 】

また、複数のタスクを並列実行し、各タスクがどの資源を占有しているかを管理するマルチタスクシステムにおいて、デッドロックを効率良く検出する方法が知られている（たとえば、特許文献2）。このようなマルチタスクシステムでは、待ち管理テーブルを設けて、各タスクの「待ち関係」を登録し、この待ち関係からデッドロックを検出し得る。

【 0 0 0 6 】

また、処理内の個別ロックに基づいてデッドロックを検出、分解する分散システムが知られている（たとえば、特許文献3）。このようなシステムでは、分散処理マネージャーが、処理を表すノードと、処理間の依存性を表すエッジとを有する待ちグラフを格納し、待ちグラフを通してプローブを伝搬し、デッドロック依存性を示すサイクリック依存性を検出し得る。

20

【 0 0 0 7 】

デッドロックの検出、解消方法として、被制御装置の占有権待ち時間が一定時間を超えた場合に、被制御装置の占有権を強制的に解放するタイムアウト方式が知られている（たとえば、特許文献2）。タイムアウトによらないデッドロックの検出方式としては、資源循環グラフを用いたデッドロックの解消方式が知られている（たとえば、非特許文献1）。資源循環グラフを用いたデッドロック検出を行う場合、全ての端末と被制御装置の制御関係を保有する必要があるため、分散システム上でそのまま用いることができない。しかし、端末、被制御装置に分散させて情報を保有した場合でも、それぞれの端末、被制御装置が保有する宛先にメッセージを送信し、メッセージが戻ってくるかを検出することで、資源循環グラフと同様にデッドロック検出を行う、Chandy - Misra - Haasアルゴリズムが知られている（たとえば、非特許文献2）。

30

【 先行技術文献 】

【 特許文献 】

【 0 0 0 8 】

【 特許文献 1 】 特開平 8 - 1 3 7 7 0 7 号 公 報

【 特許文献 2 】 特開平 6 - 3 3 7 7 9 8 公 報

【 特許文献 3 】 特開平 6 - 2 1 4 9 6 7 号 公 報

40

【 非特許文献 】

【 0 0 0 9 】

【 非特許文献 1 】 R. Holt, "Some dead lock properties of computer systems", ACM Computing Surveys, vol. 4, pp. 179-196, 1972.

【 非特許文献 2 】 K. Chandy, J. Misra, L. Haas, "Distributed deadlock detection", ACM Trans. Computer Systems, vol. 1, pp. 144-156, 1983.

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 0 】

しかしながら、Chandy - Misra - Haasアルゴリズムでは端末と被制御装

50

置が常時通信できることを仮定しているため、端末と被制御装置が通信できないことがある場合、正常にデッドロックの検出ができないことがあるという問題がある。

【0011】

よって、一つの側面として、本発明は、端末と被制御装置が通信できないことがある場合でも、デッドロックの検出が可能な分散データ処理装置、分散データ処理方法および分散データ処理プログラムを提供することを目的とする。

【課題を解決するための手段】

【0012】

複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御される複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つである分散データ処理装置が開示される。装置は、各々が前記複数の被制御装置の一つに対応する複数のノードが、第1の端末が制御権を有する第1の被制御装置に対応する第1のノードから前記第1の端末が制御権を有する第2の被制御装置に対応する第2のノードに向かって伸びる第1のリンクと、第2の端末が制御権を有する第3の被制御装置に対応する第3のノードから前記第2の端末が制御権を付与されることを待っている第4の被制御装置に対応する第4のノードに向かって伸びる第2のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第1のリンクおよび前記第2のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信するネットワーク受信部と、前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定するデッドロック判定部と、を含むことを特徴とする。

10

20

【発明の効果】

【0013】

端末と被制御装置が通信できないことがある場合でも、デッドロックを検出することができる。

【図面の簡単な説明】

【0014】

【図1】端末から被制御機器を操作するシステムの例を示す図である。

【図2】端末間のデッドロックについて説明するための図である。

【図3A】比較例における資源循環グラフの記法の例を示す図である。

30

【図3B】比較例における資源循環グラフの記法の別の例を示す図である。

【図4】資源循環グラフによるデッドロック検出について説明するための図である。

【図5】分散システム上でのデッドロック検出について説明するための図である。

【図6】資源循環グラフの例を示す図である。

【図7】Chandy-Misra-Haasアルゴリズムでのデッドロック検出の例を示す図である。

【図8】実施形態の分散データ処理装置の機能ブロックの例を示す図である。

【図9A】実施形態の資源循環グラフの記法の例を示す図である。

【図9B】実施形態の資源循環グラフの記法の別の例を示す図である。

【図10】ライン型トポロジの資源循環グラフの例を示す図である。

40

【図11】スター型トポロジの資源循環グラフの例を示す図である。

【図12】実施形態における資源循環グラフの例を示す図である。

【図13】前・次・待ノード情報の例を示す図である。

【図14】デッドロック検出メッセージの例を示す図である。

【図15】装置の制御権取得時の資源循環グラフの変形の例を説明するための図である。

【図16】装置の制御権取得時の資源循環グラフの変形の別の例を説明するための図である。

【図17】実施形態の資源循環グラフを用いたデッドロック検出の例について説明するための図である。

【図18】実施形態の資源循環グラフを用いたデッドロック解消の例について説明するた

50

めの図である。

【図 19】実施形態の資源循環グラフを用いたデッドロック検出と解消の例について説明するための図である。

【図 20】実施形態の分散データ処理装置の構成の例を示す図である。

【図 21】端末側の装置制御権獲得処理における処理の流れの例を示す図である。

【図 22】端末側の次ノード追加・変形処理における処理の流れの例を示す図である。

【図 23】端末側の待ちノード追加処理における処理の流れの例を示す図である。

【図 24】実施形態の資源循環グラフを用いたデッドロック検出処理における処理の流れの例を示す図である。

【図 25】実施形態の資源循環グラフを用いたデッドロック解消処理における処理の流れの例を示す図である。 10

【図 26】装置側の装置制御権獲得処理における処理の流れの例を示す図である。

【図 27】端末側の装置制御権解放処理における処理の流れの例を示す図である。

【図 28】デッドロック検出メッセージ送信処理における処理の流れの例を示す図である。

。

【図 29】装置側のデッドロック解消処理における処理の流れの例を示す図である。

【図 30】端末から被制御機器を操作するシステムの別の例を示す図である。

【図 31 A】デッドロックの検出、解消の例について説明するための図である（その 1）

。

【図 31 B】デッドロックの検出、解消の例について説明するための図である（その 2） 20

。

【図 31 C】デッドロックの検出、解消の例について説明するための図である（その 3）

。

【図 31 D】デッドロックの検出、解消の例について説明するための図である（その 4）

。

【図 31 E】デッドロックの検出、解消の例について説明するための図である（その 5）

。

【図 31 F】デッドロックの検出、解消の例について説明するための図である（その 6）

。

【発明を実施するための形態】 30

【0015】

以下、図面を参照して、実施形態の分散データ処理装置、分散データ処理方法および分散データ処理プログラムについて説明する。

【0016】

図 1 は、端末から被制御機器を操作するシステムの例を示す図である。ネットワーク機器の普及とモバイル端末の普及により、モバイル端末からネットワーク制御可能な装置を制御する技術が注目されている。例えば、ネットワークにより電球等のデバイスのスイッチのオン/オフの制御可能な電子基板や、ネットワーク制御可能なプロジェクタ等が知られている。

【0017】 40

ネットワーク制御可能な装置を制御するモバイル端末、タブレット端末等のモバイルコンピュータを、単に「端末」と呼ぶことがある。また、ネットワーク制御可能な機器を、「被制御機器」や「被制御装置」と呼ぶことがある。また、端末および被制御装置を単に、「装置」と呼ぶことがある。

【0018】

また、ある端末が、ある被制御装置をネットワーク制御可能なとき、端末は被制御装置の制御権を有する、とすることがある。また、このことを、端末は被制御装置を占有する占有権を有する、ということもある。これらは互いに、交換可能な表現である。ここで、制御権または占有権は、端末から被制御装置への要求に回答して、被制御装置から端末に与えられるものとしても良い。そして、端末が、被制御装置から制御権または占有権を与 50

えられるのを待っている状態を、「制御権待ち」状態、または「占有権待ち」状態と呼ぶことがある。

【0019】

以下では、「制御権」と「占有権」は互いに交換可能な用語である。

図1には、オペレータA、Bが有するモバイル端末、タブレット端末等のモバイルコンピュータ102、104を用いて、会議室など、決まった場所に設置された電子機器106、108、110を操作するシステム10を示している。図1では、オペレータAは、所有するモバイル端末102を用いて、カメラ106、プロジェクタ108を同時に操作し、カメラ106で撮像した映像を、プロジェクタ108から出力し得る。また、オペレータBは、所有するタブレット104を用いて、タブレット104に記憶された曲の中から決まった曲がオーディオ機器110から繰り返し流れるようにしても良い。

10

【0020】

これら、モバイルコンピュータ102、104からネットワーク制御可能な装置、カメラ106、プロジェクタ108、オーディオ機器110等を制御するようなシステム10においては、一般に被制御装置は状態を持つ。例えばプロジェクタ108においては、投影中状態、非投影状態、などの状態があっても良い。また、オーディオ機器110においては、音量の大きさなど、連続的な状態があっても良い。

【0021】

図1に示されているシステム10では、端末から発せられるメッセージにより、被制御装置の持つ状態を変更する構成を取る。例えば、図1においては、オペレータAのスマートフォン等のモバイルコンピュータ102の操作により、プロジェクタ108がカメラ106で撮像された映像を出力するか、それともカメラ106との接続を切って設定画面を出すかの操作が可能である。また、オペレータBの操作により、スピーカ等のオーディオ機器110から流す曲の種類を変更することが可能である。

20

【0022】

また、複数の端末により意図しない状態の変更を防ぐため、被制御装置は制御元の端末の情報を管理する。更に被制御装置は、端末からの制御権要求に基づき、端末への被制御装置の制御権の発行、または制御権待ちの通知を行う。例えば、図1においては、カメラ106、プロジェクタ108はオペレータAが所有する端末102に操作されているが、新たにオペレータAがスピーカ110の操作をしようとした場合は、スピーカ110から制御権待ちのメッセージを受け取り、オペレータBがオーディオ機器110の制御権を解放するまで、オーディオ機器110の制御権待ちをすることになる。

30

【0023】

このような構成により端末は制御先の被制御装置を常時監視する必要がなく、端末のネットワーク負荷を低減することが可能である。例えば、オペレータBの端末104は常時、オーディオ機器110と通信する必要はなく、曲の切り替え操作を行う場合にのみ、オーディオ機器と通信すればよい。

【0024】

また、システム上に多くの端末、被制御装置が存在することが想定されるため、全ての制御、非制御関係を持つことは非現実的であり、端末・被制御装置共に、自身が制御する、又は制御される端末・被制御装置の情報のみを保有する形態を取っても良い。例えば、オペレータAの端末102はプロジェクタ108、カメラ106の制御権情報を保持しているが、オーディオ機器110がオペレータBに操作されているという情報は持たない。

40

このようなシステム10は、一般に「分散システム」と呼ばれることもある。

【0025】

図2は、端末間のデッドロックについて説明するための図である。

ネットワーク制御可能な機器をモバイル端末から制御しようとする場合、端末間のデッドロックの解消は、アプリケーションの安定した操作の実現には不可欠である。デッドロックは、端末間で生じる制御の競合であり、複数の端末が、複数の機器を制御する場合に生じる。

50

【 0 0 2 6 】

例えば図 2 に示されているように、端末 A (1 0 2)、端末 B (1 0 4) で、プロジェクト 1 0 8 とオーディオ機器 1 1 0 の両者が必要なアプリケーションが実行されている状況を考える。このとき、端末 A (1 0 2) ではまずプロジェクト 1 0 8 の制御権を、端末 B (1 0 4) では、まずはオーディオ機器 1 1 0 の制御権を取得する。次に端末 A (1 0 2) はオーディオ機器 1 1 0、端末 B (1 0 4) はプロジェクト 1 0 8 の制御権を取得しようとした場合、はオーディオ機器 1 1 0 もプロジェクト 1 0 8 も既に占有されているため、端末 A、端末 B とともに制御権待ちの状態になる。しかし、端末 A (1 0 2)、端末 B (1 0 4) とともにお互いに必要な機器の制御権を保有したまま制御権待ちをしているために処理が先に進むことがなく、時間が経過しても競合が解消されることはない。このような、処理同士が競合を起こし、処理が先に進めなくなってしまう状態がデッドロックである。

10

【 0 0 2 7 】

デッドロックは自動的に解消することはないため、何らかの手段でデッドロックを検出し、解消することが必要である。

【 0 0 2 8 】

デッドロックの単純な検出、解消手段の一つは、被制御装置の占有権待ち時間が一定時間を超えた場合に、被制御装置の占有権を強制的に解放するタイムアウト方式である。しかし、タイムアウト方式では、制御権の誤解放や長時間の制御権待ちを防ぐためのタイムアウト時間の設定が困難である。また、制御権の解放を検出するために端末と被制御装置の間でポーリングが必要である。そのため、電力消費が大きくなる、といった問題から、特にモバイル端末上での実行には適さない。

20

【 0 0 2 9 】

タイムアウトによらないデッドロックの検出方式としては、資源循環グラフを用いたデッドロックの解消方式が知られている。

【 0 0 3 0 】

デッドロックの検出技法としては、待ちグラフによる手法も知られている。待ちグラフでも処理をノードとするグラフ上の循環路を検出することでデッドロックの検出を行っているため、資源循環グラフと同原理でデッドロックの検出を行っている。

【 0 0 3 1 】

図 3 A は、比較例における資源循環グラフの記法の例を示す図であり、図 3 B は比較例における資源循環グラフの記法の別の例を示す図である。

30

【 0 0 3 2 】

図 3 A、3 B では、端末、被制御装置をノードとして表し、丸で囲まれた装置は端末を、四角で囲まれた装置は被制御装置を表すこととする。更にグラフ上のリンクにより、端末と被制御装置の制御関係、制御待ち関係を表すものである。

【 0 0 3 3 】

図 3 A に示されている例では、装置 1 から端末 A に向けて矢印が伸びている。このように、被制御装置から端末に伸びるリンクによって、端末が被制御装置を制御していることを示すことがある。

40

【 0 0 3 4 】

図 3 B に示されている例では、端末 A から装置 1 に向けて矢印が伸びている。このように、端末から被制御装置に伸びるリンクによって、端末 A は、被制御装置 1 の制御権待ち状態にあることを示すことがある。

【 0 0 3 5 】

このとき、端末間のデッドロック発生時には資源循環グラフ上で循環路が生じることが知られており、資源循環グラフ上で循環路が生じているかを検査することにより、デッドロックの検出が可能である。このことを図 4 を参照しながら説明する。

【 0 0 3 6 】

図 4 は、資源循環グラフによるデッドロック検出について説明するための図である。

50

例えば、図4に示されている例では、「端末A 装置1 端末B 装置2 端末A」という循環路が生じている。このため、デッドロックが発生していることを検出することができる。

【0037】

資源循環グラフを用いたデッドロック検出を行う場合、全ての端末と被制御装置の制御関係を保有する必要があるため、分散システム上でそのまま用いることができない。しかし、端末、被制御装置に分散させて情報を保有した場合でも、それぞれの端末、被制御装置が保有する宛先にメッセージを送信し、メッセージが戻ってくるかを検出することで、資源循環グラフと同様にデッドロック検出を行う方法、たとえばChandy - Misra - Haasアルゴリズムが知られている。

10

【0038】

図5は、分散システム上でのデッドロック検出について説明するための図である。

図5に示されている例では、端末は自身が制御権待ちをしている被制御装置（装置1）の情報を、被制御装置（装置1）は自身の制御権を持つ端末の情報を記憶しておく。デッドロック検出時、端末はメッセージを送出し、またメッセージを受信した端末・被制御装置は、自身が記憶している端末、被制御装置にメッセージを伝送し、更に同一のメッセージが再度同じ端末・被制御装置に到達していないかを監視する。このとき、同一のメッセージが再度端末、又はデバイスに到達する場合は循環路が生じていることと等しいため、同一のメッセージが到達しているかを監視することにより、デッドロックの検出が可能となる。

20

【0039】

図6は、資源循環グラフの例を示す図である。

図6に示されている例では、装置2と装置3は端末Aに制御されている。また、装置1は、端末Bに制御されている。そして、端末Aは装置1の制御権待ちの状態にあり、端末Bは装置2の制御権待ちの状態にある。

【0040】

図7は、Chandy - Misra - Haasアルゴリズムでのデッドロック検出の例を示す図である。

【0041】

Chandy - Misra - Haasアルゴリズムは端末と被制御装置が常時通信できることを仮定しているため、端末と被制御装置が通信できないことがある場合、正常にデッドロックの検出ができないことがある。

30

【0042】

図7に示されている例で占有関係の状況で端末Bがスリープしている場合、端末Aから発せられたメッセージは、端末2から先には転送されないため、メッセージが端末Aに戻ってくることはない。従って、実際にはデッドロックが発生しているが、端末との通信ができないことがある場合、Chandy - Misra - Haasアルゴリズムではデッドロックを正しく検出できない場合がある。更に、デッドロックが検出されない場合は解消もされないため、端末間の競合が解消されず、システムの動作が妨げられる場合が生じる。特に、モバイル端末では、スリープや移動などで一時的に通信できないことが多発するため、Chandy - Misra - Haasアルゴリズムの仮定が成立しない場合が多く存在し得る。

40

【0043】

一般に、被制御装置は、端末に比べ通信できない場合が少ない。そこで、循環路に被制御装置のみを含むように構成された資源循環グラフを用いて、デッドロックを検出する。このような構成によって、端末と被制御装置が通信できないことがある場合でも、デッドロックの検出が可能な分散データ処理装置、分散データ処理方法および分散データ処理プログラムについて説明する。

【0044】

以下で開示される分散データ処理装置、分散データ処理方法および分散データ処理プロ

50

グラムでは、

- (C 1) 被制御装置のみを循環路に含む資源循環グラフを構成する、
 - (C 2) 被制御装置のみを循環路に含む資源循環グラフを用いた、端末と一時的に通信できない場合でもデッドロックを検出する、
 - (C 3) 被制御装置のみを循環路に含む資源循環グラフを用いた、端末と一時的に通信できない場合でもデッドロックを解消する、
 - (C 4) デッドロックの検出と解消を同時に行う、
- という特徴を有する。

【 0 0 4 5 】

< 分散データ処理装置 >

図 8 は、実施形態の分散データ処理装置 2 0 の機能ブロックの例を示す図である。分散データ処理装置 2 0 は、端末、被制御装置の一部であっても良い。

【 0 0 4 6 】

分散データ処理装置 2 0 は、デッドロック検出部 2 0 0、通信部 2 0 2、デッドロック解消部 2 0 4、資源循環グラフ更新部 2 0 6、優先度決定情報データベース (D B) 2 0 8、資源循環グラフ D B 2 1 0、占有元情報 D B 2 1 2 を含み得る。優先度決定情報データベース (D B) 2 0 8、資源循環グラフ D B 2 1 0、占有元情報 D B 2 1 2 はまとめて単に記憶部と呼ぶこともある。

【 0 0 4 7 】

デッドロック検出部 2 0 0 は、デッドロック判定部 2 0 0 0、転送メッセージ生成部 2 0 0 2 を含む。デッドロック判定部 2 0 0 0 は、メッセージの循環を検出し、デッドロックが発生しているか否かを判定しても良い。転送メッセージ生成部 2 0 0 2 は、分散データ処理装置 2 0 のユニフォーム リソース アイデンティファイア (U R I)、保存する優先度等に関する情報を含むメッセージを生成しても良い。

【 0 0 4 8 】

通信部 2 0 2 は、ネットワーク受信部 2 0 2 0、ネットワーク送信部 2 0 2 2 を含む。ネットワーク送信部 2 0 2 2 は、たとえば、デッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 で生成されたメッセージを送信しても良い。ネットワーク受信部 2 0 2 0 は、ネットワークからメッセージを受信し、デッドロック検出部 2 0 0 のデッドロック判定部 2 0 0 0 に送っても良い。

【 0 0 4 9 】

資源循環グラフ更新部 2 0 6 は、占有元端末管理部 2 0 6 0、資源追加・更新部 2 0 6 2、資源解放部 2 0 6 4 を含み得る。占有元端末管理部 2 0 6 0 は、分散データ処理装置 2 0 を占有している端末に関する情報を管理する。その際、占有元端末管理部 2 0 6 0 は、占有元情報 D B 2 1 2 を用いても良い。資源追加・更新部 2 0 6 2、は、資源が確保された時に、資源循環グラフで前・次・待ノードを追加、更新しても良い。資源解放部 2 0 6 4 は、資源が解放された時に、資源循環グラフで前・次・待ノードを更新しても良い。

【 0 0 5 0 】

優先度決定情報 D B 2 0 8 は、被制御装置の占有を解除する際の優先度を決定するための情報が格納されていても良い。優先度決定情報 D B 2 0 8 に格納されているデータは、転送メッセージ生成部 2 0 0 2 で、メッセージを生成する際に使用され得る。

【 0 0 5 1 】

資源循環グラフ D B 2 1 0 は、資源循環グラフ上、分散データ処理装置 2 0 を含んでいる被制御装置に付与される前ノード、次ノード、待ノードに関する情報が格納されていても良い。資源循環グラフ D B 2 1 0 に格納されているデータは、転送メッセージ生成部 2 0 0 2 で、メッセージを生成する際に使用され得る。

【 0 0 5 2 】

デッドロック解消部 2 0 4 は、占有を解除する端末を決定して、デッドロック解消のための占有強制解除メッセージを生成し、ネットワーク送信部 2 0 1 1 に送信させても良い。

。

10

20

30

40

50

【 0 0 5 3 】

たとえば、占有強制解除メッセージは、占有強制解除メッセージを受信した装置が、端末からの制御権要求に応答して占有権を与える通知を中止するような命令を含んでいても良い。

【 0 0 5 4 】

このように分散データ処理装置 20 は、複数の端末と、各々が、複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御される複数の被制御装置を含むシステムの複数の端末の一つまたは複数の被制御装置の一つである。

【 0 0 5 5 】

分散データ処理装置 20 のネットワーク受信部 2022 は、各々が複数の被制御装置の一つに対応する複数のノードが、第 1 の端末が制御権を有する第 1 の被制御装置に対応する第 1 のノードから第 1 の端末が制御権を有する第 2 の被制御装置に対応する第 2 のノードに向かって伸びる第 1 のリンクと、第 2 の端末が制御権を有する第 3 の被制御装置に対応する第 3 のノードから第 2 の端末が制御権を付与されることを待っている第 4 の被制御装置に対応する第 4 のノードに向かって伸びる第 2 のリンクとを用いて結合されることで構成される資源循環グラフにおいて、第 1 のリンクおよび第 2 のリンクに沿ってノード間を転送されるように、複数の被制御装置の間を転送されるデッドロック検出メッセージを受信しても良い。

10

【 0 0 5 6 】

分散データ処理装置 20 のデッドロック判定部 2000 は、デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定しても良い。

20

【 0 0 5 7 】

また、分散データ処理装置 20 のデッドロック判定部 2000 は、デッドロック検出メッセージが複数回、複数の被制御装置のうちの一つを訪問した場合に、デッドロックが発生していると判定しても良い。

【 0 0 5 8 】

分散データ処理装置 20 は、さらに、自身が第 1 のリンクの終点に対応する複数の被制御装置の一つである場合、第 1 のリンクの始点に対応する複数の被制御装置の別の一つを「前ノード」、自身が第 1 のリンクの始点に対応する複数の被制御装置の一つである場合、第 1 のリンクの終点に対応する複数の被制御装置の別の一つを「次ノード」、自身が第 2 のリンクの始点に対応する前記複数の被制御装置の一つである場合、前記第 2 のリンクの前記終点に対応する複数の被制御装置の別の一つを「待ノード」として、前ノード、次ノード、および待ノードに関する情報を記憶する記憶部 210 を含み得る。

30

【 0 0 5 9 】

資源循環グラフのノードの各々は、記憶部 210 に格納されている前ノード、次ノード、および待ノードに関する情報が付与されていても良い。

【 0 0 6 0 】

分散データ処理装置 20 は、さらに、デッドロック判定部 2000 が、デッドロックが発生していると判定したとき、デッドロックを解消するために、資源循環グラフに含まれる複数の被制御装置の一つの制御権を有する複数の端末の一つに、制御権を解消させるための通知を送るデッドロック解消部 204 を含み得る。

40

【 0 0 6 1 】

また、デッドロック検出メッセージは、資源循環グラフに含まれる複数の被制御装置の優先度に関する情報を含み、デッドロック判定部 2000 が、デッドロックが発生していると判定したとき、デッドロックを解消するために、デッドロック検出メッセージに含まれる優先度に基づいて、資源循環グラフに含まれる複数の被制御装置の一つの制御権を有する複数の端末の一つに、制御権を解消させるための通知を送るデッドロック解消部 204 を含み得る。

【 0 0 6 2 】

<< 資源循環グラフ >>

50

図 9 A は、実施形態の資源循環グラフの記法の例を示す図であり、図 9 B は、実施形態の資源循環グラフの記法の別の例を示す図である。

【 0 0 6 3 】

図 9 A に示されているように、装置が実線の矢印で結ばれている場合、同一の端末で制御されていることを示すものとする。

【 0 0 6 4 】

図 9 B に示されているように、装置が点線の矢印で結ばれている場合、矢印の根っこのノード（根ノード）によって表される装置を制御している端末は、矢印の先のノードで表される装置に対する制御権待ちの状態にあることを示すものとする。

【 0 0 6 5 】

このように、図 9 A、図 9 B に示されている資源循環グラフにおいては、端末、被制御装置のたとえば資源循環グラフ DB 2 1 0 に、以下の 2 種の情報を記憶させる。

(G 1) 同一の端末により制御されている端末・被制御装置を示す、双方向リンクリスト状のリンク（以下、前方リンクを「次ノード」、後方リンクを「前ノード」と呼ぶ）、

(G 2) 制御元の端末が制御権待ちをしている被制御装置を示すリンク（以下、「待ノード」と呼ぶ）。

【 0 0 6 6 】

よって、資源循環グラフでは、前・次ノードで連結された端末・被制御装置は同一端末によりリンク（第 1 のリンク）によって連結されていることを示し、更に前・次ノードで連結された部分グラフの根ノードは、必ず装置を制御する端末となるように構成する。また、待ノードによるリンク（第 2 のリンク）が伸ばされている被制御装置は、待ノードを保有している装置を制御している端末、すなわち、前ノードで辿ったノード（根ノードとも呼ぶ）が制御権待ちの状態であることを示す。

【 0 0 6 7 】

図 1 0 は、ライン型トポロジの資源循環グラフの例を示す図であり、図 1 1 は、スター型トポロジの資源循環グラフの例を示す図である。

【 0 0 6 8 】

ライン型トポロジは、子がないノード（葉ノード）のみに待ノードを持たせ、また前、次ノードで連結されるノードは直線状に連結する構成である。図 1 0 では、装置 1、装置 2 が待ノードである。前、次ノードで連結されるノード、つまり装置 2 と装置 3 は直線状に連結されている。

【 0 0 6 9 】

一方、スター型トポロジは端末の子ノードのみに待ノード、複数の次ノードを持たせる構成である。端末 A を親ノードとすると、装置 2 が子ノード、端末 B を親ノードとすると、装置 1 が子ノードである。

【 0 0 7 0 】

図 1 2 は、実施形態における資源循環グラフの例を示す図である。

図 1 2 では、端末 A は装置 2 と装置 3 の制御権を有し、装置 1 の制御権待ちをしており、端末 B は装置 1 の制御権を有し、装置 2 の制御権待ちをしている。

【 0 0 7 1 】

図 1 2 に示されているように、端末 A のたとえば資源循環グラフ DB 2 1 0 には、「前ノード：なし、次ノード：装置 2：待ノード：なし」の情報が格納されている。

【 0 0 7 2 】

また、装置 2 には、「前ノード：端末 A、次ノード：装置 3、待ノード：なし」の情報が格納されていても良い。この情報は、装置 2 の資源循環グラフ DB 2 1 0 に格納されていても良い。また、装置 2 の占有元情報 DB 2 1 2 には、装置 2 の制御権を有する「端末 A」の情報が格納されていても良い。

【 0 0 7 3 】

また、装置 3 には、「前ノード：装置 2、次ノード：なし、待ノード：装置 1」の情報が格納されていても良い。この情報は、装置 3 の資源循環グラフ DB 2 1 0 に格納されて

10

20

30

40

50

いても良い。

【0074】

図13は、前・次・待ノード情報の例を示す図である。この情報は、資源循環グラフDB210に格納されていても良い。図13に示されているように、資源循環グラフDB210に格納される前・次・待ノード情報は、前ノードURI、次ノードURI、待ノードURIの3つのフィールドを有し、各フィールドに「device001」等、被制御装置を特定する情報が格納され得る。また、待ノードURIのフィールドには、複数の装置のURIが格納されても良い。

【0075】

また、装置3の占有元情報DB212には、装置3の制御権を有する「端末A」の情報が格納されていても良い。

10

【0076】

前・次ノードで連結されたノードは2つの装置が同一端末により制御されていることを示す。図12の例では、装置2と装置3はともに端末Aによって制御されている。

【0077】

また、待ノードで連結されたノードは、その装置を制御する端末が制御権待ち状態にある装置である。図12の例では、装置3の待ノードである装置1は、装置3を制御する端末Aが端末1の制御権待ち状態にあることを意味する。

【0078】

このように、被制御装置のみを循環路に含む資源循環グラフは、制御、制御待ち関係を表現することが可能であり、循環路に被制御装置のみを含むため、端末との通信が一時的にできない場合でもデッドロックの検出が可能である。

20

【0079】

デッドロックは、ある被制御装置のデッドロック検出部200の転送メッセージ生成部2002で生成され、通信部202のネットワーク送信部2022から送信されたメッセージが、その被制御装置の通信部202のネットワーク受信部2020で受信され、デッドロック検出部200のデッドロック判定部2000で、自身が以前に生成したメッセージであると判定されたときに、発生していると判定され得る。

【0080】

図14は、デッドロック検出メッセージの例を示す図である。

30

図14に示されているメッセージは、「メッセージ発行元URI」、「メッセージ送信元URI」、「メッセージ送信先URI」、「既訪ノードフィールド・優先度情報」の5つのフィールドを含む。しかしながら、メッセージは、デッドロック解消のアルゴリズムに依存して、「既訪ノードフィールド・優先度情報」に優先度情報を含まなくても良い。

【0081】

「メッセージ発行元URI」フィールドには、デッドロック検出メッセージが生成された被制御装置を特定するURIが格納され得る。「メッセージ送信元URI」フィールドには、このメッセージを直線に送信した被制御装置を特定するURIが格納され得る。「メッセージ送信先URI」フィールドには、これからこのメッセージが送信される被制御装置を特定するURIが格納され得る。「既訪ノードフィールド・優先度情報」フィールドには、このメッセージが既に訪問した被制御装置と、優先度に関する情報が格納され得る。優先度は先に制御権要求を発信した被制御装置ほど高いとすれば、訪問した被制御装置の通信部202のネットワーク受信部2020が端末から制御権要求を発信した時刻であっても良い。図14に示されている例では、「既訪ノードフィールド・優先度情報」フィールドに、「(device002, appA, 2013/11/22 10:20:30)」と格納されている。これは、端末「appA」によって制御される被制御装置「device002」に2013年11月22日10時20分30秒に制御権要求を受信したことを示しても良い。

40

【0082】

ライン型トポロジは、次、待ノードにメッセージ送信を行えばデッドロックの検出が可

50

能である。一方、スター型トポロジでは、デッドロック検出のためには次、待ノードの他にメッセージを受信していない前ノードにもメッセージを送信する必要があり、煩雑な操作が必要となる。他方、装置の占有を行う際、ライン型トポロジでは、直線状に連結する構成を保持するためには、葉ノードに占有した装置を加える必要があり、端末が K 個 (K は整数とする) の装置を占有しているとき K のオーダー ($O(K)$) の通信が必要となる。他方、スター型トポロジでは端末の子ノードに新たに占有した装置を加えればよいので、1 のオーダー ($O(1)$) の通信で済む。

【0083】

資源循環グラフは、端末が被制御装置の制御権の要求を行った際、制御権を解放した際に、資源循環グラフ更新部 206 の資源解放部 2064 によって更新され得る。

10

【0084】

図 15 ~ 16 を参照して、装置の制御権取得時の資源循環グラフの変形について説明する。図 15 は、装置の制御権取得時の資源循環グラフの変形の例を説明するための図であり、図 16 は装置の制御権取得時の資源循環グラフの変形の別の例を説明するための図である。ここで、装置とは、被制御装置であっても良い。

【0085】

図 15 は端末が被制御装置を占有する場合の資源循環グラフの変形の例を示す図である。

【0086】

端末が制御権要求を行った際の資源循環グラフの更新の様子を、図 15 に示す。端末はまず被制御装置に制御権要求を行い、制御権が獲得できた場合は次ノード追加処理を、制御権待ちとなった場合は待ノード追加処理を行う。次ノード追加処理、待ノード追加処理では、それぞれライン型・スター型などの資源循環グラフのトポロジを保存するように変形操作を行う。

20

【0087】

図 15 (a) に示されているように、端末 A は装置 2 の制御権を有している。また、端末 B は装置 1 の制御権を有し、装置 2 に制御権を要求し、制御権待ち状態にある。このとき、端末 A が装置 3 に占有権 (制御権) を要求するとする。このとき図 15 (b) に示されているように、装置 3 はどの端末にも占有権 (制御権) を与えていないので、装置 3 は端末 A からの要求に回答して、端末 A に占有権 (制御権) を与える。そして、図 15 (c) に示されているように、端末 A は、葉ノードである装置 2 の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「次ノード」のフィールドに装置 3 を加える。つまり、葉ノードの次ノードに、新しく占有した装置を加える。資源循環グラフでは、装置 2 から装置 3 に、実線の矢印 (第 1 のリンク) が引かれる。

30

【0088】

図 16 は、端末が装置の制御権待ちを行う場合の資源循環グラフの変形の例を示す図である。

【0089】

図 16 (a) に示されているように、端末 A は装置 2 と装置 3 の制御権を有しているとする。また、端末 B は装置 1 の制御権を有し、さらに装置 2 に対して制御権を要求し、制御権待ち状態にあるとする。このとき図 16 (a) に示されているように、端末 A は、装置 1 の制御権を要求するとする。すると図 16 (b) に示されているように、装置 1 は、端末 A に制御権 (占有権) 待ちを通知する。次に図 16 (c) に示されているように、端末 A は、(子を持たない) 葉ノードである装置 3 の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「待ノード」のフィールドに占有待装置である装置 1 を加える。

40

【0090】

<<デッドロック検出>>

図 17 は、実施形態の資源循環グラフを用いたデッドロック検出の例について説明するための図である。図 17 に示されている例は、端末 A は装置 2 と装置 3 の制御権を有して

50

おり、端末 B は装置 1 の制御権を有し、さらに装置 2 に対して制御権を要求し、制御権待ち状態である。

【0091】

端末が被制御装置の制御権待ちとなった際、資源循環グラフ更新部 206 で図 16 に示される資源循環グラフの更新処理を行った後、デッドロック検出部 200 の転送メッセージ生成部 2002 でメッセージを生成する。そして、通信部 202 のネットワーク送信部 2022 からメッセージを送信する。もしネットワーク受信部 2020 でメッセージを受信したら、デッドロック検出部 200 のデッドロック判定部 2000 で、デッドロックが発生しているか否かを判定する。

【0092】

たとえば、端末 A がメッセージを生成、送信するとする。端末 A の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「次ノード」のフィールドは「装置 2」なので、メッセージは装置 2 に送信される。装置 2 の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「次ノード」のフィールドは「装置 3」である。よって、メッセージは、装置 2 から装置 3 に転送される。このとき、メッセージの「既訪ノードフィールド・優先度情報」フィールドには、メッセージが装置 2 にメッセージが到達した時間に関する情報、装置 2 の制御権を有する端末に関する情報等が格納されても良い。装置 3 の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「待ノード」のフィールドは「装置 1」である。よって、メッセージは、装置 3 から装置 1 に転送される。このとき、メッセージの「既訪ノードフィールド・優先度情報」フィールドには、メッセージが装置 3 にメッセージが到達した時間に関する情報、装置 3 の制御権を有する端末に関する情報等が格納されても良い。装置 1 の資源循環グラフ DB 210 に格納される前・次・待ノード情報の「待ノード」のフィールドは「装置 2」である。よって、メッセージは、装置 1 から装置 2 に転送される。

【0093】

装置 2 の通信部 202 のネットワーク受信部 2020 はメッセージを受け取ると、デッドロック検出部 200 のデッドロック判定部 2000 にメッセージを送る。装置 2 のデッドロック判定部 2000 は、受信したメッセージの「既訪ノードフィールド・優先度情報」フィールドの中に、このメッセージが既に装置 2 を訪れたことがあることを見出し、デッドロックが発生していると判定する。

【0094】

<<デッドロック解消>>

図 18 ~ 19 を参照して、資源循環グラフを用いたデッドロック解消について説明する。

【0095】

図 18 は、実施形態の資源循環グラフを用いたデッドロック解消の例について説明するための図である。デッドロックの解消時、占有権の解除を行う端末の決定に用いる情報としては、占有開始時間、デバイス占有時の優先度、前・次ノードの連結回数、待ノード回数、端末との通信可否等が含まれ得る。図 18 に示されている例では、デッドロック検出後、デッドロック解消時に再度装置に問い合わせを行う。この場合、図 14 に示されているメッセージは、「メッセージ発行元 URI」、「メッセージ送信元 URI」、「メッセージ送信先 URI」のみを含んでも良い。

【0096】

図 18 に示されているように、端末 A は装置 2 と装置 3 の制御権を有しているとする。また、端末 B は装置 1 の制御権を有し、さらに装置 2 に対して制御権を要求し、制御権待ち状態にあるとする。

【0097】

そして、装置 3 は、端末 A から制御権要求を受信し、08時15分20秒に占有されたものとする。図 17 の場合と同様に、装置 2 の通信部 202 のネットワーク受信部 2020 はメッセージを受け取ると、デッドロック検出部 200 のデッドロック判定部 2000

10

20

30

40

50

にメッセージを送る。装置2のデッドロック判定部2000は、受信したメッセージの「既訪ノードフィールド・優先度情報」フィールドの中に、このメッセージが既に装置2を訪れたことがあることを見出し、デッドロックが発生していると判定する。

【0098】

デッドロックが発生していると判定した装置2のデッドロック解消部204は、受信したメッセージに含まれる情報を用いて、循環路に含まれる各装置に、優先度決定のための情報を得るための通知を作成し、ネットワーク送信部2022から送信させる。

【0099】

装置2のネットワーク受信部2020は、優先度決定のための情報を得るための通知に対する応答を循環路に含まれる各装置から受信し、応答に含まれる情報を、優先度決定情報DB208に格納する。

10

【0100】

装置2のデッドロック解消部204は、優先度決定情報DB208に格納された優先度決定のための情報を得るための通知に対する応答に含まれる情報を用いて、被制御装置の占有権を取り消す（アポートする）端末を決定する。

【0101】

装置2のデッドロック解消部204は、アポートする端末に通知（メッセージ）を送り、被制御装置の占有権を取り消す（アポートする）処理を行わせる。その通知（メッセージ）を受信した端末は、端末が占有していたデバイスにメッセージを送り、デバイスの占有権を解除する。

20

【0102】

図18に示されている例では、優先度決定のための情報を得るための通知に対する応答に基づいて、装置2の優先度決定情報DB208には、制御権要求の発信時刻について、「装置1：端末B、7時30分31秒」、「装置2：端末A、9時30分25秒」、「装置3：端末A、8時15分20秒」等の情報が含まれているとする。今の場合、装置2のデッドロック解消部204は、装置2のデッドロック解消部204は、制御権要求の受信が最も早い装置1の優先度を最も高い装置と判定し、装置1を占有している端末Bに通知（メッセージ）を送り、装置1の占有権を取り消す（アポートする）処理を行わせる。

【0103】

このように、端末と被制御装置が通信できないことがある場合でも、デッドロックの検出、解消をすることができる。

30

【0104】

図19は、実施形態の資源循環グラフを用いたデッドロック検出と解消の例について説明するための図である。

【0105】

本例では、分散データ処理装置20は、デッドロック検出のために送るメッセージに、デッドロック解消時に占有権を解消する端末の情報を追記することにより、デッドロックの検出と解消を同時に行うこともできる。この場合、図14に示されているメッセージは、「メッセージ発行元URI」、「メッセージ送信元URI」、「メッセージ送信先URI」、「既訪ノードフィールド・優先度情報」の5つのフィールドを含む。

40

【0106】

図19に示されているように、端末Aは装置2と装置3の制御権を有しているとする。また、端末Bは装置1の制御権を有し、さらに装置2に対して制御権を要求し、制御権待ち状態にあるとする。

【0107】

そして、装置3は、端末Aから制御権要求を受信し、08時15分20秒に占有されたものとする。図17の場合と同様に、装置2の通信部202のネットワーク受信部2020はメッセージを受け取ると、デッドロック検出部200のデッドロック判定部2000にメッセージを送る。装置2のデッドロック判定部2000は、受信したメッセージの「既訪ノードフィールド・優先度情報」フィールドの中に、このメッセージが既に装置2を

50

訪れたことがあることを見出し、デッドロックが発生していると判定する。

【0108】

装置2のデッドロック解消部204は、メッセージの「既訪ノードフィールド・優先度情報」フィールドから、占有強制解除の優先度情報を抽出する。たとえば、図19に示されている例では、メッセージの「既訪ノードフィールド・優先度情報」フィールドには、メッセージが訪れた被制御装置、その被制御装置の占有権(制御権)を有する端末、制御権要求の発信時刻について、「装置1:端末B、7時30分31秒」、「装置2:端末A、9時30分25秒」、「装置3:端末A、8時15分20秒」等の情報が含まれているとする。装置2のデッドロック解消部204は、制御権要求の受信が最も早い装置1の優先度を最も高い装置と判定し、装置1を占有している端末Bに通知(メッセージ)を送り、装置1の占有権を取り消す(アポートする)処理を行わせる。

10

【0109】

このように、端末と被制御装置が通信できないことがある場合でも、デッドロックの検出、解消をすることができる。

【0110】

図20は、分散データ処理装置の構成の例を示す図である。

このコンピュータ300は、Central Processing Unit(CPU)302、Read Only Memory(ROM)304、及びRandom Access Memory(RAM)306を備えている。コンピュータ300は、さらに、ハードディスク装置308、入力装置310、表示装置312、インターフェース装置314、及び記録媒体駆動装置316を備えている。なお、これらの構成要素はバスライン318を介して接続されており、CPU302の管理の下で各種のデータを相互に授受することができる。

20

【0111】

Central Processing Unit(CPU)302は、このコンピュータ300全体の動作を制御する演算処理装置であり、コンピュータ300の制御処理部として機能する。

【0112】

Read Only Memory(ROM)304は、所定の基本制御プログラムが予め記録されている読み出し専用半導体メモリである。CPU302は、この基本制御プログラムをコンピュータ300の起動時に読み出して実行することにより、このコンピュータ300の各構成要素の動作制御が可能になる。

30

【0113】

Random Access Memory(RAM)306は、CPU302が各種の制御プログラムを実行する際に、必要に応じて作業用記憶領域として使用する、随時書き込み読み出し可能な半導体メモリである。

【0114】

ハードディスク装置308は、CPU302によって実行される各種の制御プログラムや各種のデータを記憶しておく記憶装置である。CPU302は、ハードディスク装置308に記憶されている所定の制御プログラムを読み出して実行することにより、後述する各種の制御処理を行えるようになる。

40

【0115】

入力装置310は、例えばマウス装置やキーボード装置であり、情報処理装置のユーザにより操作されると、その操作内容に対応付けられている各種情報の入力を取得し、取得した入力情報をCPU302に送付する。

【0116】

表示装置312は例えば液晶ディスプレイであり、CPU302から送付される表示データに応じて各種のテキストや画像を表示する。

【0117】

インターフェース装置314は、このコンピュータ300に接続される各種機器との間

50

での各種情報の授受の管理を行う。

【0118】

記録媒体駆動装置316は、可搬型記録媒体320に記録されている各種の制御プログラムやデータの読み出しを行う装置である。CPU302は、可搬型記録媒体320に記録されている所定の制御プログラムを、記録媒体駆動装置316を介して読み出して実行することによって、後述する各種の制御処理を行うようにすることもできる。なお、可搬型記録媒体320としては、例えばUSB(Universal Serial Bus)規格のコネクタが備えられているフラッシュメモリ、CD-ROM(Compact Disc Read Only Memory)、DVD-ROM(Digital Versatile Disc Read Only Memory)などがある。

10

【0119】

このようなコンピュータ300を用いて連携機器選択装置を構成するには、例えば、上述の各処理部における処理をCPU302に行わせるための制御プログラム(分散データ処理プログラム)を作成する。作成された制御プログラムはハードディスク装置308若しくは可搬型記録媒体320に予め格納しておく。そして、CPU302に所定の指示を与えてこの制御プログラム(分散データ処理プログラム)を読み出させて実行させる。こうすることで、分散データ処理装置が備えている機能がCPU302により提供される。

【0120】

<分散データ処理>

図21~29を参照して、分散データ処理について説明する。

20

【0121】

図21は、端末側の装置制御権獲得処理における処理の流れの例を示す図である。

また、分散データ処理装置が図20に示されているような汎用コンピュータ300である場合には、下記の説明は、そのような処理を行う制御プログラム(分散データ処理プログラム)を定義する。すなわち、以下では、下記に説明する処理を汎用コンピュータに行わせる制御プログラム(分散データ処理プログラム)の説明でもある。

【0122】

処理が開始されると、S100で端末Aのネットワーク送信部2022は、制御権を確保したい装置D(被制御装置)に制御権要求を行う。制御権要求を行うことは、制御権要求命令を送信することであり得る。本ステップの処理が終了すると、処理はS102に進む。

30

【0123】

S102で端末Aの占有元端末管理部2060は、制御権を確保したい装置D(被制御装置)から制御権を許可する通知を受信したか、すなわち装置Dの制御権が得られたか否かを判定する。本ステップでの判定の結果が“Yes”、すなわち装置Dの制御権が得られた場合には、処理はS104に進む。また、本ステップでの判定の結果が“No”、すなわち装置Dの制御権が得られない場合には、処理はS106に進む。

【0124】

S104で端末Aの資源循環グラフ更新部206の資源追加・変更部2062は、資源循環グラフの変形を行う。

40

【0125】

S104の次ノード追加・変形処理について、図22を参照して説明する。

図22は、端末側の次ノード追加・変形処理における処理の流れの例を示す図である。

【0126】

次ノード追加・変形処理が開始されると、端末Aの資源循環グラフ更新部206の資源追加・変更部2062は、端末Aの資源循環グラフDB210に格納されている情報のうち、次ノードフィールドNが空か否かを判定する。本ステップでの判定の結果が“Yes”、すなわち端末Aの次ノードフィールドNが空である場合には、処理はS210に進む。また、本ステップでの判定の結果が“No”、すなわち端末Aの次ノードフィールドNが空でない場合には、処理はS202に進む。

50

【 0 1 2 7 】

S 2 0 2 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、資源循環グラフのトポロジがライン型かスター型かを判定する。ライン型トポロジを有する資源循環グラフの例は、図 1 0 に示されている。スター型トポロジを有する資源循環グラフの例は、図 1 1 に示されている。本ステップでの判定の結果が“ライン型”、すなわち資源循環グラフのトポロジがライン型である場合には、処理は S 2 0 4 に進む。また、本ステップでの判定の結果が“スター型”、すなわち端資源循環グラフのトポロジがスター型である場合には、処理は S 2 0 6 に進む。

【 0 1 2 8 】

S 2 0 4 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、次ノードフィールド N から次ノードを辿り、終端にあるノードに制御権を獲得した装置 D (被制御装置) を追加する。本ステップの処理が終了すると、処理は S 2 0 8 に進む。

10

【 0 1 2 9 】

S 2 0 6 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、次ノードフィールド N に制御権を獲得した装置 D (被制御装置) を追加する。本ステップの処理が終了すると、処理は S 2 0 8 に進む。

【 0 1 3 0 】

S 2 1 0 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、端末 A の次ノードフィールド N に、新しく制御権を獲得した装置 D (被制御装置) をセットする。本ステップの処理が終了すると、処理は S 2 1 2 に進む。

20

【 0 1 3 1 】

S 2 1 2 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、端末 A の待ノードフィールド W が空か否かを判定する。本ステップでの判定の結果が“ Y e s ”、すなわち端末 A の待ノードフィールド W が空である場合には、処理は S 2 0 8 に進む。また、本ステップでの判定の結果が“ N o ”、すなわち端末 A の待ノードフィールド W が空でない場合には、処理は S 2 1 4 に進む。

30

【 0 1 3 2 】

S 2 1 4 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、装置 D (被制御装置) の待ノードフィールドに端末 A の待ノードフィールド W の値をセットする。本ステップの処理が終了すると、処理は S 2 1 6 に進む。

【 0 1 3 3 】

S 2 1 6 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、端末 A の待ノードフィールド W をクリアする。本ステップの処理が終了すると、処理は S 2 0 8 に進む。

40

【 0 1 3 4 】

S 2 0 8 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、制御権を獲得した装置 D に、装置 D を次ノードとしてセットした装置の情報を、通信部 2 0 2 のネットワーク送信部 2 0 2 2 から送信させる。本ステップの処理が終了すると、処理は図 2 1 の S 1 0 4 に戻る。

【 0 1 3 5 】

図 2 1 に戻って、S 1 0 4 の処理が終了すると、端末 A の処理は終了する。

図 2 1 の S 1 0 6 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、待ノード追加処理を行う。

【 0 1 3 6 】

50

端末 A の待ノード追加処理について、図 2 3 を参照して説明する。

図 2 3 は、端末側の待ちノード追加処理における処理の流れの例を示す図である。

【 0 1 3 7 】

処理が開始されると、S 3 0 0 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ DB 2 1 0 に格納されている資源循環グラフにおいて、端末 A の次ノードフィールド N が空か否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち端末 A の次ノードフィールド N が空である場合には、処理は S 3 0 2 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち端末 A の次ノードフィールド N が空でない場合には、処理は S 3 0 4 に進む。

【 0 1 3 8 】

S 3 0 2 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ DB 2 1 0 に格納されている資源循環グラフにおいて、端末 A の待ノードフィールドに、新しく制御権待ちをする装置 D (被制御装置) を加える。本ステップの処理が終了すると、待ノード追加処理は終了する。

【 0 1 3 9 】

S 3 0 4 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、資源循環グラフ DB 2 1 0 に格納されている資源循環グラフはライン型トポロジであるか否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち資源循環グラフはライン型トポロジである場合には、処理は S 3 0 6 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち資源循環グラフはライン型トポロジではない場合には、処理は S 3 0 8 に進む。

【 0 1 4 0 】

S 3 0 6 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ DB 2 1 0 に格納されている資源循環グラフにおいて、端末 A の次ノードを辿り、終端にあるノードに制御権待ちをする装置 D (被制御装置) を加える。本ステップの処理が終了すると、待ノード追加処理は終了する。

【 0 1 4 1 】

S 3 0 8 で端末 A の資源循環グラフ更新部 2 0 6 の資源追加・変更部 2 0 6 2 は、端末 A の資源循環グラフ DB 2 1 0 に格納されている資源循環グラフにおいて、装置 N の待ノードフィールドに、新しく制御権待ちをする装置 D (被制御装置) を加える。本ステップの処理が終了すると、待ノード追加処理は終了する。

【 0 1 4 2 】

図 2 1 の S 1 0 6 の処理が終了すると、処理は S 1 0 8 に進む。

S 1 0 8 で装置 A のデッドロック検出部 2 0 0 は、デッドロック検出处置を行う。

【 0 1 4 3 】

デッドロック検出处置について、図 2 4 を参照して説明する。

図 2 4 は、実施形態の資源循環グラフを用いたデッドロック検出处置における処理の流れの例を示す図である。

【 0 1 4 4 】

デッドロック検出处置が開始されると、S 4 0 0 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、端末 A の次ノードフィールド N が空か否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち端末 A の次ノードフィールド N が空である場合には、処理は S 4 0 2 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち端末 A の次ノードフィールド N が空ではない場合には、処理は S 4 0 4 に進む。

【 0 1 4 5 】

S 4 0 2 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、デッドロック検出メッセージを生成し、通信部 2 0 2 のネットワーク送信部 2 0 2 2 を介して、自身の次ノード N に送出する。次ノード N を特定するために、資源循環グラフ DB に格納されている情報を参照しても良い。本ステップの処理が終了すると、処理は S 4 0 4

10

20

30

40

50

に進む。

【 0 1 4 6 】

S 4 0 4 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、デッドロック検出メッセージを生成し、通信部 2 0 2 のネットワーク送信部 2 0 2 2 を介して、自身の待ノード W の全てに送出する。待ノード W を特定するために、資源循環グラフ DB に格納されている情報を参照しても良い。本ステップの処理が終了すると、処理は S 4 0 4 に進む。

【 0 1 4 7 】

S 4 0 4 で装置 A のデッドロック検出部 2 0 0 のデッドロック判定部 2 0 0 0 は、ノードの識別子と、ネットワーク受信部 2 0 2 0 で受信したメッセージの発行主が一致するかを判定する。本ステップでの判定の結果が “ Y e s ”、すなわちノードの識別子とメッセージの発行主が一致する場合には、処理は S 4 0 6 に進む。また、本ステップでの判定の結果が “ N o ”、すなわちノードの識別子とメッセージの発行主が一致しない場合には、処理は S 4 0 8 に進む。

10

【 0 1 4 8 】

S 4 0 6 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、デッドロック検出を通知するためのメッセージを生成し、通信部 2 0 2 のネットワーク送信部 2 0 2 2 から送信することで、デッドロック検出を通知する。本ステップの処理が終了すると、デッドロック検出処置は終了する。

【 0 1 4 9 】

S 4 0 8 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、メッセージの既訪フィールドに自身の U R I を追加する。本ステップの処理が終了すると、処理は S 4 1 0 に進む。

20

【 0 1 5 0 】

S 4 1 0 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、資源循環グラフ DB 2 1 0 を参照して、本ノード（装置 A）は次・待ノードを持つか否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち本ノード（装置 A）は次・待ノードを持つ場合には、処理は S 4 1 2 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち本ノード（装置 A）は次・待ノードを持たない場合には、デッドロック検出処置は終了する。

30

【 0 1 5 1 】

S 4 1 2 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、通信部 2 0 2 のネットワーク送信部 2 0 2 2 を介して、次・待ノード全てにメッセージを送信する。本ステップの処理が終了すると、処理は S 4 1 4 に進む。

【 0 1 5 2 】

S 4 1 4 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、資源循環グラフ DB 2 1 0 を参照して、資源循環グラフはスター型トポロジか否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち資源循環グラフはスター型トポロジである場合には、処理は S 4 1 6 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち資源循環グラフはスター型トポロジではない場合には、処理は S 4 0 4 に戻る。

40

【 0 1 5 3 】

S 4 1 6 で装置 A のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、通信部 2 0 2 のネットワーク送信部 2 0 2 2 を介して、既訪ノードフィールドに記されていない前ノード全てにメッセージを送信する。本ステップの処理が終了すると、処理は S 4 0 4 に戻る。

【 0 1 5 4 】

デッドロック検出処置が終了すると、図 2 1 の S 1 1 0 に進む。

図 2 1 の S 1 1 0 で装置 A のデッドロック検出部 2 0 0 のデッドロック検出部 2 0 0 0 は、デッドロックしていたか否かを判定する。本ステップでの判定の結果が “ Y e s ”、

50

すなわちデッドロックしていた場合には、処理はS 1 1 2に進む。また、本ステップでの判定の結果が“ N o ”、すなわちデッドロックしていない場合には、処理はS 1 1 4に戻る。

【 0 1 5 5 】

S 1 1 4で装置Aのデッドロック検出部2 0 0のデッドロック検出部2 0 0 0は、待機をし、外部から被制御装置に制御権を確保するための入力を受けると、S 1 0 0に戻る。

【 0 1 5 6 】

S 1 1 2で装置Aのデッドロック解消部2 0 4は、デッドロック解消処理を行う。

デッドロック解消処理について、図2 5を参照して説明する。図2 5に示されている処理は、図1 8に示されているような場合に相当する。

【 0 1 5 7 】

図2 5は、実施形態の資源循環グラフを用いたデッドロック解消処理における処理の流れの例を示す図である。

【 0 1 5 8 】

デッドロック解消処理が開始されると、S 5 0 0で装置Aのデッドロック解消部2 0 4は、通信部2 0 2のネットワーク送信部2 0 2 2を介して、循環路に含まれる装置（被制御装置）に優先度決定のための情報を問い合わせる。本ステップの処理が終了すると、処理はS 5 0 2に進む。

【 0 1 5 9 】

S 5 0 2で装置Aのデッドロック解消部2 0 4は、問合せへの返事に含まれる情報（問い合わせ情報）から被制御装置の占有権を取り消す（アポートする）端末を決定する。本ステップの処理が終了すると、処理はS 5 0 4に進む。

【 0 1 6 0 】

S 5 0 4で装置Aのデッドロック解消部2 0 4は、通信部2 0 2のネットワーク送信部2 0 2 2を介して、アポートする端末に通知（メッセージ）を送り、被制御装置の占有権を取り消す（アポートする）処理を行わせる。本ステップの処理が終了すると、処理はS 5 0 6に進む。

【 0 1 6 1 】

S 5 0 6でその通知（メッセージ）を受信した端末は、端末が占有していたデバイスにメッセージを送り、デバイスの占有権を解除する。本ステップの処理が終了すると、デッドロック解消処理は終了する。

【 0 1 6 2 】

図2 6は、装置側の装置制御権獲得処理における処理の流れの例を示す図である。

装置制御権獲得処理が開始されると、S 6 0 0で装置Dの通信部2 0 2のネットワーク受信部2 0 2 0は、端末Aから制御権要求命令を受信し、資源循環グラフ更新部2 0 6の占有元端末管理部2 0 6 0に送る。本ステップの処理が終了すると、処理はS 6 0 2に進む。

【 0 1 6 3 】

S 6 0 2で装置Dの資源循環グラフ更新部2 0 6の占有元端末管理部2 0 6 0は、装置Dを制御している端末があるか否かを判定する。本ステップでの判定の結果が“ Y e s ”、すなわち装置Dを制御している端末がある場合には、処理はS 6 0 6に進む。また、本ステップでの判定の結果が“ N o ”、すなわち装置Dを制御している端末がない場合には、処理はS 6 0 4に戻る。

【 0 1 6 4 】

S 6 0 4で装置Dの資源循環グラフ更新部2 0 6の占有元端末管理部2 0 6 0は、通信部2 0 2のネットワーク送信部2 0 2 2を介して、制御権要求を送った端末Aに制御権取得失敗を通知する。本ステップの処理が終了すると、処理はS 6 0 6に進む。

【 0 1 6 5 】

S 6 0 6で装置Dのデッドロック検出部2 0 0の転送メッセージ生成部2 0 0 2は、装置Dの資源循環グラフDB 2 1 0に格納されている資源循環グラフにおいて、装置Dの前

10

20

30

40

50

ノードフィールドに制御権要求した端末の情報を追加する。本ステップの処理が終了すると、処理は S 6 0 8 に進む。

【 0 1 6 6 】

S 6 0 8 で装置 D の通信部 2 0 2 のネットワーク送信部 2 0 2 2 は、制御権要求を送った端末 A に制御権取得成功を通知する。本ステップの処理が終了すると、処理は S 6 1 0 に進む。

【 0 1 6 7 】

S 6 1 0 で装置 D の資源循環グラフ更新部 2 0 6 の占有元端末管理部 2 0 6 0 は、装置 D の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、端末 A から、本装置 D を次ノードとしてセットした装置（又は端末）を受け取り、本装置 D の前ノードフィールドにセットする。本ステップの処理が終了すると、装置制御権獲得処理は終了する。

10

【 0 1 6 8 】

図 2 7 は、端末側の装置制御権解放処理における処理の流れの例を示す図である。

処理が開始されると、S 7 0 0 で端末では、ノード変数 N を空にする。本ステップの処理が終了すると、処理は S 7 0 2 に進む。

【 0 1 6 9 】

S 7 0 2 で端末は、制御権を解放する装置 D 1 に保有する前・次・待ノードを問い合わせる。本ステップの処理が終了すると、処理は S 7 0 4 に進む。

【 0 1 7 0 】

20

S 7 0 4 で端末は、装置 D 1 が次ノードを持つか否かを判定する。本ステップでの判定の結果が “ Y e s ”、すなわち装置 D 1 が次ノードを持つ場合には、処理は S 7 0 6 に進む。また、本ステップでの判定の結果が “ N o ”、すなわち装置 D 1 が次ノードを持たない場合には、処理は S 7 1 0 に進む。

【 0 1 7 1 】

S 7 0 6 で端末は、端末の資源循環グラフ D B 2 1 0 に格納されている資源循環グラフにおいて、装置 D 1 の次ノードからノードを一つ選び（以下、装置 D 2 とする）、装置 D 2 に装置 D 1 の前ノード、装置 D 2 以外の次ノード、装置 D 1 の待ノードを移動する。本ステップの処理が終了すると、処理は S 7 0 8 に進む。

【 0 1 7 2 】

30

S 7 0 8 で端末は、「ノード変数 N = 装置 D 2 の次ノード」と設定する。本ステップの処理が終了すると、処理は S 7 1 0 に進む。

【 0 1 7 3 】

S 7 1 0 で端末は、「ノード変数 N = 装置 D 1 の前ノード」と設定する。本ステップの処理が終了すると、処理は S 7 1 2 に進む。

【 0 1 7 4 】

S 7 1 2 で端末は、装置 D 1 に制御権解放メッセージを送信し、装置 D 1 において、前・次・待ノードフィールドをクリアする。本ステップの処理が終了すると、端末側の装置制御権解放処理は終了する。

【 0 1 7 5 】

40

デッドロック解消処理の別の例について、図 2 8 ~ 2 9 を参照して説明する。図 2 8 ~ 2 9 に示されている処理は、図 1 9 に示されているような場合に相当する。

【 0 1 7 6 】

図 2 8 は、デッドロック検出メッセージ送信処理における処理の流れの例を示す図である。

【 0 1 7 7 】

図 2 8 に示されている例では、処理が開始されると、S 8 0 0 で端末または被制御装置のデッドロック検出部 2 0 0 の転送メッセージ生成部 2 0 0 2 は、送信するメッセージの最後尾に装置（被制御装置）が保持する優先度情報を追記する。本例でのデッドロック検出メッセージの例は、図 1 4 に示されている。本ステップの処理が終了すると、処理は S

50

802に進む。

【0178】

S802で端末または被制御装置の通信部202のネットワーク送信部2022は、メッセージを送信する。本ステップの処理が終了すると、デッドロック検出メッセージ送信処理は終了する。

【0179】

図29は、装置側のデッドロック解消処理における処理の流れの例を示す図である。

処理が開始されると、S900でデッドロック解消部204は、デッドロック検出メッセージ中の優先度情報からアポートする端末を決定する。たとえば、デッドロック解消部204は、メッセージの「既訪ノードフィールド・優先度情報」フィールドから、占有強制解除の優先度情報を抽出する。たとえば、図19に示されている例では、メッセージの「既訪ノードフィールド・優先度情報」フィールドには、メッセージが訪れた被制御装置、その被制御装置の占有権（制御権）を有する端末、制御権要求の発信時刻について、「装置1：端末B、7時30分31秒」、「装置2：端末A、9時30分25秒」、「装置3：端末A、8時15分20秒」等の情報が含まれているとする。装置2のデッドロック解消部204は、制御権要求の受信が最も早い装置1の優先度を最も高い装置と判定する。本ステップの処理が終了すると、処理はS902に進む。

【0180】

S902でデッドロック解消部204は、アポートする端末にメッセージを送り、アポートさせる。図19に示されている例では、制御権要求の受信が最も早い装置1の優先度を最も高い装置と判定し、装置1を占有している端末Bに通知（メッセージ）を送り、装置1の占有権を取り消す（アポートする）処理を行わせる。本ステップの処理が終了すると、処理はS904に進む。

【0181】

S904でアポートする端末は、占有していた装置にメッセージを送り、装置の占有権を解除する。本ステップの処理が終了すると、デッドロック解消処理は終了する。

【0182】

<実施例>

図30～31Fを参照して、実施例のシステム30を説明する。システム30は、各ブースに複数のマイク付カメラが設置されている展示会場にて、人が集まっているブースの様子を1台ずつ設置された大音響スピーカ、大画面ディスプレイで映し出すためのシステムである。

【0183】

図30に示されているシステム30では、会議室内で、オペレータCは端末A（302）を所有し、オペレータDは端末B（304）を所有している。会議室内には、他にカメラ1（306）、カメラ2（308）、スピーカ310、ディスプレイ312が設置されている。

【0184】

上の各端末、装置は、図8に示されているような分散データ処理装置を含んでいる。

システム30のオペレータC、D（例えば展示会場の運営者）はそれぞれ携帯端末A（302）、端末B（304）を持ち、予め定められたカメラの検出範囲に入ったとき、自動的に端末が最も近くのマイク付カメラ306、308と、ディスプレイ312、スピーカ310を占有し、マイク付カメラ306、308からの映像・音声をそれぞれディスプレイ312とスピーカ310に出力する。システム30は、ユーザ位置に従って自動的に機器連携を行う機器連携アプリケーションによって動作するシステムであっても良い。

【0185】

機器連携アプリケーションが動作する端末は複数に及ぶ場合があるが、その場合でも、映し出されるカメラ映像が短時間で切り替わることを避けるため、カメラ1（306）、カメラ2（308）と、スピーカ310と、ディスプレイ312はそれぞれ一定時間、端末A（302）、端末B（304）に操作占有権を与え、端末は占有権が得られた後で、

10

20

30

40

50

スピーカ310と、ディスプレイ312に対してカメラ306、308の連携動作を行う。則ち、端末A(302)、端末B(304)は、カメラ306、308と、スピーカ310と、ディスプレイ312に対し、各1個ずつ占有を完了したとき、スピーカ310とディスプレイ312に情報を伝達するカメラを切り替える。また、機器の利用効率を上げるため、カメラ1(306)、カメラ2(308)、ディスプレイ312等の装置の占有権確保は逐次的に行うのではなく、必要な機器全てに対し、同時に占有権確保のためのメッセージを送信する構成を取る。従って、端末A(302)と端末B(304)の間で装置の占有権のデッドロックが生じうる構成である。

【0186】

また、端末A(302)、端末B(304)は、バッテリー消費を抑えるため、スリープ状態になることがある。スリープ状態である場合は、例えばデッドロック検出メッセージの送信のような操作は行うことができない。このように、システム30は、端末A(302)、端末B(304)がスリープ状態になる場合があり、かつデッドロックが生じうる構成である。

10

【0187】

以下、図30に示すような状況において、デッドロック検出・解消方法の動作例を説明する。なお、本例でのデッドロック解消ポリシーは、最も早く機器の占有をした端末がアポートされない、則ち、「最も遅く機器の占有を行った端末の占有権を取り上げる」とする。

【0188】

図31A~Fは、デッドロックの検出、解消の例について説明するための図である。

20

図31Aに示されているように、端末A(302)、端末B(304)のオペレータはそれぞれカメラ1(306)、カメラ2(308)の機器連携範囲内に入り、機器連携を開始する。このとき、端末A(302)はカメラ1(306)、スピーカ310、ディスプレイ312に対して占有権要求メッセージを送り、端末B(304)はカメラ2(308)、スピーカ310、ディスプレイ312に対して占有権要求メッセージを送る。

【0189】

図31Bに示されているように、まず端末A(302)はカメラ1(306)、端末B(304)はカメラ2(308)から占有権許可メッセージを受領し、端末A(302)、端末B(304)の資源循環グラフ更新部206の資源追加・更新部2062は、それぞれの資源循環グラフDB210に格納されている前/次ノードに関する情報を更新する。そして、端末A(302)はカメラ1(306)の占有権を持ち、端末B(304)はカメラ2(308)の占有権を持つようになる。

30

【0190】

次に図31Cに示されているように、端末A(302)は端末A(302)、端末B(304)はディスプレイ312から占有権許可メッセージを受領し、端末A(302)、端末B(304)の資源循環グラフ更新部206の資源追加・更新部2062は、それぞれの資源循環グラフDB210に格納されている前/次ノードに関する情報を更新する。このとき、端末A(302)はカメラ1(306)、スピーカ310の占有権を、端末B(304)はカメラ2(308)、ディスプレイ312の占有権を持つようになる。

40

【0191】

更に図31Dに示されているように、スピーカ310は端末B(304)に対して制御権待ちのメッセージを送り、端末B(304)はスピーカ310の制御権待ちの状態になる。

【0192】

ここで、端末B(304)はデッドロック検出メッセージをカメラ2に対して送付する。デッドロック検出メッセージはカメラ2(308)、ディスプレイ312、スピーカ310と伝わるが、資源循環グラフに循環路が生じていないため、デッドロックは検出されない。

【0193】

50

次に図 3 1 E に示されているように、ディスプレイ 3 1 2 が端末 A に対して制御権待ちのメッセージを送り、端末 A (3 0 2) はディスプレイ 3 1 2 の制御権待ちの状態になる。

【 0 1 9 4 】

このとき、端末 B (3 0 4) はバッテリー残量が少なくなり、スリープ状態になったとする。このとき、端末 A (3 0 2) はデッドロック検出メッセージを送付するが、デッドロック検出メッセージはカメラ 1 (3 0 6)、スピーカ 3 1 0、ディスプレイ 3 1 2、スピーカ 3 1 0 の順で流れるため、スピーカ 3 1 0 のデッドロック検出部 2 0 0 がデッドロックを検出する。このとき、デッドロック検出メッセージに記載された資源の確保時刻から、スピーカ 3 1 0 のデッドロック解消部 2 0 4 は、制御権解放を行うべき装置がディスプレイ 3 1 2 であると判定し、スピーカ 3 1 0 からディスプレイ 3 1 2 に制御権解放メッセージを送付させる。

10

【 0 1 9 5 】

最終的に図 3 1 F に示されているように、端末 A (3 0 2) はディスプレイ 3 1 2 の制御権を獲得し、デッドロックは解消される。

【 0 1 9 6 】

このように、端末と被制御装置が通信できないことがある場合でも、デッドロックを検出することができる。

【 0 1 9 7 】

以上の実施形態に関し、さらに以下の付記を開示する。

20

(付記 1)

複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御される複数の被制御装置を含むシステムの分散データ処理装置であって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第 1 の端末が制御権を有する第 1 の被制御装置に対応する第 1 のノードから前記第 1 の端末が制御権を有する第 2 の被制御装置に対応する第 2 のノードに向かって伸びる第 1 のリンクと、第 2 の端末が制御権を有する第 3 の被制御装置に対応する第 3 のノードから前記第 2 の端末が制御権を付与されることを待っている第 4 の被制御装置に対応する第 4 のノードに向かって伸びる第 2 のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第 1 のリンクおよび前記第 2 のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信するネットワーク受信部と、

30

前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定するデッドロック判定部と、を含む分散データ処理装置。

(付記 2)

前記デッドロック判定部は、前記デッドロック検出メッセージが複数回、前記複数の被制御装置のうちの一つを訪問した場合に、デッドロックが発生していると判定する、付記 1 に記載の分散データ処理装置。

(付記 3)

40

さらに、

自身が前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の別の一つを前ノード、自身が前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを次ノード、自身が前記第 2 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 2 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを待ノードとして、前記前ノード、前記次ノード、および前記待ノードに関する情報を記憶する記憶部を含み、

前記資源循環グラフの前記ノードの各々は、前記記憶部 (図 8、 2 1 0) に格納されて

50

いる前記前ノード、前記次ノード、および前記待ノードに関する情報が付与されている、付記 1 または 2 に記載の分散データ処理装置。

(付記 4)

さらに、

前記デッドロック判定部が、前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送るデッドロック解消部を含む、付記 1 乃至 3 のいずれか一項に記載の分散データ処理装置。

(付記 5)

前記デッドロック検出メッセージは、前記資源循環グラフに含まれる前記複数の被制御装置の優先度に関する情報を含み、

さらに、

前記デッドロック判定部が、前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記デッドロック検出メッセージに含まれる前記優先度に基づいて、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送るデッドロック解消部を含む、付記 1 乃至 3 のいずれか一項に記載の分散データ処理装置。

(付記 6)

コンピュータとして機能する複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御されるコンピュータとして機能する複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つで実行される分散データ処理方法であって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第 1 の端末が制御権を有する第 1 の被制御装置に対応する第 1 のノードから前記第 1 の端末が制御権を有する第 2 の被制御装置に対応する第 2 のノードに向かって伸びる第 1 のリンクと、第 2 の端末が制御権を有する第 3 の被制御装置に対応する第 3 のノードから前記第 2 の端末が制御権を付与されることを待っている第 4 の被制御装置に対応する第 4 のノードに向かって伸びる第 2 のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第 1 のリンクおよび前記第 2 のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信することと、

前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定することと、

を含む分散データ処理方法。

(付記 7)

前記デッドロックが発生しているか否かを判定することは、前記デッドロック検出メッセージが複数回、前記複数の被制御装置のうちの一つを訪問した場合に、デッドロックが発生していると判定する、付記 6 に記載の分散データ処理方法。

(付記 8)

さらに、

自身が前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の別の一つを前ノード、自身が前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを次ノード、自身が前記第 2 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 2 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを待ノードとして、前記前ノード、前記次ノード、および前記待ノードに関する情報を記憶することを含み、

前記資源循環グラフの前記ノードの各々は、前記前ノード、前記次ノード、および前記待ノードに関する情報が付与されている、

10

20

30

40

50

付記 6 または 7 に記載の分散データ処理方法。

(付記 9)

さらに、

前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送ることを含む、付記 6 乃至 8 のいずれか一項に記載の分散データ処理装置。

(付記 10)

前記デッドロック検出メッセージは、前記資源循環グラフに含まれる前記複数の被制御装置の優先度に関する情報を含み、

さらに、

前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記デッドロック検出メッセージに含まれる前記優先度に基づいて、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送ることを含む、付記 6 乃至 8 のいずれか一項に記載の分散データ処理方法。

(付記 11)

コンピュータとして機能する複数の端末と、各々が、前記複数の端末の一つに自身の動作を制御する制御権を付与することによって、動作が制御されるコンピュータとして機能する複数の被制御装置を含むシステムの前記複数の端末の一つまたは前記複数の被制御装置の一つで実行される分散データ処理プログラムであって、

各々が前記複数の被制御装置の一つに対応する複数のノードが、第 1 の端末が制御権を有する第 1 の被制御装置に対応する第 1 のノードから前記第 1 の端末が制御権を有する第 2 の被制御装置に対応する第 2 のノードに向かって伸びる第 1 のリンクと、第 2 の端末が制御権を有する第 3 の被制御装置に対応する第 3 のノードから前記第 2 の端末が制御権を付与されることを待っている第 4 の被制御装置に対応する第 4 のノードに向かって伸びる第 2 のリンクとを用いて結合されることで構成される資源循環グラフにおいて、前記第 1 のリンクおよび前記第 2 のリンクに沿ってノード間を転送されるように、前記複数の被制御装置の間を転送されるデッドロック検出メッセージを受信し、

前記デッドロック検出メッセージの履歴に基づいて、デッドロックが発生しているか否かを判定する、

処理をコンピュータに実行させることを特徴とする分散データ処理プログラム。

(付記 12)

前記デッドロックが発生しているか否かを判定することは、前記デッドロック検出メッセージが複数回、前記複数の被制御装置のうちの一つを訪問した場合に、デッドロックが発生していると判定する、付記 11 に記載の分散データ処理プログラム。

(付記 13)

さらに、

自身が前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の別の一つを前ノード、自身が前記第 1 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 1 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを次ノード、自身が前記第 2 のリンクの前記始点に対応する前記複数の被制御装置の一つである場合、前記第 2 のリンクの前記終点に対応する前記複数の被制御装置の別の一つを待ノードとして、前記前ノード、前記次ノード、および前記待ノードに関する情報を記憶し、

前記資源循環グラフの前記ノードの各々は、前記前ノード、前記次ノード、および前記待ノードに関する情報が付与されている、

付記 6 または 7 に記載の分散データ処理プログラム。

(付記 14)

さらに、

付記 6 または 7 に記載の分散データ処理プログラム。

10

20

30

40

50

前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送る処理をコンピュータに実行させることを特徴とする、付記 1 1 乃至 1 3 のいずれか一項に記載の分散データ処理プログラム。

(付記 1 5)

前記デッドロック検出メッセージは、前記資源循環グラフに含まれる前記複数の被制御装置の優先度に関する情報を含み、

さらに、

前記デッドロックが発生していると判定したとき、前記デッドロックを解消するために、前記デッドロック検出メッセージに含まれる前記優先度に基づいて、前記資源循環グラフに含まれる前記複数の被制御装置の一つの前記制御権を有する前記複数の端末の一つに、前記制御権を解消させるための通知を送る処理をコンピュータに実行させることを特徴とする、付記 1 1 乃至 1 3 のいずれか一項に記載の分散データ処理プログラム。

10

【符号の説明】

【0 1 9 8】

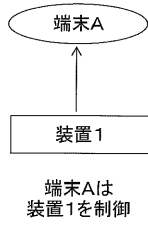
- 2 0 分散データ処理装置
- 2 0 0 デッドロック検出部
- 2 0 0 0 デッドロック判定部
- 2 0 0 2 転送メッセージ生成部
- 2 0 2 通信部
- 2 0 2 0 ネットワーク受信部
- 2 0 2 2 ネットワーク送信部
- 2 0 4 デッドロック解消部
- 2 0 6 資源循環グラフ更新部
- 2 0 6 0 占有元端末管理部
- 2 0 6 2 資源追加・更新部
- 2 0 6 4 資源解放部
- 2 0 8 優先度決定情報データベース (D B)
- 2 1 0 資源循環グラフデータベース (D B)
- 2 1 2 占有元情報データベース (D B)

20

30

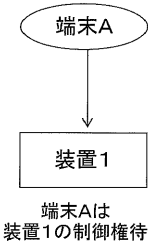
【 図 3 A 】

比較例における資源循環グラフの記法の例を示す図



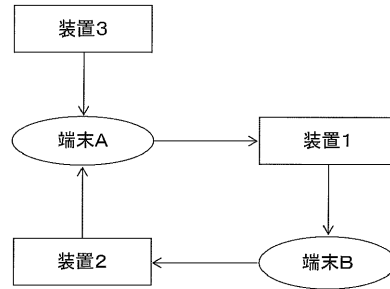
【 図 3 B 】

比較例における資源循環グラフの記法の別の例を示す図



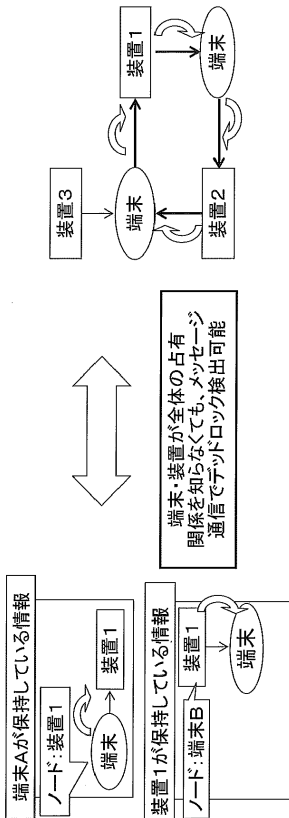
【 図 4 】

資源循環グラフによるデッドロック検出について説明するための図



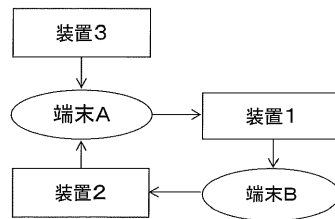
【 図 5 】

分散システム上でのデッドロック検出について説明するための図



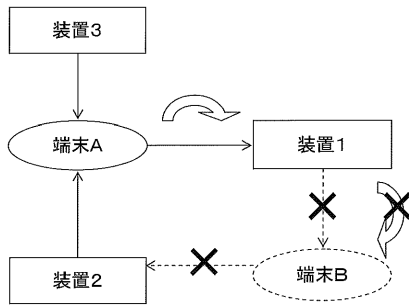
【 図 6 】

資源循環グラフの例を示す図



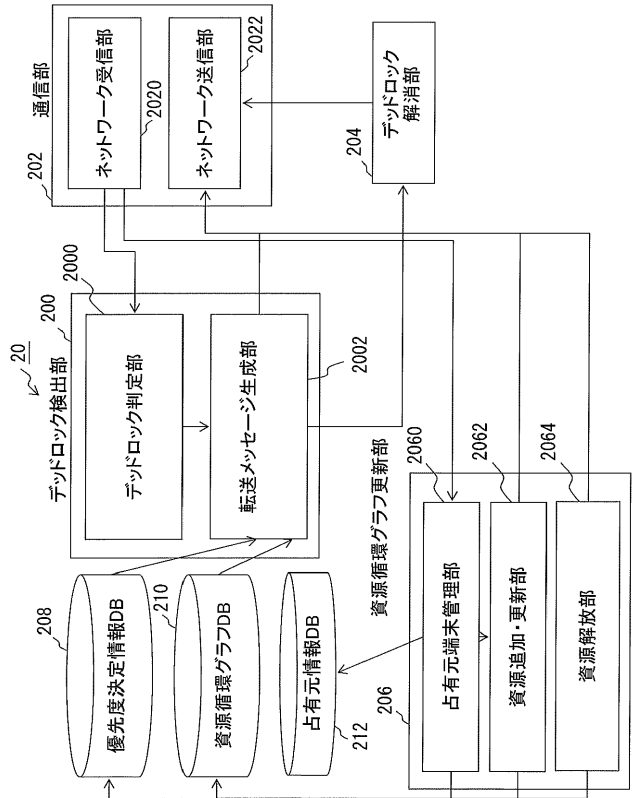
【 図 7 】

Chandy-Misra-Haasアルゴリズムでのデッドロック検出の例を示す図



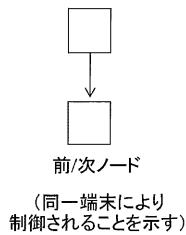
【 図 8 】

実施形態の分散データ処理装置の機能ブロックの例を示す図



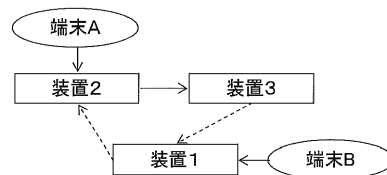
【 図 9 A 】

実施形態の資源循環グラフの記法の例を示す図



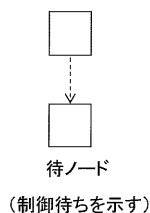
【 図 1 0 】

ライン型トポロジの資源循環グラフの例を示す図



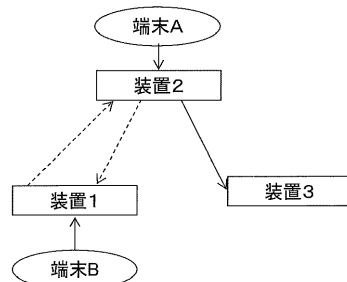
【 図 9 B 】

実施形態の資源循環グラフの記法の別の例を示す図



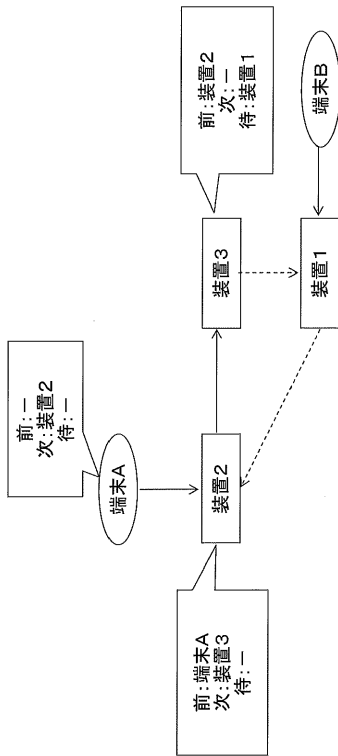
【 図 1 1 】

スター型トポロジの資源循環グラフの例を示す図



【 図 1 2 】

実施形態における資源循環グラフの例を示す図



【 図 1 3 】

前・次・待ノード情報の例を示す図

前・次・待ノード情報

前ノードURI	次ノードURI	待ノードURI
device001	device003	device004, device005

【 図 1 4 】

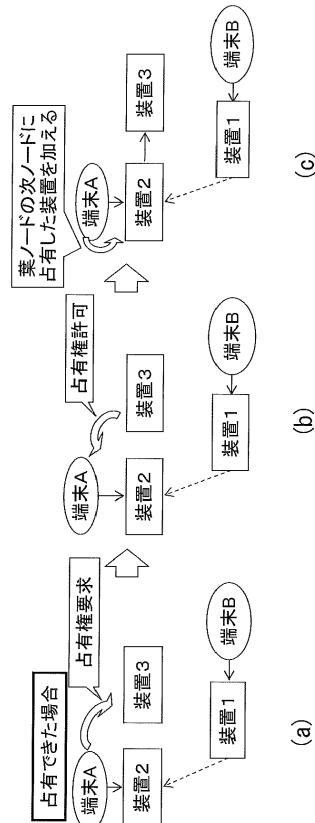
デッドロック検出メッセージの例を示す図

デッドロック検出メッセージ

メッセージ発行元URI	メッセージ送信元URI	メッセージ送信先URI	既訪ノードフィールド・優先度情報
device001	device005	device006	(device002, appA, 2013/11/22 10:20:30), (device003, appB, 2013/11/22 15:20:40)

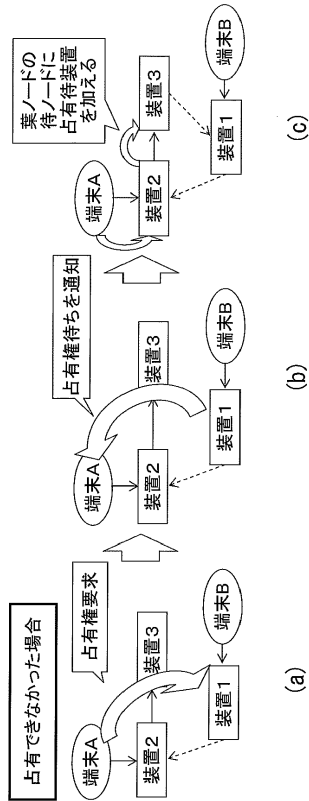
【 図 1 5 】

装置の制御権取得時の資源循環グラフの変形の例を説明するための図



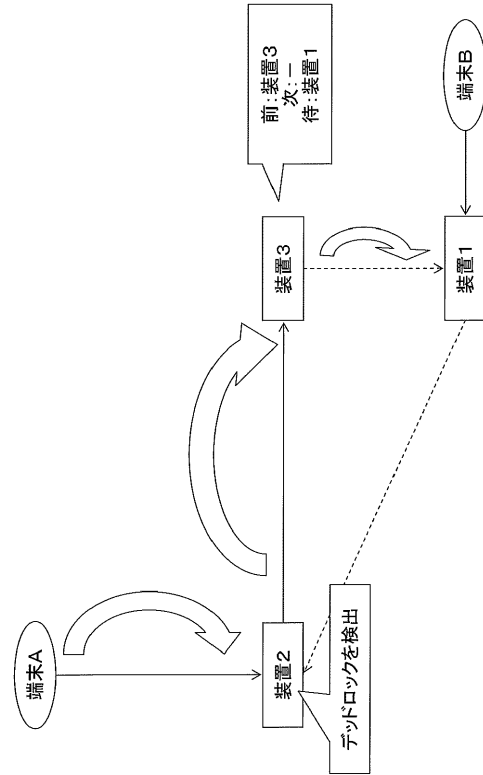
【 図 1 6 】

装置の制御権取得時の資源循環グラフの変形の別の例を説明するための図



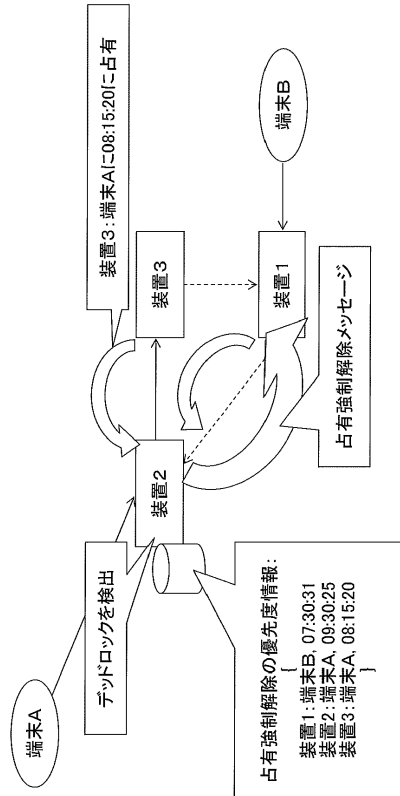
【 図 1 7 】

実施形態の資源循環グラフを用いたデッドロック検出の例について説明するための図



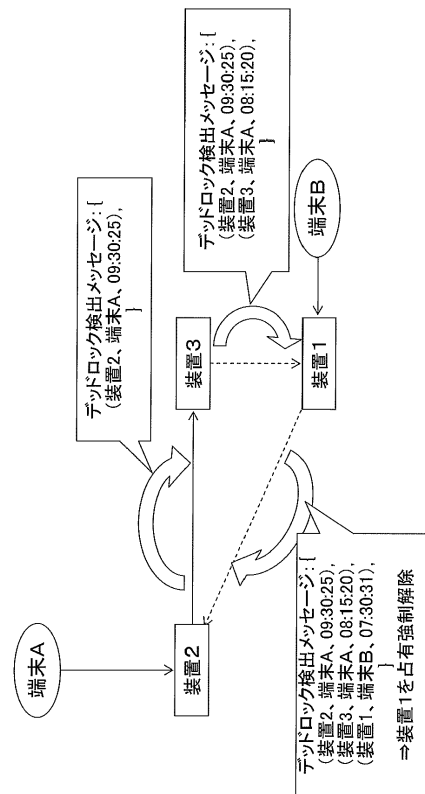
【 図 1 8 】

実施形態の資源循環グラフを用いたデッドロック解消の例について説明するための図



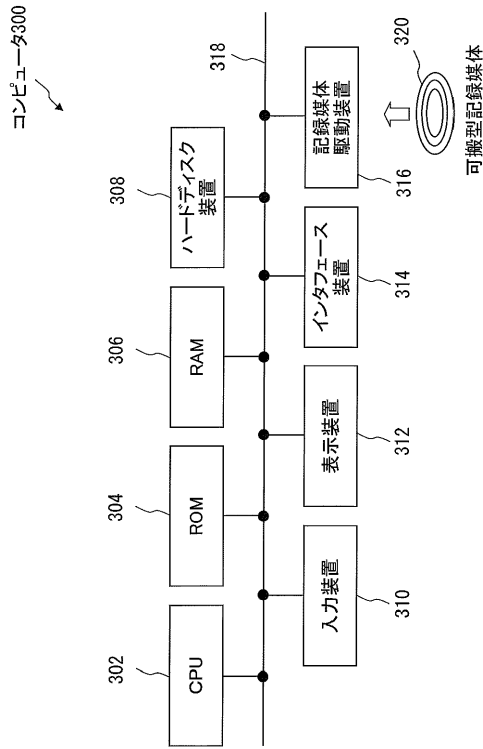
【 図 1 9 】

実施形態の資源循環グラフを用いたデッドロック検出と解消の例について説明するための図



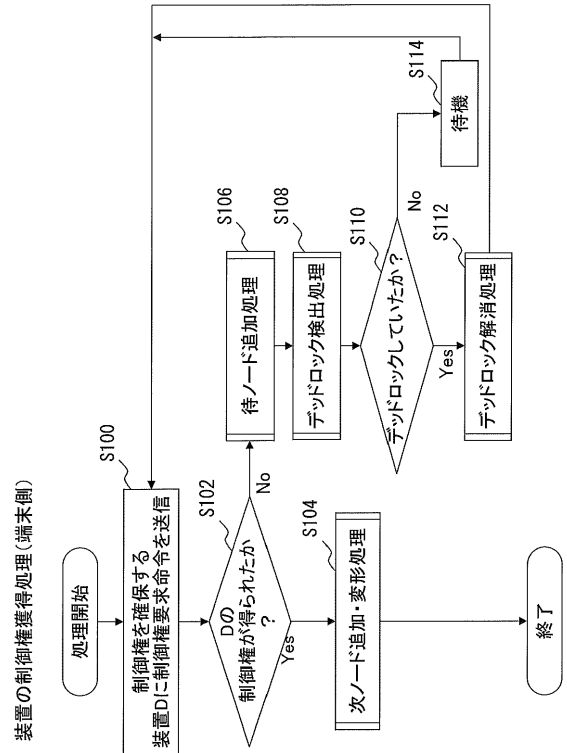
【図20】

実施形態の分散データ処理装置の構成の例を示す図



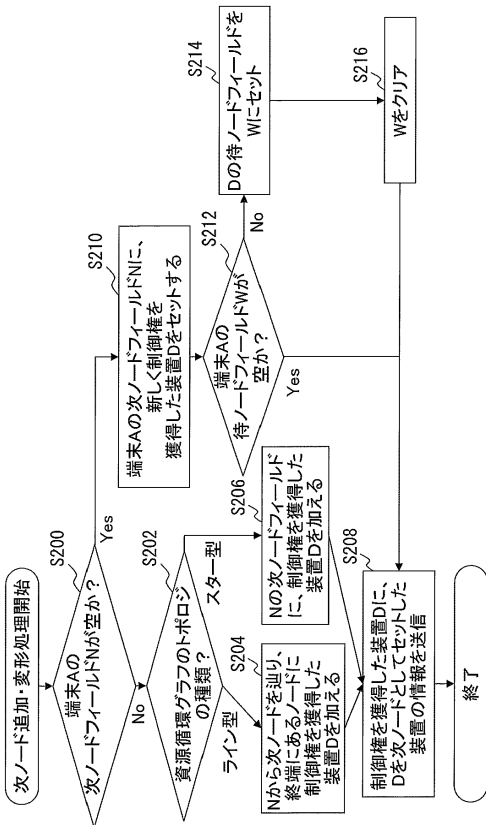
【図21】

端末側の装置制御権獲得処理における処理の流れの例を示す図



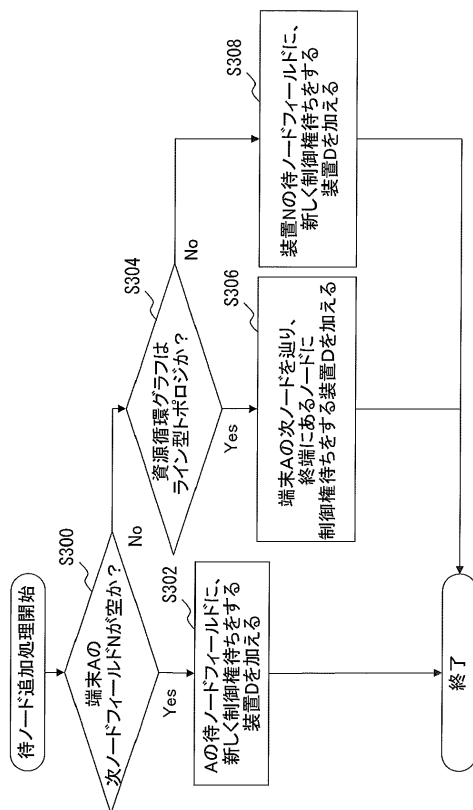
【図22】

端末側の次ノード追加・変形処理における処理の流れの例を示す図



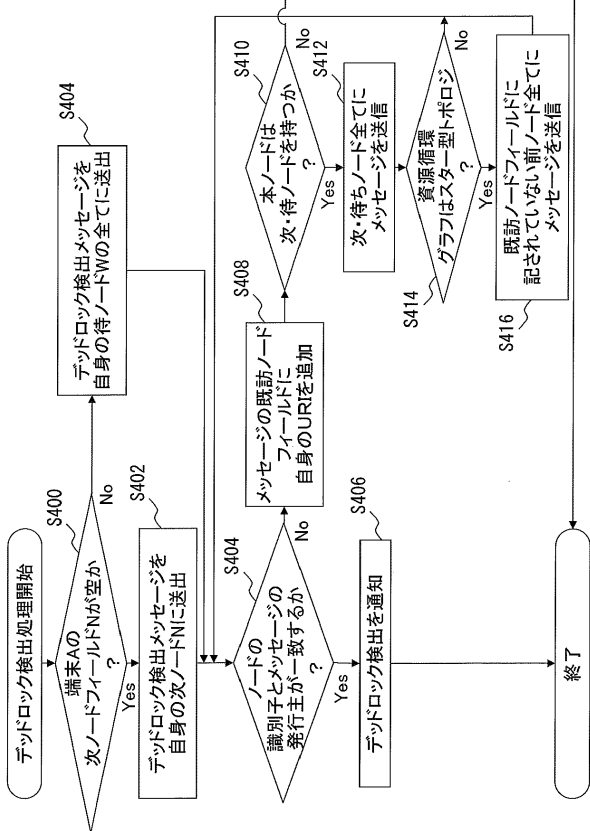
【図23】

端末側の待ちノード追加処理における処理の流れの例を示す図



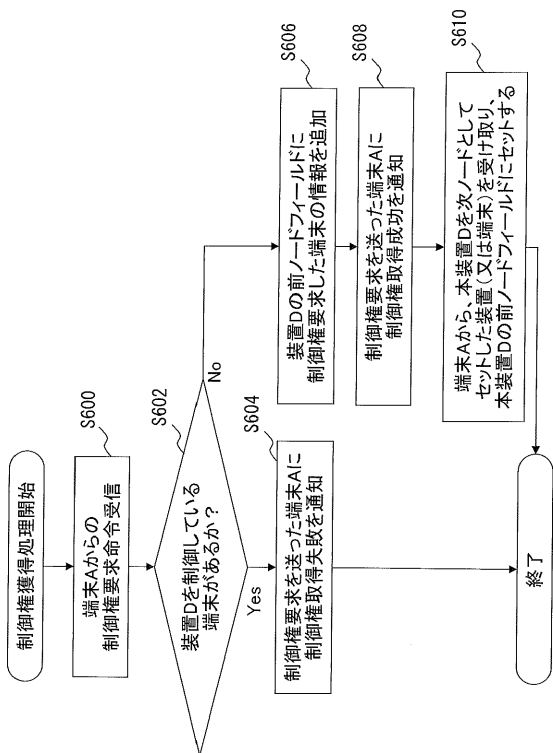
【 図 2 4 】

実施形態の資源循環グラフを用いたデッドロック検出処理における処理の流れの例を示す図



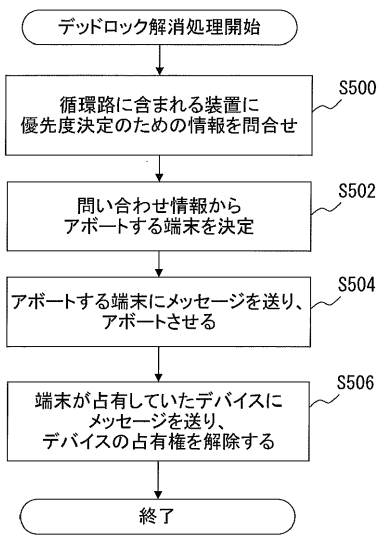
【 図 2 6 】

装置側の装置制御権獲得処理における処理の流れの例を示す図



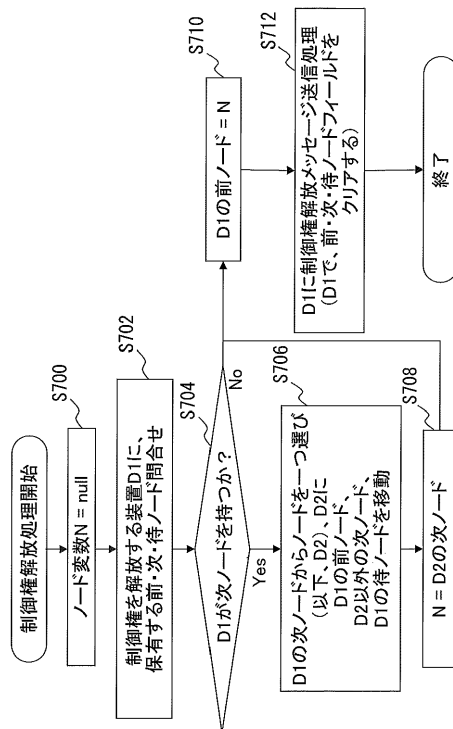
【 図 2 5 】

実施形態の資源循環グラフを用いたデッドロック解消処理における処理の流れの例を示す図



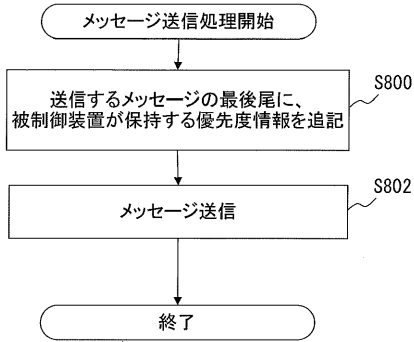
【 図 2 7 】

端末側の装置制御権解放処理における処理の流れの例を示す図



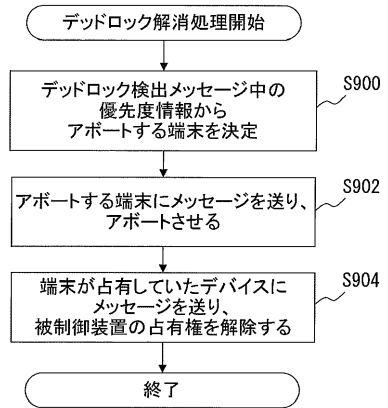
【図28】

デッドロック検出メッセージ送信処理における処理の流れの例を示す図



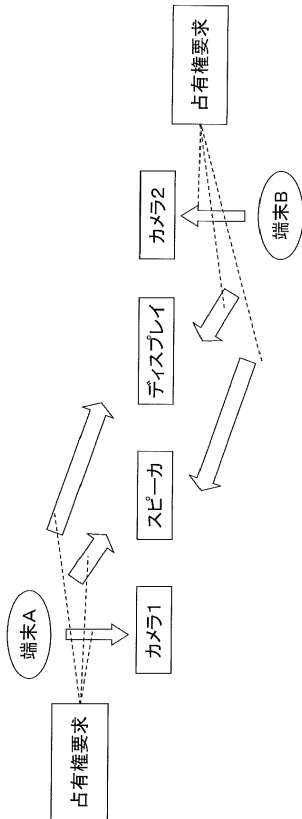
【図29】

装置側のデッドロック解消処理における処理の流れの例を示す図



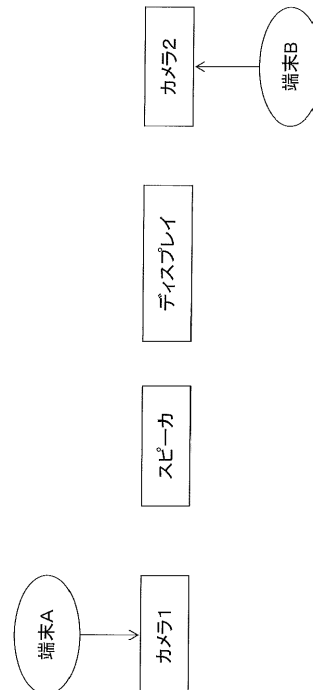
【図31A】

デッドロックの検出、解消の例について説明するための図(その1)



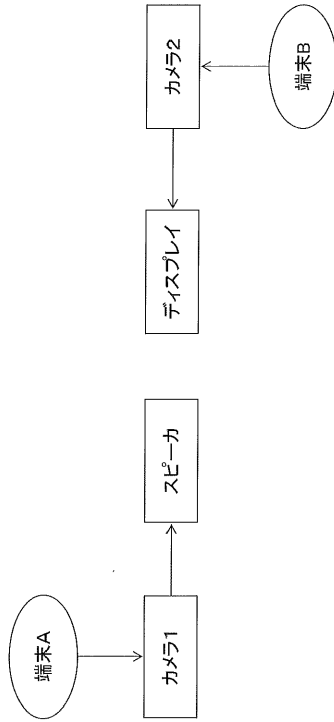
【図31B】

デッドロックの検出、解消の例について説明するための図(その2)



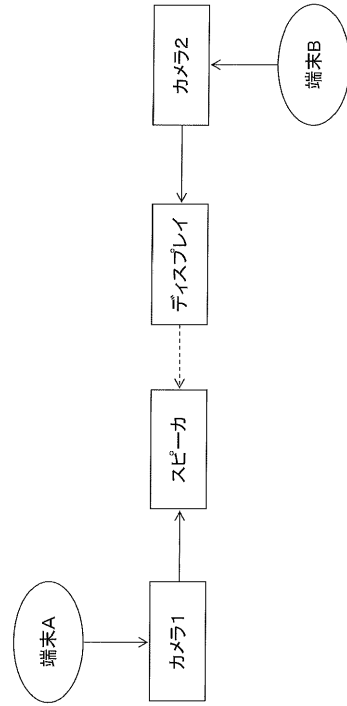
【図 3 1 C】

デッドロックの検出、解消の例について説明するための図(その3)



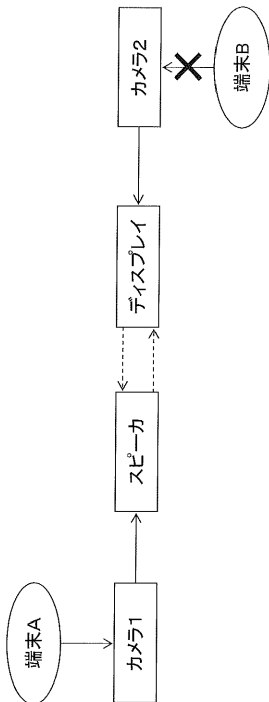
【図 3 1 D】

デッドロックの検出、解消の例について説明するための図(その4)



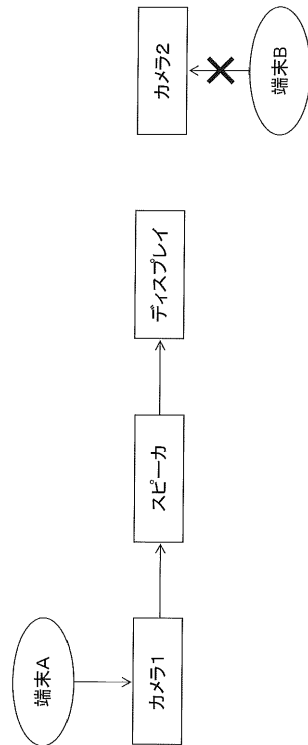
【図 3 1 E】

デッドロックの検出、解消の例について説明するための図(その5)



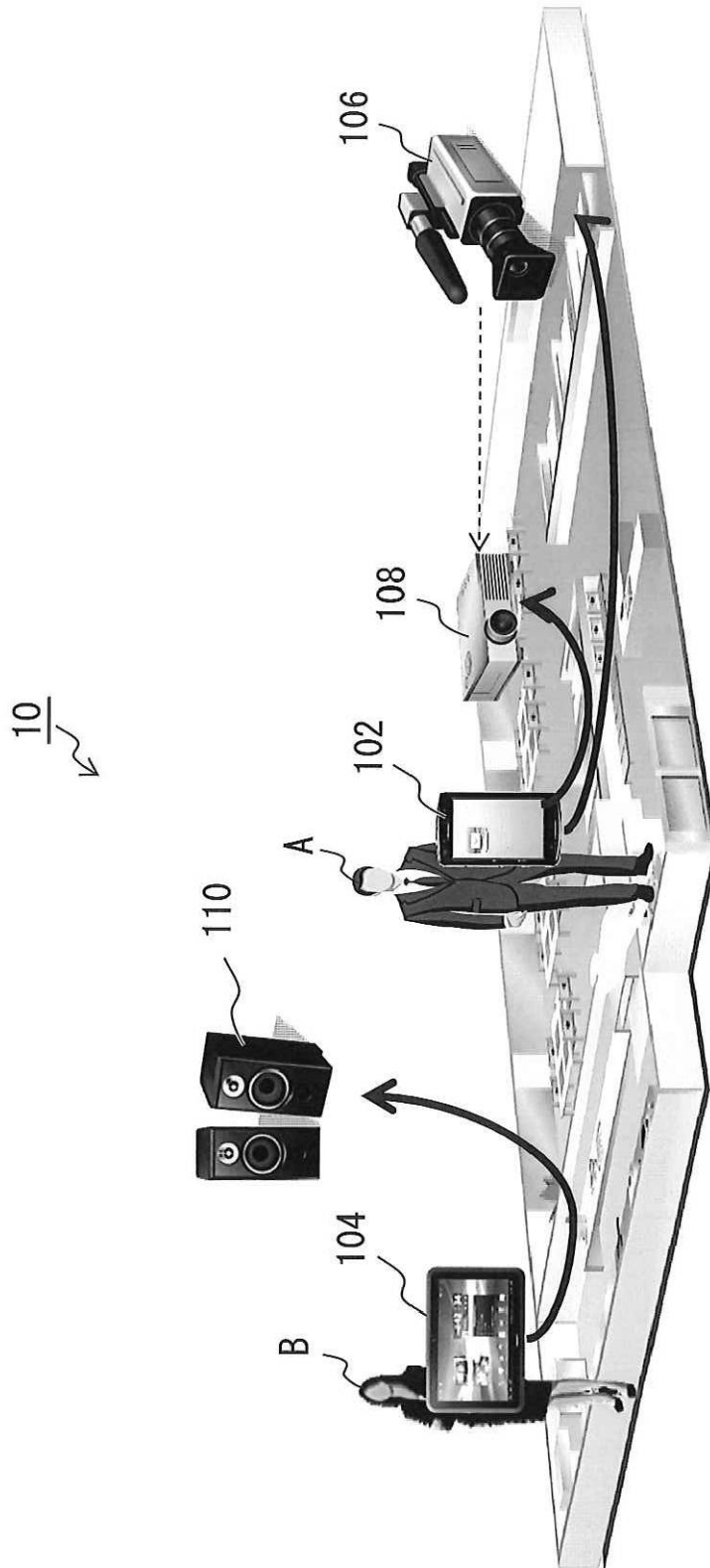
【図 3 1 F】

デッドロックの検出、解消の例について説明するための図(その6)



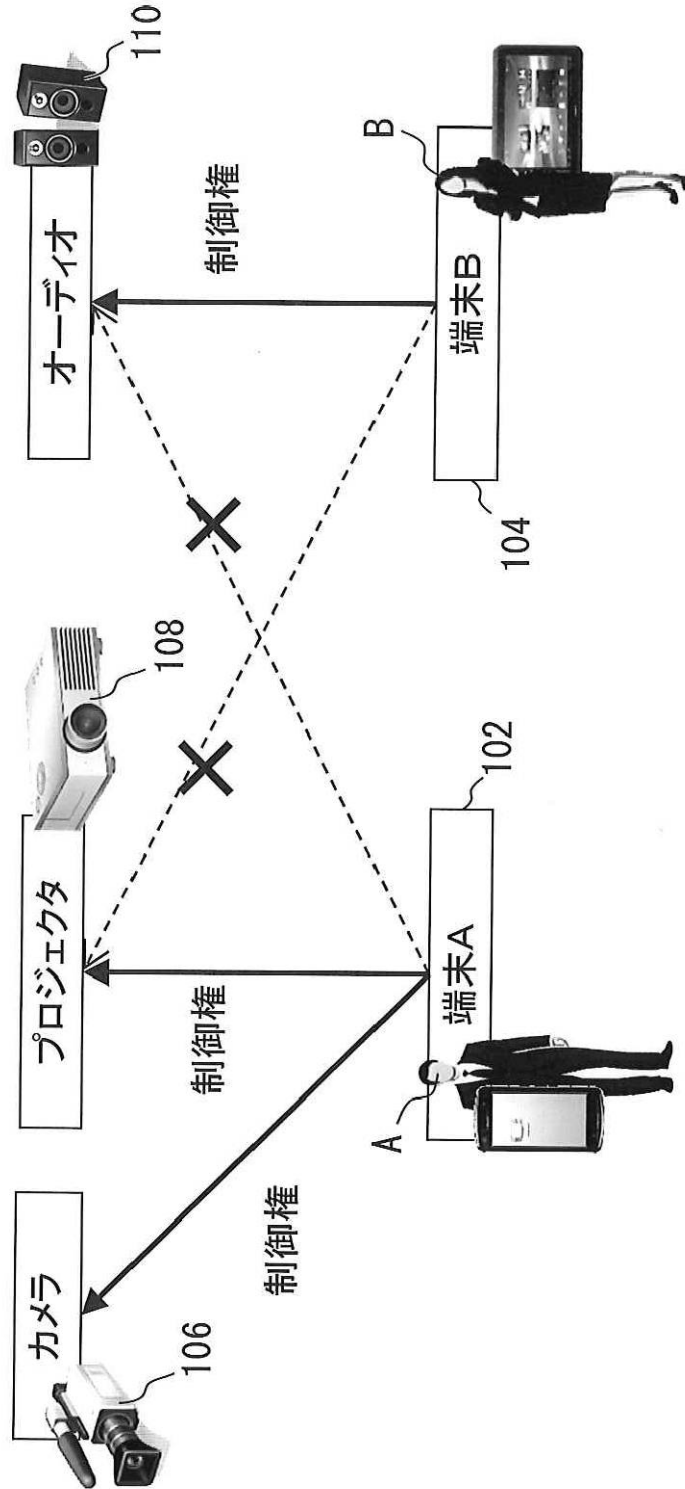
【図 1】

端末から被制御機器を操作するシステムの例を示す図



【図2】

端末間のデッドロックについて説明するための図



【図30】

端末から被制御機器を操作するシステムの別の例を示す図

