

(19) 日本国特許庁 (JP)

## (12) 特 許 公 報 (B2)

(11) 特許番号

特許第4079335号  
(P4079335)

(45) 発行日 平成20年4月23日 (2008. 4. 23)

(24) 登録日 平成20年2月15日 (2008. 2. 15)

(51) Int. Cl.

F I

H 0 4 L 12/28 (2006. 01)

H 0 4 L 12/28 2 0 0 Z

請求項の数 5 (全 23 頁)

(21) 出願番号	特願平10-518602	(73) 特許権者	トムソン コンシューマ エレクトロニクス インコーポレイテッド
(86) (22) 出願日	平成9年10月16日 (1997. 10. 16)		アメリカ合衆国 インディアナ州 4 6 2
(65) 公表番号	特表2001-508605 (P2001-508605A)		9 0 - 1 0 2 4 インディアナポリス ノース・メリディアン・ストリート 1 0 3
(43) 公表日	平成13年6月26日 (2001. 6. 26)		3 0
(86) 国際出願番号	PCT/US1997/018750	(74) 代理人	弁理士 木越 力
(87) 国際公開番号	W01998/017033		
(87) 国際公開日	平成10年4月23日 (1998. 4. 23)	(74) 代理人	弁理士 青山 耕三
審査請求日	平成16年10月5日 (2004. 10. 5)		
(31) 優先権主張番号	60/028, 651	(72) 発明者	ホーランドー, カール フランシス
(32) 優先日	平成8年10月16日 (1996. 10. 16)		アメリカ合衆国 インディアナ州 インディアナポリス グラツシー・クリーク・コート 2 7 1 3
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	60/031, 249		
(32) 優先日	平成8年11月12日 (1996. 11. 12)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ネットワーク・バス上で接続された装置を制御する方法

## (57) 【特許請求の範囲】

## 【請求項 1】

(a) ネットワーク・バス上で接続された第 1 の装置を制御する方法であって、  
 (b) 前記ネットワーク・バス上の前記第 1 の装置と相互接続された少なくとも第 2 の装置と通信し、それによって前記第 1 の装置と前記第 2 の装置の間の関係を形成するステップと、  
 (c) 少なくとも 1 つの動作モードが前記ネットワーク・バスを介して前記第 2 の装置によって制御可能な前記第 1 の装置の動作モードを制御するステップと、  
 (d) 前記関係の持続性に関する情報と前記関係を終了させる条件に関する所定の条件とを記憶するステップと、  
 (e) 第 3 の装置から問い合わせデータを受信するステップと、  
 (f) 前記関係の持続性に関する情報と前記問い合わせデータとを処理するステップと、  
 (g) 前記終了条件および前記問い合わせデータのうちの一方に応答して前記第 1 の装置と前記第 2 の装置との間の関係を終了させるステップと、  
 からなり、  
 前記第 1 の装置と前記第 2 の装置との間の前記関係の持続性に関する情報が少なくとも 2 つの異なる持続性レベルの 1 つからなる、前記方法。

## 【請求項 2】

前記第 1 の装置と前記第 2 の装置との間の関係を終了させるステップは、前記第 1 の装置

と前記第 3 の装置の間に新しい関係を形成するステップを含むものである、請求項 1 に記載の方法。

【請求項 3】

さらに、前記関係を終了させるステップは、許可を得るために前記第 2 の装置にポーリングするステップに応答して実行される、請求項 1 に記載の方法。

【請求項 4】

さらに、前記第 2 の装置の状態をモニタするステップを含む、請求項 3 に記載の方法。

【請求項 5】

さらに、前記第 1 の装置に含まれている情報を変更するよう前記第 2 の装置をイネーブルするステップを含む、請求項 4 に記載の方法。

【発明の詳細な説明】

発明の属する技術分野

本発明は、例えばデジタル・データバスのような相互接続を介して、例えば消費者用電子装置等のような複数の装置（デバイス）の間で通信を行うためのシステムに関する。また、本発明は、特に、そのような装置の間のインターオペラビリティ（相互運用性、相互操作性、相互情報交換性）を管理（統御）するための構成に関する。

発明の背景

例えばテレビジョン受信機または受像機、表示装置、ビデオカセット・レコーダ（記録装置）（VCR）、直接放送衛星（DBS）受信機、およびホーム・コントロール装置（例えばセキュリティ（安全）システムまたは温度制御装置）のような電子装置（デバイス）を相互接続するために、データバスを用いることができる。データバスを用いた通信はバス・プロトコルに従って行われる。バス・プロトコルの例として、コンシューマ・エレクトロニクス・バス（Consumer Electronics Bus）即ちCEBusおよびIEEE 1394 ハイ・パフォーマンス（高性能）シリアル・バス（High Performance Serial Bus）がある。

バス・プロトコルは、典型的には、制御情報とデータの両方の通信を行うために設けられる。例えば、CEBus 制御情報は、エレクトロニクス・インダストリーズ・アソシエーション（EIA、電子産業協会）規格書IS-60で規定されたプロトコルを有する“コントロール・チャネル（制御チャネル）”上で通信される。特定のアプリケーション用の制御情報は、CAL（共通アプリケーション言語、Common Application Language）として知られているプログラミング言語の形式を用いて規定することができる。

消費者用電子装置は、ますます複雑になり、増加し続ける多数の機能を備えている。完全なオーディオ・ビデオ（A/V）システムを構築するには、それらの複雑な装置を共にデータバスを介して結合する必要がある一方で、その構築には多数の問題が生じる。例えば、1つの装置の或る複数の機能が、データバスに結合される1つまたは2つ以上の装置との対話（インタラクション）を要求することがある。別の装置における特定の動作を完了または完成（完結）させるためには、1つの装置の或る機能が必要かもしれない。種々の装置の各要求の間に競合（コンフリクト、衝突）が生じることがある。欧州特許出願0604166は、伝送信号のデータフィールドにおいて規定されているよりも多いデータの伝送を中断することなく実施することのできる通信システムを開示するものである。

現在、2つ以上の装置が共通の資源（リソース）の使用、例えばネットワーク・バス上に接続された第3の装置の使用について（その使用を求めて）競合（compete）したときには、時間的に最初のものがその装置を制御する。その最初の装置が共通の資源の捕捉を解除するまでは、他の装置は共通の資源にアクセスすることができない。その場合でも、単純なネットワーク、例えばテレビジョン、ケーブルボックス（cable box）およびVCRで構成されたネットワークにおいては問題が生じない。そのような単純なネットワークでは、VCRはケーブルボックスによって“制御”され、即ちVCRはケーブルボックスが供給しているものだけを記録する。しかし、もっと複雑なネットワークでは、2つの装置、例えばケーブルボックスとデジタル放送受信機が、共用の資源、即ちVCRの制御権について（制御権をめぐる）競合することがある。

### 発明の概要

多くの制御アプリケーションにおいては、装置（デバイス）に対する一般的なアクセスを区別（差別）することが望ましい。この区別は、通常、１つの装置に便利性を与える。そのような場合、短時間（短期間）または長時間（長期間）の間、２つの装置の間に“ ロッキング（locking、閉錠）”といわれる特別な関係（relationship）が成立する。ロッキングの関係は、１つの装置が第２の装置（即ち“ ロックされた装置 ”）に対するアクセスを制御することを可能にする。

一般的に、２つのタイプのロッキング方法（アプローチ）がある。第１の方法では、ロックされた装置が、自己が別の装置に対してアンロック（unlocked、開錠）され得るか（非ロック状態にできるか）および／またはロックされ得るか（ロック状態にできるか）を判断する。その装置をロックすることを求める要求は、そのロックされた装置に対して直接行われる。そのロックされた装置が新しい要求を満たすことができない場合には、ロックされた装置はそれをロックした装置（即ちロッキング装置）をポーリング（poll）して、現在のロック状態を終了できるかどうかおよび新しいロッキングを完成させる（consume）ことができるか否かを確認する。そのロッキング装置が応答しない場合には、例えば失敗（failure）した場合には、ロックされた装置はそのロックを終了させて新しいロックを設定する。第２の方法においては、別の装置をロックしようとする装置が、ネットワークを通して、所望の装置とロックを形成している任意の装置に対してロックを終了させることを求める要求を伝達（ブロードキャスト、放送）する。

### 発明の概要

一般的に、本発明によれば、第１の電子装置は、データバスに結合するための手段と、データバスに結合された第１の電子装置と第２および第３の電子装置の間でのデータバスを介した通信を制御する制御手段と、を具えている。この制御手段は、それら装置間の相互の通信関係を管理（統御）する第２および第３の装置から受取った制御および問合せ（クエリ、質問）データを処理する。

本発明の１つの特徴によれば、ネットワーク・バスによって相互接続された少なくとも第２の装置と通信する通信手段を有する装置（第１の装置）が設けられており、それによって上述の上記第１の装置と上記第２の装置の間で関係が形成される。また、その通信手段は第３の装置から問合せデータを受取ることができるものである。さらに、その装置は、上記関係の持続性（persistence、パーシスタンス）に関する情報を記憶するための第１の手段と、上記関係を終了させるための条件に関する情報を記憶するための第２の手段と、上記第２の装置からの制御データにตอบสนองして上記装置の動作モードを制御する制御手段とを有する。この制御手段は、上記関係の情報および上記問合せデータを処理することができ、また、上記終了条件および上記問合せデータのうちの一方にตอบสนองして上記関係を終了させることができる。

本発明の別の特徴によれば、関係の情報は、上記関係が上記問合せデータにตอบสนองして終了する第１の状態と、上記関係が上記問合せデータに関係なく維持される第２の状態の２つの状態のうちの一方向を含んでいる。

本発明のさらに別の特徴によれば、上記問合せデータにตอบสนองする上記関係の終了は、上記消費者電子装置と上記第３の装置の間に新しい関係を形成することを含んでいる。

本発明のさらに別な特徴によれば、制御手段は、上記関係を終了させる許可を得るために、上記問合せデータの受領にตอบสนองして上記第２の装置をポーリングすることができる。

本発明のさらに別な特徴によれば、ネットワーク・バスによって相互接続された複数の装置を有するシステムが設けられている。各装置は、通信手段と、情報を記憶する第１と第２の手段と、上述のような上記装置の動作モードを制御する制御手段と、を有する。

本発明の方法の特徴によれば、ネットワーク・バス上で接続された消費者電子装置のための方法は、ネットワーク・バスによって相互接続された少なくとも第２の装置と通信し、上記消費者電子装置の動作モードを制御し、上記関係の持続性に関する情報と上記関係を終了させるための条件に関する所定の条件とを記憶し、第３の装置から問合せデータを受取り、上記関係の重要性（重要度、重み、意味）に関する上記情報と上記問合せデ

10

20

30

40

50

ータとを処理し、上記終了条件と上記問合せデータのうちの一方に応答して上記関係を終了させることからなる。

#### 【図面の簡単な説明】

図１Ａは、データバスを用いて互いに結合された電子装置のシステムをブロック図形式で示している。

図１Ｂは、図１Ａに示されているシステムの一部の詳細をブロック図形式で示している。

図２～図７は、図１に示されたシステムの動作を示すフロー図を示している。

図面において、異なる図面における同じ参照番号は同じまたは同様の機能を示している。

#### 発明の詳細な説明

図１Ａは、データバス１３０を介して互いに結合された、参照番号１１０、１００および１２０によって識別される３つの電子装置（デバイス）Ａ、ＢおよびＣを含んだシステムを示している。装置Ａ、Ｂ、およびＣは、例えばオーディオおよび／またはビデオ信号処理装置およびホーム・コントロール装置のような消費者用電子装置を含んだ種々のタイプの電子装置であればよい。消費者用電子装置の具体例には、テレビジョン（ＴＶ）受信機（受像機）、表示装置、ケーブルボックス、ビデオ・テープレコーダ（ＶＴＲまたはＶＣＲ）、直接放送衛星受信機（例えばＤＳＳ衛星受信機）、コンパクト・ディスク（ＣＤ）ユニット、デジタル・ビデオディスク（ＤＶＤ）ユニット、等が含まれる。ホーム・コントロール装置には、セキュリティ（安全）装置（例えば、カメラ、錠）、遠隔制御スイッチング装置および環境制御装置（例えば、暖房、冷房）等が含まれる。本発明の原理を取入れたシステムは、図１Ａに示されているような３つの装置より多い数の装置を含んでもよい。また、各装置は、相異なるタイプの装置でもよい。例えば、本発明によるシステムは、ＴＶ、ＶＣＲ、ケーブルボックス、衛星受信機、オーディオ・システムおよびホーム・コントロール装置を含めた家庭全体の多数の装置を含むものであってもよい。

図１Ａに示された例において、データバス１３０は、各装置間のデータおよび制御情報を通信するための複数の信号バス（経路）を含んでいる。このバス構成は、シリアル（直列）でもまたはパラレル（並列）でもよい。データバス上のデータ通信は、例えば上述のＣＥＢｕｓまたはＩＥＥＥ１３９４のプロトコルのような種々のバス・プロトコルの中の任意のものを用いて構成することができる。バス１３０を介して通信されるデータはビデオおよびオーディオ・データを含んでいる。例えば、デジタル・ビデオ信号を受信する衛星受信機は、バス１３０を介してデジタルのビデオおよびオーディオ・データを、記録用のデジタル・テープレコーダ（ＤＴＲ）に転送することもできる。

装置１００は、装置１００をバス１３０に結合するためのバス・インタフェース・ユニット１０２を含んでいる。バス・インタフェース１０２は、バス１３０からデータを受信するのに適し、またバス１３０上にデータを送信するのに適したハードウェアを含んでいる。例えば、インタフェース１０２は、バス１３０からデータを受信して、装置１００がデータ処理準備完了状態になるまで一時記憶し、またバス１３０上での送信が準備完了状態になるまでデータを記憶するレジスタまたはバッファメモリを含んでもよい。

インタフェース１０２は制御マイクロコンピュータ（μＣ）１０４によって制御される。用語“マイクロコンピュータ”は、１つまたはそれ以上の集積回路および／または別個の複数の構成要素を用いて実装構成されたマイクロプロセッサ、マイクロコンピュータ、制御器（コントローラ）および制御コンピュータのような装置を含めた種々の装置を表すことを意図しているが、これらの装置に限定されるものではない。マイクロコンピュータ１０４は、装置１００内におけるインタフェース１０２と、例えば信号処理ユニット（図１Ａには図示せず）のようなその他のユニットとの間でデータを渡す処理を制御する。例えば、装置１００がＤＴＲであり、バス・インタフェース・ユニット１０２が記録用のデジタル・ビデオデータを受信する場合には、μＣ１０４は、インタフェース１０２が、装置１００内の、意図通りに記録を行うような信号処理回路にビデオおよびオーディオ・データを渡すことを保証する。また、μＣ１０４は、インタフェース１０２が、適切なバス・プロトコルに従ってバス１３０を介してデータを通信するように、インタフェース１０

10

20

30

40

50

2を制御する。

マイクロコンピュータ104は、メモリ106に記憶された制御プログラムおよびデータによる制御の下で所要のバス・プロトコルを実行する。メモリ106は、例えば、RAM、ROM、EEPROMを含む種々のタイプのメモリ、およびハード・ドライブまたはその他のタイプのバルク(bulk)記憶装置を含んでいる。制御プログラムは、バス・インタフェース102の機能、例えば通信をイネーブルまたはディセーブル(disable)し、装置100内の他の動作に対してバス動作を調整し、バス通信のデータをフォーマット化する等の各機能を制御するためのルーチンを含んでいればよい。

μC104によって行われるバス・インタフェース102の制御の1つの特徴は、装置100内の資源の利用に対する競合する複数の要求を解決(resolve)する方法を決定することである。例えば、装置110がケーブルボックスであり、装置120がDSS衛星受信機であり、ユニット100がビデオ・テープレコーダ(VTR)である場合を想定する。ユーザ(使用者)は、特定の時間にケーブルボックス110を介して特定のプログラム(番組)の記録を開始するようにVTR100をプログラムすることができる。VTR100がケーブルボックス110を介してプログラムを記録している間に、それと競合する命令(コマンド)が、DSSユニット120にVTR100に対するアクセスを試行させる。通常のシステムでは、DSSユニット120により試みられるアクセスを無視することによって、またはVTR100が利用できない状態にあることをユーザ(使用者)に伝えることによって、上述の問題を解決する。

本発明の特徴によれば、図1Aに示されたシステムは、後で詳細に説明するような方法で資源に対する競合を解決するロック・マネージャ(管理装置)を実現する。図1Aに示された典型例の実施形態において、μC104は、バス制御プログラム内に含まれている“ロック・マネージャ”ルーチンによる制御の下で動作して、装置100が後で詳細に説明する方法で競合を確実に解決するようにする。“ロック機構”1070のルーチンもバス制御プログラム内に含まれており、ロック機構1070の動作については後の図2の詳細な説明において説明する。ロック・マネージャは、バス130に結合される各装置間で通信関係またはロッキング条件(コンディション)を設定し、またその関係がどのような条件で変更されるかを決定するように構成されている。ロック・マネージャの動作には、後で詳細に説明するような、メモリ106の部分107に記憶された各パラメータが必要である。競合を解決するときに必要なロック関係に関連するパラメータは、図1Aにおけるメモリ106のロック・マネージャ領域107内に記憶される。そのような3つのパラメータは、パーシスタンス(持続性)、ロックド(ロックされた)アドレス、およびアンロック(開錠)条件である。図1Bは、メモリ領域107における別々の3つのエントリー(項目)1071、1072、および1073に記憶されたその3つのパラメータを示している。図1Aには示されていないが、装置110および120はそれぞれ、ユニット100内のユニット102、104、106、および107によって形成されるものとして記載した機能と同様の機能を形成する各機能を含んでいる。ユニット110および120内の構成は装置100内に示された構成と同様でもよいが、同様の機能は他の構成を用いて形成することもできる。

ロック・マネージャの動作の一例として、上述の競合の例について考える。ロック・マネージャを用いて、ケーブルボックス110からのプログラムをVTR100を用いて記録するには、ロック要求をVTR100に送信するケーブルボックス110のバス制御機能が必要である。このロック要求は、VTR100に対して、ケーブルボックス110との或る関係、即ちロック条件を設定するように依頼するものである。マイクロコンピュータ104は、バス・インタフェース・ユニット102を介してロック要求を受信し、VTR100が既に別の装置にロック状態されている否かを判断する。否であれば、即ち、VTRがロッキング可能である場合には、VTR100は、ロック機構1070において、ロックが設定されたことを示す応答をケーブルボックス110に送信する。ケーブルボックス110からの所望のプログラムの記録が進行し、そのロック関係によって、上述のパーシスタンス(持続性)およびアンロック条件のようなパラメータによって規定される或る

10

20

30

40

50

条件以外の場合にはその記録動作への割込みが防止される。この或る条件は、後で詳細に説明するようにロック関係に関連するものである。各パラメータの値は、ロッキング装置、即ちロック条件を開始するユニット（この例ではケーブルボックス 110）によって設定され、ロックされた装置（この例における VTR）におけるメモリ 106 の領域 107 のそれぞれの部分に転送されて記憶される。DSS ユニット 120 が VTR 100 の使用を要求するときには、DSS ユニット 120 内のロック・マネージャ機能が VTR 100 にロック要求を送る。先に形成された現在のロック関係が VTR 100 とケーブルボックス 110 の間に存在するので、VTR 100 内のロック・マネージャは、VTR 100 に記憶された現在のロック条件のパラメータに従って DSS 120 からのロック要求を処理し、後で説明する方法で、競合する要求を解決（処理）する。

10

図 2 は、ロック・マネージャ 107 と協調動作するロッキング機構 1070 の動作を示している。ロック・マネージャ 107 は、“マスター装置”が開始する関係またはロックを管理する。ロック機構 1070 は、2つの装置のロッキングを共に実行する。マスター（主）装置がロックの開始を希望するときには、マスター装置はスレーブ（従）装置に命令を送信する。装置はロック制御に関係する種々の状態の中の任意の 1つの状態になっていればよい。そのような 4つの状態、アンロック状態、ロック状態、アンロック要求のペンディング（未処理）状態、およびロック・ポインタ変更状態が、図 2 における楕円 210、220、230、および 240 によって示されている。各状態間の遷移は各楕円部分間をつなぐ線によって示されている。図 2 の上部に示されているように、ロック・マネージャは、ロックおよびアンロック要求を含む各要求を受取る。図 2 の下部に示されているように、ロック機構は、要求に応答して、“grant\_\_lock（grant\_\_lock、ロック許可）”または“lock\_\_in\_\_use（lock\_\_in\_\_use、ロック使用中）”のような応答を発生することができる。また、ロック機構は、アンロック要求および装置エラー/リセット指示（indication）を発生することができる。

20

図 2 に示されているように、このシステムは、ロック要求に応答してアンロック状態 210 からロック状態 220 に遷移して、grant\_\_lock（grant\_\_lock）応答を発生する。ロック状態 220 からアンロック状態 210 への復帰（リターン）は、ロックの解除に応答して発生する。ロック解除は、状態 230 と 240 を経由した遷移によって発生する。このシステムは、アンロック要求に応答して、状態 220 からアンロック要求ペンディング状態 230 に移行する。状態 230 において、システムは、パーシスタンス・パラメータおよびアンロック条件のようなパラメータを処理して、ロックが解除されるべきかどうかを決定する。ロックが解除（dissolve）されない場合には、lock\_\_in\_\_use（lock\_\_in\_\_use）メッセージが送信されて、システムがロック状態 220 に復帰する。ロックが解除できる場合には、アンロック・メッセージが発生されて、システムは状態 240 に遷移し、状態 240 において、ロックされていたアドレスまたはロッキング・ポインタが、現在ロックを要求している装置のアドレスまたはポインタに変更される。その後、システムはロック状態 220 に遷移して、grant\_\_lock（grant\_\_lock）メッセージを発生する。

30

種々の条件下における、即ち装置間の種々のタイプのトランザクションの期間におけるロック・マネージャ 107 およびロック機構 1070 の動作が、図 3 ~ 図 7 に示されている。図 3 において、装置 B がロックされておらず、装置 B は、装置 A からのロック要求に応答して装置 A とのロックを設定する。

40

図 4 において、装置 B は、前のロック要求に応答して装置 C にロックされており、装置 A からの競合するロック要求を受け取る。そのパーシスタンス・パラメータは、アンロッキングして装置 A と再ロッキングする前に、装置 C に対してアンロッキング承認（approval）を要求する必要があることを意味する 0（ゼロ）に設定されている。装置 B はアンロック条件パラメータの評価だけを行って、装置 C とのロックを解除するか否かを決定しなければならない。

図 5 において、装置 B は、装置 C からの前のロック要求に応答して装置 C にロックされており、パーシスタンス・パラメータが 1 に設定されている。装置 B は、“ディスインヘリ

50

ット・ロック (disinherit lock、ロック承継排除) ” 要求ともいわれるアンロック要求を装置 C に送信する。装置 C におけるロック・マネージャは、ロックを解除すべきか否かを決定する。図 4 において、装置 C は、ロックを維持するよう決定し、“リソース・イン・ユース (resource in use、資源使用中)” のメッセージを送信し、このメッセージは装置 B によって装置 A に中継される。

図 6 に示されているトランザクションは、装置 C の装置 B とのロックを解除することによって装置 C が装置 A のロック要求に応答することを除けば、図 4 におけるトランザクション (即ちパーシスタンス (持続性) = 1) と同様である。こうして、装置 C はアンロック・メッセージを装置 B に送信し、その後、装置 B は装置 A とのロックを許可する。

図 7 は装置 B が装置 C にロックされていて、そのパーシスタンス (持続性) が 0 (ゼロ) に等しい場合のトランザクションを示しており、装置 A は、全てのロッキング・マネージャに呼かけ (hail) て、装置 B とロックしたいという装置 A の希望を示す。装置 C は、装置 A に対して、装置 C が装置 B とのロックを解除することを通知する。次いで、パーシスタンス (持続性) パラメータが 0 (ゼロ) に設定されているので、装置 A は装置 B にロック要求を送信し、装置 B は装置 A にロックし、装置 B とのロックを解除するように装置 C に命令するメッセージを装置 C に送信することができる。

さらに、ロッキング装置またはマスター装置だけがロック・マネージャ 107 に要求すること、およびスレーブ装置またはロック可能な装置がロック機構 1070 を使用するだけであることは、本発明の範囲内の事項である。そのようなシステムの動作は本明細書に記載したシステムと同様である。

上述した事項に加えて、ロック・マネージャの特徴を含んだ本発明のロック・マネージャの特徴に関して、後でさらに詳細に説明する。単に便宜上、次の詳細な説明は、C A L プログラミング言語の使用に基づく態様を含んでいる。しかし、それに代えて別の任意のプログラミング言語を用いてもよいことに留意すべきである。

#### ロッキングを用いるアプリケーションの例

次に、上述したアプリケーションに加えて、ロッキング関係 (relationship) が必要なアプリケーションについて説明する。第 1 の例において、装置 (“装置 A”) は、幾つかのデータ・パケットを遠隔装置 (“装置 B”) にダウンロードすることを希望する。装置 A は、そのダウンロードが完了するまでは、装置 B 中に存在するデータが上書き (overwrite) されるのを好まない。通常、この機能は、両装置におけるセグメント化されたネットワーク・サービス (Segmented Network Service) を要求する。しかし、本発明に従って、装置 A は一時的ロッキング関係を設定し、この一時的ロッキング関係によって、装置 A が装置 B に書込むことのできる唯一の装置であることが保証される。データ転送の後、装置 A は、ロッキング関係を終了させ、装置 B を “アンロック” (開錠) する。

第 2 の例において、ケーブルボックスのような外部装置はプログラム・タイマ・イベント (事象) を実行する。ケーブルボックスは、V C R 輸送機構を “ロック” する。これによって、そのセッションが終了するかまたはオーバライド (無効化) 条件が発生するまで、V C R が記録モードに維持される。ケーブルボックスは、そのイベントが終了したときにストップ命令 (Stop) を V C R に送信して V C R をアンロックする。

第 3 の例において、装置は、或る期間に O S D メッセージを表示することを希望する。その装置は、ディスプレイ (表示装置) を自己にロックして、関連する表示情報を送信する。この表示装置は、或る期間に情報を表示する。ロックされた装置は、ロッキング装置自体によって、または例えばタイムアウト等の条件付きイベントによってアンロックされる。

また、特別な形式のロッキング、“リスニング (聴取)” も実行される。種々の装置を制御し、種々の装置とインターオペレーティング (相互運用、相互操作、相互情報交換) するには、それらの状態を知る機能が必要である。リスニング機能を含んでいる装置は、例えば家庭内に置かれた装置等の特別なグループの装置によって送信された各同報レポート (報告) 間の区別 (判別) を行う。

例えば、状態変更情報が或る装置から別の装置へと自動的に送信される。V C R が表示装

10

20

30

40

50

置に対してモーション（動き）モード状態を更新し、または温度センサが環境マネージャに対して更新情報を送信する。この方法（アプローチ）には2つの問題がある。第1の問題は、そのレポートの行き先を決定することである。コンパニオン（連係、協調）の問題は、受信装置がその機能にとってどのレポート情報が重要かを決定することである。このような状況が生じるのは、同じ装置タイプ、例えばサーモスタットまたはセキュリティ・センサ（安全検知器）のような同じ装置タイプの複数のインスタンス（事柄）が存在していて、同じ家庭内の種々のサブ・システムに報告するときである。各情報ソース（源）間の区別を行う必要がある。リスニング機能はこの機能を実現する。

リスニングの種々のアプリケーションが存在する。幾つかのアプリケーションは他のアプリケーションよりも動的（ダイナミック）である。環境制御は、安定な（静的）リスニング・アプリケーションの例である。そのようなシステムでは、リスニング機能は、初期設定機能の後は相対的に静的である。これらのシステムでは、リスニングを行うのに信頼性が高く安価な手段が必要である。例えば、セキュリティおよび特別なアクセスのような問題は重要である。このシステムの装置構成は、設置（インストール）のときに形成され、高頻度で変更されることはまれである。

これはA/Vシステムと全く対照的である。A/Vシステムは非常にダイナミック（動的）である。装置は、比較的高頻度で設置され取外される。ビデオ・ソース（源）および要求機能によって、システム装置構成が迅速に規則的に変化する。再生、記録、表示メッセージおよびタイマ・イベントの全てが、装置のリスニングおよびロッキングの変更を要求する。リスニング・ソース（聴取源）装置を瞬間的にVCRからDBS装置に変更する必要がある。また、非常に多くの装置間リスニングが存在する。ケーブルボックスがVCRの状態をリスニングしている間に、TVは、表示のために1つの装置、環境システムをリスニングすることができる。

種々のレベルの望ましいロッキングおよびリスニングが存在する。多くのインスタンス（事柄）において、アプリケーションは装置レベルのロッキングだけを要求する。しかし、オブジェクト（対象）レベルでロックすることが望ましい場合もある。（即ち、装置が2つ以上の（1つより多い）オブジェクトで構成されることがある。例えば、装置が、装置自体の内部における別々の制御可能な機能を有することがある。）このような1つの状況は、VCRアプリケーションにおいて、他の装置にタイマ・イベント（事象）をエディット（編集）または付加させている間に、トランスポート機構をロックすることが望ましい場合である。同様に、適正な表示を保証するために、TVにおけるディスプレイ・オブジェクトをロックすることが望ましいが、他の通信に応答する装置の機能をロックすることは望ましくない。

ロッキング属性（attribute）は、関係（リレーショナル）、条件付き（conditional）および継承（inherited）の各特性を有する。条件付きロッキング特性の例は、オブジェクト（object）が装置によって既にロックされているときである。アンロック状態のオブジェクトはロック可能である。しかし、その装置はロック可能でないかもしれない。そのような状況において、その装置は、オブジェクトの1つをロックした同じ装置によってのみロック可能である。継承されたロッキングでは、装置内の全てのオブジェクトがオブジェクトのロック状態（status）を継承すると推定される。

#### ロック・マネージャ

他の装置をロックすることを希望する全ての装置は、装置によって開始された活性状態の全てのロックを追跡し且つ遠隔装置をロッキングしアンロッキングする責務のある“ロック・マネージャ”を持っていなければならない。CALを用いて実装構成された、そのようなロック・マネージャの構成を次に説明する。



ロック・マネージャ (Lock Manager)				(16)データリスト・メモリ (DataList Memoery)
これにより、装置によって設定されるロックのための 中央クリアリング・ハウスが形成される。				
I V	R/W	タイプ	名称	コンテキスト機能
a (61)	R	n	number_of_locks (ロックの数)	メモリ・サイズ (size_of_memory)
b (62)	R	n	Length_of_lock _pointer (ロック・ポインタ の長さ)	レコードの長さ (length_of_record)
C* (43)	R/W	n	current_lock (現在のロック)	現在インデックス (current_index)
I* (6c)	R/W	d	lock_pointer (ロック・ポインタ)	メモリ・ブロック (memory_block)

上述の区分けされた各パラメータについて以下に説明する。

#### 1. ロック\_\_ポインタ (Lock\_\_pointer) 定義

データ・メモリ (Data Memory) のメモリ\_\_ブロック (memory\_block) インスタンス変数 (I V) の各レコード (記録) は、ネットワーク上の遠隔位置にあるロックされた装置を指すポインタである。ロック\_\_ポインタ (lock\_pointer) データは、3つのフィールド、(1) <デバイス\_\_ネットワーク\_\_アドレス> (<Device\_\_Network\_\_Address>)、(2) <ロックド\_\_コンテキスト\_\_ID> (<Locked\_\_Context\_\_ID>)、および (3) <ロックド\_\_オブジェクト\_\_ID> (<Locked\_\_Object\_\_ID>)、を含んでいる。左側の空欄のフィールドは、ナル (null: 空、0) と考えられる。

デバイス・ネットワーク・アドレス (Device Network Address) フィールドは、ロックされた装置 (デバイス) のネットワーク・アドレスである。C E B u s アプリケーションにおいて、M A C アドレスはネットワーク・アドレスとして使用される。

ロックド\_\_コンテキスト\_\_ID (Locked\_\_Context\_\_ID) フィールドは2つのバイトを含んでいる。第1のバイトはコンテキスト番号またはコンテキスト・ワイルドカード (cont

10

20

30

40

50

ext wildcard) “ D E ” の何れかである。 D E ワイルドカードは、ロックが装置中の同じコンテキストの全てのインスタンスまたは全ての例示 (instanciation) に対応することを示唆する。第2のバイトは、コンテキスト\_\_クラス\_\_ I D (Context\_\_Class\_\_ID) である。残りのバイトはナル (空) である。

このロックド\_\_オブジェクト\_\_ I D (Locked\_\_Object\_\_ID) フィールドは2つのバイトに分割されている。第1のバイトはオブジェクト\_\_ナンバー (Object\_\_Number) またはオブジェクト・ワイルドカード “ 0 0 ” の何れかである。オブジェクト・ワイルドカードが使用されるときには、第2のバイトはオブジェクト\_\_クラス (Object\_\_Class) である。ワイルドカードが使用されないときには、第2のバイトはナル (空、0) である。これは、同じオブジェクト・クラス (Object\_\_Class) 内の全てのオブジェクトのロックングすることを可能にする。

10

## 2. アンロック・メッセージ (Unlock Message) 定義

ロックされた装置は、アンロック・メッセージをロックング・マネージャに送信する。3つの定義されたアンロック・メッセージ、(1) デバイス\_\_アンロック (Device\_\_Unlock、装置アンロック)、(2) コンテキスト\_\_アンロック (Context\_\_Unlock)、および(3) オブジェクト・アンロック (Object\_\_Unlock)、が存在する。

アンロック・メッセージはマクロ (Macro)、U 4 を用いて送信される。U 4 マクロは2つのアーギュメント (argument、引き数、変数) フィールドを含んでいる。第1のアーギュメントはデバイス\_\_ネットワーク\_\_アドレス (Device\_\_Network\_\_Address、装置ネットワーク・アドレス) を含んでいる。第2のアーギュメントはコンテキストおよびオブジェクト情報を含んでいる。マクロ・プロトタイプ (原型) は次の通りである。

20

“ <context\_\_ID> <object\_\_ID> U4 <data token> <size of data> <escape token> <Device\_\_Network\_\_Address> F5 > <data\_\_token> <size of data> <escape token> <Context\_\_Number> <Context\_\_Class\_\_ID> <Object\_\_ID> <Object\_\_Class> “

〔訳、data token : データ・トークン、size of data : データのサイズ、escape token : エスケープ・トークン〕

コンテキストがロックされたときには、コンテキスト\_\_ナンバー (Context\_\_Number) およびコンテキスト\_\_クラス\_\_ I D (Context\_\_Class\_\_ID) が第2のアーギュメント・フィールドに置かれる (オブジェクト\_\_ I D (Object\_\_ID) およびオブジェクト\_\_クラス (Object\_\_Class) は与えられない)。オブジェクト情報はナルである。

30

また、オブジェクトがロックされたときには、コンテキスト\_\_ナンバー (Context\_\_Number)、コンテキスト\_\_クラス\_\_ I D (Context\_\_Class\_\_ID) およびオブジェクト\_\_クラス (Object\_\_Class) は、第2のアーギュメントで与えられる。

### a) 装置アンロック・メッセージ (Device Unlock Message)

デバイス\_\_アンロック (Device\_\_Unlock、装置アンロック) メッセージは、ロックされていた装置の <デバイス\_\_ネットワーク\_\_アドレス> (<Device\_\_Network\_\_Address>) を含んでいなければならない。第2のアーギュメントはそのメッセージ中に含まれていない。

“ 00 04 55 34 F5 F5 F4 34 F6<Device\_\_Network\_\_Address>F5 “

ロック・マネージャ・オブジェクト中に置かれたロックング・ポインティング・ベクトルの例

40

8

	デバイス・ネットワーク ・アドレス	Locked__Context__ID (ロケット・コンテキスト・ID)		Locked__Object__ID (ロケット・オブジェクト・ID)	
	1～4バイト目	5バイト目	6バイト目	7バイト目	8バイト目
ロック タイプ	Locked__Address (ロケット・アドレス)	Context __Number または Wild Card (コンテキスト・ ナンバー または ワイルドカード)	Context __Class __ID (コンテキスト・ クラス・ID)	Object __ID または Wild Card (オブジェクト ・ID または ワイルドカード)	Null または Object __Class (ナール または オブジェクト・ クラス)
装置 ロック	Locked__Address (ロケット・アドレス)				
コンテキスト ・ロック	Locked__Address (ロケット・アドレス)	A0	(9F)* 00.. 9E	01	
特定の コンテキスト	Locked__Address (ロケット・アドレス)	(DF)* A0.. DE	(9F)* 00.. 9E	01	
同コンテキ スト・クラ スの全 コンテキスト	Locked__Address (ロケット・アドレス)	DE	(DF)* A0.. DE	01	
装置中 の全コン テキスト	Locked__Address (ロケット・アドレス)	DE	00	01	
オブジェクト ・ロック	Locked__Address (ロケット・アドレス)	(DF)* A0.. DE	(DF)* A0.. DE	(3F)* 01 ..3E	<Any Value> (＜任意値＞)
コンテキスト ・クラス 中の全 オブジェクト	Locked__Address (ロケット・アドレス)	(DF)* A0.. DE	(DF)* A0.. DE	00	<Any Value> (＜任意値＞)

#### ロッキング装置

装置をロックする機能は、前に設定されたロックに依存する。装置のロッキングには、他の全てのロックを上書きすることが必要である。

ロッキング実装構成には次のテーブルにおける変数（I V）が使用される。ロッキング\_\_アドレス（Locking\_\_Address）I Vは全ての実装構成中に存在しなければならない。パースタンス（持続性）のデフォルト・レベルは“0”である。次のサブセクションはI Vの詳細な説明を含んでいる。

10

20

30

40

50

I V	R/W	タイプ	名称
B (42)	R/W	N	persistence (パ-シスタンス、持続性)
E (45)	R/W	D	Key__Entry (キー__エンリー)
G* (47)	R/W	D	Locking__Address (ロッキング__アド-レス)
O (4F)	R/W	D	Unlcok__Condition (アンロック__コンデ-ション)

10

a) パーシスタンス (持続性) I V

パーシスタンス (持続性) I V は装置のアンロックの振る舞いに影響を与える。現在、3つの定義されたパーシスタンス・レベル、(0) ルース (疎)、(1) タイト (密)、および (2) リスニング (聴取) が存在する。パーシスタンス (持続性) I V はルース (Loose) のデフォルト・レベルを有する。装置がアンロック状態に反転したときには、装置はパーシスタンスにデフォルト・レベルを設定する。

20

ルース・ロッキング・レベルはパーシスタンス・レベルが“0”のときに発生する。ルース・ロッキングの結果、装置はそれを制御しようとする各装置の間の区別 (差別) を行う。しかし、その結合 (binding) はルースである。これは、ロックされた装置が、前のロッキング装置が先に形成された現在のロックを解除しなければならないという確信の下で新しいロックに対する要求を許可することができることを意味する。ロッキング・レベルがルースであるときには、ロッキング装置は要求があったときに自動的にロックを解除する。その結果、前のロックは速やかに解除され、新しいロックが許可される。

30

ルースにロックされた装置は、ロッキング\_\_アドレス (Locking\_\_Address) を変更することを求める要求を受信すると、完了メッセージを返送 (リターン) して、前のロッキング装置に対してロックを解除するよう通知を送信する。ルース・ロッキングは任意の装置がロッキング\_\_アドレス I V (Locking\_\_Address IV) を変更できると推定する。

タイト・ロッキングは、パーシスタンス・フィールドが“1”に設定されたときに生じる。これによって、そのロッキング装置だけがロッキング\_\_アドレス I V (Locking\_\_Address IV) を変更するまたは変更する許可を与えることが可能になる。これは、第3の装置がロック状態の装置とのロックを形成することを希望するときには有用である。この場合、第3の装置はロック状態の装置に問合せる。このロック状態の装置は、ロッキング装置のロック・マネージャ・オブジェクト (Lock Manager Object) にアンロック・メッセージを送信する。このロッキング装置は、ロックを形成できるかまたはロックを形成できないかの何れかである。即ち、ロッキング装置は前のロックを解除するときに完了トークン (Completed Token)、F E を返送 (リターン) する。それ以外の場合には、ロッキング・マネージャは、エラー - 資源使用中の (8) コードを有する F D トークンを返送 (リターン) する。

40

パーシスタンス I V を“2”に設定すると、リスニングといわれるロッキングの形式が生じる (上述の通り)。リスニングにおいては、“ロックされた”装置は自己を別の装置に結合する。装置はそのアプリケーションに応じて種々のレベルのパーシスタンスを実現することができる。

50

## b) キー\_\_エントリー I V (Key\_\_Entry IV)

キー\_\_エントリー I V (Key\_\_Entry IV) は、ロックされた装置に入力される単純なキーとして作用する。この I V は保護された変数にアクセスすることを可能にする。このオプション (任意選択) は、例えば特別な診断動作およびセットアップ動作のようなプライベートな (個人的な) 機能を一般のアクセスから保護する。また、それは、装置がリスニング・アプリケーションにおける各ソース (源) の間の区別を行うことを可能にする。

キー\_\_エントリー I V (Key\_\_Entry IV) の使用法は 2 つある。第 1 の使用法は、タイト・ロック (パーシスタンス = 1) のときに設定される。ロック装置は、装置のプライベート領域に対するアクセスを許可するデータ・タイプ変数キー\_\_エントリー (Key\_\_Entry) 値を与える。この値は、説明の便宜上、“ロック\_\_キー (lock\_\_key)” として定義される。

10

第 2 の使用法は、リスニング・アプリケーション (パーシスタンス = 2) におけるものである。リスニング機能は、キー\_\_エントリー I V (Key\_\_Entry IV) を追加のディスクリミネータとして使用する。このキー\_\_エントリー (Key\_\_Entry) 機能についてはこのドキュメントの後の方で説明する。

第 1 のレベルのプライバシーに加えて、製造業者は、キー\_\_エントリー I V (Key\_\_Entry IV) を、認証 (authentication) を必要とする保護変数となるように選択することができる。これによって、特別なアクセスおよび認証の発生 (発行) が分離される。キー\_\_エントリー I V (Key\_\_Entry IV) は、装置がアンロック状態にあるときまたはパーシスタンスが“ルース”に設定されていないときには、不活性状態にある。デフォルト・キー\_\_エントリー値は“ナル (0)”である。装置がアンロックされるかまたはパーシスタンス・レベルが変更されたときには、キー\_\_エントリー I V (Key\_\_Entry IV) はデフォルト値に反転する。その結果、ロック装置だけがキー\_\_エントリー I V (Key\_\_Entry IV) 値を供給するかまたは読取ることができるようになる。それによって、1 つの装置だけがプライベートなデータまたは機能にアクセスすることが可能になる。

20

アプリケーションに応じて、“ロック\_\_キー (lock\_\_key)” は、ROM 内に位置する固定されたキー、または“トラステッド・キー (trusted key)” のような書込み可能なキーであり得る。通常、唯一の有効キー\_\_エントリー (Key\_\_Entry) 値が存在するが、複数の有効キー\_\_エントリー I V (Key\_\_Entry IV) の値も存在し得る。製造業者はロック\_\_キー I V (lock\_\_Entry IV) を書込み可能であるように選択することができる。保護された装置は、キー\_\_エントリー I V (Key\_\_Entry IV) を無効エントリー (例えば論理長さを 0) または偽値に設定することによって、ロックの解除 (disarm) によって、またはロック・パーシスタンス < 1 の変更によって、再びプライベートになる。

30

## c) ロッキング\_\_アドレス I V (Locking\_\_Address IV)

ロッキング\_\_アドレス I V (Locking\_\_Address IV) は、データ・タイプ変数であり、その装置をロックしたネットワーク・アドレスである。C E B u s アプリケーションにおいて、M A C アドレスはネットワーク・アドレスとして使用される。

ロッキング装置は、ネットワーク・アドレスを適正なロッキング\_\_アドレス I V (Locking\_\_Address IV) に書込んだときに、装置とのロック状態を開始する。いったんロックが設定された後は、ロッキング\_\_アドレス I V (Locking\_\_Address IV) が無効アドレス (例えば、論理長さが 0) に設定されることによってロックが解除される。ロッキングを支援しない装置に対してロック要求すると、結果としてリターン・エラー - 無効 I V (Error-Invalid IV) が生じる。

40

## d) アンロック\_\_コンディション I V (Unlock\_\_Condition IV)

第 3 の変数はアンロック\_\_コンディション I V (Unlock\_\_Condition IV) であり、このアンロック\_\_コンディション I V によって、ユーザはロックを自動的に解除する条件を指定することが可能になる。アンロック条件は、装置において見つけられた I V ' s を用いることができる (これはイベント・マネージャ・オブジェクト (Event Manager Object) と同様である。)

アンロッキング\_\_コンディション I V (Unlocking\_\_Condition IV) は、装置に対するア

50

ンロック条件を定義するブール式（論理式）を含んでいる。そのブール式は、C A L 文法（グラマ）で定義されたC A L 表現である。このプログラムは、例えばタイマ時間の経過（終了）またはスレッシュホールドを横切る（cross）ような或る条件が装置内で発生するのを待つ。その装置は、指定されたアンロック条件が生じたときにロックを解除（解決）する。アンロック条件はロックされた装置に存在することに留意されたい。パーシスタンス・レベルはルース（疎）またはタイト（密）の何れであってもよい。

条件はセットアレイ（setArray）方法を用いてアンロック\_\_コンディション（Unlock\_\_Condition）にロードされる。この場合、ロードされるデータは、実際のC A L 条件プログラムである。このプログラムはアンロック作用を生じさせる条件を監視する（watch）条件を含んでいる。

10

まさにイベント・マネージャ・オブジェクトの場合のように、アンロッキング\_\_コンディション I V（Unlocking\_\_Condition IV）が正確に動作するためのC A L 文法には、2つの僅かな相違がある。第1の相違は、監視（ウォッチ）すべきインスタンス変数を識別することである。単純な I V 識別子（identifier）ではなくて、I V のコンテキストおよびオブジェクト位置が与えられなければならない。例えば、（カレント\_\_バリュース > マックス\_\_バリュース（current\_\_value > max\_\_value、現在値 > 最大値）をテストする代わりに、アンロッキング\_\_コンディション（Unlocking\_\_Condition、アンロッキング条件）が（オーディオ・コンテキスト、ボリューム・コントロール、カレント\_\_バリュース > オーディオ・コンテキスト、ボリューム・コントロール、マックス\_\_バリュース（Audio Context, Volume Control, current\_\_value > Audio Context, Volume Control, max\_\_value）をテストする。全てのバイトを要求することによって、ステートメント（文）は正確に構文解析（パース：parse）することができる。

20

アンロック\_\_コンディション（Unlocked\_\_Condition）に含まれているブール式（ブール表現）は、最大で2つの関係性を評価することを可能にする。例えば、クロック・オブジェクト（Clock Object）は、時間が12に等しく月が6に等しいときにアンロックすることができる。ブール式における第1の I V はレポートされた値である。標準の関係（リレーショナル）オペレータ（演算子）と共に、追加のオペレータ、D E L T A オペレータが定義される。アンロック\_\_コンディション I V（Unlock\_\_condition IV）が（セットアレイ（setArray）方法を用いて）書き込まれたときには、オブジェクトは I V 中に置かれた条件の有効性をチェックしなければならないことに留意されたい。その条件が無効である場合には、エラーバッド（不良）アーギュメント・タイプ（Error-Bad Argument Type）（12）が返送（リターン）され、アンロック\_\_コンディション（Unlock\_\_Condition）は発生しない。

30

#### ロッキング実装構成

セットアレイ（setArray）方法を使用して、ロッキング\_\_アドレス I V（Locking\_\_Address I V）に有効宛先アドレスを設定することにより装置がロックされる。C E B u s アプリケーションの場合において、ネットワーク・アドレスは、M S B、L S Bの順に現れるユニット・アドレス（Unit address）およびハウス・コード（House Code）で構成される。ユニット・コードUnit Code 0034、ハウス・コードHouse Code 001Aを用いたロッキング・メッセージの例では、次のようなメッセージが得られる。

40

```
<context> <object> 46 471 Ff5 Ff5 Ff4 34 Ff6 00 34 00 1A
(<context> <object> setArray 'GA' <DEL> <DEL> <DATA> 34 <ESCAPE> 0034 001A)
〔訳、DEL：削除、ESCAPE：エスケープ〕
```

オブジェクト・ロッキング\_\_アドレス I V（Locking\_\_Address IV）を無効アドレス（例えば、論理長さを0）に設定することによって、それがアンロックされる。

```
(<context> <object> setArray 'GA' <DEL> <DATA> 30<ESCAPE>)
```

エラー - 無効 I V は、この機能が支援（支持）されないときに返送（リターン）される。

#### 1. オブジェクト・ロッキング実装構成（Object Locking Implementation）

オブジェクト・レベルでのロッキングは、個々のオブジェクトにロッキング機構を設定することによって実現される。

50

## 2. コンテキスト・ロッキング実装構成 (Context Locking Implementation)

オブジェクト・レベル・ロッキングはコンテキストに拡張することができる。コンテキスト・レベルでのロッキングは、コンテキスト・コントロール・オブジェクトにロッキング機構を設定することによって実現される。

ロッキング\_\_アドレス I V (Locking\_\_Address IV) に有効値を設定することによってコンテキストがロックされる。コンテキストにおける例示 (instantiate、インスタンス化) されたオブジェクトは、同じロッキング\_\_アドレス (Locking\_\_Address) を継承する。(これは、パーシスタンス<>2のときだけ適用される。) コンテキスト内のオブジェクトが既にロックされている場合には、オブジェクト・ロッキング\_\_アドレス I V (Locking\_\_Address IV) 値がコンテキスト・ロッキング\_\_アドレス (Locking\_\_Address) 値と同じでない場合には、コンテキストはロックできない。

10

ロッキング装置のネットワーク・アドレスを、コンテキスト・コントロール・オブジェクト (Context Control Object) 中に置かれたコンテキスト\_\_ロッキング\_\_アドレス I V (Context\_\_Locking\_\_Address IV) に書込むことによって、コンテキスト・ロックが開始される。

<context\_\_ID> 01 01 46 47 F5 F5 F4 34 F6 <Locking\_\_Address>

コンテキスト\_\_パーシスタンス (Context\_\_Persistence)、コンテキスト\_\_キー\_\_エン트리 (Context\_\_Key\_\_Entry)、コンテキスト\_\_ロッキング\_\_アドレス (Context\_\_Locking\_\_Address)、およびコンテキスト\_\_アンロック\_\_コンディション I V s (Context\_\_Unlock\_\_Condition IVs) は、その振る舞い (behavior) を、オブジェクト I V s パーシスタンス (Object IVs Persistence)、キー\_\_エン트리 (Key\_\_Entry)、ロッキング・アドレス (Locking\_\_Address)、およびアンロック\_\_コンディション (Unlock\_\_Condition) からそれぞれ継承する。

20

## 3. デバイス・ロッキング実装構成

装置レベルにおけるロッキングは、ノード・コントロール・オブジェクト (Node Control Object) に、装置ロッキング機構を設定することによって実現される。このノード・コントロール・オブジェクトは、自己の個別のオブジェクト・ロッキング機構を持つことができる。装置レベル・ロッキング変数は次の通りである。

	ノード・コントロール・オブジェクト		(01)ノード・コントロール	
	装置ロックを可能にするノード・コントロール・オブジェクト における変数			
I V	R／W	タイプ	名称	コンテキスト機能
o (6Fh)	R	d	object_list (オブジェクト・リスト)	コンテキストにおいて 使用されるオブジ ェクトのリスト
z (6a)	R／W	N	Device_Persistence (デバイス・パーシスタンス)	パーシスタンス (Persistence)
v (76)	R／W	D	Device_Key_Entry (デバイス・キー・エントリー)	キー・エントリー (Key_Entry)
T* (54)	R／W	D	Device_Locking _Address (デバイス・ロッキング・アドレス)	ロッキング・アドレ (Locking_Address)
V (56)	R／W	D	Device_Unlock _Condition (デバイス・アンロック・コンディション)	アンロック・コンデ (Unlock _Condition)

デバイス\_\_パーシスタンス (Device\_Persistence)、デバイス\_\_キー\_\_エントリー (Device\_Key\_Entry)、デバイス\_\_ロッキング\_\_アドレス (Device\_Locking\_Address)、およびデバイス\_\_アンロック\_\_コンディション I V s (Device\_Unlock\_\_Condition I V s) は、その振る舞いを、オブジェクト I V s パーシスタンス (Object\_IVs Persistence)、キー\_\_エントリー (Key\_Entry)、ロッキング\_\_アドレス (Locking\_Address)、およびアンロック\_\_コンディション (Unlock\_\_Condition) からそれぞれ承継する。装置ロックは、ロッキング装置のネットワーク・アドレスをデバイス\_\_ロッキング\_\_アドレス I V (Device\_Locking\_Address IV) に書込むことによって開始される。

00 01 46 54 F5 F5 F4 34 F6 <Locking\_Address>

10

20

30

40

50



## ロッキング・シナリオ (Locking Scenarios)

デモンストレーションのために、ロッキング装置は、別の装置をロックすることを希望する装置であるものとする。このロックされる装置はロックしようとする装置である。ロッキング要求は、ロックされるべき装置またはロックを保持するロック・マネージャの何れかに対して行うことができる。通常の条件下において、ロッキング要求は、新しいアドレスをロッキング\_\_アドレス I V (Locking\_\_Address IV) に書き込む試みによって行われる。その要求は直接的または条件付きであってもよい。

### a) 直接ロッキング要求

ユニット・コード 0 0 3 4 (Unit Code 0034)、ハウス・コード 0 0 1 A (House Code 001A) を用いたロッキング・メッセージの一例の結果として、次のメッセージが得られる

```
<context> <object> 46 4741 Ff5 Ff5 Ff4 34 Ff6 00 34 00 1A
```

```
( <context> <object> setArray ' G ' <DEL> <DEL> <DATA> ' 4 ' <ESCAPE> 0034 001A )
```

装置は、それがロック可能かどうかを決定しなければならない。装置、オブジェクト、コンテキストがロックされていない場合には、完了トークン F E が返送 (リターン) される。ロッキング装置は、ロッキング・ポインタをロック・マネージャ・オブジェクト中に置く。

### b) 条件付きオブジェクト・ロッキング要求

ユニット・コード 0 0 3 4 (Unit Code 0034)、ハウス・コード 0 0 1 A (House Code 001A) を用いたロッキング・メッセージの一例の結果、次のメッセージが得られる。

```
<context> <object> 56 47 F7 <context> <object> 46 471 F5f5 Ff5 Ff4 34 Ff6 00 34 00 1A EB 44 47 F8
```

```
<context> <object> I ( IF " G " <BEGIN> <context> <object> setArray ' G ' <DEL> <DEL> <DATA> ' 4 ' <ESCAPE> 0034 001A <ELSE> setArray " G " <END> )
```

〔訳、IF ~ : ~ の場合は、BEGIN : 開始する、ELSE : その他の場合は〕

アレイ " G " が使用中でない場合には、装置はロックされる。それ以外の場合には、装置がロックされることはない。この方法は資源ロッキングと共に用いられる。装置が使用中の場合には、要求する装置は、同報 (broadcast) メッセージを送信することによって装置のロックを断念 (放棄) するようロッキング装置に要求 " 呼びかけ (hail) " することができる。

### c) ロック・マクロ U 2 (LOCK Macro U2) の使用

ロック・マクロ U 2 (LOCK Macro U2) を用いてコンテキストとオブジェクトをロックする。マクロ・メッセージはマクロ I D を含んでおり、そのマクロ I D の後にアーギュメント・リストが続く。2つの必須のアーギュメント、(1) ロッキング\_\_アドレス (Locking\_\_Address)、および (2) パーシスタンス (persistence) が存在する。アーギュメント・リストは、それが満たされて (詰め込まれて) いない場合には、ナル (0、空) と見なされる。ロックが設定された後、キー\_\_エン트리 (Key\_\_Entry) およびアンロック\_\_コンディション I V s (Unlock\_\_Condition IVs) が更新される。

```
<context__ID> <object__ID> " U2 " <data token> <size of data> <escape token> <locking__address> F5 <persistence> F5 <Key__Entry>
```

例えば、コンテキスト 1 1 をロックする C A L マクロ (CAL Macro) メッセージについて :

```
" 11 011 55U 322 F4 34 F6 0F 32 23 34 F5 01 "
```

アーギュメント 0Ff32 2334 はロッキング・アドレスであり、パーシスタンスは 1 に設定される。この時点で、ロッキング装置は、エントリー\_\_キー I V (Entry\_\_Key IV) データ および アンロック\_\_コンディション I V (Unlock\_\_Condition IV) データを送信することができる。

ロッキング\_\_アドレス (Locking\_\_Address) が与えられないときには、それをマクロ・メッセージのソース・アドレス (出所アドレス) であると仮定する。

```
" 11 01 55 32 F5 F5 01 "
```

## ロッキング解除要求

### a) ロッキング・マスターに対するロッキング解除要求

(1) ディスインヘリット (DISINHERIT、承継排除) (オプション 1) の使用

オブジェクト、コンテキストまたは装置のレベルの何れかにおいてオブジェクトがロックされている場合に、そのオブジェクトは、妨げとなるロック装置に対して、そのロックを放棄するように要求する。次の通り、メッセージがロッキング・マネージャ・オブジェクト (Locking Manager Object) に送信される。

<Universal Context> <Locking Manager Object>・disinherit

<lock\_\_pointer (6C)> F4 [number of bytes] F6 <device\_\_address>

<context\_\_id> <object>

10

[訳、disinherit: 承継排除する、number of bytes: バイト数]

disinherit the lock with <device> <context> <object>

[訳、disinherit the lock with ~: ~とのロック承継排除する]

コンテキスト\_\_ID (context\_\_ID) は、それが A 0 でない限り、コンテキスト\_\_ナンバー (context\_\_number) を含んでいる。それが A 0 の場合には、コンテキスト・ナンバーはオプションとして (任意選択的に) 含めることができる。

(2) アンロック・マクロ U 4 1 (UNLOCK Macro U41) (オプション 2 1) の使用  
アンロック・マクロ U 1 4 (UNLOCK Macro U14) はイベント・マネージャ・オブジェクト (EVENT Manager Object) に送信される。そのマクロは、所望のロッキング・ポインタをマッチ (match: 整合、一致) させて、それを削除するように要求する。マッチが存在する場合には、ロッキング・マネージャは、要求を許可するかまたは要求を拒否する。ロッキング装置を見つけることができない場合には、ロックされた装置はロックを解除する。詳細についてはセクション 2. 1. 2 を参照されたい。

20

### リスニング実装構成

リスニング機能はレポート方法のコンパニオン (対の一方、関係関係にあるもの) である。リスニングは、他の装置またはネットワーク資源の状態に関係のある (関心を持つ) オブジェクトにおいて実行される。ネットワーク資源の例は時間である。

リスニング機能によって、装置は状態情報を含んだ各同報メッセージ間の区別ができるようになる。リスニング機能は、ロッキング機能の特別なアプリケーションである。リスニング・モードにおいて、オブジェクトは、コンテキストまたは装置のロッキング状態を承継しない。(これは、装置またはコンテキストがロックされたときに、リスニング機能が正しく実行されるのに必要である。)

30

リスニングにおいて、装置は自己を別の装置に結合 (バインド) する。これは、装置が自己のロッキング\_\_アドレス I V (locking\_\_Address IV) を設定することができることを意味する。ロッキング\_\_アドレス I V (locking\_\_Address IV) は、読出しのみ (リードオンリー) 形または読出し / 書込み (リード / ライト) 形の何れかである。書込みオプション (任意選択) は、実際には、ルース・ロッキングのタイプである。パーシスタンス I V は " 2 " に設定される。キー\_\_エン트리 I V (Key\_\_Entry IV) は、特別なアクセス・ディスクリミネータとして作用する。次のテーブルはリスニング I V リストを含んでいる。

40

オブジェクト・リスニング I V リスト				
I V	R/W	タイプ	名称	コンテキスト機能
G* (47)	R/W**	D	Locking__Address (ロッキング・アドレス)	オブジェクト・ リスニング・ アドレス
B* (42)	R	N	Persistence (パーステナス)	2 = リスニング
E (45)	R/W	D	Key__Entry (キー・エントリ)	特別なアクセスを 可能にする

10

20

ロッキング\_\_アドレス (Locking\_\_Address) が読み出しのみ (リードオンリ) 変数であるときには、装置は自己のロッキング・アドレスを設定する。ロッキング\_\_アドレス (Locking\_\_Address) が外部から設定されるときには、ロッキング装置は、そのロッキング・マネージャ (Locking\_\_Manager) にロック\_\_ポインタ (lock\_\_pointer) を置かない。

リスニングには3つの相異なる実装構成が存在する。第1の方法では、ソース・ネットワーク・アドレスを用いて、リスニング・オブジェクトに対するアクセスをゲート処理する。そのオブジェクトは、ロッキング\_\_アドレス I V (Locking\_\_Address I V) に対応するソース装置アドレスに有利な形で区別を行う (を優先させる)。この場合、メッセージは適切なインスタント・バリアブル (Instant Variable、瞬間変数) に書込まれる。

30

第2と第3のリスニング実装構成では、キー\_\_エントリ装置が使用される。この実装構成では、マクロ・メッセージが必要である。マクロ・メッセージ、L 1 を使用して、キー\_\_エントリ (Key\_\_Entry) 値およびデータが同時に与えられる。リスニング装置は、マクロが正しいソース・アドレスからのものであること、および与えられたキー\_\_エントリ (Key\_\_Entry) 値が所望の値とマッチすることを検証 (ベリファイ) する。

#### 1. 複数ソース・リスニング実装構成

リスニング・オブジェクトが、区別 (差別) することなく、全ての同報ソース (源) に対してリスニング (聴取) することを希望することもある。そのような1つの状況は、火災を検出していることを火災センサが同報するときに発生する。火災センサは、その情報をネットワーク上で同報する。リスニング装置は、火災を報告する可能性のある全ての資源をリスニングすることを希望する。

40

一般的に、複数ソース・リスニング・アプリケーションは、装置が任意の装置 (センサ) の状態を受信できるようにすることによって実行される。これは、ロッキング\_\_アドレス I V (Locking\_\_Address I V) に同報アドレスを設定することによって規定される。C E B u s アプリケーションにおいては、リスニング・ディスクリミネータが、特別ハウス・コード (House Code) 内の全てのユニット\_\_アドレス (Unit\_\_Address) またはネットワーク上の全てのアドレスの何れかに設定される。メッセージに含まれているアラーム・オーサ・ソース (alarm author source、警報発生源) またはゾーン・アドレス (zone addr

50

ess) は、メッセージ中に含まれており、そのメッセージに含まれているソース\_\_アドレス (source\_\_address) フィールドまたは状態ベクトルから抽出することができる。

2. キー\_\_エンタリー (Key\_\_Entry) およびロッキング\_\_アドレス (Locking\_\_Address) を介したリスニング

このようにして、ロック\_\_キー (lock\_\_key) 値は、装置が個々のコンテキストおよびオブジェクトの間の区別を行うことを可能にするセレクト値 (選択値) となる。例えば、タイム・コンテキストにおいて、タイム・タイプ識別子 (time type identifier、時間タイプ識別子) をマクロ・メッセージにおいて与えることができる。この値は所望の時間タイプについてチェックされる。装置がレポーティング・コンテキストの複数のインスタンス (例示) を有する場合には、装置はキー\_\_エンタリー (Key\_\_Entry) フィールド中にコン

10

<context\_\_ID> <object\_\_ID> 53 31 F4 <size of data> <escape token> 5

<Locking\_\_Address> F5 <data token> <size of data> <escape token>

<Key\_\_Entry> F5 <Argument Listmessage>

リスニングにおいて、装置は自己を別の装置に結合する。リスニングの場合、装置は自己のロッキング\_\_アドレス I V (Locking\_\_Address IV) を設定することができる。ロッキング\_\_アドレス I V (Locking\_\_Address IV) は読出しのみ (リードオンリ) 形または書込み / 読出し形の何れかである。

3. キー\_\_エンタリー (Key\_\_Entry) を介したリスニング

20

また、リスニングはロッキング\_\_アドレス (Locking\_\_Address) を “00 00 00 00” に設定することによって実行される。この場合は、ディスクリミネータだけがキー\_\_エンタリー (Key\_\_Entry) 値である。これによって、オブジェクトはアクセス・キーに基づいて区別を行うことが可能となる。これは次のように行われる。

<context\_\_ID> <object\_\_ID> 53 31 F5 00 00 00 00 F5 <data token> <size of data> <escape token> <Key\_\_Entry> F5 <Argument\_\_List>

ロッキング\_\_アドレス (Locking\_\_Address) が読出しのみ (リードオンリ) 形であるときには、装置は排他的に自己のロッキング・アドレスを設定する。ロッキング\_\_アドレス I V (Locking\_\_Address IV) が R / W のときには、ロッキングはルース (疎) である。ロッキング装置はロッキング\_\_ポインタ (locking\_\_pointer) をロッキング・マネージャ (Locking Manager) 中に設定しない。

30

<context\_\_ID> <object\_\_ID> L1 <data token> <size of data> <escape token> <Locking\_\_Address> F5 F5 <message>

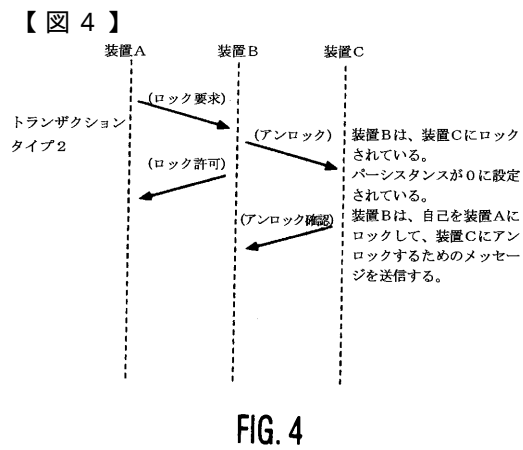
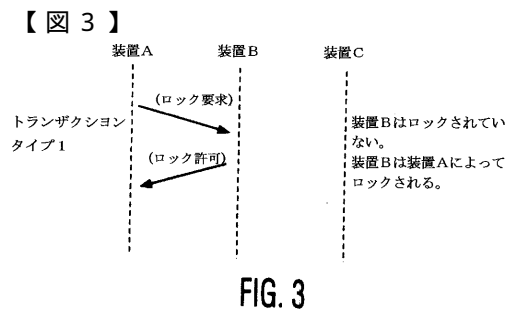
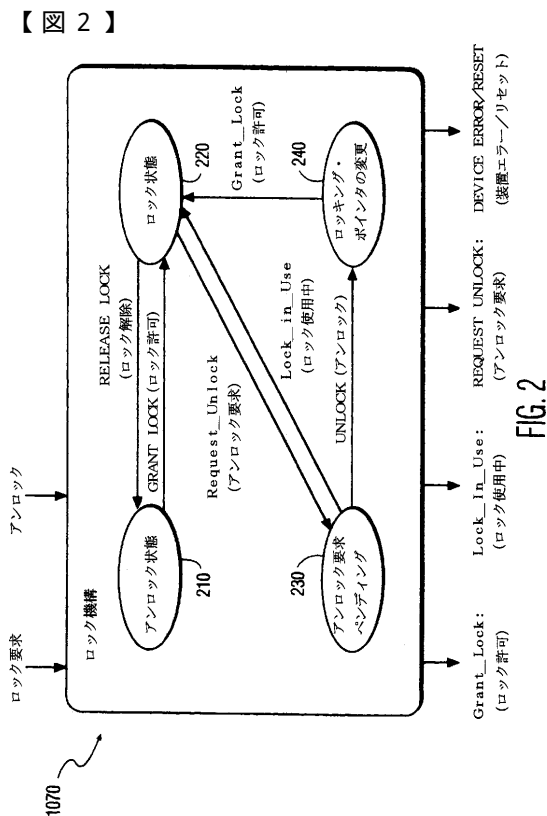
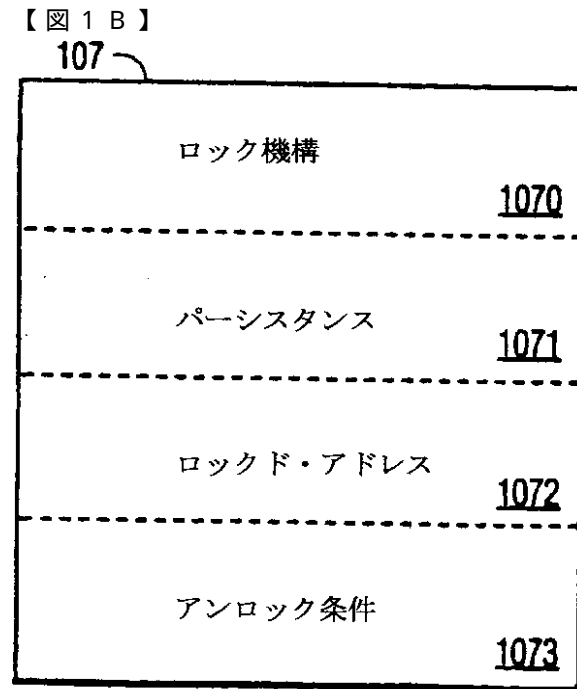
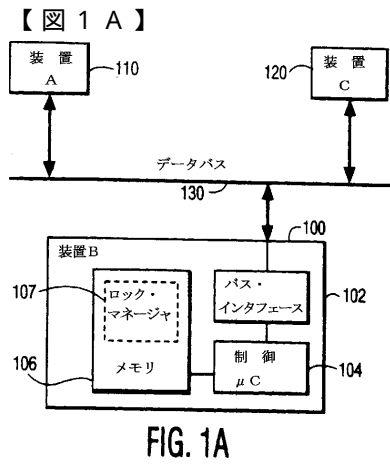
固有の識別子が、1つの装置において、複数のコンテキストまたはオブジェクトの間の区別を可能にすることが望ましいような幾つかのインスタンス (事例) がある。その1つの例は、複数のセンサを含んだ温度センサ装置である。各センサを対応するリスニング・オブジェクトに結合することが望ましいことがある。これは、キー\_\_エンタリー = ナル (0) (key\_\_entry = null) である場合には区別が存在しなくなるようなキー\_\_エンタリー (key\_\_entry) 法を用いて実行される。キー\_\_エンタリー (key\_\_entry) が或る値に設定されるときには、リスニング装置が、オブジェクトを更新する前に、適正なキー\_\_エンタリー (key\_\_entry) 値を要求する。

40

例: リスニング変数は、ユーザがディスクリミネータ (discriminator) を設定するのを可能とするキー\_\_エンタリー\_\_ナンバー (key\_\_entry\_\_number) のようなディスクリミネータを有する。データおよびディスクリミネータはマクロ\_\_ID (Macro\_\_ID) キー\_\_エンタリー F5 リスニング・データ (key\_\_entry F5 Listening DATA) で送信される。キー\_\_エンタリー (key\_\_entry) が与えられない場合には、ディスクリミネータはネットワーク・アドレスである。

本発明を多数の実施形態に基づいて詳細に説明したが、以上の記載を読んで理解すれば、この分野の専門家が上述の実施形態に対する多数の代替構成例を思い付き、そのような代替構成例が本発明の請求の範囲内に含まれることは明らかである。

50



【図 5】

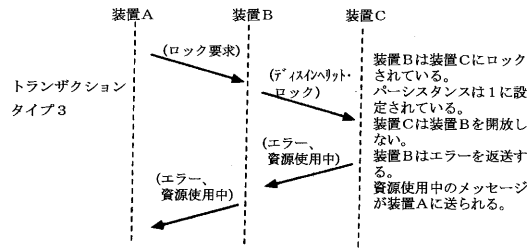


FIG. 5

【図 6】

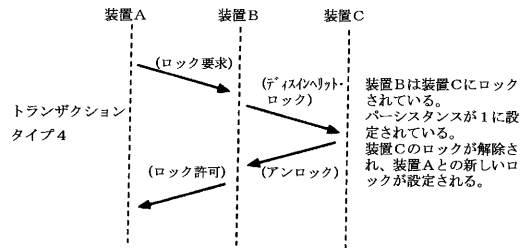


FIG. 6

【図 7】

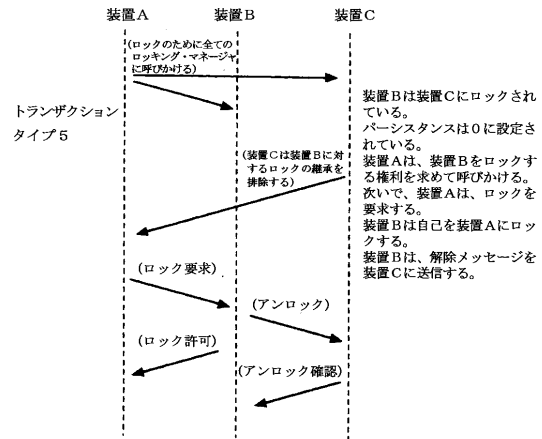


FIG. 7

---

フロントページの続き

審査官 矢頭 尚之

(56)参考文献 特開平 0 7 - 2 5 0 0 7 2 ( J P , A )  
特開平 0 8 - 0 0 8 9 1 3 ( J P , A )  
特開平 2 - 2 1 7 5 4 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)  
H04L 12/28  
H04L 29/02