

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3963385号

(P3963385)

(45) 発行日 平成19年8月22日(2007.8.22)

(24) 登録日 平成19年6月1日(2007.6.1)

(51) Int. Cl.		F I			
<b>G06F 21/22</b>	<b>(2006.01)</b>	G06F	9/06	660C	
<b>G06F 1/00</b>	<b>(2006.01)</b>	G06F	1/00	370E	

請求項の数 11 (全 16 頁)

(21) 出願番号	特願2003-540769 (P2003-540769)	(73) 特許権者	390009531
(86) (22) 出願日	平成14年10月10日(2002.10.10)		インターナショナル・ビジネス・マシー ズ・コーポレーション
(65) 公表番号	特表2005-507519 (P2005-507519A)		INTERNATIONAL BUSIN ESS MASCHINES CORPO RATION
(43) 公表日	平成17年3月17日(2005.3.17)		アメリカ合衆国10504 ニューヨーク 州 アーモンク ニュー オーチャード ロード
(86) 国際出願番号	PCT/EP2002/012190	(74) 代理人	100086243
(87) 国際公開番号	W02003/038570		弁理士 坂口 博
(87) 国際公開日	平成15年5月8日(2003.5.8)	(74) 代理人	100091568
審査請求日	平成16年6月23日(2004.6.23)		弁理士 市位 嘉宏
(31) 優先権主張番号	01480108.8	(74) 代理人	100108501
(32) 優先日	平成13年10月30日(2001.10.30)		弁理士 上野 剛史
(33) 優先権主張国	欧州特許庁 (EP)		

最終頁に続く

(54) 【発明の名称】 デジタル・エンコードされた製品の使用を制御する方法およびシステム

(57) 【特許請求の範囲】

【請求項1】

コンピュータと該コンピュータに接続されたメータリング・サーバとを備えるシステムで用いられ、前記コンピュータ上で実行されるデジタル・エンコードされた製品の使用を制御するための方法(300)であって、

前記コンピュータには、前記製品のリストを保管する制御カタログが備えられ、

前記コンピュータ上で動作する実行エージェントが、

前記コンピュータ上で前記製品の実行要求があったとき、前記製品のリストが前記制御カタログに含まれない場合に、当該製品の開始をイネーブルするステップ(328, 330)と、

前記製品のリストが前記制御カタログに含まれる場合に、前記メータリング・サーバに対して実行要求を送るステップ(328, 334)と、

前記メータリング・サーバからの応答に応じて前記実行要求を許可するか否かを決定するステップ(310~315)と、

前記制御カタログに含まれない製品の表示を前記コンピュータで保管されるログに追加するステップ(332)と、

前記ログを前記メータリング・サーバに定期的を送るステップ(366)とを実行し、

前記メータリング・サーバ上で動作するライセンスング・エージェントが、

前記メータリング・サーバに備えられた許可カタログに保管された使用条件に前記実行要求に係る前記製品の実行が従うか否かを判定するステップ(380, 384~385)

10

20

と、

前記ステップ(380, 384~385)における判定に応じて前記応答を前記コンピュータに返送するステップ(382)と、

前記ログにリストされた前記製品が少なくとも前記メータリング・サーバに備えられたグローバル・カタログに含まれるか否かを判定して、含まれる場合に前記制御カタログ及び前記許可カタログを更新するステップ(389, 390)とを実行する方法。

【請求項2】

前記コンピュータ上で動作するオペレーティング・システムのカーネル拡張モジュールが、

前記実行要求に関連するプロセスの開始を検出して前記実行エージェントに前記実行要求を報告するステップ(306)と、 10

前記プロセスの開始が検出されると前記プロセスを中断するステップ(308)とを実行し、

前記ステップ(310~315)では、前記実行エージェントが前記メータリング・サーバからの応答に応じて前記実行要求を許可した場合に前記カーネル拡張モジュールに前記プロセスを再開させ、前記メータリング・サーバからの応答に応じて前記実行要求を不許可とした場合に前記プロセスを打ち切らせる請求項1に記載の方法(300)。

【請求項3】

前記メータリング・サーバには、予め定められた複数のコンピュータが関連しており、前記ステップ(328, 334)では、前記実行要求を示す要求メッセージを、前記コンピュータから複数のコンピュータに関連するメータリング・サーバに送っており、 20

前記ステップ(380, 384~385)は、前記複数のコンピュータのすべてについて前記使用条件を表すライセンス情報に前記実行要求に係る前記製品の実行が従うか否かを判定するステップ(380)と、前記実行要求に対して許可を与えた場合に前記メータリング・サーバによって現在許可されている実行許可を表すメータリング・テーブルに当該許可に係る製品をリストとして追加するステップ(384~385)とを有する請求項1又は2に記載の方法(300)。

【請求項4】

前記実行エージェントでは、前記応答が実行許可の拒否を示しているとき、少なくとも1つの代替メータリング・サーバに前記実行要求を送るステップ(336, 340, 342)をさらに実行する請求項3に記載の方法(300)。 30

【請求項5】

前記製品の実行が終了すると、前記実行エージェントでは実行の終了を示す終了メッセージを、前記許可を与えたメータリング・サーバに送るステップ(398, 318, 349)を実行し、

前記ライセンス・エージェントでは、前記終了メッセージに回答して、前記メータリング・テーブルから前記製品のリストを除去するステップ(387)を実行する請求項3又は4に記載の方法(300)。

【請求項6】

前記実行エージェントでは、前記メータリング・サーバに、実行許可中の製品に係る検査を行うための検査メッセージを定期的を送るステップ(364)を実行し、 40

前記ライセンス・エージェントでは、前記検査メッセージを事前に設定された期間中に受け取らないとき、前記メータリング・テーブルから前記コンピュータで動作する製品に係るリストを除去するステップ(392)を実行する請求項5に記載の方法(300)。

【請求項7】

前記実行エージェントでは、当該コンピュータにインストールされた製品を示すインストール情報を定期的に収集するステップ(368, 370)と、

前記メータリング・サーバに前記インストール情報を送るステップ(372)とを実行 50

する請求項3ないし6のいずれかに記載の方法(300)。

【請求項8】

単一の中央管理サーバから前記メータリング・サーバの管理を行うための管理メッセージをメータリング・サーバに送るステップと、

前記管理メッセージを受けた際前記ライセンス・エージェントは前記中央管理サーバに保管されたマスタ・カタログから当該メータリング・サーバに係る許可カタログを抽出して展開するステップ(394)とを有する請求項3ないし7のいずれかに記載の方法(300)。

【請求項9】

コンピュータ・アプリケーション・プログラム(220, 225, 260)であって、  
データ処理システム(100)で実行する際、請求項1ないし8のいずれかに記載の方法を実行するために前記データ処理システムの作業メモリに直接にロード可能なコンピュータ・アプリケーション・プログラム。

10

【請求項10】

請求項9に記載のコンピュータ・アプリケーション・プログラムが記録されたコンピュータ読み取り可能な記録媒体(160w, 160s)。

【請求項11】

コンピュータ(105w)と該コンピュータに接続されたメータリング・サーバ(105s)とを備え、前記コンピュータ上で実行されるデジタル・エンコードされた製品の使用を制御するためのデータ処理システム(100)であって、

20

前記コンピュータには、前記製品のリストを保管する制御カタログ(240)と、  
前記コンピュータ上で前記製品の実行要求があったとき、前記製品のリストが前記制御カタログに含まれない場合に、当該製品の開始をイネーブルする手段(220, 225)と、

前記製品のリストが前記制御カタログに含まれる場合に、前記メータリング・サーバに対して実行要求を送る手段(210w, 225)と、

前記メータリング・サーバからの応答に応じて前記実行要求を許可するか否かを決定する手段(225)と、

前記制御カタログに含まれない製品の表示を前記コンピュータで保管されるログに追加する手段(225)と、

30

前記ログを前記メータリング・サーバに定期的に送る手段(225, 210w)とが備えられ、

前記メータリング・サーバには、前記メータリング・サーバに備えられた許可カタログに保管された使用条件に前記実行要求に係る前記製品の実行が従うか否かを判定する手段(260, 265)と、

前記手段(260, 265)による判定に応じて前記応答を前記コンピュータに返送する手段(260, 210s)と、

前記ログにリストされた前記製品が少なくとも前記メータリング・サーバに備えられたグローバル・カタログに含まれるか否かを判定して、含まれる場合に前記制御カタログ及び前記許可カタログを更新する手段(260)とが備えられているデータ処理システム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、デジタル・エンコードされた(デジタル方式でエンコードされた)製品の使用を制御する方法およびシステムに関する。

【背景技術】

【0002】

ソフトウェア・プログラムなどのデジタル・エンコードされた製品は、回数制限なくコピーを完全に再生することができる。しかし、プログラムの無許可のコピーには、通常、その権利者の知的財産権の侵害が伴う。このため、使用されるプログラムが、対応する

50

許可された使用条件に従っていることを検証することが非常に重要である。この問題は、多数のコンピュータを有する大組織で特に深刻であり、近年、この種の製品の無制御の配布をさらに促進するインターネットの広範囲の拡散によって悪化してきた。

【 0 0 0 3 】

上述の検証は、通常は、すべてのコンピュータを手作業で監査することによって実行される。しかし、この手順は、時間がかかり（特に大規模な場所で）、エラーが発生しやすい。

【 0 0 0 4 】

許可された使用条件をプログラムが遵守しているかどうかを評価するいくつかの自動化ツールも、提案されている。通常、これらのツールでは、各コンピュータにインストールされるエージェントが活用される。エージェントによって、コンピュータで動作するプログラムが定期的に検査される。プログラム使用状況レポートを生成するために、検証の結果がログに記録される。さらに、その許可された使用条件に違反するプログラムが動作中である場合に、このツールによって、めいめいのプロセスを強制終了することができ（プログラムの実行を停止するために）、そのプログラムを二度と開始できなくすることができる。

【 0 0 0 5 】

しかし、上で説明した解決策では、プログラムの使用をリアルタイムでは監視せず、その結果、実際のライセンス検証を実行することは不可能である。さらに、許可された使用条件に強制的に従わせるためにプログラムの実行を強制終了する時に、対応するデータに損傷が発生する場合があります。これによって、エンドユーザが、次の（許可された）使用のためにコンピュータにそのプログラムを再インストールすることも強制される。

【 0 0 0 6 】

当技術分野で提案された、テクニカル・ライセンス（technical licensing）と称する異なる解決策には、ライセンス管理システムへの呼び出しを含むように各プログラムを修正することが含まれる。エンドユーザが、プログラムを開始するたびに、対応する要求が、ライセンス管理システムに転送される。ライセンス管理システムによって、そのプログラムの実行が、めいめいのデジタル証明書に組み込まれている許可された使用条件によって示される限度内であるかどうかを検証され、対応する戻りコードがプログラムに供給される。検証の結果が肯定である場合には、プログラムがその実行を継続することができ、そうでない場合には、プログラムは停止を強制される。

【 0 0 0 7 】

しかし、上で説明した解決策は、各プログラムが権利者によって修正されることを必要とする。さらに、この解決策は、非常に厳格であり、異なる要件に簡単には適合できない。テクニカル・ライセンス技術は、ライセンス管理システムをサポートするプログラムだけに適用され、異なる製品の使用を制御するには完全に非効率的である。

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 8 】

本発明の目的は、上述の短所を克服することである。この目的を達成するために、請求項 1 に記載の方法が提案される。

【 課題を解決するための手段 】

【 0 0 0 9 】

短く言えば、本発明は、デジタル・エンコードされた製品の使用を制御するコンピュータによって実施される方法であって、コンピュータでのデジタル・エンコードされた製品の実行を要求するステップと、前記製品の開始の前に前記実行要求をインターセプトするステップと、前記製品の許可された使用条件を前記実行要求が遵守していることを検証するステップと、前記検証の結果に従って前記製品の開始をイネーブルするか妨げるステップとを含む方法を提供する。

10

20

30

40

50

## 【0010】

さらに、本発明は、その方法を実行するコンピュータ・プログラム・アプリケーション、このアプリケーションを保管されたプログラム製品、およびデジタル・エンコードされた製品の使用を制御するデータ処理システムも提供する。

## 【0011】

本発明による解決策のさらなる特徴および長所は、添付図面に関する、純粹に非制限的な指摘として与えられる、本発明の好ましい実施形態の以下の説明から明瞭になる。

## 【発明を実施するための最良の形態】

## 【0012】

特に図1に関して、本発明を使用することができるデータ処理システム100が示されている。システム100に、複数のワークステーション105wが含まれる。ワークステーション105wは、一連のサブセットにグループ化され、メータリング(metering)サーバ105sが、各サブセットのワークステーション105wに関連する。メータリング・サーバ105sおよび対応するワークステーション105wは、ネットワーク110(たとえば、イントラネット)を介して互いに接続される。異なるメータリング・サーバ105sは、異なるネットワーク120(たとえば、インターネット)を介して管理サーバ115(リモート位置にインストールされる)と通信する。さらに、管理サーバ115は、資産管理システム125に接続され(もう一つのネットワーク127を介して)、資産管理システム125には、ワークステーションおよびそのユーザに関する情報が保管される。

10

20

## 【0013】

たとえばパーソナル・コンピュータ(PC)からなる各ワークステーション105wは、通信バス130wに並列に接続される複数のユニットによって形成される。詳細に言うと、マイクロプロセッサ( $\mu$ P)135wによって、ワークステーション105wの動作が制御され、RAM140wは、マイクロプロセッサ135wによって作業メモリとして直接使用され、ROM145wに、ワークステーション105wのブートストラップ用の基本プログラムが保管される。さらに、複数の周辺ユニットが、(めいめいのインターフェースによって)バス130wに接続される。具体的には、大容量記憶装置は、磁気ハードディスク150wおよびCD-ROM160wを読み取るドライバ155wからなる。さらに、ワークステーション105wに、入力装置165w(たとえば、キーボードおよびマウスからなる)、および出力装置170w(たとえば、モニタおよびプリンタからなる)が含まれる。ネットワーク・インターフェース・カード(NIC)175wが、ワークステーション105wをネットワーク110に接続するのに使用される。

30

## 【0014】

メータリング・サーバ105s(たとえば、ミッドレンジ・コンピュータからなる)は、同様に、バス130s、複数のマイクロプロセッサ135s、RAM140s、およびROM145sによって形成され、メータリング・サーバ105sには、さらに、ハードディスク150s、CD-ROM160sのドライバ155s、入力装置165s、出力装置170s、およびNIC175s(ネットワーク110および120にアクセスするための)が含まれる。

40

## 【0015】

同様の考慮事項が、システムが異なるアーキテクチャを有する場合、ワークステーション、メータリング・サーバ、および管理サーバが異なるネットワークで互いに接続される場合、他のネットワークが使用される場合、各ワークステーションまたはメータリング・サーバが異なる構造を有するか異なる構成要素を含む場合、資産管理システムが設けられない場合などにあてはまる。

## 【0016】

図2を検討すると、ワークステーションおよびメータリング・サーバの作業メモリ140wおよび140sの部分的内容が示されている。情報(プログラムおよびデータ)は、通常は、めいめいのハードディスクに保管され、プログラムが動作中である時に(少なく

50

とも部分的に)作業メモリにロードされ、プログラムは、当初は、CD-ROMからハードディスクにインストールされる。

【0017】

オペレーティング・システム205wおよびオペレーティング・システム205sによって、それぞれワークステーションおよびメータリング・サーバのソフトウェア・プラットフォームが提供される。スタック210w(ワークステーション側)およびスタック210s(メータリング・サーバ側)によって、一緒にネットワーク通信を定義するように働くプロトコル層の組が処理される。インターネットを介する情報の効果的な伝送を保証するために、情報が、暗号化され(セキュリティを保証するため)、ファイヤウォールを通過できるプロトコル(HTTPなど)を使用して伝送される。

10

【0018】

特にワークステーションを検討すると、オペレーティング・システム205wに、主モジュール215(カーネルと称する)が含まれ、この主モジュール215によって、オペレーティング・システムの他の部分が必要とする必須サービスのすべてが提供される。カーネル拡張モジュール220によって、一般に使用可能な標準インターフェースを使用して、オペレーティング・システムに機能が追加される。

【0019】

実行時エージェント225が、バックグラウンドで動作して、ワークステーションでのアプリケーション・プログラム230の実行を制御する。実行時エージェント225は、メータリング・サーバと情報を交換するために、スタック210wとインターフェースする。実行時エージェントは、登録処理中にワークステーション上に静かにインストールされる。この登録処理では、ワークステーションのエンドユーザが、関連するメータリング・サーバにホストされるウェブ・ページに接続することによって、エンドユーザ自身を登録する。

20

【0020】

実行時エージェント225によって、複数のメモリ構造が制御される。具体的に言うと、サーバ・テーブル235に、そのワークステーションに関連する可能な代替メータリング・サーバのIPアドレスが含まれる(ワークステーションに直接に接続される主メータリング・サーバのほかに、実行時エージェントの構成テーブル内で識別される)。制御カタログ240に、ワークステーションでの実行を制御されなければならないプログラムがリストされ、その一方で、内部ログ245によって、制御カタログ240に含まれないが、ワークステーションで実行されたプログラムがリストされる。さらに、実行時エージェント225によって、実行中テーブル250への情報の入力が制御され、実行中テーブル250には、ワークステーションにおいて現在実行中の、対応する実行許可が与えられたプログラムが(許可を与えたメータリング・サーバの識別子と一緒に)リストされる。実行時エージェント225によって、プログラム・インベントリ255(メータリング・サーバへの送信用)も生成され、このプログラム・インベントリ255には、ワークステーションにインストールされたプログラムに関する情報が含まれる。

30

【0021】

メータリング・サーバに移ると、バックグラウンドで動作するライセンス・エージェント260によって、関連するワークステーションが制御される。ライセンス・エージェント260は、ワークステーションおよび管理サーバと情報を交換するために、スタック210sとインターフェースする。

40

【0022】

ライセンス・エージェント260によって、許可カタログ265が管理され、許可カタログ265には、各プログラムの許可された使用条件に関する情報が含まれ、たとえば、許可カタログ265によって、プログラムを実行できるワークステーションの最大処理能力または作業メモリの最大容量、プログラムのライセンス数(同時に実行を許可されるプログラムのインスタンスの最大個数を定義する)、その他の情報が指定される。さらに、ライセンス・エージェントによって、メータリング・テーブル270への情報の

50

入力が制御され、メータリング・テーブル 270 には、ワークステーションで現在実行中の、メータリング・サーバによって実行の許可を与えられたプログラムがリストされる。ライセンスエージェント 260 によって、関連ワークステーションでの実行を一般に許可することができる既知のプログラムを含むグローバル・カタログ 273 も管理される。

**【 0023 】**

さらに、ライセンスエージェント 260 によって、リポジトリ 275 が管理され、このリポジトリ 275 には、関連ワークステーションに配布される異なる情報（実行時エージェント・コードの最新版、または代替メータリング・サーバのリストおよび制御されるプログラムのリストなど）、またはワークステーションから収集される異なる情報（ソフトウェア使用のリアルタイム・レポートおよびワークステーションにインストールされたプログラムのインベントリ情報など）が含まれる。

10

**【 0024 】**

同じような考慮点が、プログラムおよびデータが異なる形で構成される場合、他のモジュールまたは機能が設けられる場合、データが同等のメモリ構造に保管される場合、異なる使用条件が想定される場合、情報が異なるプロトコルを使用して伝送される場合、ライセンスエージェントがウェブ・サーバに組み込まれる場合などにあてはまる。代替案では、実行時エージェントが、ワークステーションに保管されたログオン・スクリプトによってインストールされるか、ソフトウェア配布によって展開される。

**【 0025 】**

図 3 ~ 5 からわかるように、一緒に方法 300 を構成する一連のルーチンが、ワークステーションおよびメータリング・サーバで実行されて、ワークステーションにインストールされたプログラムの使用が制御される。

20

**【 0026 】**

具体的には、カーネル拡張モジュールが、ワークステーションのブートストラップ中にロードされる。図 3 を参照すると、カーネル拡張モジュールによって実施されるルーチンが、ブロック 302 で開始され、ブロック 304 のアイドル・ループでサイクルして、イベントの発生を待つ。プログラムの実行要求が検出される場合に、ブロック 306 ~ 308 が実行され、対応する応答メッセージが実行時エージェントによって返される場合に、ブロック 310 ~ 315 が実行され、プログラムの終了が検出される場合に、ブロック 318 が実行され、どの場合でも、このルーチンは、その後、ブロック 304 に戻って、新しいイベントの発生を待つ。逆に、ワークステーションがシャットダウンされた場合には、ルーチンは、最終ブロック 322 で終了する。

30

**【 0027 】**

ワークステーションのブートストラップには、実行時エージェントのロードも含まれ、これによって、ブロック 324 から始まるルーチンが実施される。このルーチンは、ブロック 326 のアイドル・ループでサイクルして、イベントの発生を待つ。実行要求が通知される場合には、ブロック 328 ~ 334 が実行され、対応する応答メッセージがメータリング・サーバから受け取られる場合には、ブロック 336 ~ 346 が実行され、プログラム終了が通知される場合には、ブロック 348 ~ 350 が実行され、タイムアウトが満了した場合には、ブロック 364 ~ 374 が実行され、どの場合でも、このルーチンは、ブロック 326 に戻って、新しいイベントの発生を待つ。逆に、ワークステーションがシャット・ダウンされた場合には、このルーチンは、最終ブロック 375 で終了する。

40

**【 0028 】**

それと同時に、ライセンスエージェントが、メータリング・サーバのブートストラップ中にロードされる。図 5 を参照すると、ライセンスエージェントによって、ブロック 376 で開始されるルーチンが実施され、その後、ブロック 378 のアイドル・ループでサイクルして、イベントの発生を待つ。実行要求メッセージが受け取られる場合には、ブロック 380 ~ 386 が実行され、終了メッセージが受け取られる場合には、ブロック 387 が実行され、情報がワークステーションから受け取られる場合には、ブロッ

50

ク 388 ~ 390 が実行され、検査メッセージが、事前に設定された周期の間に各関連するワークステーションから受け取られない場合には、ブロック 392 ~ 393 が実行され、管理サーバから管理メッセージが受け取られる場合には、ブロック 394 が実行され、どの場合でも、このルーチンは、その後、ブロック 378 に戻って、新しいイベントの発生を待つ。逆に、メータリング・サーバがシャット・ダウンされた場合には、このルーチンは、最終ブロック 396 で終了する。

#### 【 0029 】

図 3 に戻ると、エンドユーザが、たとえばめいめいのアイコンをマウスでダブル・クリックすることによって、ワークステーションでのプログラムの実行を要求する時に ( ブロック 397 )、カーネルによって、新しいプロセスが開始される。プロセスの開始が、カーネル拡張モジュールに通知され ( いわゆるカーネル・フック技法を使用して )、ブロック 306 で、カーネル拡張モジュールによって、その情報が実行時エージェントに報告される。カーネル実行モジュールによって、ブロック 308 でプロセスが中断される。

10

#### 【 0030 】

通知にตอบสนองして、実行エージェントによって、ブロック 328 で、プログラムが制御カタログに含まれるかどうかを検証される。そうでない場合には、この方法はブロック 330 に移り、実行時エージェントによって、カーネル拡張モジュールに、プログラムを開始できることが通知される。プログラムは、ブロック 332 で内部ログに追加される。それに応答して、カーネル拡張モジュールによって、プログラムがその実行を開始することをイネーブルするために、ブロック 312 でプログラムに関連するプロセスが再開される。逆に、プログラムが制御カタログに含まれる場合には、ブロック 334 で、実行時エージェントによって、対応する実行要求メッセージ ( ワークステーションの処理能力または作業メモリ容量などのプログラムの実行環境を定義する構成情報を含む ) がメータリング・サーバに送られる。

20

#### 【 0031 】

図 5 を参照すると、メータリング・サーバで動作するライセンスング・エージェントは、実行要求メッセージを受け取る時に、ブロック 380 で、プログラムの実行が、許可カタログに保管されている許可された使用条件に従うかどうか ( たとえば、実行環境が許可される環境と一致し、実行中のプログラムのインスタンスの数が許容される最大値に達していないかどうか ) が検証される。ライセンスング・エージェントは、ブロック 382 で、対応する応答メッセージをワークステーションに返す。その後、ブロック 384 でテストを実行して、実行の許可が与えられたかどうかを判定する。そうである場合には、ライセンスング・エージェントによって、ブロック 385 でプログラムがメータリング・テーブルに追加され、その後、ブロック 386 に進むが、そうでない場合には、直接にブロック 386 に進む。ブロック 386 を検討すると、メータリング・サーバによって、製品の使用可能ライセンスの事前に設定された比率 ( 80 % など ) に達した時にシステム管理者に通知の電子メールが送られ、メータリング・サーバが製品のライセンスを使い果たした時にもう 1 つの通知の電子メールが送られる。

30

#### 【 0032 】

図 3 に戻ると、実行時エージェントが応答メッセージを受け取るや否や、ブロック 336 でテストを行って、実行の許可が与えられたかどうかを判定する。そうである場合には、この方法は、ブロック 338 ( 下で説明する ) に進む。逆の場合に、実行エージェントが、ブロック 340 で、代替メータリング・サーバがサーバ・テーブルにリストされているかどうかを判定する。1 つまたは複数の代替メータリング・サーバが使用可能である場合には、実行時エージェントによって、各代替メータリング・サーバに順番に実行要求メッセージが送られて、上で説明したものと同一の検証が実行され、その後、この方法はブロック 338 に進む。逆に、代替メータリング・サーバが使用可能でない場合には、この方法は、直接にブロック 338 に進む。

40

#### 【 0033 】

ブロック 338 を検討すると、実行時エージェントによって、応答メッセージがカーネ

50

ル拡張モジュールに転送され、その後、判断ブロック344に進む。実行の許可が与えられた場合には、ブロック346で、プログラムを実行中テーブルに追加する（逆の場合には、この方法はブロック326に直接に戻る）。

#### 【0034】

カーネル拡張モジュールによって、応答メッセージが受け取られるや否や、ブロック310で、実行の許可が与えられたかどうかを判定するテストが実行される。そうである場合には、カーネル実行モジュールによって、ブロック312でプログラムに関連するプロセスが、プログラムの実行を開始できるようにするために再開される。逆に、実行の許可が拒否された場合には、カーネル拡張モジュールによって、ブロック315で、プログラムの開始を防ぐためにプロセスが打ち切られる。

10

#### 【0035】

プログラムの実行が終了される時（ブロック398）に、イベントが、カーネル拡張モジュールに通知され、カーネル拡張モジュールによって、ブロック318で、その情報が実行時エージェントに報告される。この通知に回答して、ブロック348で、実行時エージェントによって、プログラムが実行中テーブルに含まれるかどうかを検査される。そうである場合には、実行時エージェントによって、ブロック349で対応する終了メッセージがメータリング・サーバに送られ、ブロック350に移り、プログラムが、実行中テーブルから削除される（逆の場合には、この方法はブロック326に直接に戻る）。図5を参照すると、ライセンシング・エージェント（メータリング・サーバ側）が、終了メッセージを受け取る時に、プログラムが、めいめいのライセンスを解放するために、ブロック387でメータリング・テーブルから削除される。

20

#### 【0036】

図4を参照すると、たとえば数分の、タイムアウトが満了した時には、必ず（ブロック364）、実行時エージェントが、ワークステーションでのプログラムの実行の許可を与えたメータリング・サーバのそれぞれ（実行中テーブルで示される）に、検査メッセージを送る。この方法は、その後、ブロック366に進み、実行時エージェントによって、内部ログ全体を関連するメータリング・サーバに送る。

#### 【0037】

その後、判断ブロック368でテストを行って、より長いタイムアウト（たとえば、数十分の）が満了したかどうかを判定する。そうである場合には、ブロック370で、インストールされたプログラムに関する情報を収集するために、実行時エージェントによって、ワークステーションのハードディスクがスキャンされる。ブロック372に継続して、そのように収集されたインストール情報（プログラム・インベントリに保管される）が、メータリング・サーバに送られる。ブロック374で、ユーザの介入を必要とせずに、実行時エージェントによって、サーバ・テーブル、制御カタログ、または実行時エージェントのコードあるいはこれらの組合せに対するアップグレードが（メータリング・サーバから）ダウンロードされる。

30

#### 【0038】

図5を参照する。メータリング・サーバで動作するライセンシング・エージェントに移ると、関連ワークステーションから受け取るすべての情報（内部ログまたはプログラム・インベントリなど）が、ブロック388で保管される。メータリング・サーバによって、内部ログが受け取られる時に、この方法では、ブロック389で、内部ログにリストされた各プログラムが、グローバル・カタログに含まれるかどうかを検査する。そうである場合には、この方法は、ブロック390に進み、プログラムが、ワークステーションに配布される制御カタログおよび許可カタログ（対応する許可される使用条件に関する制限なし）に追加され、この形で、プログラムの実行の次の要求のすべてが、必ずイネーブルされて、システム管理者による分析のためにプログラムの使用を追跡できるようになる。逆の場合には、この方法は、ブロック378に直接に戻る。

40

#### 【0039】

ブロック392を検討すると、検査メッセージが、事前に設定された期間（たとえば1

50

0分)の間に、一般に関連するワークステーションから受け取られない場合に、このワークステーションで動作するすべてのプログラムが、ブロック392でメータリング・テーブルから削除され(めいめいのライセンスを解放するために)、通知の対応する電子メールが、ブロック393でシステム管理者に送られる。

【0040】

さらに、管理メッセージが管理サーバから受け取られる時に、必ず、対応するアクションがブロック394で実行される。たとえば、各メータリング・サーバの許可カタログを、管理サーバに保管されたマスタ・カタログから抽出することができ、各許可カタログが、対応するメータリング・サーバに展開される(これによって、ネットワーク状で転送される情報の量が減る)。さらに、管理サーバによって、メータリング・サーバから情報(10 対応するワークステーションにインストールされた、またはそこで動作する、あるいはその両方のプログラムに関する情報)を収集することができる。管理サーバによって、異なるレポートを作成することができる。詳細には、システム管理者が、製品の使用(開始時刻および終了時刻によってフィルタリングされる)、特定の製品の詳細な使用状況、インストールされ、使用され、ライセンスを交付された製品の比較ビューに関するレポートを要求することができる。セキュリティ・プロファイルも、管理サーバで定義され、その結果、ワークステーションの各エンドユーザが、そのユーザが使用しているプログラムに関する情報だけにアクセスできるようになる。この形で、管理サーバによって、使用状況データ、調達データ、およびインベントリ・データが収集され、管理される単一のリポジトリが実施される。20

【0041】

同様の考慮点が、同等の方法が実行される場合(たとえば、上で説明した動作を並列に実行する複数の並行プロセスを使用する場合)、他の機能が設けられる場合、プログラムの実行が別のプログラムによって要求される場合、実行要求メッセージに異なる情報が含まれる場合、タイムアウト値が異なるかシステム管理者によってカスタマイズできる場合、同等の検査メッセージが定期的にメータリング・サーバに送られる(たとえば、実行中テーブルの対応するレコードを含む)場合、通知の電子メールがシステム管理者に送られない場合などにあてはまる。代替案では、提案される方法が、オーディオおよびビデオの録音・録音物、電子ブック、またはマルチメディア作品など、同等のデジタル・エンコードされた製品の使用の制御にも使用される。30

【0042】

より一般的に、本発明によって、デジタル・エンコードされた製品の使用を制御するコンピュータ実施される方法が提供される。この方法は、コンピュータでのデジタル・エンコードされた製品の実行を要求するステップから開始され、この実行要求は、製品の開始の前にインターセプトされる。実行要求が、製品の許可された使用条件に従うことが、検証され、製品の開始が、検証の結果に応じてイネーブルまたは防止される。

【0043】

本発明の解決策を用いると、コンピュータで動作するプログラムを、リアルタイムで制御できるようになる。この形で、非常に有効なライセンシング検証が、低コストで実行される。40

【0044】

提案される方法の動作は、極めて安全である。というのは、プログラムが、開始そのものを妨げられるため、既にデータを処理しており潜在的に危険な動作を実行している間に強制終了されるようなことはないからである。さらに、プログラムの保全性が、許可の拒否によって影響されず、その結果、プログラムの実行を、後に簡単にイネーブルできる(コードを再インストールする必要なしに)。

【0045】

本発明の解決策によって、テクニカル・ライセンスによるものまたはそうでないもののいずれかの、すべてのタイプのプログラムの使用を制御する統合された解決策がもたらされる。したがって、許可された使用条件の実施を制御する特殊な論理を含まないプログラ50

ムであっても、制御することができ、それと同時に、提案される方法によって、ライセンス管理システムを活用する既存の解決策もサポートされる。

【0046】

上で説明した本発明の好ましい実施形態によって、さらなる利益も提供される。

【0047】

具体的には、実行要求が、めいめいのプロセスの開始を検出することによってインターセプトされ、プロセスが、中断され、検証の結果に従って再開されるか打ち切られる。

【0048】

この手順は、非常に単純であると同時に、非常に有効である。

【0049】

有利なことに、オペレーティング・システムは、プロセスの開始をカーネル拡張モジュールに通知し、カーネル拡張モジュールは、実行要求を、ワークステーションで動作する実行時エージェントに報告する。それと同時に、カーネル拡張モジュールは、プロセスを中断する（実行要求に対する応答を待つ）。

【0050】

カーネル拡張モジュールおよび実行時エージェントを設けることによって、本発明の解決策の実施形態が、プラットフォーム独立になり、あらゆる種類のオペレーティング・システムに簡単に移植可能になる。

【0051】

本発明の好ましい実施形態では、実行要求が、メータリング・サーバによって処理され、メータリング・サーバによって、それに関連する複数のワークステーションが制御される。

【0052】

提案された解決策を用いると、ライセンスの非常に柔軟な管理が可能になる。たとえば、ワークステーションでの各プログラムのインスタンスの最大個数の並行使用を、非常に単純な形で単一の点から制御することができる。

【0053】

有利なことに、関連するサーバが、すべての使用可能なライセンスを既に許可している場合に、ワークステーションは、代替メータリング・サーバに連絡することを許可される。

【0054】

したがって、各製品の使用可能なライセンスの総量を、組織内のすべてのワークステーション（異なるメータリング・サーバによって制御される場合でも）の間で共用することができる。

【0055】

しかし、本発明の解決策は、異なる形で実行要求をインターセプトし（たとえば、エンドユーザ・アクションを監視するため）、カーネル拡張モジュールまたは実行時エージェントだけを使用し、ワークステーションによって直接に実行要求を処理し（メータリング・サーバなしで）、ワークステーションに代替メータリング・サーバに連絡させずに、実行される。

【0056】

制御カタログに、計量されるプログラムがリストされることが好ましい（他のプログラムの実行は、想定される方法によって影響されない）。

【0057】

この特徴によって、システム管理者によってカスタマイズされたライセンスング・ポリシーを実施することが可能になる。

【0058】

制御カタログに含まれないプログラムは、必ず実行を許可されるが、内部ログに追加され、内部ログは、定期的にメータリング・サーバに送られる（ワークステーションの許可カタログおよび制御カタログを更新するために）。

10

20

30

40

50

## 【 0 0 5 9 】

異なるワークステーションによって供給される内部ログを用いると、新しい製品の使用を簡単に追跡できるようになる（使用されるや否や、メータリング・サーバに報告されるライセンス交付されないプログラムと共に）。

## 【 0 0 6 0 】

代替案では、制御カタログが設けられず（必ず制御されるすべてのプログラムについて）、制御カタログに含まれないプログラムの実行は、メータリング・サーバに即座に通知され、許可カタログおよび制御カタログが、内部ログに従って異なる形で更新される（グローバル・カタログの使用もなしで）か、制御カタログにリストされたプログラムだけが、ワークステーションで実行される（許可されるならば）。

10

## 【 0 0 6 1 】

さらに、プログラムが実行を終了する時に、終了メッセージが、ワークステーションから、めいめいの実行許可を与えたメータリング・サーバに送られ、そのプログラムは、実行中テーブル（ワークステーション側）およびメータリング・テーブル（メータリング・サーバ側）から削除される。

## 【 0 0 6 2 】

この手順によって、プログラムが実行を終了するや否やの対応するライセンスの直接解放が保証され、ライセンスは、他のワークステーションによる使用のために即座に使用可能になる。

## 【 0 0 6 3 】

有利なことに、各ワークステーションは、実行の許可を与えためいめいのメータリング・サーバに検査メッセージを送り、ワークステーションからの検査メッセージが事前に設定された期間の間に受け取られない場合に、メータリング・サーバは、そのワークステーションによって使用されたライセンスを自動的に解放する。

20

## 【 0 0 6 4 】

この形で、たとえばネットワーク問題のゆえに、ワークステーションがライセンスを解放できない場合に、ライセンスが、メータリング・サーバ側で自動的に解放される。

## 【 0 0 6 5 】

しかし、本発明の解決策は、異なる形での各プログラムの終了の検出（たとえば、各メータリング・サーバの制御下でのポーリング手順）によって、検査メッセージの定期的伝送さえなし（ライセンスが、毎晩など、事前に設定された期間の後に自動的に解放される）で、適切に実施される。

30

## 【 0 0 6 6 】

有利なことに、各ワークステーションは、定期的に、インストール情報を収集し、その情報が、関連するメータリング・サーバに送られる。

## 【 0 0 6 7 】

想定された特徴によって、各ワークステーションにインストールされた製品を記憶し、監視することが可能になる（使用されないものであっても）。

## 【 0 0 6 8 】

本発明の好ましい実施形態では、単一の管理サーバによって、中央から、複数のメータリング・サーバが制御される。

40

## 【 0 0 6 9 】

提案された3ティア・アーキテクチャ（ワークステーション、メータリング・サーバ、および管理サーバ）によって、集中化された点からの非常に柔軟なライセンスの管理が可能になる。

## 【 0 0 7 0 】

たとえば、管理サーバによって、すべてのワークステーションに関するプログラム使用およびインベントリ情報に関する履歴レポートを提供するために、すべてのメータリング・サーバから情報が収集される。さらに、管理サーバによって、集中化された点からライセンシング情報を維持するための管理インターフェースが提供される。

50

## 【0071】

したがって、過剰応諾状態（未使用の製品に関して使用可能なライセンスが多すぎる場合）を検出することによって、費用を節約することが可能であり、あるいは、潜在的な侵害状態を事前に検出する（ライセンス不足になる前に訂正措置を講じるために）ことが可能である。

## 【0072】

提案されたアーキテクチャは、高いスケーラビリティを特徴とする。たとえば、小さい会社では、1つのメータリング・サーバと1つの管理サーバだけが必要である（同一のコンピュータにインストールすることもできる）。逆に、大きい会社では、複数のメータリング・サーバ（単一の管理サーバによって制御される）を、1つまたは複数のネットワーク内に設けることができる。その一方で、管理サーバを、リモート位置にインストールすることができ、管理サーバによって、異なる施設のメータリング・サーバを制御することができる（たとえば、他の会社にソフトウェア管理サービスを提供する組織で）。

10

## 【0073】

代替案では、インストール情報が、要求時に収集され、提案される方法が、プログラムの実行だけを制御する（インストールは制御しない）、あるいは、システムが、異なるアーキテクチャを有する（管理サーバなしも可能である）。

## 【0074】

有利なことに、本発明による解決策は、CD-ROMで提供されるコンピュータ・プログラム・アプリケーションを用いて実施される。このアプリケーションは、各ワークステーションおよび関連するメータリング・サーバにインストールされるプログラムからなる。

20

## 【0075】

代替案では、プログラムが、フロッピディスクで提供されるか、ハードディスクに事前にロードされるか、他のコンピュータ可読媒体に保管されるか、ネットワーク（通常はインターネット）を介してワークステーションおよびメータリング・サーバに送られるか、ブロードキャストされるか、より一般的に、作業メモリに直接にロード可能な他の形で提供される。しかし、本発明による方法は、ハードウェア構造を用いて、たとえば半導体材料のチップに集積されて、実行される。

## 【0076】

当然、ローカル要件および特定の要件を満足するために、当業者は、上で説明した解決策に多数の修正および変更を加えることができ、そのすべてが、請求項によって定義される本発明の保護の範囲に含まれる。

30

## 【図面の簡単な説明】

## 【0077】

【図1】本発明の方法を使用することができるデータ処理システムを示す基本ブロック図である。

【図2】本発明のシステムに含まれるワークステーションおよびメータリング・サーバの作業メモリの部分的内容を示す図である。

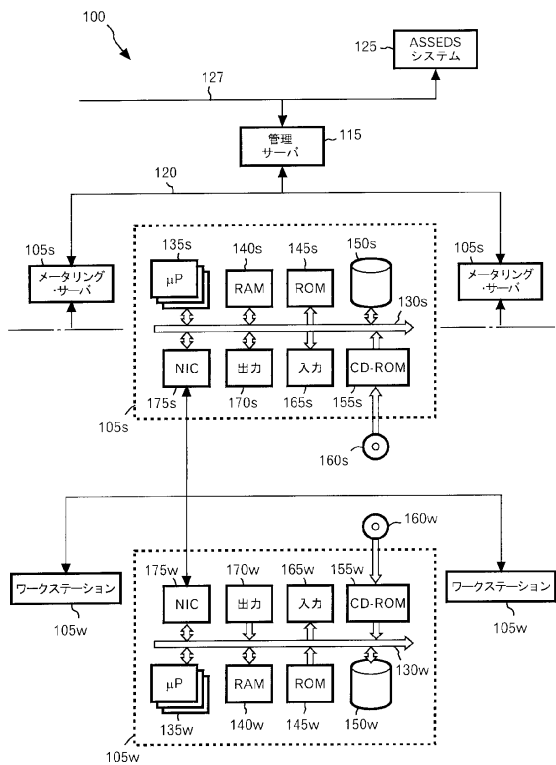
【図3】ソフトウェア・プログラムの使用を制御する方法の論理を示す流れ図である。

40

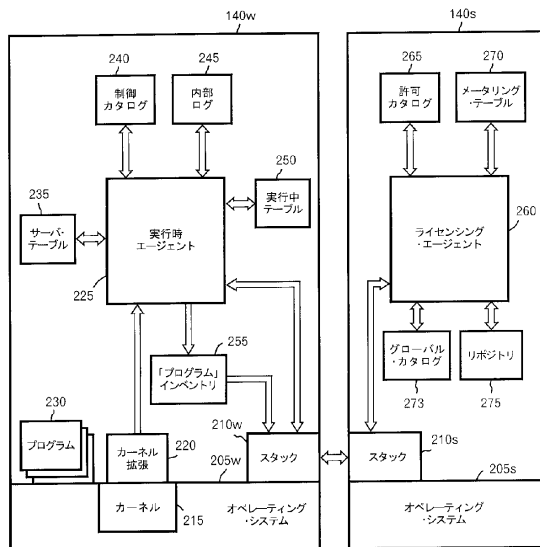
【図4】ソフトウェア・プログラムの使用を制御する方法の論理を示す流れ図である。

【図5】ソフトウェア・プログラムの使用を制御する方法の論理を示す流れ図である。

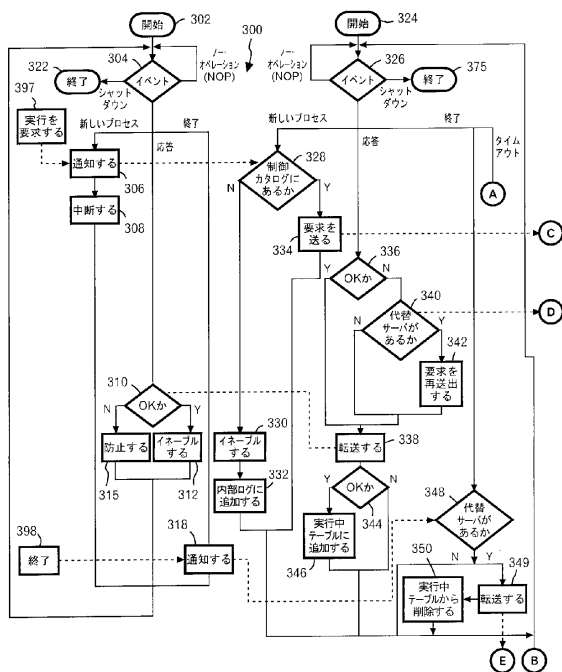
【図1】



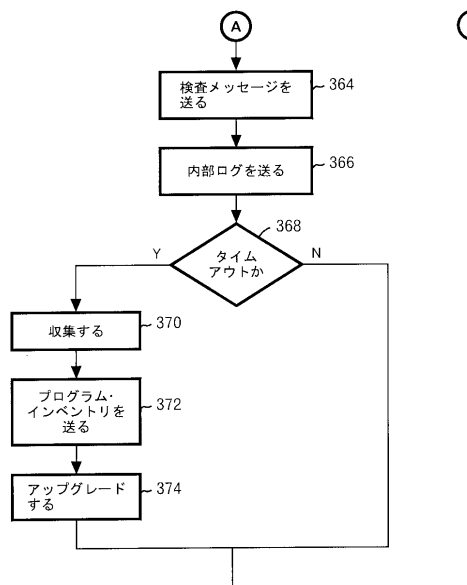
【図2】



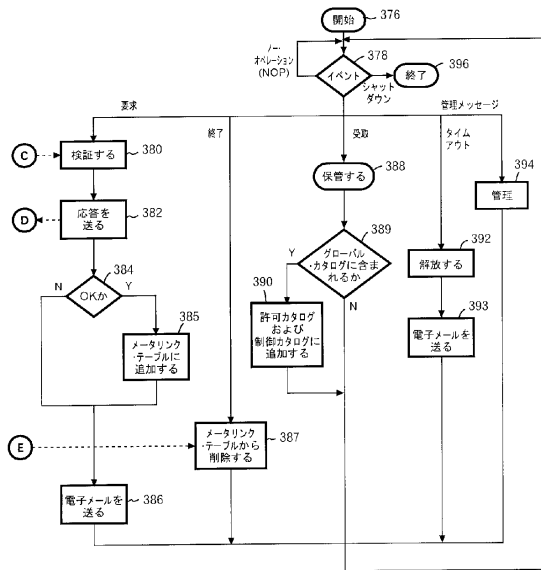
【図3】



【図4】



【 図 5 】



## フロントページの続き

- (72)発明者 パストレッリ、ベルナルド  
イタリア I - 176100 ラクイラ ヴィア・ヴェンティ・セッテンプレ 13/A
- (72)発明者 ディ・グイリオ、ドメーニコ  
イタリア I - 00142 ローマ ヴィア・アレッシオ・バルドヴィネッティ 85
- (72)発明者 ロッシ、リッカルド  
イタリア I - 00149 ローマ ヴィア・レオポルド・ルスポリ 119

審査官 石川 正二

- (56)参考文献 特開平06 - 223040 (JP, A)  
特開2001 - 222424 (JP, A)  
特開平10 - 171649 (JP, A)  
特開平09 - 237189 (JP, A)  
特開2001 - 142689 (JP, A)  
特開2001 - 236219 (JP, A)  
米国特許第05758069 (US, A)  
米国特許第06021438 (US, A)  
特開2000 - 305776 (JP, A)  
特開2000 - 057035 (JP, A)  
米国特許第05790664 (US, A)  
鈴木信夫, ネットワーク時代のライセンス管理、ライセンスサーバが切り札に, 日経エレクトロニクス, 日本, 日経BP社, 1990年11月26日, no. 514, p. 157 - p. 163, JPN402004421

(58)調査した分野(Int.Cl., DB名)

G06F 21/22

G06F 1/00