

(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2004-341967
(P2004-341967A)

(43) 公開日 平成16年12月2日(2004.12.2)

(51) Int. Cl.⁷
G06F 9/445

F I
G06F 9/06 610Q

テーマコード(参考)
5B076

審査請求 未請求 請求項の数 3 O L (全 22 頁)

(21) 出願番号	特願2003-139750 (P2003-139750)	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目6番地
(22) 出願日	平成15年5月19日(2003.5.19)	(74) 代理人	100075096 弁理士 作田 康夫
		(72) 発明者	須藤 裕実 神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部 内
		Fターム(参考)	5B076 AA04 AC01 AC05

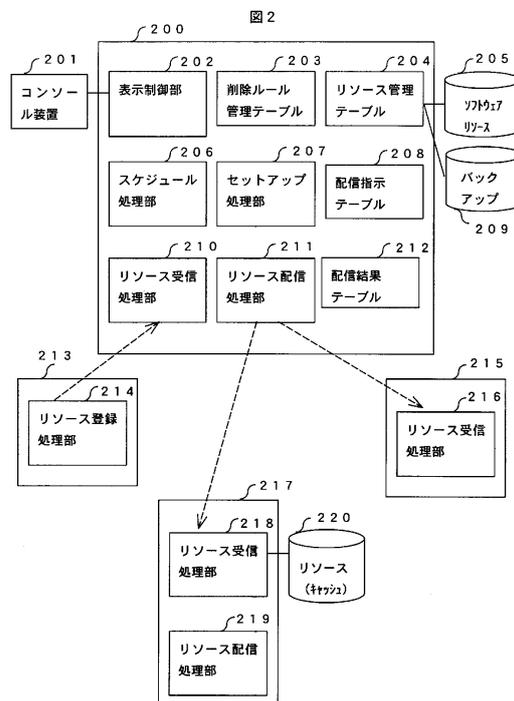
(54) 【発明の名称】 ソフトウェアリソース管理方法

(57) 【要約】

【課題】ソフトウェア配信システムにおいて配信するソフトウェアリソースの増大に伴い管理が煩わしくなっている。

【解決手段】ソフトウェア配信システムのソフトウェアリソースを決められたルールに基づいて自動的に削除する機能と必要に応じて自動的に回復する機能を提供する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

ソフトウェアリソース、ソフトウェアリソースのリソース管理テーブル、ソフトウェアリソースの削除ルール管理テーブル、ソフトウェアリソースの配信結果テーブルを持ち、リソース管理テーブル、削除ルールテーブル、配信結果テーブルを基にソフトウェアリソースの削除および回復を行うことを特徴とするソフトウェアリソース管理方法。

【請求項 2】

請求項 1 に記載のソフトウェアリソース管理方法において、リソース管理テーブル、削除ルールテーブルを基にソフトウェアリソースの削除をできないようにすることを特徴とするソフトウェアリソース管理方法。

10

【請求項 3】

請求項 1 に記載のソフトウェアリソース管理方法において、リソース管理テーブルを基に削除したソフトウェアリソースを上位のバージョンまたはリビジョンのソフトウェアリソースを適用し回復することを特徴とするソフトウェアリソース管理方法。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は複数台のコンピュータとネットワークにより構成するソフトウェア配信管理システムにおいて、配信するソフトウェアリソースを管理するための方法に関する。

【0002】

20

【従来の技術】

従来より、複数のコンピュータをネットワークで接続し、一方のコンピュータからネットワークを経由してもう一方のコンピュータにソフトウェアやデータなどのソフトウェアリソースを配信する機能をもつソフトウェア配信管理システムが発案され、さまざまなベンダーより製品および実用化がなされている。

【0003】

ソフトウェア配信管理システムの概要の一例を図 1 を用いて簡単に説明する。ソフトウェアリソース 103 (以下リソースと略す) の登録システムから、リソースを配信管理サーバ 101 に登録する。リソースは媒体 102 に格納される。配信管理サーバから、中継サーバ 105 やクライアント 106 ~ 110 に対してリソースの配信をネットワーク 104 30 を経由して行うものである。中継サーバ 105 を設置し、中継サーバにリソースをキャッシングし、中継サーバからクライアント 108 ~ 110 に対してリソースを配信することにより、分散的で効率良いリソース配信が可能である。各クライアントは、配信結果の情報を配信管理サーバに送付することにより、配信管理サーバでは、どのクライアントに対してどのリソースを配信したかどうかを一元的に管理できるようになる。

【0004】

配信管理サーバに保持するリソースについて従来では手動により削除を行っていた(例えば、特許文献 1 参照。)。例えば、特許文献 1 においては配信する情報としてルールベースというデータを扱っているが、GUI 上の削除ボタンによって手動により削除を行っている。

40

【0005】**【特許文献 1】**

特開 2002 - 351664 号公報

【0006】**【発明が解決しようとする課題】**

パーソナルコンピュータやワークステーションが普及浸透すると共に、配信管理サーバにおいて配信する必要のあるリソースが多種多様化してきていること、過去に配信したリソースのバージョンアップが繰り返し行われることにより、管理しなければならないリソースの数も増え、配信不要な例えば古いソフトウェアなどを削除する必要が生じてきている。

50

【0007】

特に、近年では、コンピュータウィルスの流行により、ウィルスを検知および駆除するためのソフトウェアやパターンファイルのバージョンアップが日々行われており、管理するリソースの数は日増しに急増傾向になってきている。ただし、リソースは、もともと重要なデータであったりソフトウェア資産価値を含んでいる事情もあり、その性質上手動により確認しながら削除する必要があるが、誤操作により必要なソフトウェアリソースを削除してしまうことがある。また、一旦削除してしまったソフトウェアリソースは再度登録システムから登録しなおす必要があり手間がかかっていた。本発明では、これらの課題を解決することを目的とする。

【0008】

【課題を解決するための手段】

上記目的を達成するために、ソフトウェアリソース管理方法において、リソースをあらかじめ設定したルールに基き、自動的に削除できるようにすることで解決できる。さらに、配信管理サーバにおいては、配信処理の関係でリソースが必要となったり、登録したリソースが重要な情報であったり資産価値をもつ側面も持つことから、一気に削除してしまうのではなく、必要に応じて一時的に保管管理、バックアップや元の状態に回復できるようにする。

【0009】

【発明の実施の形態】

以下、本発明の実施例を図面に基づいて詳細に説明する。図2は、本発明における実施例である配信管理システムの構成を示す図である。配信管理サーバとして計算機200上に配信管理サーバのプログラム202、206、207、210、211を動作させる。計算機213上にはリソースの登録システムとして、リソース登録処理のプログラム214を動作させる。なお、登録システムは独立した計算機上に配置する必要はなく、配信管理サーバ200上でも動作させることができる。コンピュータアーキテクチャの違いにより扱うリソースをシステム毎に分ける必要がある場合に登録システムを個別に存在させる必要がある。計算機217上には中継サーバとして、中継処理を行うためのプログラム218、219を動作させる。

【0010】

なお、本説明では省略しているが中継サーバはシステム構成上の上位システムである配信管理サーバからの配信指示を受信する処理が存在する点を除けば基本的に配信管理サーバとほぼ同様のシステム構成となる。計算機215上にはクライアントとして、リソースを受信するためのプログラム216を動作させる。

【0011】

リソースを登録し配信するまでの基本的な動作を説明する。なお実施の形態における流れ図には特に記載はしないがリソースを扱う部分には排他制御が施されており、リソースの登録処理やリソースの配信処理において競合が発生しても不正な状態にならないものとする。

【0012】

最初にリソースの登録処理について図5を用いて説明する。サーバ側はリソース登録システムからの登録要求を待っている状態(506)で、リソース登録システムからリソースの登録要求(501)を受信したら、サーバ側が持つリソースの一覧をリソースが持つ属性と共にすべての情報を引き渡す(507、502)。

【0013】

リソース登録システム上で登録するリソースの名称、バージョン、リビジョンを設定し(503)、受け取った一覧情報を検索して既に同一のリソースが登録されていないかどうかを確認する(504)。リソース名称やバージョン、リビジョンの設定は、リソースの管理や配信するリソースの見分けを容易につけるために設定する。リビジョンとはマイナーバージョンの意味である。リソース名称、バージョン、リビジョンを組み合わせた名称で同一名称の既に登録済みのリソースがある場合は再度設定し直す(504、503)。

10

20

30

40

50

【0014】

予め、重複するリソースを参照できるようにするために、サーバから受け取ったリソース情報の一覧を基にリソースの一覧をコンソールなどに表示し参照できるようにすることもできる。なお、サイズ、登録日はユーザが設定しても良いが、OSのAPIなどを使用して求め、属性値を設定する。また、リソースについては、複数のファイルにより構成する場合は管理しやすいようにするため一つのファイルにアーカイブして登録する。アーカイブしたファイルは、例えばネットワークが公衆回線であった場合はセキュリティ維持のために暗号化したり、例えばネットワークが低速回線であった場合は転送量を減らすため圧縮することも可能であり、配信後にクライアント側で圧縮、暗号を解除する。

【0015】

既に登録済みのリソースが存在しない場合は、アーカイブしたファイルを新たなリソースとしてリソースの属性情報と共にサーバ側に送信する(505)。サーバ側はリソースとそのリソースの属性情報を受信し、リソースはディスクに格納し、属性情報はリソース管理テーブルに設定する(508)。

【0016】

図3にリソース管理テーブルを示す。それぞれの情報は、登録システム側で設定した属性情報が入る。ファイルパス307には格納先ディスクのパスが設定される。リソース名称、バージョン、リビジョンの組み合わせにより、ユニーク性を持つ。例えば、図3に示すようにリソース名称が同一でもバージョン、リビジョンが異なれば、別のリソースとして存在するものとする(308、309、310)。リソースが正しく格納できたら、登録を完了し再び登録要求の待ち状態となる(506)。登録システム側は登録処理を終了する。

【0017】

次に、リソースの配信処理について図6を用いて説明する。配信処理はサーバからどのクライアントにどのリソースを配信するかを決定し配信指示情報を作成する(601)。配信指示は詳細に説明しないが、配信するあて先と配信するリソース(リソース名称、バージョン、リビジョン)により構成される情報である。作成の際にはリソース管理テーブルから配信するリソースを選択する。そして、配信指示情報を基に配信するクライアントに対して配信指示を渡す(602)。

【0018】

クライアント側では、サーバ側からの配信指示を待っている状態となっており(606)、配信指示を受けたらサーバからリソースを受信する(603、607)。配信が何らかの理由により失敗した場合は何回かリトライを試み配信が成功するまで繰り返す(604、603)。クライアント側では、リソースの受信が完了したら、例えばリソースがアプリケーションの場合はインストールプログラムを起動しインストールを実施し、例えばユーザデータであれば所定の位置にファイルをコピーするなど、リソースに合わせたインストール処理を行う。インストールが完了したら、サーバ側にインストール結果情報を送信する(608)。

【0019】

サーバ側はインストール結果情報を受信したら図4に示す配信結果テーブルのインストール回数(402)を加算し、最終アクセス日(404)を実行した日に設定して更新する(605)。配信が終了したら、クライアント側は再びサーバからの配信指示の待ち状態となる(608、606)。サーバ側も再び配信指示作成待ちとなる(605、601)。なお、サーバ側の配信処理はマルチスレッドを用いることにより、並行して複数のクライアントあての配信処理ができるものとする。

【0020】

リソースの配信結果についての配信結果テーブルを図4に示す。通番401は、図3のリソース管理テーブルの通番301とリンクしており、何のリソースが配信されたかどうかが把握できる。クライアントへのインストールの場合は、インストール回数402がクライアントにインストールした分更新され、どれぐらいのクライアントに配信が行われたか

10

20

30

40

50

を把握できる。通番 1 を例にすると、「A」という名称のリソースは、20 台のクライアントに配信されたことが分かる。

【0021】

図 6 では、サーバ側でクライアントに対し配信するリソースを選択するサーバ側が主導して配信を行う方法であるが、図 7 にはクライアント側が欲しいリソースを選択して配信するクライアント側が主導する配信の方法について説明する。まず最初に、サーバ側で登録されているリソース管理テーブルを参照し接続している全クライアント宛に配信指示情報を作成しておき(701)、クライアントからの配信要求の待ち状態となる(702)。クライアント側では、サーバに接続し配信可能なリソースの一覧情報を受け取る。さらに、そのリソース情報の中から受信したいリソースをユーザが選択して配信要求を行う(707)。クライアントが配信要求したリソースについて、サーバからクライアントに対して配信を行う(703、708、704)。リソースを配信後、クライアントでインストールを行い、インストール結果をサーバ側に送信する(709、705)。サーバ側では、リソースの追加や削除、接続するクライアントに変更が生じたら配信指示を変更し、クライアント側からの配信要求の待ち状態となる(706、702)。

10

【0022】

次に、図 8 には中継サーバを経由して配信する方法について説明する。中継サーバを経由して配信したいクライアントとリソース管理テーブル内の配信したいリソースから配信指示情報を作成し(801)、中継サーバに対して配信指示を渡す(802)。中継サーバ側では、サーバ側からの配信指示を待っている状態となっており(806)、配信指示を受けたらリソースを受信する(803、807)。

20

【0023】

リソースの受信が完了したら、中継サーバが持つリソース管理テーブルとディスクに格納する(808)。そして、下位システムであるクライアントに対して配信指示情報を転送し、サーバ側に、ダウンロード結果情報を送信する(809)。そして、下位クライアントからの配信指示があれば、サーバ側に渡す(810)。サーバ側は図 4 に示す配信結果テーブルのダウンロード回数(403)を加算し、下位クライアントからの配信結果の通知があればインストール回数(402)を加算し、最終アクセス日(404)を実行した日に設定して更新する(805)。転送が終了したら、中継サーバ側は再びサーバからの配信指示の待ち状態となる(809、806)。サーバ側も再び配信指示作成待ちとなる(805、801)。

30

【0024】

これにより、中継サーバの下位のクライアントは中継サーバからリソースを配信できる。中継サーバ上のリソースはキャッシュの役割を果たし、これにより分散的に効率よく配信できる。中継サーバへ何回ダウンロードしたかどうかを確認するには、配信結果テーブルのダウンロード回数 403 を参照することにより把握できる。通番 1 を例にすると、「A」という名称のリソースは、10 台の中継サーバに対してダウンロードしたことが分かる。また、インストール回数の情報も含めて、当該リソースへの最終アクセスがいつ行われたのかを把握できる。以上が基本的なリソースの登録から配信するまでの説明である。

【0025】

次に、本発明である配信管理システムにおけるリソースの管理方法について説明する。まず、リソースをどのように扱うかを図 23 を使用して説明する。登録されたリソースは、通常状態で配信に使用される(2301)。リソースに対してルールを設け、新たにリソースを登録した時などのイベントが生じたときにルールに従い完全に削除(2304)、削除への猶予期間を設けた一時保管状態(2302)、削除禁止(2303)にする。

40

【0026】

削除および一時保管状態になるには、例えばディスク容量の制限やリソースの個数の上限などを設定したルールに抵触した場合である。削除禁止になるには、登録した日からある一定の年月を経過しても削除処理されなかった場合である。削除禁止は古いリソースにもかかわらず配信には使用されるが、リソースの元となるファイルが古い故に製造中止など

50

により回復が困難であるため削除されないようにすることを想定している。一時保管状態になったリソースについては、完全に削除するまでの猶予期間を設けたもので、その間に使用された場合に、まだニーズがあるものとして猶予期間を延長したり再び通常状態にするために再登録し、使用されない場合は削除する。一時保管は一旦削除扱いとした後も配信に使用される場合に対処できるようにすることを想定している。

【0027】

図9には、本発明のためのリソース管理テーブルの内容を示す。図3と比較すると、リソースの配信可能な期限を設定する使用期限日907、一時保管状態にしたリソースをいつまで保管するかを設定する保管期限日908、全リソースにおける当該リソースのプライオリティであることを示す重要度909、当該リソースの削除の状態を示すステータス910、当該リソースの削除方法912、当該リソースの回復方法913の情報が追加されている。図9には回復方法に設定するパラメータの例914を示す。通番の‘2’のリソースについても例914と同様に何らかの値が設定されているが説明は省略する。

10

【0028】

図10には、図9のリソース管理テーブルに設定される定義を示す。定義1001は、図9のステータス910に設定される定義の一覧である。‘1’から‘5’の定義を持つ。‘1’は通常の状態であるリソースであることを示す。‘2’は新たに配信対象にはできないが、下位システムからの配信要求に備えてリソースの実体であるファイルを一時的に保管しておくリソースであることを示す。‘3’は‘2’と同じであるが、リソースの実体を圧縮した状態で保管しておくリソースであることを示す。‘4’は新たに配信対象にできるが、リソースの実体であるファイルを削除した状態で保管しておくリソースであることを示す。‘5’は絶対に削除されずに保管しておくリソースであることを示す。なお、ステータスを図23に当てはめると‘1’は通常状態、‘2’、‘3’、‘4’は一時保管状態、‘5’は削除禁止状態である。

20

【0029】

定義1002は、図9のリソースの削除方法912に設定される定義の一覧である。‘1’から‘4’の定義を持つ。‘1’はリソースの実体であるファイルは残す一時保管状態ステータスにするリソースであることを示す。‘2’は‘1’と同じであるがリソースの実体であるファイルを圧縮する一時保管状態ステータスにするリソースであることを示す。‘3’はリソースの実体であるファイルを削除する一時保管状態ステータスにするリソースであることを示す。‘4’はリソースの実体であるファイルを削除すると共にリソース管理テーブルからも削除するリソースであることを示す。‘1’および‘2’のリソースについては配信処理において新たに配信指示することができないものとし、配信指示が中継サーバおよびクライアントに渡ったあとのタイミングでリソースが削除されてしまったケースに対処するためのものである。また、‘3’のリソースについては、定義1003のリソースの回復方法で示す回復方法で回復し、主としてディスク容量の増大化を防ぐことを意識したものである。設定時は当該リソースの利用頻度のある程度予想し回復方法を適切な方法に設定する必要がある。

30

【0030】

定義1003は、図9のリソースの回復方法912に設定される定義の一覧である。‘1’から‘6’の定義を持つ。‘1’はインターネットを経由してリソースであるファイルをダウンロードして回復する方法をとるリソースであることを示す。‘2’はリソースの作成者宛てもしくは作成した企業宛てに向けてメールを送りリソースであるファイルを手して回復する方法をとるリソースであることを示す。‘3’はCD-ROMなどのインストールメディアからリソースであるファイルを戻して回復する方法をとるリソースであることを示す。‘4’はリソースの登録の際にあらかじめ別のディスクに取得しておいたバックアップからリソースを取得して回復する方法をとるリソースであることを示す。‘5’は中継システム上のキャッシュとして保持しているリソースを吸い上げて回復する方法をとるリソースであることを示す。‘6’は、当該リソースを回復せず、当該リソースの最も上位のリビジョンのリソースを代わりに配信する方法をとるリソースであることを

40

50

示す。この回復方法は、コンピュータウィルスの対策ソフトウェアなどのように日々リソースが更新され、常に最新のリソースを適用した方がシステムの運用上効果的なリソースについて指定する。なお、本発明においては、リビジョンのみを適用しているが、バージョンを適用することもできる。

【0031】

図11は削除ルール管理テーブルの内容を示す。テーブルは、リソースがどのような状況になった場合に削除するのかを示すルールの内容1101、ルールを適用するか否かを示す適用1102、リソースが削除に到るまでにどのタイミングで警告を出すかを示す警告1103、ルール適用時にどの範囲でリソースを削除するかの削除対象1104の列で構成されている。削除のルールは9個のルール1105～1113により構成されている。なお、ルールの内容の一部に‘n’で示す箇所があるが、その部分には整数値が入り値を自由に設定できることを示している。ルール1105で例えると、‘最大登録数n個を超えたときに削除する。’について、‘n’が100であれば100個の以上登録できないようにする意味である。これはルールの設定時に‘n’の部分に‘100’を設定することで可能となる。

10

【0032】

ルール1105では、n個のリソースが登録されたらリソースを削除することを示している。ルール1106では、登録したリソースの合計のサイズがnMBを超えたらリソースを削除することを示している。ルール1107では、新しいバージョンのリソースを登録したら、古いバージョンおよびリビジョンのリソースを削除することを示している。ルール1108では、新しいリビジョンのリソースを登録したら、古いリビジョンのリソースを削除することを示している。ルール1109では、一時保管状態であるリソース管理テーブルの使用期限日に到達したリソースを削除することを示している。ルール1110では、中継サーバにリソースをn回ダウンロードしたらリソースを削除することを示している。ルール1111では、クライアントからインストール完了の報告をn回受けたらリソースを削除することを示している。ルール1112では、中継システムへのダウンロードおよびクライアントからのインストール報告の最終更新がn日間無かったときにリソースを削除することを示している。ルール1113では、クライアント管理テーブルの登録日からn年が経過したリソースについて削除できないようにすることを示している。

20

【0033】

図12には、図11の削除ルール管理テーブルに設定される定義を示す。定義1201は、削除ルール管理テーブルの1102に設定される削除ルール適用の定義の一覧である。‘Y’または‘N’の定義を持つ。‘Y’はルールの適用を意味し、‘N’はルールを適用しない意味をもつ。定義1202は、削除ルール管理テーブルの1103に設定されるリソースが削除されるまでの警告のタイミングの閾値の定義の一覧である。‘1’から‘6’の定義を持つ。

30

【0034】

定義1203は、削除ルール管理テーブルに設定されるリソースの削除対象1104の定義の一覧である。‘1’から‘8’の定義を持つ。‘1’はリソース管理テーブルの登録日を参照し古いリソースから削除することを示す。‘2’はリソース管理テーブルのサイズを参照し大きいサイズのリソースから削除することを示す。‘3’はリソース管理テーブルのサイズを参照し小さいサイズのリソースから削除することを示す。‘4’はリソース管理テーブルの重要度を参照し重要度の値の大きいリソースから削除することを示す。重要度の値の設定方法については、図15に示す登録処理で説明する。‘5’は登録したリソースの名称と同一のリソースをリソース管理テーブルから検索し古いバージョンおよびリビジョンであるリソースをすべて削除することを示す。‘6’は登録したリソースの名称と同一のリソースをリソース管理テーブルから検索し古いリビジョンであるリソースをすべて削除することを示す。‘7’は登録したリソースの名称と同一のリソースをリソース管理テーブルから検索し古いバージョンおよびリビジョンのうち最新のバージョン以外のリソースをすべて削除することを示す。‘8’は登録したリソースの名称と同一のリ

40

50

ソースをリソース管理テーブルから検索し古いリビジョンのうち最新のリビジョン以外のリソースをすべて削除することを示す。

【0035】

図11の範囲1114と1115に示す枠内は図12に示す定義の値の設定可能な組合せの一覧を示しており、斜線の部分は設定ができないことを示す(設定が不要である)。例えば、ルール1105では、適用1102に‘Y’と‘N’のいずれかの定義を設定でき、警告1103に‘1’と‘2’のいずれかの定義を設定でき、削除対象1104に‘1’と‘4’のいずれかの定義を設定することができる。また、ルール1113では、適用1102に‘Y’と‘N’のいずれかの定義を設定でき、警告1103に‘5’と‘6’のいずれかの定義を設定でき、削除対象1104には定義を設定できない。

10

【0036】

図13は、ルールを設定するセットアップ処理を示す。本処理は、配信管理システムが動作する際に必ず設定される処理である。本説明においてはセットアップが行わずにサーバ機能を動作させることはできないものとする。セットアップ処理を開始した場合、まずサーバの全機能を一旦停止する(1301)。ここでは、セットアップ処理を実行する際にリソースの登録処理や配信処理などが動作中の可能性があることからサーバの機能が停止した後に処理を開始するものとし、サーバの全機能が停止したらセットアップ処理を行う。最初にリソースの削除ルールの設定を行う。削除ルール管理テーブルのルール1105~1113の適用の判断、および各ルールを適用する場合の警告機能の選択、削除対象リソースの組み合わせを決定し、必要に応じてルールの内容1101に含まれる‘n’に値を設定する(1303)。

20

【0037】

図14にコンソール上に表示する設定インタフェースの例を示す。画面1401では、削除ルール管理テーブルのルール1106についての削除方法についてGUIを用いて設定を行っている。削除ルールを適用するか否かのチェックボックス、最大個数を入力するためのエディットボックスなどから構成され、ユーザに設定してもらう。

【0038】

入力後の状態の例として図24に示すテーブルを用いて説明すると、ルール2401は、最大個数が100個を越えたときに重要度の低いリソースを削除し、25個、50個、75個、90個のそれぞれのリソースが登録された場合に警告メッセージを出すことを示す(ルールの内容のnには100を設定したものとする。)。ルール2403は、リソースの登録の際に同一名称のリソースの古いバージョンとリビジョンのリソースをすべて削除することを示す。ルール2404は、リソースの登録の際に同一名称のリソースの古いリビジョンのリソースのうち最新のリビジョンを1つだけ残してすべて削除することを示す。ルール2407は、100台のクライアントからインストール報告があった場合に当該リソースを削除し、25台、50台、75台、90台のそれぞれのインストール報告があった場合に警告メッセージを出すことを示す(ルールの内容のnには100を設定したものとする。)。ルール2408は、60日の間にクライアントまたは中継サーバからインストール報告やダウンロード報告が無かった場合に当該リソースを削除し、10日、20日、30日のそれぞれの日数が経過した場合に警告メッセージを出すことを示す(ルールの内容のnには60を設定したものとする。)。ルール2409は、登録日から5年経過した場合に当該リソースを削除できないようにし、6ヶ月、1年、3年のそれぞれの日数が経過した場合に警告メッセージを出すことを示す(ルールの内容のnには5を設定したものとする。)。以上のようにすることで削除ルールと削除対象を組み合わせた削除ルールを作成できる。

30

40

【0039】

次に、登録済みのリソースのリソース管理テーブルの情報について変更できるようにする(1305)。変更できる情報は、リソース管理テーブルの削除ルールに関係する使用期限日907、重要度909、削除方法912、回復方法913の情報についてである。これにより、リソースの削除のタイミングを短縮したり延長するなどリソースの登録以降に

50

再調整できる。さらに、ここでは定義 1003 の回復方法のうち '4' のバックアップから回復する方法において使用するバックアップ先のパス名を設定する。パス名には比較的大容量の固定ディスクなどのパスを設定する。セットアップ処理では、いずれかの処理 1303 および 1305 のみを選択することができる (1302、1304)。図 13 には特に記載していないが、この処理ではサーバ処理の基本的な動作の設定も行うことができるが省略する。

【0040】

次に、リソースの削除機能について説明する。図 15 はリソースの登録処理を示す。図 5 に示すリソースの登録と異なる点は、リソースを登録する際に削除ルールに関連する使用期限、重要度、削除方法、回復方法の属性の設定と、リソースを登録する際に設定された削除ルールに抵触するリソースがあった場合にリソースの削除を行う点で、他は同様である。リソース登録処理 1501 ~ 1502、サーバ側リソース受信処理 1508 ~ 1509 については、図 5 に示したリソース登録処理と同様であるため説明は省略する。

10

【0041】

リソース登録システム上で登録するリソースの属性の設定部分から説明する (1503)。ここではリソース名称、バージョン、リビジョン、登録日時の設定の他、使用期限日、重要度、削除方法、回復方法の設定を行う。使用期限日には当該リソースを配信できる使用可能な日付を設定する。この属性値は削除ルール管理テーブルのルール 1109 で参照され、期限を過ぎると当該リソースが削除の対象となる。重要度には、1 から n (通常状態のステータスであるリソースの全体個数) の重複しない値がリソース毎に設定されている。値が小さいほど重要度が高く、値が大きいほど重要度が低く、最大値が設定されているリソースが削除対象の最優先となる。登録時は 1 から n + 1 の間の値を 1 つ選択する。サーバ側では、当該リソースの重要度を基にリナンバリングする (設定した重要度以降の値を持つリソースの重要度を + 1 する)。削除方法は定義 1002 に示す定義から 1 つ選択する。削除方法にて '3' を選択した場合は回復方法を定義 1003 に示す定義から 1 つ選択する。さらに、回復方法に応じて、パラメータを設定する。回復方法が '1' の場合は、URL 名を設定する。'2' の場合はメールアドレスを設定する。'4' の場合は、何も設定しない。これはサーバ側で設定してあるバックアップのパス名を設定するためである。'5' の場合は、接続された中継サーバの名称を設定する。削除方法にて '4' および '6' を選択した場合は、回復方法を設定する必要は無い。'6' については、頻

20

30

【0042】

次に、受け取った一覧情報内を検索して既に同一のリソースが登録されていないかどうかを確認する (1504)。このとき、登録済みのリソースが存在しても、その登録済みのリソースが一時保管状態であれば、リソースの再登録を行うため属性の設定 (1504 から 1503) は行わずに、リソースの登録処理へ移行する。一時保管状態かどうかは、リソース管理テーブルのステータスの値が '2'、'3'、'4' の場合である。既に登録済みかどうかは、ステータスの値が '1' および '5' の場合で判断できる。リソース名称、バージョン、リビジョンを組み合わせた名称で同一名称の既に登録済みのリソースがある場合は再度設定し直す (1504、1503)。

40

【0043】

既に登録済みのリソースが存在しない場合は、アーカイブしたファイルを新たなリソースとしてリソースの属性情報と共にサーバ側に送信する (1505)。サーバ側はリソースとそのリソースの属性情報を受信し、リソースが既に登録されたもので一時保管状態かどうかを確認する (1510)。リソースが存在しない場合は、本リソースの登録結果によ

50

る削除ルールに抵触するリソースの削除を行う。この処理については、後述する図 18 で詳細に説明する。なお、ここで削除対象となるのは、削除ルール管理テーブルのうちリソースの登録に関連するルール 1105、1106、1107 および 1108 が適用される。削除したリソースがあれば、リソース登録処理側にそのリソース情報を渡す(1511)。

【0044】

処理 1510 で、リソースが存在する場合は、処理 1511 を行わない。リソース登録処理より登録要求のあったリソースとそのリソースの属性情報について、リソースはディスクに格納し、属性情報はリソース管理テーブルに設定する(1512)。このとき、当該リソースの削除方法が '1' で回復方法が '4' であれば予め定義済みのバックアップのパスにバックアップのリソースを作成する。そして、回復方法にファイルパス名をパラメータとして設定する。このバックアップのパス名は、本リソースが削除状態のステータスになった状態で回復が必要となった場合に参照する。

10

【0045】

以上の処理が終了したら登録を完了し再び登録要求の待ち状態となる(1508)。リソース登録システム側は、サーバ側で削除されたリソースがあればそれを表示し(1506、1507)、登録処理を終了する。

【0046】

図 16 にはリソースの配信処理を示す。図 6、図 7、図 8 で示すリソースの配信処理と異なる点は、リソースを配信する際に一時保管状態のステータスのリソースは新たに配信の対象にできないこと、リソースの配信を実行し中継システムからダウンロードが成功した報告があった場合、またはクライアントからインストールが完了した報告があった場合に、設定された削除ルールに抵触するリソースがあった場合にリソースの削除を行う点で、他は同様である。図 16 のステップ 1601 はリソースの配信指示を作成する部分であるが、配信する際に削除中のリソースは新たに配信の対象としないようにする。ここはリソースの配信処理である図 6 のステップ 601、図 7 のステップ 701、図 8 のステップ 801 にそれぞれ置き換えることができる。配信の対象かどうかは、リソース管理テーブルのステータスの値が '2' と '3' である場合に、そのリソースを配信対象としないようにすることで可能である('4' は対象とする)。

20

【0047】

図 16 のステップ 1602 は、図 6 のステップ 605、図 7 のステップ 705、図 8 のステップ 805 における配信結果情報の更新処理後に、配信結果による削除ルールに抵触するリソースの削除を行う。この処理については、後述する図 18 で詳細に説明する。なお、ここで削除対象となるのは、削除ルール管理テーブルに示されるルールのうち配信に関連するルール 1110 および 1111 が適用される。以上、図 16 は図 6、図 7、図 8 の配信処理にいずれにも共通的に置き換えることができ、説明を省略している部分は同様の処理となる。

30

【0048】

図 17 にはスケジュール処理を示す。まず、削除ルールに抵触するリソースの削除を行う(1701)。この処理については、後述する図 18 で詳細に説明する。なお、ここで削除対象となるのは、削除ルール管理テーブルに示されるルールのうち日時の超過に関連するルール 1109 および 1112 が適用される。次に、一時保管状態の確認を行う。具体的には、リソース管理テーブルの保管期限がスケジュール処理を実行した日時より過ぎているリソースの検索とそのリソースの削除を行う。リソースの実体の削除とリソース管理テーブルの当該リソースの情報を合わせて削除する。この際に残ったリソースのリソース管理テーブルの通番、および重要度をリナンバリングする。また、配信結果テーブルからも当該リソースの情報を削除する。さらに、リソース管理テーブルの削除方法が '1' で回復方法が '4' であればバックアップ用に作成したバックアップのリソースも削除する(1702)。この処理で削除したリソースについては図 22 のメッセージ出力処理を利用しメッセージを出力する。

40

50

【 0 0 4 9 】

次に、削除ルール管理テーブルのルール 1 1 1 3 である登録日から一定の期間に達したリソースかつ通常状態のステータスであるリソースについて検索を行い、リソース管理テーブルのステータスの値を ' 1 ' から ' 5 ' とし、削除できないようにする。さらに使用期限に 0 を設定する (1 7 0 3)。この処理で削除禁止にしたリソースについては図 2 2 のメッセージ出力処理を利用しメッセージを出力する。

【 0 0 5 0 】

次に、削除ルールの警告の値に従い、警告の閾値に達したリソースが存在したら、警告メッセージの処理を行う (1 7 0 4)。警告メッセージの処理は、図 2 2 で詳細に説明する。

10

【 0 0 5 1 】

以上の処理が終わったらウェイト状態となる。ウェイト間隔は 2 4 時間とし、図 1 3 には記載していないが、本スケジュール処理の開始時間はセットアップ処理で設定できるものとする。例えば、リソースの削除処理と配信処理の競合を防ぐことを目的とし、夜間に配信を実行する運用をとった場合は昼にスケジュール処理を動作するように設定できる (1 7 0 5 から 1 7 0 1)。

【 0 0 5 2 】

図 1 8 には削除ルール管理テーブルに基づいたリソースの削除処理を示す。まず、削除するリソースが存在するか削除ルール管理テーブルを参照し削除ルールに沿って、リソース管理テーブルの各属性値を参照し、ルールに抵触するリソースを検索する (1 8 0 1)。

20

【 0 0 5 3 】

削除ルール管理テーブルのルール 1 1 0 5 では、リソース管理テーブルに登録されるステータスが ' 1 ' の状態であるリソース数が、n 個に達しているときにリソースの削除を行う。削除するリソースは、削除対象の定義に従い選択される。例えば、' 1 ' の場合は、ステータスが ' 1 ' で登録日の一番古いリソースが削除の対象として選択される。さらに、' 4 ' の場合は、ステータスが ' 1 ' で重要度の値が一番大きいリソースが削除の対象として選択される。

【 0 0 5 4 】

削除ルール管理テーブルのルール 1 1 0 6 では、リソース管理テーブルに登録されたリソースのサイズの合計が、n MB に達しているときにリソースの削除を行う。削除するリソースは、削除対象の定義に従い選択される。例えば、' 2 ' の場合は、リソース管理テーブルの中の、ステータスが ' 1 ' でサイズの一番大きいリソースが削除の対象として選択される。さらに、' 3 ' の場合は、ステータスが ' 1 ' でサイズの一番小さいリソースが削除の対象として選択される。

30

【 0 0 5 5 】

削除ルール管理テーブルのルール 1 1 0 7 では、リソース登録システムにより登録されたリソースのバージョンを基にリソースの削除を行う。削除するリソースは、削除対象の定義に示される定義に従い選択される。例えば、' 5 ' 場合は、ステータスが ' 1 ' でリソース名称の同じリソースが削除の対象として選択される。さらに、' 6 ' の場合は、ステータスが ' 1 ' でリソース名称の同じかつ 1 つ下のバージョン以外のリソースが削除の対象として選択される。(1 つ下のバージョンとは、例えば登録したリソースのバージョンが 7 の場合必ず 6 のバージョンというわけではなく、7 未満のうち一番高いバージョンの意味である。)

40

削除ルール管理テーブルのルール 1 1 0 8 についてもルール 1 1 0 7 と同様であり、対象がリソース管理テーブルのリビジョンを対象として削除するリソースを選択する。削除ルール管理テーブルのルール 1 1 0 9 では、リソース管理テーブルの使用期限日が過ぎているリソースについて削除を行う。このルールは削除対象の定義がなく、ステータスが ' 1 ' でかつ使用期限日が過ぎているリソースが削除の対象として選択される。

【 0 0 5 6 】

削除ルール管理テーブルのルール 1 1 1 0 では、配信結果テーブルのダウンロード回数を

50

参照し、n回以上のダウンロードを行っているリソースについて削除を行う。このルールも削除対象の定義がなく、ステータスが'1'でかつn回以上のダウンロードが行われているリソースが削除の対象として選択される。

【0057】

削除ルール管理テーブルのルール1111も同様であり、これは配信結果テーブルのインストール回数を参照し、リソースを選択する。削除ルール管理テーブルのルール1112では、配信結果テーブルの最終アクセス日から現在の日付からn日(nヶ月、n年)経過しているリソースについて削除を行う。このルールは削除対象の定義がなく、ステータスが'1'でかつ最終アクセス日からn日が過ぎているリソースが削除の対象として選択される。

10

【0058】

リソース登録処理から削除処理が実行された場合は、処理の関連性からルール1105、1106、1107、1108を適用する。リソースの配信処理から削除処理が実行された場合は、ルール1110、1111を適用する。スケジュール処理から削除処理が実行された場合は、ルール1109、1112を適用する。

【0059】

削除するリソースがなければ、そのまま終了するが、削除するリソースが存在すれば、そのリソースのリソース管理テーブル上の削除方法を参照する。削除方法が'1'であれば(1802)、リソース管理テーブルのステータスを'1'から'2'に更新する(1806)。削除方法が'2'であれば(1803)、リソース管理テーブルのファイルパスに示すファイルを圧縮処理し、ステータスを'1'から'3'に更新する(1807)。削除方法が'3'であれば(1804)、リソース管理テーブルのファイルパスに示すファイルを削除し、ステータスを'1'から'4'に更新する(1807)。削除方法が'1'、'2'、'3'の値で、値に応じて削除処理を行った場合は、保管期間の設定を行う(1809)。

20

【0060】

保管期間には削除した日の1ヶ月後の日時を設定する。図13には記載していないが、1ヶ月の期間はセットアップ処理で別の期間に自由に設定できるものとする。長い期間を設定すれば、万が一配信要求があっても回復する可能性があるがある程度の容量を必要とする。短い期間を設定すれば、万が一配信要求があっても回復できなくなる。システムの性質や扱うリソースの性質により適度な値を設定すればよい。重要度には0を設定する。この処理で削除処理したリソースについては図22のメッセージ出力処理を利用しメッセージを出力する。

30

【0061】

削除方法が'4'であれば、リソース管理テーブルおよび実体のファイルを即時に削除する(1805)。この際に残ったリソースのリソース管理テーブルの通番、および重要度をリナンバリングする。リソース管理テーブルの削除方法が'1'で回復方法が'4'であればバックアップ用に作成したバックアップのリソースも削除する。また、配信結果テーブルからも削除するリソースの情報を削除する。この処理で削除したリソースについては図22のメッセージ出力処理を利用しメッセージを出力する。リソースの削除処理(1805、1809)について終了したら再度1801に戻り、削除するリソースがあるかどうかを検索する。削除するリソースが無くなるまで、繰り返し処理する。

40

【0062】

次に、リソースの回復機能について説明する。図19に、リソースの回復について示す。リソースの回復処理はリソースの配信処理である図6のステップ603、図7のステップ703、図8のステップ803の部分に回復のための処理を追加することで実現可能である。まず、下位システム(中継サーバ、クライアント)からリソース配信の要求があったときに、リソース管理テーブルを参照し、リソースが一時保管状態かどうかを判断する(1901)。一時保管状態かどうかは、リソース管理テーブルのステータスの値が'1'または'5'以外の状態になっていることで判断できる。当該リソースが一時保管状態で

50

ない場合は、そのままリソースの配信を実行する(1907)。

【0063】

当該リソースが一時保管状態の場合は、図20に示すリソースの回復処理を実行する(1902)。リソースの回復処理が正しく完了したかどうかを判定し(1903)、回復が行われたら、リソースの配信を行う(1907)。リソースの回復処理の詳細は図20と図21で説明する。

【0064】

回復が行われなかった場合は、下位システム(中継サーバ、クライアント)に配信処理の保留通知を行う(1904)。下位システムはこれにより、サーバからの指示があるまでは、下位システムからはアクションを行わない。そして、ステップ1902で行ったリソースの回復処理の続きを行う。回復方法によっては一回の回復処理で回復しないケースがあるため、ここで回復するまで回復処理を継続して行う(1905)。回復処理が正しく行われたら、下位システムに対して保留解除の再開通知を行う(1906)。その後、リソースの配信を行う(1907)。

10

【0065】

リソースの配信以降の処理は、リソースの配信処理である図6、図7、図8と同様であるため省略する。図19では省略しているが、回復中に他の中継サーバやクライアントから配信要求が来た場合を考慮し、ステップ1901から1907の処理については、リソース名称、バージョンおよびリビジョンの組み合わせた名称による排他制御を施すことにより、他の中継サーバやクライアントから配信要求はウェイト状態となるようにする。

20

【0066】

図20および図21には詳細な回復処理の説明を示す。回復対象となるリソースのリソース管理テーブルのステータスの値が'2'であれば(2001)、保管期限をさらに1ヶ月延長して(2005)、回復完了状態で元の処理に戻る。本ケースは、リソース管理テーブルのファイルパスで示されるファイルがそのまま残っているため、そのファイルをそのまま使用して配信処理を行う。回復対象となるリソースのリソース管理テーブルのステータスの値が'3'であれば(2002)、リソース管理テーブルのファイルパスで示されるファイルの圧縮を解凍し元のファイルの状態に戻す(2003)。そして、リソース管理テーブルのステータスを'2'のステータスに変更し(2004)、保管期限をさらに1ヶ月延長して(2005)、回復完了状態で元の処理に戻る。

30

【0067】

本ケースは、リソース管理テーブルのファイルパスで示されるファイルが圧縮解凍されるため元の配信可能な状態に戻り、その復元したファイルを使用して配信処理を行う。配信処理の要求が今後あることが考えられるため、以降は圧縮状態のステータスには戻さず保管期限まで一時保管する。ステップ2005で示す1ヶ月の期限については、図18での説明と同様にセットアップ処理で変更できるものとする。回復対象となるリソースのステータスの値が'2'と'3'以外であれば'4'であると見なし、リソース管理テーブルの回復方法を参照する。回復方法が'6'であれば(2006)、当該リソースの最上位のリビジョンのリソースをリソース管理テーブルから検索し、そのリソースを配信リソースとするため配信指示情報のリソース名称を最上位のリビジョンのリソースに変更する(2007)。

40

【0068】

そして、ステップ2005の処理へ移行し当該リソースの保管期限を一ヶ月間延長する。リソースの配信が正しく行われたあとに、最上位のリビジョンのリソース情報で配信結果テーブルを更新する。回復方法が'3'であれば(図21の2101)、リソースを作成したインストールメディアからファイルを回復するため、設定したドライブのマウント先にアクセスを行いファイルが存在するか確認する(2106)。なお、設定したパラメタのドライブにあらかじめメディアをマウントしておけば、スムーズに回復を行うことができる。

【0069】

50

回復方法が ' 2 ' であれば (2 1 0 2)、リソースの作者宛てもしくは企業宛てにリソースを送信してもらう旨の内容でメールを作成し設定したメールアドレスにメールを送信する (2 1 0 7)。返信メールにはリソースを添付してもらい自動的にリソースファイルにコピーするようにメールソフトを設定する。

【 0 0 7 0 】

回復方法が ' 5 ' であれば (2 1 0 3)、リソースが配信される過程で保管される中継サーバのキャッシュからファイルを吸い上げてリソースを回復するため、設定した中継サーバのアドレスに対して接続しリソースを吸い上げる (2 1 0 8)。設定する中継サーバのアドレスは、キャッシュがされている可能性が高い中継サーバのアドレスを設定しておくことで回復する可能性が高くなる。また、設定した中継サーバにリソースが存在しなかった場合は、設定しなかった中継サーバにも接続し回復を行う。

10

【 0 0 7 1 】

回復方法が ' 1 ' であれば (2 1 0 4)、設定した URL に向けてインターネット経由でリソースをダウンロードし回復するため、設定した URL に接続を行いリソースをダウンロードする (2 1 0 9)。

【 0 0 7 2 】

回復方法が ' 1 '、' 2 '、' 3 '、' 5 ' において、回復できたかどうかを判断し (2 1 1 0)、回復できたら、リソース管理テーブルのファイルパスの示すパスにファイルコピーする (2 1 1 1)。インストールメディアから回復する場合などのように必要であれば、リソースの登録処理のようにアーカイブを行い 1 つのファイルにしたうえでファイルコピーをする。メールの送信のように、即回復ができなかった場合は、回復未完状態で元の処理に戻る。

20

【 0 0 7 3 】

回復方法が ' 1 '、' 2 '、' 3 '、' 5 '、' 6 ' 以外であれば、' 4 ' とみなし、バックアップのパス名に示すパスよりファイルコピーを行う (2 1 0 5)。ファイルコピーが完了したら、リソース管理テーブルのステータスを ' 1 ' の通常状態にする。さらに、使用期限日を回復を実行した時点の日付に 1 ヶ月足した値とする。また、保管期限日はすべて 0 の値にする。重要度は最下位の値とする。その他の属性については、リソースの登録時点の値とする (2 1 1 2)。そのあと、回復完了状態で元の処理に戻る。

【 0 0 7 4 】

図 2 2 には、メッセージ出力の方法を示す。まず、メッセージ出力要求が、警告処理から行われたものか、削除処理から行われたものかを判定する (2 2 0 1)。警告処理から行われた場合は、削除ルール管理テーブルのルールと警告で指定された警告の閾値に該当するリソースをリソース管理テーブルを基に検索する (2 2 0 3)。該当するリソースがあれば、リソースおよび警告内容を、メッセージ出力バッファに追記形式で格納し (2 2 0 4)、他に警告対象となるリソースが無いか検索する (2 2 0 3 に戻る)。すべてのリソースについての検索が終わったら、検索処理を終えループを抜ける。削除処理から行われた場合は、削除処理で削除したリソース名称、バージョン、リビジョンをメッセージ出力バッファに格納する (2 2 0 2)。

30

【 0 0 7 5 】

メッセージ出力バッファにメッセージが格納されているかどうかを確認する (2 2 0 5)。何もメッセージがなければそのまま終了する。何らかのメッセージが格納されているならば、メッセージ出力先を選択する。メッセージ出力であれば (2 2 0 6)、コンソール装置にメッセージ出力バッファに格納された内容を出力する (2 2 0 7)。ログ出力であれば (2 2 0 8)、メッセージ出力バッファに格納された内容をログ出力専用のファイルに出力する (2 2 0 9)。メール送信であれば (2 2 1 0)、メッセージ出力バッファに格納された内容をメールに書きサーバのユーザ宛てに送信する (2 2 1 1)。なお、メッセージの出力先の定義について本説明では省略するが、セットアップ処理であらかじめ定義できているとする。以上、説明した機能がそれぞれ動作することによりリソースの削除と回復が行われるようになる。

40

50

【0076】

本発明では、ステータスが一時保管状態のリソースについて再登録した場合や回復処理を施した場合に通常状態のステータスにしているが、このとき削除ルールを適用していない。ここは、再登録や回復処理の後に図18の削除処理を実行することで、削除ルールを適用できる。また、本発明では、図20のステップ2005で一時保管状態を示すリソース管理テーブルのステータスの‘2’と‘3’については保管期限を延長する処理としたが、‘4’のステータスのように再度登録し通常のリソースの状態にすることで再登録の手間を省くことができる。図20のステップ2005の部分を図21のステップ2112のように再登録する処理にすることで可能である。属性の再設定値についても‘4’のケースと同様である。ただし、回復方法を‘6’の最上位のリビジョンを適用する方法については、その性質上再登録処理の対象外とする。

10

【0077】

また、本発明では、図9のリソースの回復方法913を1つ設定するようにした。ここに2つ以上の回復の方法と回復方法の優先順位を設定することで、より回復を効果的に行うことができるようになる。図20と図21に示す回復処理で処理2006～2110を、設定された回復方法913の設定数だけ優先順位に従いループ処理することで可能である。

【0078】

また、本発明では、削除ルール管理テーブルのルールのうち新たなバージョンやリビジョンのリソースを登録した場合(図11のルール1107や1108)やインストール結果の情報の更新の場合(図11の1111)が発生したときに削除するルールについて共通ルールの扱いとしたが、バージョンアップの頻度が高いリソースなどの特質をもつリソースや全クライアントではなく一部のクライアントにだけ配信するリソースなどの特質を考えた場合にリソース毎にルールの設定を行いたい場合もある。そこで、使用期限日などの属性と同様にリソース管理テーブル側にローカルの位置付けで削除のルールの情報を保持するようにし、リソースの登録処理で削除ルールを設定できるようにする(図15のステップ1503に追加)。リソースの削除処理にリソース毎のルールに抵触するかを判断する処理を追加することで可能である(図18のステップ1801に追加)。

20

【0079】

【発明の効果】

本発明によるリソースの自動削除方法を採用することにより、誤ってリソースを削除してしまう可能性が大幅に低下するとともに、ルールに従って自動的に削除することにより不必要にディスク容量を圧迫することを防ぎ安定した運用を行うことが可能となった。また、リソースの個数を自動的に制限できるようにすることで管理の手間もかからないようになった。また、配信結果によりリソースを削除できるようにすることで、ライセンス数のコントロールも可能となった。その一方で、ただ単に削除するだけではなく回復の手段も各種用意することでルールの設定ミスなどにより誤って削除した場合も容易に回復でき、さらに希少価値の高いリソースを削除できないようにする手段も用意することで再登録の必要が無くなったことにより、操作性および運用性が向上する。

30

【図面の簡単な説明】

【図1】本発明のシステムの構成を示す図である。

【図2】本発明のプログラムモジュールの構成を示す図である。

【図3】リソース管理テーブルの一例を示す図である。

【図4】本発明で利用する配信結果テーブルの一例を示す図である。

【図5】リソースの登録処理を示すフローチャートである。

【図6】サーバ側主導によるリソースの配信処理を示すフローチャートである。

【図7】クライアント側主導によるリソースの配信処理を示すフローチャートである。

【図8】中継サーバへのリソースの配信処理を示すフローチャートである。

【図9】本発明で利用するリソース管理テーブルの一例を示す図である。

【図10】リソース管理テーブルで使用する定義一覧を示す図である。

40

50

- 【図 1 1】本発明で利用する削除ルール管理テーブルの一例を示す図である。
- 【図 1 2】削除ルール管理テーブルで使用する定義一覧を示す図である。
- 【図 1 3】セットアップ処理を示すフローチャートである。
- 【図 1 4】設定画面の例を示す図である。
- 【図 1 5】リソースの自動削除処理を含めたリソースの登録処理を示すフローチャートである。
- 【図 1 6】リソースの自動削除処理を含めたリソースの配信処理を示すフローチャートである。
- 【図 1 7】スケジュール処理を示すフローチャートである。
- 【図 1 8】リソースの削除処理を示すフローチャートである。
- 【図 1 9】リソースの自動回復処理を含めたリソースの配信処理を示すフローチャートである。
- 【図 2 0】リソースの回復処理を示すフローチャートである。
- 【図 2 1】リソースの回復処理を示すフローチャートである。
- 【図 2 2】警告メッセージ出力処理を示すフローチャートである。
- 【図 2 3】本発明によるリソースの扱いを説明する図である。
- 【図 2 4】本発明で利用する削除ルール管理テーブルの一例を示す図である。

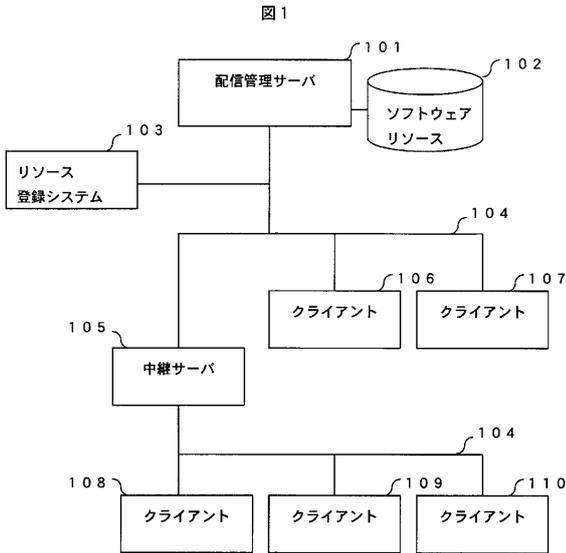
10

【符号の説明】

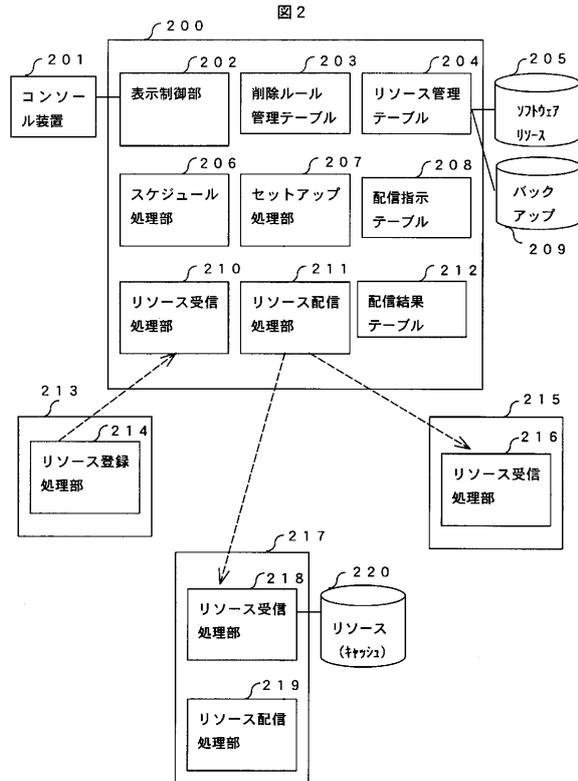
1 0 1 : 配信管理サーバ、1 0 2 : ソフトウェアリソース、1 0 3 : リソース登録システム、1 0 4 : ネットワーク、1 0 5 : 中継サーバ、1 0 6 : クライアント、1 0 7 : クライアント、1 0 8 : クライアント、1 0 9 : クライアント、1 1 0 : クライアント、2 0 0 : 配信管理サーバ、2 0 1 : コンソール装置、
2 0 2 : 表示制御部、2 0 3 : 削除ルール管理テーブル、2 0 4 : リソース管理テーブル、2 0 5 : ソフトウェアリソース、2 0 6 : スケジュール処理部、
2 0 7 : セットアップ処理部、2 0 8 : 配信指示テーブル、2 0 9 : バックアップ、2 1 0 : リソース受信処理部、2 1 1 : リソース配信処理部、2 1 2 : 配信結果テーブル、2 1 3 : リソース登録システム、2 1 4 : リソース登録処理部、2 1 5 : クライアント、2 1 6 : リソース受信処理部、2 1 7 : 中継サーバ、2 1 8 : リソース受信処理部、2 1 9 : リソース配信処理部、2 2 0 : リソース(キャッシュ)

20

【 図 1 】



【 図 2 】

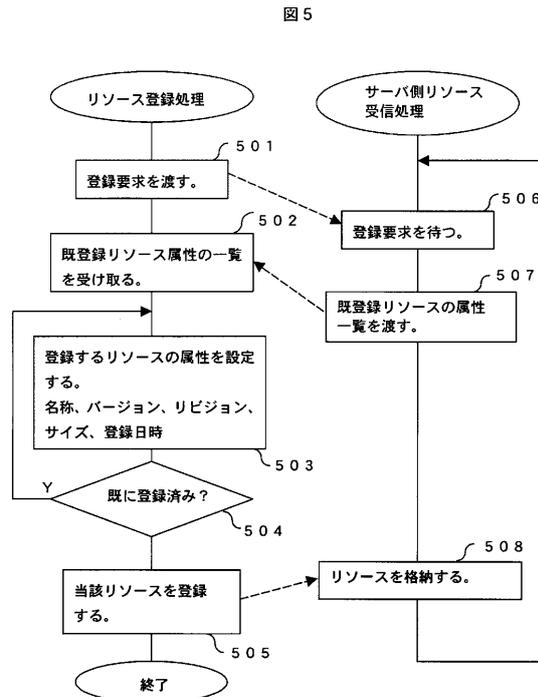


【 図 3 】

図3

通番	リソース名称	バージョン	リビジョン	サイズ	登録日	ファイルパス
1	A	7	6	300	YY/MM/DD	/a76
2	B	1	1	500	YY/MM/DD	/b11
3	C	0	9	100	YY/MM/DD	/c09
4	C	1	0	150	YY/MM/DD	/c10
5	C	1	1	120	YY/MM/DD	/c11

【 図 5 】



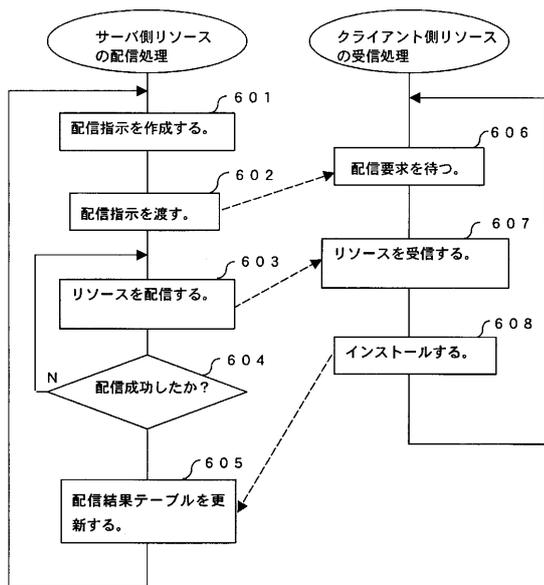
【 図 4 】

図4

通番	インストール回数	ダウンロード回数	最終アクセス日
1	20	10	YY/MM/DD
2	50	5	YY/MM/DD
4	5	0	YY/MM/DD
5	50	5	YY/MM/DD

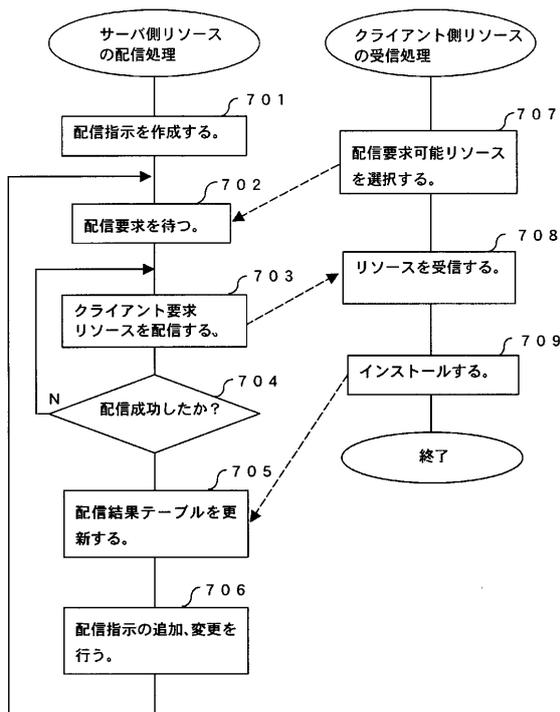
【 図 6 】

図 6



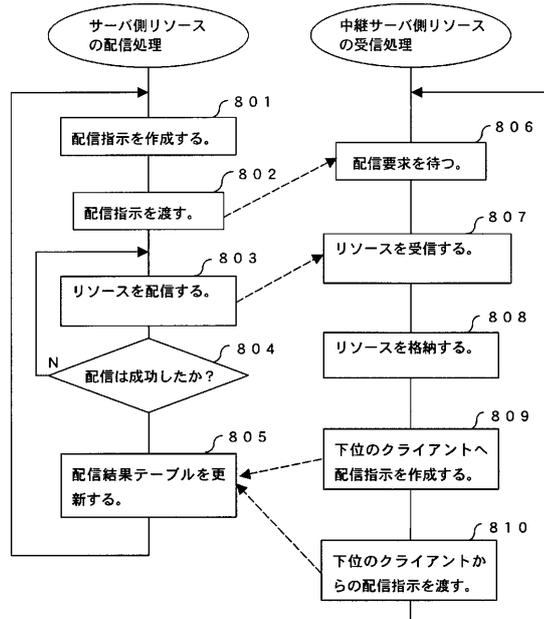
【 図 7 】

図 7



【 図 8 】

図 8



【 図 9 】

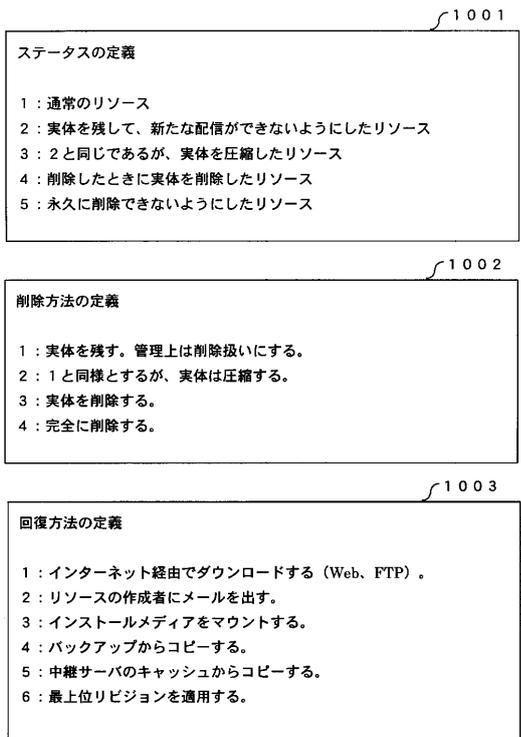
図 9

通番	リソース名称	バージョン	リビジョン	サイズ	登録日	使用期限日	保管期限日	重要度	ステータス	ファイルパス	削除方法	回復方法
1	A	7	6	300	YY/MM/DD	YY/MM/DD	YY/MM/DD	2	1	/a76	1	
2	B	1	1	500	YY/MM/DD	YY/MM/DD	YY/MM/DD	1	1	/b11	3	...
3	C	0	9	100	YY/MM/DD	YY/MM/DD	YY/MM/DD	3	2	/c09	3	

4:/backup/c10
914

【図 10】

図 10



【図 11】

図 11

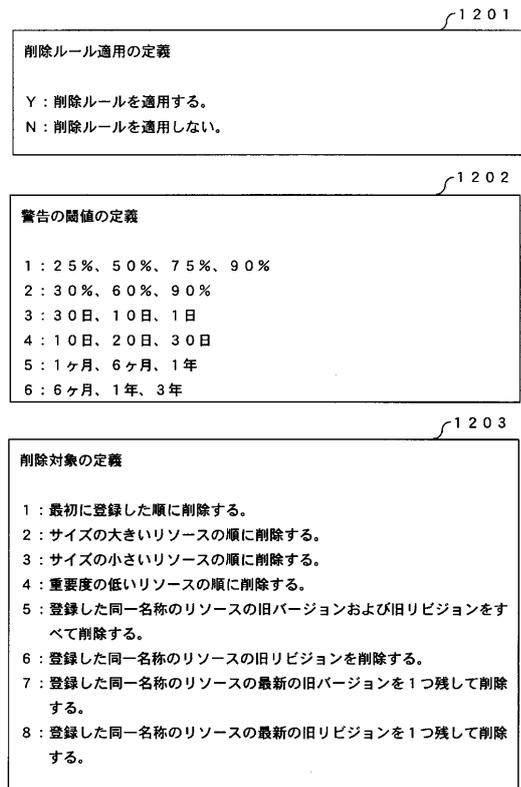
	1101	1102	1103	1104
ルールの内容	適用	警告	削除対象	
1105 最大登録数 n 個を超えたときに削除する。	Y, N	1, 2	1, 4	
1106 最大管理サイズ n MB を超えた場合に削除する。	Y, N	1, 2	1, 2, 3, 4	
1107 新バージョンを登録したときに削除する。	Y, N	/	5, 7	
1108 新リビジョンを登録したときに削除する。	Y, N	/	6, 8	
1109 使用期限に達したリソースを削除する。	Y, N	3	/	
1110 n 回ダウンロードしたら削除する。	Y, N	1, 2	/	
1111 n 回のインストール報告を受けたら削除する。	Y, N	1, 2	/	
1112 最終アクセス日時から n 日間更新がないときに削除する。	Y, N	4, 5, 6	/	
1113 登録日より n 年が経過したリソースは削除しない。	Y, N	5, 6	/	

1114

1115

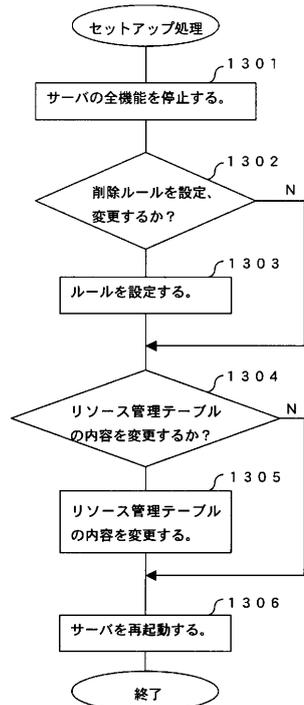
【図 12】

図 12



【図 13】

図 13



【 図 1 4 】

図 1 4

1401

リソース削除のルールを設定を行います。(2/9)

最大管理サイズに達した場合に登録済みのリソースを削除する。

最大容量 MB

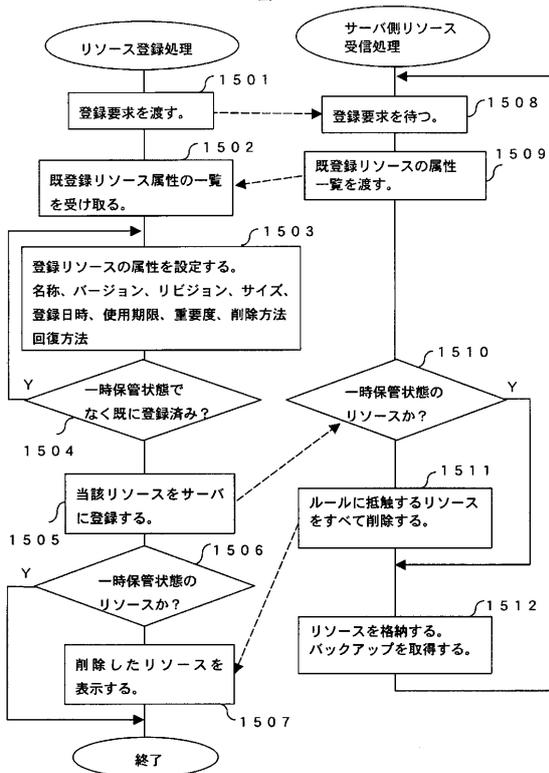
- 最初に登録したリソースの順に削除する。
- サイズの大きいリソースの順に削除する。
- サイズの小さいリソースの順に削除する。
- 重要度の低いリソースの順に削除する。

警告メッセージ

↓

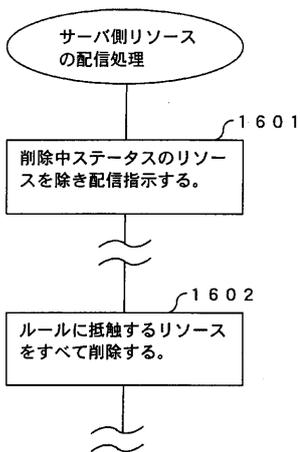
【 図 1 5 】

図 1 5



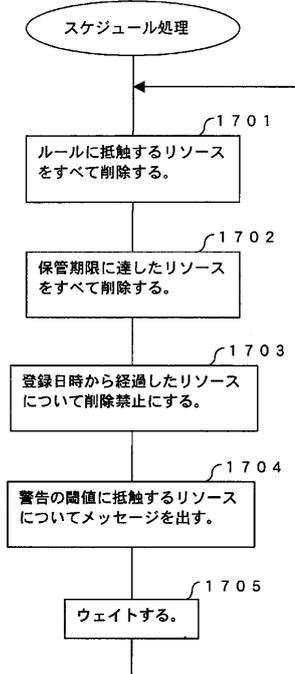
【 図 1 6 】

図 1 6



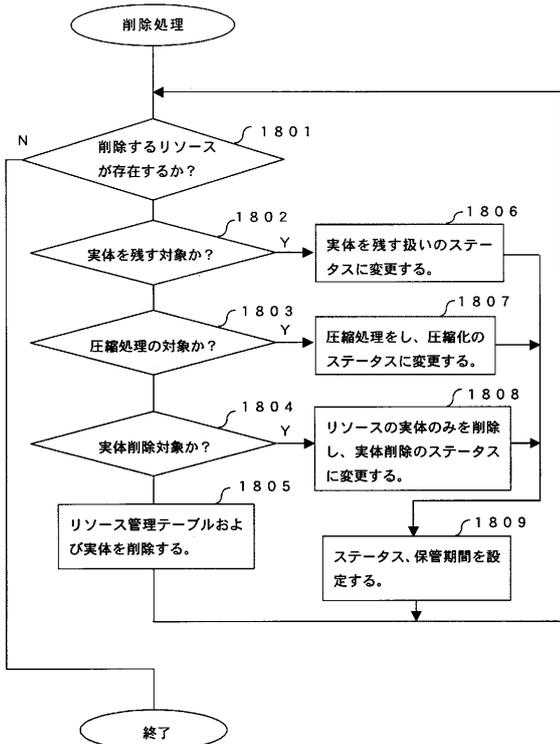
【 図 1 7 】

図 1 7



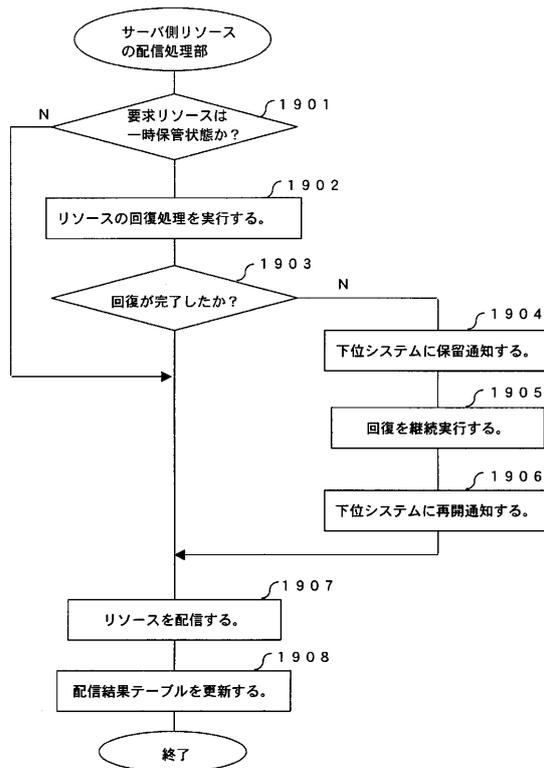
【図18】

図18



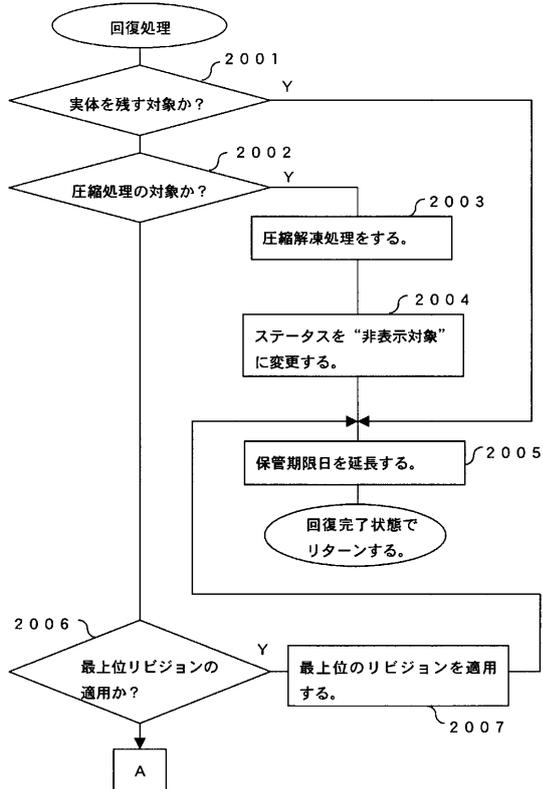
【図19】

図19



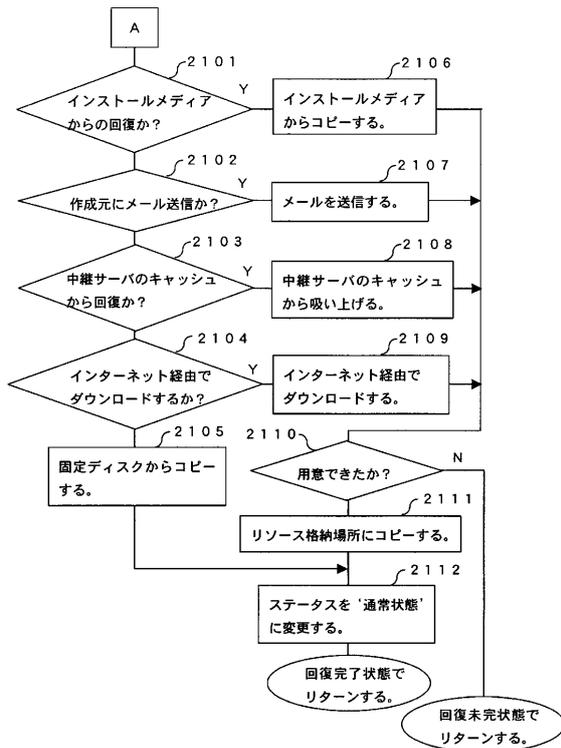
【図20】

図20



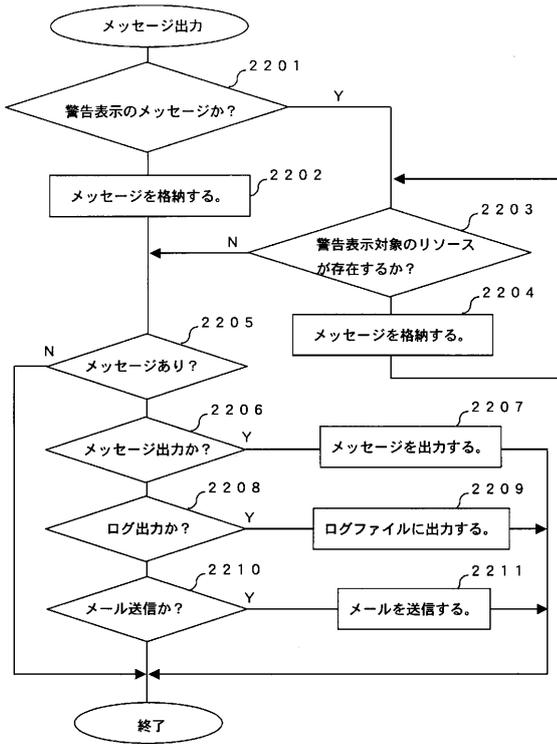
【図21】

図21



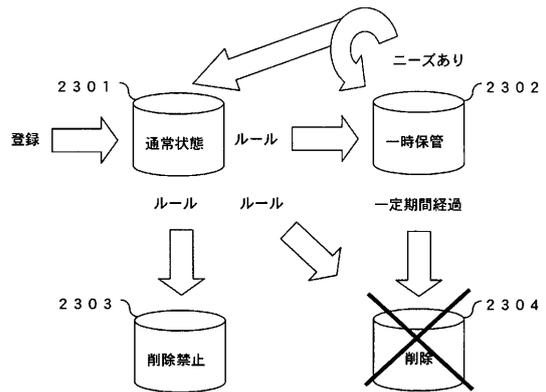
【 図 2 2 】

図 2 2



【 図 2 3 】

図 2 3



【 図 2 4 】

図 2 4

	ルールの内容	適用	警告	削除対象
2 4 0 1	最大登録数 n 個を超えたときに削除する。	Y	1	4
2 4 0 2	最大管理サイズ n MB を超えた場合に削除する。	N	—	—
2 4 0 3	新バージョンを登録したときに削除する。	Y	/	5
2 4 0 4	新リビジョンを登録したときに削除する。	Y	/	8
2 4 0 5	使用期限に達したリソースを削除する。	N	—	/
2 4 0 6	n 回ダウンロードしたら削除する。	N	—	/
2 4 0 7	n 回のインストール報告を受けたら削除する。	Y	1	/
2 4 0 8	最終アクセス日時から n 日間更新がないときに削除する。	Y	4	/
2 4 0 9	登録日より n 年が経過したリソースは削除しない。	Y	6	/