

(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2012-247885

(P2012-247885A)

(43) 公開日 平成24年12月13日(2012.12.13)

(51) Int.Cl.

G06F 9/48 (2006.01)

F I

G06F 9/46 452A

テーマコード (参考)

審査請求 未請求 請求項の数 7 O L (全 15 頁)

(21) 出願番号 特願2011-117561 (P2011-117561)
 (22) 出願日 平成23年5月26日 (2011. 5. 26)

(71) 出願人 000233491
 株式会社日立システムズ
 東京都品川区大崎一丁目2番1号
 (74) 代理人 100080001
 弁理士 筒井 大和
 (74) 代理人 100113642
 弁理士 菅田 篤志
 (74) 代理人 100117008
 弁理士 筒井 章子
 (74) 代理人 100147430
 弁理士 坂次 哲也
 (74) 代理人 100147050
 弁理士 中原 亨

最終頁に続く

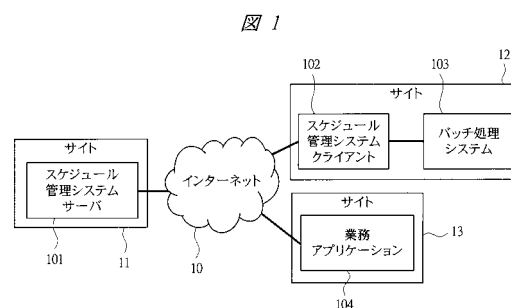
(54) 【発明の名称】 スケジュール管理システム、スケジュール管理システムの制御方法及びプログラム

(57) 【要約】

【課題】スケジュール管理システムクライアントが管理するバッチが実行可能かスケジュール管理システムサーバに問い合わせることで、インターネット上にポートを公開する必要なくセキュアな環境でバッチ処理システムを稼働させる手段を提供する。

【解決手段】スケジュール管理システムサーバ101に対して所定のネットワークプロトコルポートのみを開放する。このネットワークプロトコルポートを介して、該当するカレンダー情報にアクセスを行うことで不要なポートを開放することを避ける。

【選択図】図1



【特許請求の範囲】**【請求項 1】**

スケジュール管理システムサーバが稼働する第 1 のサイトと、スケジュール管理システムクライアントとバッチ処理システムが稼働する第 2 のサイトと、を含み、前記第 1 のサイトと前記第 2 のサイトがインターネットを介して接続されるスケジュール管理システムであって、

業務アプリケーションが動作する該スケジュール管理システム外のサイトと接続する論理的なインターフェースを前記第 1 のサイトは有することを特徴とするスケジュール管理システム。

【請求項 2】

10

請求項 1 記載のスケジュール管理システムにおいて、

前記スケジュール管理システムサーバは、バッチ処理の開始処理を行う前記スケジュール管理システムクライアントから出力される呼び出しでバッチ処理可否判定処理を開始し、

前記バッチ処理可否判定処理の結果を、前記スケジュール管理システムクライアントに戻すことを特徴とするスケジュール管理システム。

【請求項 3】

請求項 2 記載のスケジュール管理システムにおいて、

前記バッチ処理可否判定処理に際し、前記スケジュール管理システムサーバは自身が有する 1 又は 2 以上のリポジトリからスケジュールの基礎情報を用いることを特徴とするスケジュール管理システム。

20

【請求項 4】

請求項 3 記載のスケジュール管理システムにおいて、

前記バッチ処理可否判定処理を受信した前記スケジュール管理システムクライアントは所定の場合にバッチ処理を実行することを特徴とするスケジュール管理システム。

【請求項 5】

スケジュール管理システムサーバが稼働する第 1 のサイトと、スケジュール管理システムクライアントとバッチ処理システムが稼働する第 2 のサイトと、を含み、前記第 1 のサイトと前記第 2 のサイトがインターネットを介して接続されるスケジュール管理システムの制御方法であって、

30

前記スケジュール管理システムクライアントが前記スケジュール管理システムサーバに処理を行うべきバッチの有無の確認を要求するバッチ処理確認ステップと、

前記バッチ処理確認ステップでの要求を受けて、前記スケジュール管理システムサーバが処理を行うべきバッチの有無を確認し、前記スケジュール管理システムクライアントに前記確認の結果を送信するバッチ処理確認ステップと、を有することを特徴とするスケジュール管理システムの制御方法。

【請求項 6】

スケジュール管理システムサーバ上で実行の判定を行うプログラムであって、

該プログラムは、前記スケジュール管理システムサーバとコンピュータネットワークで接続されるスケジュール管理システムクライアント上で動作するバッチ実行モジュールから、企業 ID とバッチ ID を取得し、バッチ処理の要否を判定し、前記コンピュータネットワークを介して前記バッチ実行モジュールに前記判定の結果を返送することを特徴とするプログラム。

40

【請求項 7】

請求項 6 記載のプログラムにおいて、

前記判定の結果が肯定的なものである場合には、前記スケジュール管理システムサーバ内のバッチ実行スケジュールリポジトリの最終実施日を更新することを特徴とするプログラム。

【発明の詳細な説明】**【技術分野】**

50

【 0 0 0 1 】

本発明は、企業の営業日や休業日を定めたカレンダー情報を集中管理する方法、及び、システムのバッチ処理実行スケジュールを集中管理する方法と、複数の企業カレンダー情報に則ってバッチ処理を実行する機能と、企業カレンダー情報を他アプリケーションに提供する機能に関する。

【 背景技術 】**【 0 0 0 2 】**

利用者がインターネット上にあるアプリケーションやハードウェアリソースを任意に選択し、利用するクラウドコンピューティングが普及している。これに伴い、企業システムの業務システムがインターネット上のアプリケーションに置き換わる、あるいは企業システム自体がインターネット上のハードウェアリソース上で稼働するようになっている。

10

【 0 0 0 3 】

これにより、アプリケーションやハードウェアのメンテナンス等、システムの運用コストの削減や、システム障害による業務停止のリスクを避ける等の効果が期待できる。

【 0 0 0 4 】

このような状況を背景に、定期的に大量データを編集するようなバッチ処理を実行するシステムと、それらの実行スケジュールを管理しバッチ処理を起動させる自動起動システムをインターネット上のハードウェアリソースを利用して稼働させることが想定されている。

【 0 0 0 5 】

特開平 9 - 6 4 5 7 号公報（特許文献 1）は、特定の日付、時刻、曜日・時刻でプログラムを自動起動することにより定型業務の自動実行を可能とする、プログラム自動制御装置を開示する。

20

【 0 0 0 6 】

特開平 1 0 - 6 9 5 7 8 号公報（特許文献 2）には、本日実行すべきデータ処理業務を確認する作業に連動して、この確認作業によって本日実行すべき業務である認識したデータ処理業務の起動を指令できるようにする旨が開示されている。

【 先行技術文献 】**【 特許文献 】****【 0 0 0 7 】**

【特許文献 1】特開平 9 - 6 4 5 7 号公報

【特許文献 2】特開平 1 0 - 6 9 5 7 8 号公報

30

【 発明の概要 】**【 発明が解決しようとする課題 】****【 0 0 0 8 】**

しかし、クラウドコンピューティングにおいては、バッチシステムとスケジュール管理システム間での通信にインターネットを利用することとなり、どちらか一方に公開用のポートを開ける必要がある。このため、セキュリティリスクを増大させるといった問題が発生する。

【 0 0 0 9 】

このようなシステムをクラウドコンピューティング環境で利用する場合、システム間情報連携を実現するセキュアなシステム構成の検討が必要となる。

40

【 0 0 1 0 】

本発明の目的は、スケジュール管理システムクライアントが管理するバッチが実行可能かスケジュール管理システムサーバに問い合わせることで、インターネット上にポートを公開する必要なくセキュアな環境でバッチ処理システムを稼働させる手段を提供することにある。

【 0 0 1 1 】

本発明の前記並びにその他の目的と新規な特徴は、本明細書の記述及び添付図面から明らかになるであろう。

50

【課題を解決するための手段】**【0012】**

本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、次の通りである。

【0013】

本発明の代表的な実施の形態に関わるスケジュール管理システムは、スケジュール管理システムサーバが稼働する第1のサイトと、スケジュール管理システムクライアントとバッチ処理システムが稼働する第2のサイトと、を含み、これらの第1のサイトと第2のサイトがインターネットを介して接続され、業務アプリケーションが動作する該スケジュール管理システム外のサイトと接続する論理的なインターフェースを第1のサイトは有することを特徴とする。

10

【0014】

このスケジュール管理システムにおいて、スケジュール管理システムサーバは、バッチ処理の開始処理を行うスケジュール管理システムクライアントから出力される呼び出しでバッチ処理可否判定処理を開始し、バッチ処理可否判定処理の結果を、スケジュール管理システムクライアントに戻すことを特徴としても良い。

【0015】

このスケジュール管理システムにおいて、バッチ処理可否判定処理に際し、スケジュール管理システムサーバは自身が有する1又は2以上のリポジトリからスケジュールの基礎情報を用いることを特徴としても良い。

20

【0016】

このスケジュール管理システムにおいて、バッチ処理可否判定処理を受信したスケジュール管理システムクライアントは所定の場合にバッチ処理を実行することを特徴としても良い。

【0017】

本発明の代表的な実施の形態に関わるスケジュール管理システムの制御方法は、スケジュール管理システムサーバが稼働する第1のサイトと、スケジュール管理システムクライアントとバッチ処理システムが稼働する第2のサイトと、を含み、第1のサイトと第2のサイトがインターネットを介して接続され、スケジュール管理システムクライアントがスケジュール管理システムサーバに処理を行うべきバッチの有無の確認を要求するバッチ処理確認ステップと、バッチ処理確認ステップでの要求を受けて、スケジュール管理システムサーバが処理を行うべきバッチの有無を確認し、スケジュール管理システムクライアントに確認の結果を送信するバッチ処理確認ステップと、を有することを特徴とする。

30

【0018】

本発明の代表的な実施の形態に関わるプログラムは、スケジュール管理システムサーバ上で動作するものであって、スケジュール管理システムサーバとコンピュータネットワークで接続されるスケジュール管理システムクライアント上で動作するバッチ実行モジュールから、企業IDとバッチIDを取得し、バッチ処理の要否を判定し、コンピュータネットワークを介してバッチ実行モジュールに判定の結果を返送することを特徴とする。

【0019】

このプログラムにおいて、判定の結果が肯定的なものである場合には、スケジュール管理システムサーバ内のバッチ実行スケジュールリポジトリの最終実施日を更新することを特徴としても良い。

40

【発明の効果】**【0020】**

スケジュール管理システムサーバがスケジュール管理システムクライアントに実行命令を出すのではなく、スケジュール管理システムクライアントが管理しているバッチが実行可能か否かをスケジュール管理システムサーバに問い合わせることで、バッチ処理システム環境でインターネット上にポートを公開する必要はなく、セキュアな環境でバッチ処理システムの稼働が可能となる。

50

【 0 0 2 1 】

また、インターネット上の様々なアプリケーションは、スケジュール管理システムサーバで管理された情報を元に、各企業に適したユーザインターフェース等のサービスを提供可能となる。さらに、企業独自のカレンダー情報が手に入ることから、「 3 月第 2 営業日」等の抽象表現での日付指定が業務アプリケーション上で可能となる。

【 図面の簡単な説明 】

【 0 0 2 2 】

【 図 1 】 本発明を実現するスケジュール管理システムの全体概要図である。

【 図 2 】 図 1 で説明した各サイトの開放ポート一覧に関する図である。

【 図 3 】 スケジュール管理システムサーバの構成図である。

10

【 図 4 】 スケジュール管理システムクライアントの構成図である。

【 図 5 】 スケジュール管理システムサーバで公開される Web サービスの一覧に関する図である。

【 図 6 】 定休日情報リポジトリの一例を示す図である。

【 図 7 】 例外休業日情報リポジトリの一例を示す図である。

【 図 8 】 バッチ実行スケジュールリポジトリの一例を表す図である。

【 図 9 】 本発明におけるバッチ実行モジュールの動作を示すフローチャートである。

【 図 1 0 】 本発明における実行判定モジュールの動作を示すフローチャートである。

【 図 1 1 】 本発明におけるカレンダー情報提供モジュールの動作を示すフローチャートである。

20

【 図 1 2 】 本発明における日付提供モジュールの動作を示すフローチャートである。

【 発明を実施するための形態 】

【 0 0 2 3 】

以下の実施の形態においては、便宜上その必要があるときは、複数のセクションまたは実施の形態に分割して説明する。しかし、特に明示した場合を除き、それは互いに無関係なものではなく、一方は他方の一部又は全部の変形例、詳細、補足説明などの関係にある。また、以下の実施の形態において、要素の数など（個数、数値、量、範囲などを含む）に言及する場合、特に明示した場合及び原理的に明らかに特定の数に限定される場合などを除き、その特定の数に限定されるものでなく、特定の数以上でも以下でも良い。

【 0 0 2 4 】

30

さらに、以下の実施の形態において、その構成要素は、特に明示した場合及び原理的に明らかに必須であると考えられる場合を除き、必ずしも必須のものではないことは言うまでもない。

【 0 0 2 5 】

以下、図を用いて本発明の実施の形態を説明する。

【 0 0 2 6 】

（実施の形態）

本発明に関わるスケジュール管理システムの概略動作は以下のようになる。

【 0 0 2 7 】

スケジュール管理システムクライアントは、スケジュール管理システムサーバに対して一定間隔でバッチ処理の実行の可否について問い合わせを行う。問い合わせを受けたスケジュール管理システムサーバは、自身が管理するバッチ実行スケジュールと企業のカレンダー情報を参照し、処理を行うべきバッチがあるかを確認する。

40

【 0 0 2 8 】

処理を行うバッチの有無の判定結果は、スケジュール管理システムサーバからスケジュール管理システムクライアントに伝達される。処理を要するバッチがあった場合には、スケジュール管理システムクライアントは指定されたバッチ処理を実行する。

【 0 0 2 9 】

次にインターネット上の業務アプリケーションが企業独自のカレンダー情報を取り込む場合について説明する。

50

【 0 0 3 0 】

またインターネット上にある業務アプリケーションは、複数の企業が汎用化された同じ業務アプリケーションを利用し、迅速かつ安価にアプリケーションを導入できる。一方で、インターネット上にある業務アプリケーションでは個別の企業の営業日や休業日を登録することはできず、画面情報にも反映されないため、利用者にとっては不便なユーザインターフェースとなっている。

【 0 0 3 1 】

これに対し、本発明では、業務アプリケーションは必要に応じて、特定企業の特定の期間についてスケジュール管理システムサーバに問い合わせる。問い合わせを受けたスケジュール管理システムサーバは自身が管理するカレンダー情報を参照し、指定された期間の営業日、休日といったカレンダー情報をアプリケーションに返す。

10

【 0 0 3 2 】

アプリケーションでは受け取ったカレンダー情報を元に画面の表示などを行う。

【 0 0 3 3 】

図 1 は、本発明を実現するスケジュール管理システムの全体概要図である。

【 0 0 3 4 】

本発明に関わるシステムは、インターネット 1 0 によって相互に接続される、スケジュール管理システムサーバ 1 0 1 が稼働するサイト 1 1 と、スケジュール管理システムクライアント 1 0 2 とバッチ処理システム 1 0 3 が稼働するサイト 1 2、インターネット上で公開される業務アプリケーション 1 0 4 が稼働するサイト 1 3 により構成される。

20

【 0 0 3 5 】

インターネット 1 0 とは、インターネットプロトコル技術を用いて相互接続されたコンピュータネットワークである。

【 0 0 3 6 】

従来、企業が持つ複数のバッチ処理システムは社内 LAN (Local Area Network) など、外部と隔離された環境に配置される。これらの複数のバッチ処理システムはスケジュール管理システムからの起動命令で各バッチ処理が稼働する。インターネット 1 0 を経由する場合、スケジュール管理システムがバッチ処理システムに命令を出すために必要な TCP 等のトランスポート層のプロトコルで用いるネットワークポートをインターネット上に公開する必要がある。これはバッチ処理システムが配置された全環境で必要となる為、バッチ処理システムが多いほどセキュリティのリスクを増大させることとなっていた。

30

【 0 0 3 7 】

また特に手段を講じなければ、End To End の経路は、保証されるものではなく、経路上のデータの安全性は保証されない。さらには、不足の第三者がインターネット 1 0 経由でサイト 1 1、1 2、1 3 に不正にアクセスを試みることも可能であるため、各サイトは一定の安全措置を採る必要がある。

【 0 0 3 8 】

本発明は、これらのリスクを解消するための物である。

【 0 0 3 9 】

スケジュール管理システムサーバ 1 0 1 は、企業の営業日と休業日を定義したカレンダー情報と、バッチ処理システムの実行スケジュールを管理するシステムサーバである。

40

【 0 0 4 0 】

スケジュール管理システムクライアント 1 0 2 は、スケジュール管理システムサーバ 1 0 1 が管理するスケジュールに基づいてバッチ処理システムを実行する上位ソフトウェアモジュールである。

【 0 0 4 1 】

バッチ処理システム 1 0 3 は、スケジュール管理システムクライアント 1 0 2 により実行され、インターネット上のハードウェアを利用する際に、自社で運用していたスケジュール監視システム (図示せず) をスケジュール管理システムサーバ 1 0 1 が提供するスケ

50

ジュール管理システムに置き換える下位ソフトウェアモジュールである。

【0042】

なお、サイト12上のスケジュール管理システムクライアント102及びバッチ処理システム103の対は複数存在すると思われるが、説明の簡略化のために一組に制限している。サイト12上にスケジュール管理システムクライアント102及びバッチ処理システム103の対が複数存在する場合も本発明の射程に含まれるのは言うまでもない。

【0043】

また、スケジュール管理システムクライアント102及びバッチ処理システム103が動作するサイト12が複数存在することも発明者は想定している。これらも本発明の射程に含まれる。

【0044】

サイト13上で動作する業務アプリケーション104は、Salesforce.com（登録商標）やGoogle Apps（登録商標）などインターネット上で公開される業務アプリケーションである。

【0045】

図2は、図1で説明した各サイトの開放ポート一覧に関する図である。ここで、「ポート番号」はTCP（トランスミッションコントロールプロトコル）またはUDP（ユーザデータグラムプロトコル）で定義される使用ポートアドレスのことを言う。

【0046】

サイト11では、スケジュール管理システムクライアント102からの問い合わせと、業務アプリケーション104からの問い合わせをWebサービスとして受け付ける。このため、インターネット10を経由してのサイト11へのポート番号80（HTTP）、若しくはポート番号443（HTTPS）の通信が許可されている。

【0047】

サイト12では、スケジュール管理システムクライアント102がスケジュール管理システムサーバ101のWebサービスに問い合わせるために、サイト12からインターネット10を経由してのポート番号80、若しくはポート番号443が解放されている。

【0048】

サイト13に関してもサイト12と同様である。これにより、不適切なポートを経由しての外部からのアクセスをサイト12とサイト13は遮断することができる。

【0049】

図3は、スケジュール管理システムサーバ101の構成図である。

【0050】

スケジュール管理システムサーバ101は、定休日情報リポジトリ302と、例外休業日情報リポジトリ303と、バッチ実行スケジュールリポジトリ304と、実行判定モジュール305と、カレンダー情報提供モジュール306と、日付提供モジュール307と、を含んで構成される。

【0051】

定休日情報リポジトリ302は、企業毎の定休日を管理するリポジトリ（データ格納部）である。

【0052】

例外休業日情報リポジトリ303は、企業毎の定休日以外の休業日を管理するリポジトリである。

【0053】

これらの定休日情報リポジトリ302及び例外休業日情報リポジトリ303は、該企業の定休日など、スケジュールの基礎情報を提供する。すなわち該当日が定休日情報リポジトリ302及び例外休業日情報リポジトリ303から上げられる情報に含まれていなければ、営業日であるとの判断をなすことができる。

【0054】

バッチ実行スケジュールリポジトリ304は、バッチ処理システム103の稼働スケジ

10

20

30

40

50

ジュールを管理するリポジトリである。

【 0 0 5 5 】

実行判定モジュール 3 0 5 は、スケジュール管理システムクライアント 1 0 2 からの問い合わせに対して、対象のバッチがその時実行出来るか否かを判定するソフトウェアモジュールである。

【 0 0 5 6 】

カレンダー情報提供モジュール 3 0 6 は、業務アプリケーション 1 0 4 からの問い合わせに対して、休業日情報を返すソフトウェアモジュールである。

【 0 0 5 7 】

日付提供モジュール 3 0 7 は、業務アプリケーション 1 0 4 からの問い合わせに対して、抽象的な表現を具体的な日付に変換するソフトウェアモジュールである。

【 0 0 5 8 】

スケジュール管理システムサーバ 1 0 1 が持つ実行判定モジュール 3 0 5 と、カレンダー情報提供モジュール 3 0 6 と、日付提供モジュール 3 0 7 は S a a S 等の W e b サービスとしてインターネットに公開されている。

【 0 0 5 9 】

図 4 は、スケジュール管理システムクライアント 1 0 2 の構成図である。

【 0 0 6 0 】

スケジュール管理システムクライアント 1 0 2 は、バッチ実行モジュール 4 0 2 とプロパティファイル 4 0 3 を含んで構成される。

【 0 0 6 1 】

バッチ実行モジュール 4 0 2 は、スケジュール管理システムサーバ 1 0 1 の実行判定モジュール 3 0 5 にバッチ処理システム 1 0 3 の起動可否を確認し、可能であった場合はバッチ処理システム 1 0 3 を起動するソフトウェアモジュールである。

【 0 0 6 2 】

プロパティファイル 4 0 3 は、バッチ実行モジュール 4 0 2 が管理するバッチ処理システム 1 0 3 を表す I D とバッチ処理システム 1 0 3 を所有する企業 I D が格納されたデータ格納部である。

【 0 0 6 3 】

図 5 は、スケジュール管理システムサーバ 1 0 1 で公開される W e b サービスの一覧に関する図である。

【 0 0 6 4 】

実行判定モジュール 3 0 5 は、スケジュール管理システムクライアント 1 0 2 のバッチ実行モジュール 4 0 2 から渡される企業 I D とバッチ I D を引数として取得し、バッチ I D が表すバッチ処理システム 1 0 3 がその時起動可能であるかを判定し、その可否情報をスケジュール管理システムクライアント 1 0 2 に返す。

【 0 0 6 5 】

カレンダー情報提供モジュール 3 0 6 は、業務アプリケーション 1 0 4 から渡される企業 I D と、確認期間開始日と確認期間終了日で指定される期間情報を元に、企業 I D に該当する企業に関して、指定された期間内の休業日一覧を呼び出し元の業務アプリケーション 1 0 4 に返す。

【 0 0 6 6 】

日付提供モジュール 3 0 7 は業務アプリケーション 1 0 4 から渡される企業 I D と、日付確認の対象月を指定する確認指定月情報と、確認指定月情報で指定された月の中で何番目の営業日（または休業日）を検索するかを指定する確認数え日と、日付確認の対象が営業日か休業日かを指定する営業日休業日フラグを元に、定休日情報リポジトリ 3 0 2 と例外休業日情報リポジトリ 3 0 3 を参照し、具体的な日付情報を返す。例えば、業務アプリケーション 1 0 4 から、企業 I D として対象の企業 I D、確認指定月として 1、確認数え日として 3、営業日休業日フラグとして t r u e を受け取った場合、日付提供モジュール 3 0 7 は企業 I D の定休日情報リポジトリ 3 0 2 と例外休業日情報リポジトリ 3 0 3 の情

10

20

30

40

50

報を参照し、企業IDに該当する企業の1月の第3営業日にあたる日付情報を呼び出し元の業務アプリケーション104に返す。

【0067】

図6は、定休日情報リポジトリ302の一例を示す図である。

【0068】

ここでは、企業IDとその企業の定休日（定期休業曜日）が登録されている。図6の場合、Company Aは土曜日と日曜日が定休日であり、Company Bは火曜日が定休日であることが登録されている。

【0069】

図7は、例外休業日情報リポジトリ303の一例を示す図である。

10

【0070】

ここでは、企業IDとその企業の定休日以外の休業日（例外休業日）が登録されている。

【0071】

図7の場合、Company Aは3月10日と5月20日、Company Bは12月31日が休業日であることが登録されている。

【0072】

図8は、バッチ実行スケジュールリポジトリ304の一例を表す図である。

【0073】

ここでは、バッチ処理システム103を表すバッチIDとその実行開始時刻、及び、実施周期、周期で指定された日にちが営業日の場合、バッチ処理システム103を起動させるか否かを表す営業日実施フラグ、周期で指定された日にちが休業日の場合、バッチ処理システム103を起動させるか否かを表す休業日実施フラグが登録されている。なお、図中の「最終実施日」は、バッチ処理を最後に行った日をいう。便宜上スケジュール管理システムクライアント102からスケジュール管理システムサーバ101への問い合わせがあった最終の日となると思われるが、「最終実施日」を如何様にするかは設計事項である。なお、後述する図10ではステップS1109、S1111がこの「最終実施日」の更新に当たる。

20

【0074】

図8では、Batch Aは毎週水曜日の11時に実行、Batch Bは毎月最終日が営業日の場合のみ12時に実行、Batch Cは2011年の2月18日から2日おきで、営業日に該当する日のみ実施することを表している。

30

【0075】

なお、定休日情報リポジトリ302、例外休業日情報リポジトリ303及びバッチ実行スケジュールリポジトリ304は、各企業の担当者により予めこれらの情報が登録されているものとし、登録手段に関して言及しないものとする。

【0076】

図9は、本発明におけるバッチ実行モジュール402の動作を示すフローチャートである。

【0077】

バッチ実行モジュール402はLinux（登録商標）のcronやWindows（登録商標）のタスクスケジューラなどにより定期的に行われるモジュールである。

40

【0078】

起動したバッチ実行モジュール402はプロパティファイル403からバッチ処理システム103を管理する企業を表す企業IDと、バッチ処理システム103を表すバッチIDを取得し（ステップS1001）、スケジュール管理システムサーバ101の実行判定モジュール305を公開したWebサービスに問い合わせを行う（ステップS1002）。

【0079】

スケジュール管理システムサーバ101からの戻り値がtrueの場合（ステップS1

50

003: Yes)、バッチ処理システム103を実行可能である。このため、プロパティファイル403から取得したバッチIDに該当するバッチ実行モジュール402を実行する(ステップS1004)。スケジュール管理システムサーバ101からの戻り値がfalseの場合(ステップS1003: No)、バッチ処理システム103は実行不可であるため、処理を終了することとなる。

【0080】

図10は、本発明における実行判定モジュール305の動作を示すフローチャートである。

【0081】

実行判定モジュール305はWebサービスとして公開され、スケジュール管理システムクライアント102のバッチ実行モジュール402から呼び出されることで処理が開始する。

【0082】

処理開始後、実行判定モジュール305は、まずバッチ実行モジュール402から取得した企業ID情報をキーに、定休日情報リポジトリ302と例外休業日情報リポジトリ303を参照し、休業日一覧を取得する(ステップS1101)。

【0083】

次に、実行判定モジュール305は、処理実行時の日時とバッチ実行モジュール402から取得したバッチIDの情報をもとに、バッチ実行スケジュールリポジトリ304を参照する(ステップS1102)。

【0084】

ステップS1102で参照した結果を元に実行判定モジュール305は処理実施日がバッチ実行スケジュールリポジトリ304の実施周期で指定された日に該当し、且つ、処理実施時刻がバッチ実行スケジュールリポジトリ304の始日時を過ぎている、且つ、バッチ実行スケジュールリポジトリ304の最終実行日時が実施日よりも過去の日付であるか(=営業日であるか)を判定する(ステップS1103)。

【0085】

営業日である場合(ステップS1104: Yes)、バッチ実行スケジュールリポジトリ304から取得した情報の営業日実施フラグを参照する。trueである場合(ステップS1105: Yes)は、呼び出し元のバッチ実行モジュール402にtrue(ステップS1107)を返し、false(ステップS1105: No)である場合は、呼び出し元のバッチ実行モジュール402にfalseを返す(ステップS1110)。

【0086】

また、実行判定モジュール305は、処理実施日が休業日であった場合(ステップS1104: No)、バッチ実行スケジュールリポジトリ304から取得した情報の休業日実施フラグを参照する。休業日実施がtrueである場合は(ステップS1106: Yes)、呼び出し元のバッチ実行モジュール402にtrueを返し(ステップS1108)、falseである場合は(ステップS1106: No)、呼び出し元のバッチ実行モジュール402にfalseを返す(ステップS1112)。

【0087】

なお、呼び出し元のバッチ実行モジュール402にtrueを返した場合、バッチ実行スケジュールリポジトリ304内のバッチIDに該当するデータに関して、最終実施日を処理実施日に書き換える(ステップS1109及びステップS1111)。

【0088】

図11は、本発明におけるカレンダー情報提供モジュール306の動作を示すフローチャートである。

【0089】

カレンダー情報提供モジュール306はWebサービスとして公開される。業務アプリケーション104がカレンダー情報提供モジュール306を呼び出すことでカレンダー情報提供モジュール306の処理が開始される。

10

20

30

40

50

【 0 0 9 0 】

カレンダー情報提供モジュール 3 0 6 の処理開始後、カレンダー情報提供モジュール 3 0 6 は、まず業務アプリケーション 1 0 4 から取得した企業 ID 情報をキーに、定休日情報リポジトリ 3 0 2 と例外休業日情報リポジトリ 3 0 3 を参照し、該当企業の休業日を取得する（ステップ S 1 2 0 1 ）。

【 0 0 9 1 】

次に、カレンダー情報提供モジュール 3 0 6 は、ステップ S 1 2 0 1 で取得した休業日情報から、業務アプリケーション 1 0 4 から取得した確認期間開始日と確認期間終了日で指定された期間に含まれる休業日の一覧を作成する（ステップ S 1 2 0 2 ）。

【 0 0 9 2 】

その後カレンダー情報提供モジュール 3 0 6 は、作成した休業日一覧を呼び出し元の業務アプリケーション 1 0 4 に返す（ステップ S 1 2 0 3 ）。

【 0 0 9 3 】

業務アプリケーション 1 0 4 はその画面表示に送られた休業日一覧を反映するなどして、利用者へのサービスの情報として利用する。

【 0 0 9 4 】

図 1 2 は、本発明における日付提供モジュール 3 0 7 の動作を示すフローチャートである。

【 0 0 9 5 】

日付提供モジュール 3 0 7 は W e b サービスとして公開される。日付提供モジュール 3 0 7 は業務アプリケーション 1 0 4 から呼び出されることで処理が開始する。

【 0 0 9 6 】

日付提供モジュール 3 0 7 の処理開始後、日付提供モジュール 3 0 7 は、まず業務アプリケーション 1 0 4 から取得した企業 ID 情報をキーに、定休日情報リポジトリ 3 0 2 と例外休業日情報リポジトリ 3 0 3 を参照し、休業日一覧を取得する（ステップ S 1 3 0 1 ）。

【 0 0 9 7 】

次に業務アプリケーション 1 0 4 から取得した確認指定月、確認数え日、営業日休業日フラグを元に、具体的な日付を算出する（ステップ S 1 3 0 2 ）。例えば、業務アプリケーション 1 0 4 から、企業 ID は C o m p a n y A、確認指定月は 1、確認数え日は 3、営業日休業日フラグとして t r u e を受け取ったとき、定休日情報リポジトリ 3 0 2 の内容が図 6、例外休業日情報リポジトリ 3 0 3 の内容が図 7 の状態とすると、処理実施年度の 1 月の中で、土曜日と日曜日を除く 3 つ目の日付を、呼び出し元の業務アプリケーション 1 0 4 に返す（ステップ S 1 3 0 3 ）。このように、日付提供モジュール 3 0 7 を使用することで、業務アプリケーション 1 0 4 は抽象的に表現された日付情報から具体的な日付情報を取得することができる。

【 0 0 9 8 】

このように処理することで、インターネット上に不要なポートを公開する必要なくセキュアな環境でバッチ処理システムを稼働させることが可能となる。すなわち、1) 企業が持つ複数のバッチ処理システムと、それらの管理と実行を行うスケジュール管理システムをインターネット上のハードウェアリソース上で稼働させる、2) 企業が持つシステムをインターネット上にあるアプリケーションに置き換えるという問題を解決することができる。

【 0 0 9 9 】

以上、本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は前記の実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更が可能であることは言うまでもない。

【 0 1 0 0 】

例えば、スケジュール管理システムサーバ 1 0 1 中には定休日情報リポジトリ 3 0 2、例外休業日情報リポジトリ 3 0 3、バッチ実行スケジュールリポジトリ 3 0 4、実行判定

10

20

30

40

50

モジュール 305、カレンダー情報提供モジュール 306、日付提供モジュール 307 が存在する。これらの役割を併せ持つソフトウェアモジュールに処理を一元化する場合や、いくつかの処理を行うモジュールを用いるなど、一般的な応用事例については当然に本発明の射程に含まれる。

【0101】

また上記では、処理すべきバッチの有無の判定結果はその如何にかかわらずスケジュール管理システムサーバ 101 からスケジュール管理システムクライアント 102 へ伝達された。しかし、処理すべきバッチがあるときのみスケジュール管理システムサーバ 101 からスケジュール管理システムクライアント 102 に伝達するようにしても良い。

【産業上の利用可能性】

10

【0102】

本発明を採用することで、インターネット上で提供されるハードウェアリソースを利用してバッチ処理システムをセキュアに稼働させる為の手段、または、インターネット上で公開されているアプリケーションを提供する事業者が、利用者が所属する企業の営業日と休業日情報に合わせたサービスを提供するための手段として利用することが可能となる。

【符号の説明】

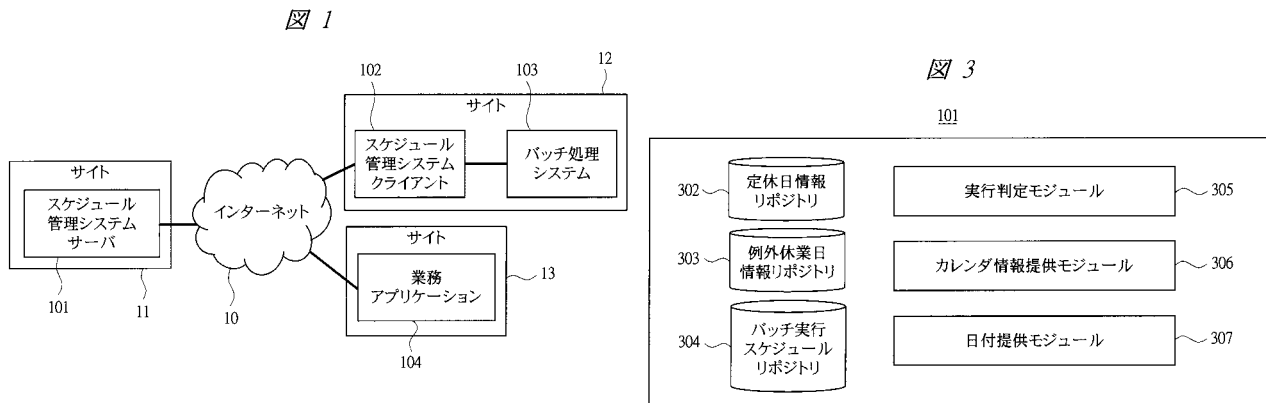
【0103】

- 10：インターネット、11、12、13：サイト、
 101：スケジュール管理システムサーバ、
 102：スケジュール管理システムクライアント、103：バッチ処理システム、
 104：業務アプリケーション、302：定休日情報リポジトリ、
 303：例外休業日情報リポジトリ、304：バッチ実行スケジュールリポジトリ、
 305：実行判定モジュール、306：カレンダー情報提供モジュール、
 307：日付提供モジュール、402：バッチ実行モジュール、
 403：プロパティファイル。

20

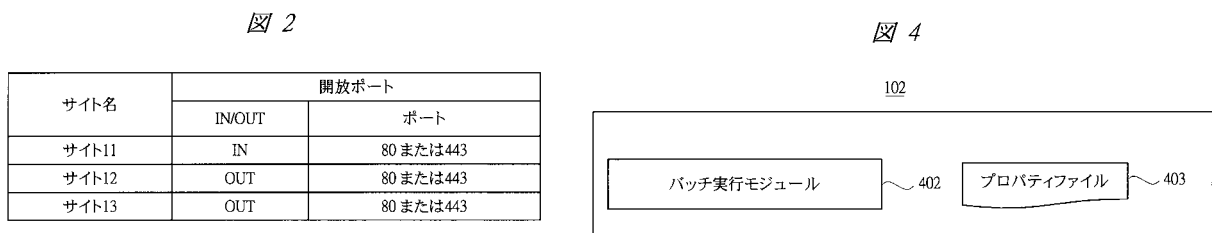
【図1】

【図3】



【図2】

【図4】



【 図 5 】

図 5		
モジュール名	引数	戻り値
実行判定モジュール	企業ID	実行可否 (true/false)
	バッチID	
カレンダー情報提供 モジュール	企業ID	休業日一覧
	確認期間開始日	
	確認期間終了日	
日付提供モジュール	企業ID	指定された日付
	確認指定月	
	確認数え日	
	営業日休業日フラグ (true: 営業日/false: 休業日)	

【圖 7】

企業ID	例外休業日
CompanyA	3月10日
CompanyA	5月20日
CompanyB	12月31日

【 図 8 】

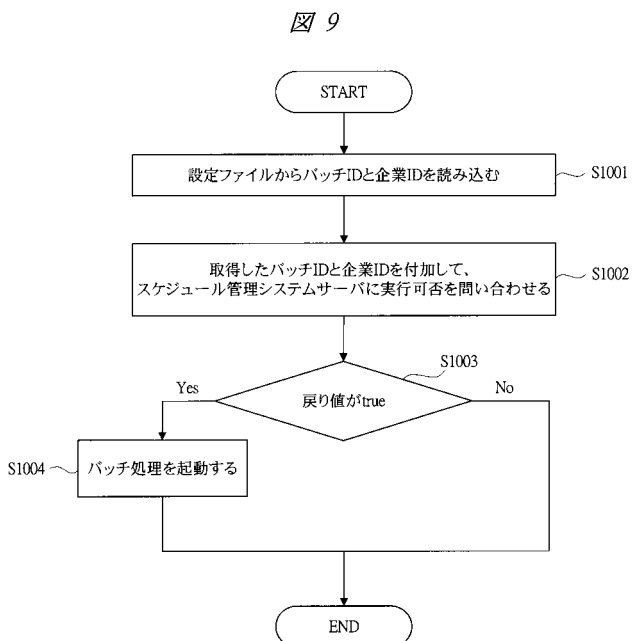
バッチID	実行開始時刻	実施周期	営業日実施	休業日実施	最終実施日
BatchA	11:00	毎週水曜日	true	true	2011/2/2
BatchB	12:00	毎月最終日	true	false	2011/1/31
BatchC	23:00	2011/02/18から 2日おき	true	false	2011/2/22

【 図 6 】

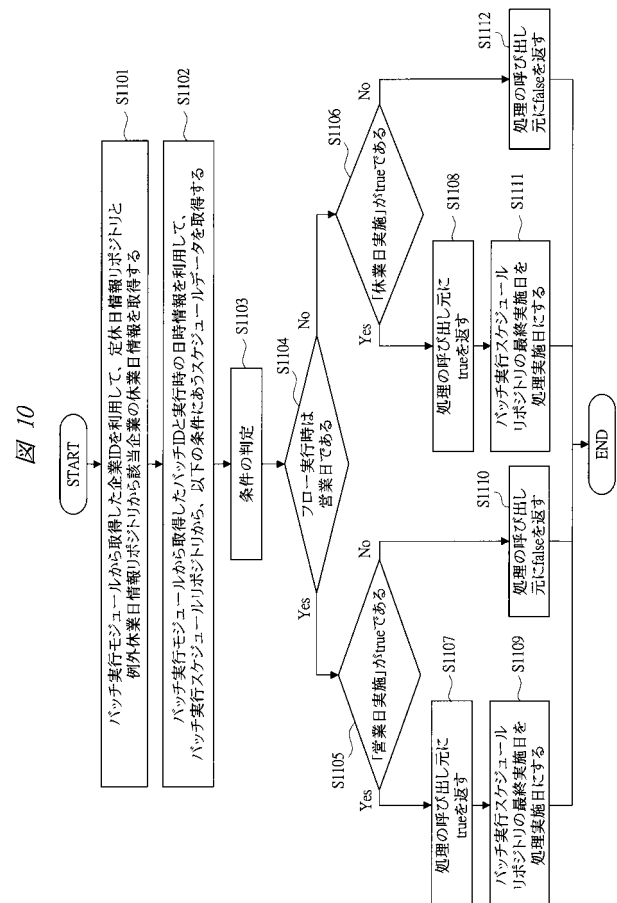
企業ID	定期休業曜日
CompanyA	土日
CompanyB	火

图 8

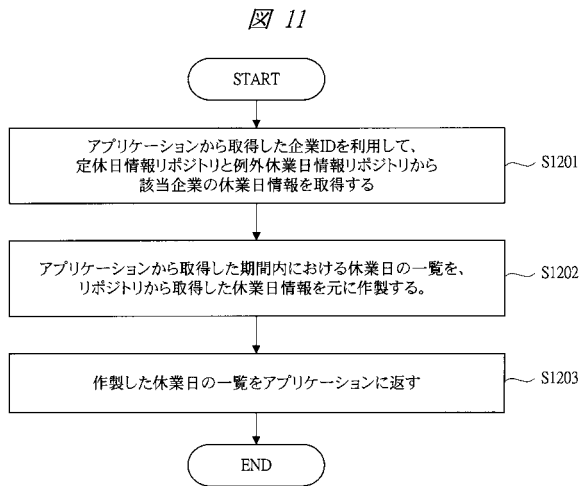
【 図 9 】



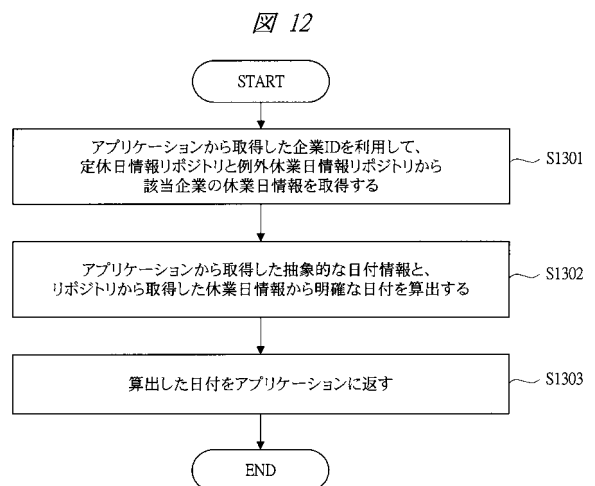
【 図 1 0 】



【図 1 1】



【図 1 2】



フロントページの続き

- (72)発明者 渡邊 淳弘
東京都品川区大崎一丁目2番1号 株式会社日立情報システムズ内
- (72)発明者 菊池 一也
東京都品川区大崎一丁目2番1号 株式会社日立情報システムズ内