

(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2008-234646

(P2008-234646A)

(43) 公開日 平成20年10月2日 (2008. 10. 2)

| | | | | | | |
|----------------|--------------|-------------------|----------------|--------------|----------------|------------------|
| (51) Int. Cl. | | F I | | | テーマコード (参考) | |
| G 0 6 F | 15/00 | (2006. 01) | G 0 6 F | 15/00 | 3 9 0 | 5 B 0 1 9 |
| G 0 6 Q | 10/00 | (2006. 01) | G 0 6 F | 17/60 | 1 6 4 | 5 B 1 8 5 |
| G 0 6 F | 15/02 | (2006. 01) | G 0 6 F | 15/02 | 3 5 5 A | |

審査請求 未請求 請求項の数 12 O L (全 75 頁)

| | | | |
|--------------|----------------------------|----------|------------------------|
| (21) 出願番号 | 特願2008-66659 (P2008-66659) | (71) 出願人 | 000006747 |
| (22) 出願日 | 平成20年3月14日 (2008. 3. 14) | | 株式会社リコー |
| (31) 優先権主張番号 | 11/724, 757 | | 東京都大田区中馬込 1 丁目 3 番 6 号 |
| (32) 優先日 | 平成19年3月15日 (2007. 3. 15) | (74) 代理人 | 100070150 |
| (33) 優先権主張国 | 米国 (US) | | 弁理士 伊東 忠彦 |
| | | (72) 発明者 | テツロウ モトヤマ |
| | | | アメリカ合衆国、ニュージャージー州、07 |
| | | | 006、ウェスト コードウェル、デドリ |
| | | | ック プレイス 5、リコー アメリカス |
| | | | コーポレーション内 |
| | | (72) 発明者 | アヴェリー カーティス フォング |
| | | | アメリカ合衆国、ニュージャージー州、07 |
| | | | 006、ウェスト コードウェル、デドリ |
| | | | ック プレイス 5、リコー アメリカス |
| | | | コーポレーション内 |

最終頁に続く

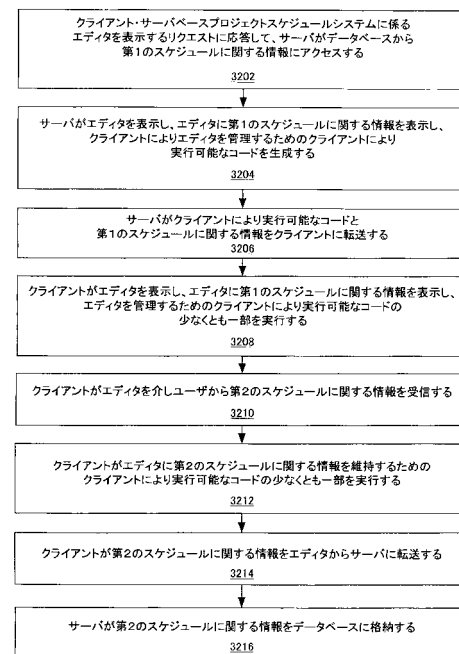
(54) 【発明の名称】 ネットワーク上でプロジェクトスケジュールを管理するためのプロジェクトタスク管理システムのドキュメント・オブジェクト・モデル (DOM) 要素のクラスオブジェクトラップ

(57) 【要約】 (修正有)

【課題】クライアント・サーバベースのプロジェクトスケジュール管理システムによりプロジェクトスケジュールを効果的に管理する技術の提供。

【解決手段】クライアント・サーバベースプロジェクトスケジュール管理システムは、プロジェクトの各メンバーによる各種スケジューリングタスクを実行するため、ウェブブラウザを介しアクセス可能な複数のエディタを有する。クライアントにより実行可能なコードがサーバにより生成され、スケジュールに関する情報と共にクライアントに転送される。クライアントは、このコードを実行し、各エディタを表示し、新規の又は更新された情報を管理及び維持する。タスクの各行は対応するオブジェクトにより表され、エディタ要素は各オブジェクトにより直接アクセス可能なオブジェクト属性である。データベースクエリが、使用される情報と置換されるプレースホルダーを含むコンスタント文字列に基づきサーバにより生成される。

【選択図】図32



【特許請求の範囲】**【請求項 1】**

プロジェクトスケジュールシステムにおいてタスクを管理するためのコンピュータにより実現される方法であって、

表示テーブルの各行がプロジェクトスケジュールに係るタスクと前記プロジェクトスケジュールシステムに係るエディタの各要素に対応する表示値とに対応する前記表示テーブルの行に影響を与えるイベントにตอบสนองして、クラスオブジェクトが前記イベントに基づき前記エディタの要素に対応する値を操作するステップを有し、

前記エディタの 1 以上の要素は、前記クラスオブジェクトの属性であり、

前記エディタの要素に対応する値を操作するステップは、前記クラスオブジェクトの 1 以上のメンバーファンクションが、

前記要素に対応する前記クラスオブジェクトの 1 以上の属性に直接アクセスするステップと、

前記イベントに基づき前記クラスオブジェクトの 1 以上の属性のそれぞれの値を直接操作するステップと、

を実行することから構成される方法。

【請求項 2】

前記値を操作するステップは、クライアントにより実行され、

当該方法はさらに、

前記クライアントが前記 1 以上の属性の少なくとも 1 つに対する値をサーバに送信するステップと、

前記サーバが前記 1 以上の属性の少なくとも 1 つに対する値をデータベースに格納するステップと、

を有する、請求項 1 記載の方法。

【請求項 3】

前記 1 以上の属性のそれぞれの値を操作するステップは、前記クラスオブジェクトの 1 以上の属性の少なくとも 1 つに対する値を変更する、請求項 1 記載の方法。

【請求項 4】

前記イベントは、前記エディタとのユーザのやりとりに基づく、請求項 3 記載の方法。

【請求項 5】

前記 1 以上の属性のそれぞれの値を操作するステップは、前記クラスオブジェクトの 1 以上の属性の少なくとも 1 つの表示を変更する、請求項 1 記載の方法。

【請求項 6】

前記イベントは、前記エディタとのユーザのやりとりに基づく、請求項 5 記載の方法。

【請求項 7】

前記クラスオブジェクトがインスタンス化されるクラスは、クライアント上に前記エディタを表示及び管理するよう動作する前記エディタの JavaScript プロセッサコンポーネントのコンポーネントから構成される、請求項 1 記載の方法。

【請求項 8】

前記クラスオブジェクトは、前記表示テーブルに対応する、請求項 1 記載の方法。

【請求項 9】

前記クラスオブジェクトは、前記表示テーブルの行に対応する、請求項 1 記載の方法。

【請求項 10】

前記クラスオブジェクトは、前記表示テーブルの行のセルに対応する、請求項 1 記載の方法。

【請求項 11】

前記クラスオブジェクトは、前記表示テーブルの行のセルに含まれる要素に対応する、請求項 1 記載の方法。

【請求項 12】

請求項 1 乃至 11 何れか一項記載の方法を 1 以上のプロセッサに実行させる 1 以上の命

10

20

30

40

50

令シーケンスを有するコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般にプロジェクト管理に関し、より詳細には、クライアント・サーバネットワークアーキテクチャによりプロジェクトスケジュールを管理するためのタスク管理システムに関する。

【背景技術】

【0002】

製品開発プロジェクトは、典型的には、監視及び管理のための労力をかなり要する。さらに、コンピュータソフトウェア開発プロジェクトは、本来的には管理することが困難である。この困難さは、ソフトウェアパッケージから構成される大量のタスク及び関連する提出物と、当該タスク及び提出物に係る大量の事務処理及びプロジェクトファイルとに部分的に起因する。他の原因となるファクタは、ソフトウェアパッケージの開発サイクルの期間中に各タスクと提出物との間に確立された複雑な相互依存性群に関するものである。さらなる他の原因となるファクタは、開発中のソフトウェアに関する設計仕様を生成及び維持する必要である。

10

【0003】

開発プロジェクトの管理は、典型的には、プロジェクト文書へのアクセス、スケジュールなどを整理、維持及び制御することを含む。さらに、企業組織内では、しばしば同時に実行される開発プロジェクトが複数存在し、このため、文書管理作業が大きく拡大する。従来、マスタープロジェクトスケジュールの管理は、手作業によりデータをスケジューリングアプリケーションに入力し、手作業によりスケジュール間のリンクを生成し、手作業により各開発者のタスクスケジュールをマスタープロジェクトスケジュールに集約させることを伴ってきた。これは、監視及びクオリティの制御がほとんどないため、面倒で間違いやすいタスクである。

20

【0004】

マスタープロジェクトスケジュールはしばしば流動的なものであり、このため管理には、タスク状態及び関連するスケジュール更新を開発者に求めることとなる。しばしば、開発者によって管理に対して提供されるフィードバックは、ほとんど監視されておらず、厳格なポリシー、手順又は検証プロセスに従うものでない。このため、プロジェクトスケジュールの実際の状態は、各タスクの進捗が主観により支配され、しばしばタスクを割り当てられた各個人による独立的な進捗レポートとなってしまうため、確定することが困難となる。

30

【0005】

パブリック及びプライベートネットワークは、プロジェクトチームの各メンバーがプロジェクトに関するすべての情報を取得するための有用かつシンプルな通信機構を提供する。プロジェクト及びプロジェクトメンバーに関するウェブページを維持することは、プロジェクトチームの各メンバーが当該プロジェクトの状態を判断するためスケジュールに容易にアクセスすることを可能にする。しかしながら、プロジェクトチームの各メンバーのスケジュールを生成及び更新することに関するタスクと、各メンバーのすべてのスケジュールを1つのプロジェクトスケジュールに統合することに関するタスクは、簡単なものではない。

40

【0006】

プロジェクトスケジュール管理の1つのアプローチは、HTMLエディタを使用して、すべてのスケジュールを手作業により生成することに関するものである。プロジェクトマネージャは、主要なプロジェクトタスクを表示するプロジェクトスケジュール全体を生成することができる。プロジェクトタスクは、いくつかのサブタスクに分割可能であり、これらのサブタスクが非公式にプロジェクトチームの各メンバーに割当て可能である。その後、プロジェクトチームの各メンバーは、個別のメンバースケジュールの形式により統一

50

されていないリスクはあるが、メンバースケジュールを生成することが可能である。すべてのメンバーのスケジュールから、プロジェクトマネージャは当該プロジェクトスケジュールに関するすべてのサブタスクスケジュールを集計し、全体的なプロジェクトスケジュールを更新することが可能となる。さらに、すべてのスケジュールは、各タスクのすべてのスケジュールの履歴を維持することが可能である。小さなグループでは、スケジューリングは退屈なものである。大きなグループでは、スケジューリングは、特にプロジェクトタスクスケジュールによるすべてのサブタスクスケジュールの集計は、複雑かつ退屈なものである。

【 0 0 0 7 】

プロジェクトスケジュール管理の他のアプローチは、ネットワークベースプロジェクトスケジュール管理システムを介し相互依存した各プロセスを自動化することに関する。例えば、手続き型のコンピュータプログラミング言語が、ネットワークベースプロジェクトスケジュール管理システムを開発するのに利用可能である。このような自動化されたシステムでは、グローバルファンクションがプロジェクトスケジュール管理システムのすべてのタスクを実行可能である。しかしながら、当該システムの各機能を変更又は訂正することは、プログラムにおいて変更が必要となる部分を特定するなど、困難な問題を生じさせる可能性がある。さらに、システムの一部の変更はシステムの他の部分に影響を与える可能性があり、正しく実装されない場合には、システムの機能を中断させる可能性がある。さらに、システムに対するグローバル変更の実装やシステムに対するコンポーネントの追加が各コードモジュールへの変更又は新たなコードの開発を必要とする可能性がある場合、複数のエディタページなど同様のシステムコンポーネント及び機能に対して、個別のコードが開発される必要が生ずるかもしれない。

【 0 0 0 8 】

本セクションで説明されたアプローチは、求めることが可能であるが、必ずしも以前に想到又は追求されてきたアプローチである必要はない。特段の記載がない場合、本セクションにおいて説明されるアプローチの何れも単に本セクションに含めることによって従来技術とみなされると考えられるべきでない。

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 9 】

本発明の課題は、クライアント・サーバベースのプロジェクトスケジュール管理システムによりプロジェクトスケジュールを効果的に管理するための技術を提供することである。

【 0 0 1 0 】

より詳細には、本発明は、エディタを構成するのに利用されるオブジェクトのためのシステム及び方法に関する。当該オブジェクトは、直接アクセス及び操作可能なエディタのラップされた DOM (Document Object Model) 要素を含む。

【 課題を解決するための手段 】

【 0 0 1 1 】

クライアント・サーバベースプロジェクトスケジュール管理システムにおけるプロジェクトスケジュールの管理は、クライアントとサーバのそれぞれにより実行される機能とそれらの間の通信に関する。実施例によると、スケジュール管理システムに係るエディタを閲覧するリクエストを受信することに応答して、サーバはデータベースからのスケジュールに関する情報にアクセスする。その後、サーバは当該情報をエディタに表示し、クライアントによりエディタを管理するためのクライアントにより実行可能なコードを生成し、クライアントに転送する。これにより、クライアントは、エディタに関連して抽出された情報を表示し、エディタを介しユーザから入力されるデータを管理及び維持するため、サーバにより生成されたコードを実行することが可能となる。

【 0 0 1 2 】

実施例によると、クライアントがエディタを介し入力データを受信すると、クライアン

10

20

30

40

50

トは、サーバにより生成されたクライアントにより実行可能なコードの一部を実行することによって、当該情報を管理及び維持する。ある時点において、クライアントはデータベースに格納するため、この新たな情報をサーバに転送する。以降においてクライアントがエディタに係るウェブページをリクエストすることに応答して、サーバはデータベースからの該当するエディタに固有の情報にアクセスし、抽出した情報をエディタに係るウェブページに表示するためのさらなるクライアントにより実行可能なコードを生成する。当該情報及びコードは、ウェブページに情報を表示するためクライアントに転送される。

【 0 0 1 3 】

実施例によると、サーバ動作は、サーバによる P H P (H y e r t e x t P r e p r o c e s s o r) スクリプトコードの実行に基づき、クライアント動作は、サーバにより生成された J a v a S c r i p t コードのクライアントによる実行に基づく。

10

【 0 0 1 4 】

実施例によると、ネットワークベースプロジェクトスケジュール管理システムにおけるデータベースクエリの自動生成は、リクエストに係る特定のエディタに関連した特定の“コンスタントクエリ”文字列をプロジェクトスケジュール管理システムから特定することに関する。コンスタントクエリ文字列は1以上のプレースホルダー文字列を有し、各プレースホルダー文字列は、データベースに送信するためのデータベースクエリを生成するため、プレースホルダー文字列が何れの変数値と置換されるか特定する。データベースクエリは、何れかのプレースホルダー文字列に対応する変数値と置換することによって自動生成され、これにより、生成されたクエリはデータベースに送信され、結果を返すことが可能となる。さらに、“プレースホルダー文字列のための定数”が、何れかの埋め込まれたプレースホルダー文字列をコンスタントクエリ文字列から検索し、当該プレースホルダー文字列を置換するための適切な値を特定するのに役立つよう利用されてもよい。プレースホルダー文字列を置換するのに使用される値は、ユーザエディタセッション及び/又はデータベースから取得されてもよい。実施例によると、コンスタントクエリ文字列は、プロジェクトスケジュール管理システムの各エディタコンポーネントの基本的特徴に係るクラスパッケージにおいて定義される。

20

【 0 0 1 5 】

実施例によると、プロジェクトスケジュールシステムにおけるスケジュールタスクの管理は、スケジュール情報が格納されているテーブルの行に影響を与えるイベントに応答することに関し、テーブルの各行はプロジェクトスケジュールに係るタスクに対応し、プロジェクトスケジュールシステムに係るエディタの各要素に対応する各値を格納する。このようなイベントに応答して、影響を受けた行に対応するクラスオブジェクトが、当該イベントに基づきエディタの各要素に対応する各値を操作する。エディタの各要素はクラスオブジェクトの属性として構成されているため、クラスオブジェクトは、これらの要素に対応し、イベントに基づきクラスの属性値を直接操作可能なオブジェクト属性に直接アクセス可能である。このような属性の操作は、属性の表示を変更しうる属性値の変更を含むかもしれない。

30

【 発明の効果 】

【 0 0 1 6 】

本発明によると、クライアント・サーバベースのプロジェクトスケジュール管理システムによりプロジェクトスケジュールを効果的に管理するための技術を提供することが可能となる。

40

【 発明を実施するための最良の形態 】

【 0 0 1 7 】

クライアント・サーバベースのプロジェクトスケジュール管理システムによりプロジェクトスケジュールを管理するための技術が記載される。これらの技術は、主としてソフトウェア開発プロジェクトを参照して説明されるが、当業者は、本発明の効果がまた、他の開発プロジェクトに当該技術を適用するときにも利用可能であると認識すべきである。以下では、説明のため多数の具体的詳細が、本発明の完全なる理解を提供するため与えられ

50

る。しかしながら、本発明はこれらの具体的詳細なしに実現可能であるということは明らかであろう。他の例では、本発明を不必要に不明りょうにすることを回避するため、周知の構成及び装置はブロック図により示される。

【 0 0 1 8 】

[タスク割当てエディタ]

図 1 A は、本発明の実施例によるタスク割当てエディタの画面図である。タスク割当てエディタ 1 0 2 は、ユーザがプロジェクトにおいて完了されるべきプロジェクトタスクを作成するのを支援する。いくつかの組織では、組織に関連して実行されるすべてのプロジェクトに共通したデフォルトプロジェクトタスクが存在する。これらのプロジェクトタスクには、プロジェクトメンバーに割り当てられるサブタスクが関連付けされる。典型的には、プロジェクトマネージャが、プロジェクトメンバーに各タスクを設定及び割り当てる。プロジェクトマネージャは、このタスク割当てエディタ 1 0 2 を利用して、プロジェクトのプロジェクトタスクを設定し、各プロジェクトタスクについてサブタスクを生成し、これらサブタスクをメンバーに割り当てることが可能である。プロジェクトマネージャがタスクの追加及び割当てをしている間、タスク割当てに関する情報は、タスク割当てエディタ 1 0 2 に格納及び維持される。マネージャがタスク割当てエディタ 1 0 2 を用いてセッションを完了すると、タスク割当て情報がデータベースにわたされ、格納され、維持される。

10

【 0 0 1 9 】

ユーザが図 1 A のタスク割当てエディタ 1 0 2 の“終了”ボタンを選択することに応答してなど、タスク割当てセッションが完了すると、タスク割当てウェブページ 1 0 4 が、各プロジェクトメンバーに割り当てられるタスクを表示するため、ウェブサーバにおいて自動生成される。図 1 B は、本発明の実施例によるタスク割当てウェブページのサンプルの画面図である。タスク割当てエディタ 1 0 2 を介し入力及び編集されるタスク及びタスク割当て情報は、ウェブブラウザに表示される際、ウェブページにある形式により表示される。すべてのタスクとタスクの割当ては、好ましくは、各行がタスクに対応し、タスク割当てエディタ 1 0 2 とタスク割当てウェブページ 1 0 4 に表示される 1 以上のデータベーステーブル内に格納される。

20

【 0 0 2 0 】

一実施例によると、タスク割当てマネージャ 1 0 2 (図 1 A) は、各種処理を実行するのに利用可能なボタン(“詳細追加”、“上への行の追加”、“下への行の追加”、“削除”及び“終了”など)を含む。“終了”ボタンは、エディタセッションを終了し、データベースに格納及び維持するタスク割当て情報を送信する。その他のボタンは、タスクに対応する行のチェックボックスを選択することによって選択される必要があるタスクに対して各処理を実行する。“詳細追加”ボタンは、マネージャがサブタスクを追加し、プロジェクトメンバーに割り当てることができるように、プロジェクトタスクの下方に行を追加する。“上への行の追加”及び“下への行の追加”ボタンは、マネージャがさらなるプロジェクトタスクを追加できるように、又はさらなるサブタスクを追加及び割り当てできるように、選択されたタスク(プロジェクトタスク又はサブタスク)に対応する行の上下に行を追加する。追加される行数は、“下への行の追加”ボタンの隣の“行数”メニュー選択によって設定される。“削除”ボタンは、選択されたタスクを削除し、プロジェクトからプロジェクトタスクを削除し、又はプロジェクトメンバーへのサブタスクの割当てを削除する。

30

40

【 0 0 2 1 】

[プロジェクトスケジュールエディタ]

図 2 A は、本発明の実施例によるプロジェクトスケジュールエディタの画面図である。プロジェクトスケジュールエディタ 2 0 2 は、タスク割当てエディタ 1 0 2 (図 1 A) において作成されたプロジェクトタスクのスケジュールを設定するのに使用される。プロジェクトタスクは、プロジェクトスケジュールエディタ 2 0 2 において作成及びスケジュールリングされてもよい。しかしながら、一実施例では、サブタスクはプロジェクトスケジュー

50

ールエディタ 202 を使用してプロジェクトメンバーに割り当てるために、プロジェクトタスクに追加することはできない。おそらく、プロジェクトマネージャは、タスク割当てエディタ 102 の後にプロジェクトスケジュールエディタ 202 を使用するであろう。マネージャは、プロジェクトスケジュールエディタ 202 を使用して、タスク割当てエディタ 102 に追加された主要なプロジェクトタスクの初期的なプロジェクトスケジュールを設定することができる。プロジェクトマネージャがタスクを追加及びスケジュールリングしながら、プロジェクトタスクのスケジュールリングに関する情報がプロジェクトスケジュールエディタ 202 に格納及び維持される。マネージャがプロジェクトスケジュールエディタセッションを終了すると、プロジェクトタスクのスケジュール情報がデータベースにわたされ、格納され、維持される。

10

【0022】

ユーザが図 2 A のプロジェクトスケジュールエディタ 202 の“終了”ボタンを選択することなどに応答して、プロジェクトスケジュールセッションが終了すると、プロジェクトスケジュールウェブページ 204 は、プロジェクトスケジュールのテーブルを表示するため、ウェブサーバにおいて自動生成される。各プロジェクトメンバーのスケジュールがプロジェクトサブタスクについて作成及び/又は更新される場合、プロジェクトスケジュールエディタ 202 は、すべてのサブタスクスケジュールと共に各プロジェクトタスクスケジュールを表示する。プロジェクトスケジュールエディタ 202 は、サブタスクをそれを割り当てられたプロジェクトメンバーと共に表示する。エディタセッションを終了することによって、又は図 2 A のプロジェクトスケジュールエディタ 202 の“併合”を選択することによって、各プロジェクトタスクのすべてのサブタスクスケジュールが、プロジェクトタスクのスケジュールを更新するため自動的に併合又は集計され、プロジェクトタスクスケジュールがデータベースにおいて更新される。

20

【0023】

図 2 B は、本発明の実施例によるプロジェクトスケジュールウェブページのサンプルの画面図である。プロジェクトスケジュールウェブページ 204 は、タスク又はサブタスクが割り当てられたメンバーと共に、プロジェクトタスク及びそのサブタスクのスケジュールを表示するため生成される。プロジェクトスケジュールウェブページ 204 は、プロジェクトチームがタスクのスケジュールにおいて生じた変更を確認することができるように、各プロジェクトタスク及びサブタスクの以前のすべてのスケジュール（以前の日付の取消線などと共に）を表示する。プロジェクトスケジュールエディタ 202 を介し入力及び編集されたプロジェクトスケジュール情報は、ウェブブラウザに表示される際にウェブページの形式により表示される。すべてのプロジェクトタスクのスケジュール及びサブタスクのスケジュールは、好ましくは、各行がタスクに対応し、プロジェクトスケジュールエディタ 202 及びプロジェクトスケジュールウェブページ 204 に表示される 1 以上のデータベーステーブルに格納される。

30

【0024】

一実施例によると、プロジェクトスケジュールエディタ 202（図 2 A）は、各種処理を実行する各ボタン（“上への行の追加”、“下への行の追加”、“削除”、“併合”及び“終了”）を含む。“終了”及び“併合”ボタンは、プロジェクトスケジュールエディタセッションを終了し、データベースに格納及び維持されるプロジェクトタスクスケジュール情報を送信する。“併合”ボタンは、プロジェクトスケジュールがデータベースにおいて更新されるように、各メンバースケジュールをプロジェクトスケジュールに併合する。“併合”ボタンは、プロジェクトスケジュールエディタを更新されたタスクスケジュールによるプロジェクトスケジュールウェブページに再表示する。その他のボタンは、タスクに対応する行のチェックボックスを選択することによって選択されるタスクに対する各処理を実行する。これらの処理は、プロジェクトタスクに対してのみ実行可能であり、メンバーに割り当てられたサブタスクに対しては実行できない。“上への行の追加”及び“下への行の追加”ボタンは、マネージャがさらなるプロジェクトタスクを追加し、タスクのスケジュールを設定することができるように、選択されたプロジェクトに対応する行の

40

50

上下に行を追加する。追加される行数は、“下への行の追加”ボタンの隣の“行数”メニュー選択によって設定される。“削除”ボタンは、選択されたプロジェクトタスクを削除する。

【0025】

[メンバースケジュールエディタ]

図3Aは、本発明の実施例によるメンバースケジュールエディタの画面図である。メンバースケジュールエディタ302(“個人スケジュールエディタ”とも呼ばれる)は、各プロジェクトメンバーのスケジュールを作成するのに使用される。一実施例によると、メンバースケジュールエディタ302は、メンバースケジュールが以前に作成されていた場合、終了していないタスクのみを表示する。メンバーのタスクは、プロジェクトサブタスク及び/又はプロジェクトに関連しないタスクである可能性がある。メンバーは、メンバースケジュールエディタ302を介しスケジュールを設定し、スケジュールを変更し、タスクの結果を更新することが可能である。メンバーの各タスクは、タスクの詳細をスケジュールリングするため、より低いレベルのタスクに分解可能である。より低いレベルのタスクの追加又は変更は、より上位のレベルのタスクのスケジュールに影響を与えるかもしれない。従って、より上位のレベルのタスクスケジュールは、“更新”ボタンが選択されると更新される。メンバーがタスクスケジュールを追加又は変更しながら、タスクのスケジュールリングに関する情報がメンバースケジュールエディタ302に格納及び維持される。メンバーがメンバースケジュールエディタ302のセッションを終了すると、タスクスケジュール情報がデータベースにわたされ、格納及び維持される。図3Aは、ドロップダウンリストにより割り当てられたタスクを示す。

10

20

【0026】

ユーザが図3Aのメンバースケジュールエディタ302の“終了”ボタンを選択などすることに対応して、メンバースケジュールセッションが終了すると、メンバースケジュールウェブページ304(図3Bの画面において“タスクスケジュール”とラベル付けされた)が、メンバースケジュールのテーブルを表示するため、ウェブサーバにおいて自動生成される。図3Bは、本発明の実施例によるメンバーのスケジュールウェブページのサンプルの画面図である。メンバースケジュールエディタ302を介し入力及び編集される個人スケジュール情報は、ウェブブラウザに表示される際にウェブページの形式により表示される。すべてのタスクのスケジュールは、各行がタスクに対応するテーブル内に表示される。メンバースケジュールウェブページ304は、プロジェクトチームがタスクのスケジュールに生じた変更を確認することができるよう、各プロジェクトタスク及びサブタスクの以前のスケジュール(以前の日付の取消線などと共に)を表示する。

30

【0027】

メンバースケジュールエディタ302では、各種処理を実行するのに使用されるボタン(“詳細追加”、“最も下への行の追加”、“上への行の追加”、“下への行の追加”、“削除”、“更新”及び“終了”)がテーブルの近傍に配置される。“終了”ボタンは、メンバースケジュールエディタセッションを終了し、データベースに格納及び維持されるタスクスケジュール情報を送信する。“更新”ボタン及び“最も下への行の追加”ボタンを除いて、その他のボタンは、タスクに対応する行のチェックボックスを選択することによって選択されるタスクに対する処理を実行する。“詳細追加”ボタンは、メンバーがタスクのさらなる詳細を与えるため、タスクにサブタスク(1レベル下位のタスク)を追加できるように、タスクの下方に行を追加する。“上への行の追加”及び“下への行の追加”ボタンは、メンバーが同一レベルにおいてスケジュールにさらなるタスクを追加できるように、選択されたタスクに対応する行の上下に行を追加する。追加される行数は、“下への行の追加”ボタンの隣の“行数”メニュー選択によって設定される。“削除”ボタンは、選択されたタスクを削除する。“削除”ボタンはまた、タスクを削除し、メンバーのスケジュールからタスクに係る下位レベルのすべてのタスクを削除する。“最も下への行の追加”ボタンは、追加される行数が“行数”メニュー選択において設定されるスケジュールの最も下に1以上の最も高いレベルの行を追加する。“更新”ボタンは、下位レベル

40

50

のタスクスケジュールにより上位レベルのすべてのタスクスケジュールを更新し、新たな日付を表示するため、メンバースケジュールエディタ 3 0 2 の表示を更新する。

【 0 0 2 8 】

タスクのスケジュール情報は、プランの開始及び終了日と、実際の開始及び終了日とを有する。プランの日付と実際の日付は、メンバースケジュールエディタ 3 0 2 におけるタスクについて設定及び変更可能である。しかしながら、タスクが最初にスケジュールされるとときには、プランの日付のみがプロジェクトスケジュールエディタ 2 0 2 (図 2 A) のプロジェクトタスクに対して設定可能でない。併合時に、プランの日付は自動更新され、実際の日付はプロジェクトサブタスクのプランの日付と実際の日付のメンバーのスケジュールの情報に基づき、自動設定される。図示しないが、プロジェクトスケジュールエディタ 2 0 2 は、プランされた日付が変更可能となるように、変更することができる。しかしながら、プロジェクトタスクのプランされた日付においてどのような変更がなされても、それはプロジェクトサブタスクのメンバーのスケジュールのプランされた日付を併合することによって上書きされる。データベースの情報は、プロジェクトマネージャがプロジェクトエディタセッションを終了したとき、又はプロジェクトスケジュールエディタ 2 0 2 の “ 併合 ” ボタンを介し、プロジェクトタスクの実際の日付を更新するのに使用される。

10

【 0 0 2 9 】

図 4 は、本発明の実施例による各エディタ (タスク割当て、プロジェクトスケジュール、メンバースケジュール) の 1 つにプロジェクトメンバーがログオンするためのログインウェブページの画面図である。メンバーはプロジェクト番号とメンバー名を入力し、適切なエディタを選択し、その後にエディタにアクセスするためこの情報を送信する。プロジェクトスケジュール管理システムは、この入力を確認し、当該メンバーがプロジェクトの有効なメンバーであるか判断し、選択されたエディタに対するアクセス権を有する。そうでない場合、メンバーはエディタへのアクセスを拒絶される。より強固なセキュリティのため、ログインウェブページ及びエディタは、セキュア HTTP (HTTPS など) を介し実行可能であり、ログインページはログイン前にパスワードを要求することができる。

20

【 0 0 3 0 】

[プロジェクトスケジュール管理システム]

図 5 は、本発明の実施例が実現可能な動作環境を示す図である。図示された動作環境は、ここに記載されるプロジェクトスケジュール管理システムの全体的なシステム構成を説明する。この一例となる動作環境は、複数のワークステーション、1 以上のウェブサーバ及び 1 以上の付属するデータベースから構成され、これらすべては、通信のためソフトウェア開発ネットワークに直接的に又は間接的に接続される。

30

【 0 0 3 1 】

一般に、ウェブサーバ 5 0 7 及び 5 3 0 は、エディタの表示及び管理のためのリソースを有する。ウェブサーバ 5 0 7 及び 5 3 0 は、データ 5 0 8、5 3 8 などのタスク割当て及びタスクスケジュール情報を格納、維持及び管理するため、それぞれデータベース 5 0 6 及び 5 3 6 とやりとりする。2 つのウェブサーバと 2 つのデータベースの図示は、単なる一例に過ぎない。すなわち、ここに記載されるようなプロジェクトスケジュール管理システムにおいて使用されるウェブサーバとデータベースの個数は、実現形態に応じて異なるかもしれない。コンピュータワークステーション 5 0 1 と 5 0 2 のウェブブラウザは、エディタを表示するため、ウェブサーバ 5 0 7 及び 5 3 0 のリソースにアクセスする。プロジェクトメンバー又はマネージャは、ネットワーク 5 0 0 (LAN 又は WAN) を介しエディタにアクセス可能である。プロジェクト管理システムは、プロジェクト、部門、組織レベルなど、組織内の各レベルにおいてプロジェクトを管理するのに利用可能である。

40

【 0 0 3 2 】

ワークステーション 5 0 1、5 0 2 は、典型的には、1 以上のブラウザを有する図 3 5 のコンピュータシステム 3 5 0 0 により示されるように構成されたコンピュータシステムであり、例えば、技術者 / 開発者が製品開発プロジェクトに係るタスクを終了するため使

50

用される。このようなタスクの関連する非限定的な具体例として、プロジェクトの開始、タスクスケジュールの準備及び維持、ソフトウェアアーキテクチャの設計、仕様書の作成、ソフトウェアコードの作成、ソフトウェアコードの実装及びテスト、各種タスクプロジェクトの検査などがあげられる。さらに、プロジェクトマネージャは、プロジェクトの進捗状況を確認及び管理するための情報にアクセスするため、ワークステーション501、502を利用する。開発者及びマネージャは、アクセスポイント524を介しウェブサーバ507、530、データベース506、536、携帯装置20及びラップトップ522など、その他の接続されたコンポーネントにネットワーク500を介し通信を送信する。ウェブサーバ507及び530からウェブページにアクセス可能なワークステーション501、502、携帯装置520及びラップトップ522は、ブラウザのエディタを管理するためウェブページが有するJavaScriptを処理可能である。ブラウザは、JavaScriptを処理することが可能である。

10

20

30

40

50

【0033】

ウェブサーバ507、530は、適切なプロトコル(HTTP(Hypertext Transfer Protocol)、TCP/IP(Transmission Control Protocol/Internet Protocol)など)を利用して、ワークステーション501、502において開発者やマネージャなどのユーザにウェブページを構成するファイル(HTML(Hypertext Markup Language)ファイル、XML(Extensible Markup Language)ファイルなど)を提供するコンピュータハードウェア及びソフトウェアの組み合わせである典型的なウェブサーバを示す。非限定的な具体例として、PHPスクリプトを実行するためのモジュールを有するApacheウェブサーバが、ウェブサーバ507及び530のウェブサーバアプリケーションとして利用可能である。一般に、開発プロジェクトのライフサイクル期間中に交換及び管理される情報の大部分は、ネットワーク500を介しウェブサーバ507、530により提供される。さらに、ここに記載される技術の各特徴は、本発明の実現がこのような実現形態に限定されるものではないが、ウェブサーバ507、530上で実現及び実行されるかもしれない。これらの技術は、ワークステーション501、502などの他の何れかの処理システム、又は図35に示されるような同様に構成されたコンピュータシステム上で実現可能である。

【0034】

データベース506、536は、開発プロジェクトに関連するデータ508、538を格納し、ネットワーク500を介し送信されるクエリを通じてワークステーション501、502における認証された個人による情報へのアクセスを提供する典型的なデータベースを示す。データベース501、502に格納されているデータのタイプは無限であり、非限定的な具体例として、プロジェクト開始フォーム、メンバー及びプロジェクトタスクスケジュール、仕様書、ソフトウェアコード、検査レポート、ウェブページファイル並びに文書ディレクトリ及びインデックスがあげられる。

【0035】

ネットワーク500は、ワークステーション501、502、ウェブサーバ507、530及びデータベース506、536などの各種接続されたコンポーネントの間の情報の交換を実現するため、パケット交換ネットワークなどの従来のネットワークを示す。ネットワーク500は、企業内に実装される従来技術によるイーサネット(登録商標)、高速イーサネット(登録商標)、トークンリングなどのローカル・エリア・ネットワーク(LAN)や802.11a及び802.11b(IEEE(Institute of Electrical and Electronics Engineers)のワーキンググループにより開発された)などのワイヤレスLANであるかもしれない。さらに、ネットワーク500はまた、VPN(Virtual Private Network)を介したりリモートユーザとの通信を実現するため、インターネットなどのワイド・エリア・ネットワーク(WAN)であってもよく、又はLANとWANの組み合わせを表すものであってもよい。さらに、ネットワーク500は、以下に限定されるものではないが、

電線若しくはケーブル又は光若しくは無線接続を含む各種媒体を利用して構成可能である。

【 0 0 3 6 】

図 6 は、本発明の実施例が実現可能な自動化されたスケジューリングシステムのソフトウェアコンポーネントを含む通信アーキテクチャを示す図である。クライアントプロセッサ 6 0 2 はウェブブラウザに対応し、サーバプロセッサ 6 0 4 はウェブサーバ 5 0 7 や 5 3 0 (図 5) などのウェブサーバに対応する。プロジェクトメンバー又はマネージャは、ユーザインタフェース 6 0 1 を介しクライアントプロセッサ 6 0 2 とやりとりする。クライアントプロセッサ 6 0 2 は、ログインウェブページ (図 4) 及び各種エディタウェブページ (図 1 A、2 A、3 A) を管理及び維持する。クライアントプロセッサ 6 0 2 は、これらのウェブページにおいて発生するすべてのイベントを処理する。一実施例によると、クライアントプロセッサ 6 0 2 は、HTTP プロトコルを介しサーバプロセッサ 6 0 4 とやりとりする。一実施例によると、クライアントプロセッサ 6 0 2 は、セキュアな HTTPS プロトコルを介しサーバプロセッサ 6 0 4 とやりとりする。

10

【 0 0 3 7 】

サーバプロセッサ 6 0 4 は、ログインウェブページ (図 4) 及びエディタウェブページ (図 1 A、2 A、3 A) を表示するため、クライアントプロセッサ 6 0 2 に情報を提供する。サーバプロセッサ 6 0 4 はまた、クライアントプロセッサ 6 0 2 がこれらのページにより情報を送信すると、ログイン及びエディタウェブページの情報を処理する。データベース 6 0 6 は、プロジェクト及びタスクスケジューリング情報のレポジトリとなる。サーバプロセッサ 6 0 4 は、データベースにおける情報を取得、追加又は更新するため、データベース 6 0 6 とやりとりする。一実現形態によると、サーバプロセッサ 6 0 4 はデータベース 6 0 6 とやりとりする。しかしながら、他のデータベース及びプロトコルが利用可能である。

20

【 0 0 3 8 】

[クライアント・サーバインタフェース]

図 7 は、本発明の実施例によるシステムのクライアントプロセッサとサーバプロセッサとの間のインタフェースを示す図である。HTTP / HTTPS GET リクエストは、クライアントプロセッサ 6 0 2 にホーム、ログイン (図 4)、プロジェクトスケジュールエディタ (図 2 A)、メンバースケジュールエディタ (図 3 A) 及びタスク割当てエディタ (図 1 A) の各ウェブページをサーバプロセッサ 6 0 4 から取得させる。HTTP / HTTPS POST リクエストは、クライアントプロセッサ 6 0 2 にログイン (図 4) 及びエディタウェブページ (図 1 A、2 A、3 A) に入力された情報を処理のため、サーバプロセッサ 6 0 4 に送信させる。適用可能な HTTP / HTTPS GET リクエスト及び HTTP / HTTPS POST リクエストが、以降においてより詳細に説明される。

30

【 0 0 3 9 】

HTTP / HTTPS GET Project / Dept / Division Home Page リクエストは、サーバプロセッサ 6 0 4 にそれぞれデパートメント又はディビジョンに係るプロジェクトホームページをクライアントプロセッサ 6 0 2 に対してリターンさせる。ホームページは、スケジュール、タスク割当て及びエディタへのログインのための各ウェブページにリンク及び閲覧するためのリンク (ハイパーリンクなど) を含む。

40

【 0 0 4 0 】

HTTP / HTTPS GET current project directory / schedule .html リクエストは、サーバプロセッサ 6 0 4 に図 2 B に示されるような現在のプロジェクトのプロジェクトスケジュールを含むウェブページをクライアントプロセッサ 6 0 2 に対してリターンさせる。

【 0 0 4 1 】

HTTP / HTTPS GET current project directory / task Assignment .html リクエストは、サーバプロセッサ 6 0 4 に図

50

1 Bに示されるような現在のプロジェクトのプロジェクトタスクのタスク割当てを含むウェブページをクライアントプロセッサ602に対してリターンさせる。

【0042】

HTTP/HTTPS GET project_member_directory/schedule.htmlリクエストは、サーバプロセッサ604に図3Bに示されるような現在のプロジェクトのプロジェクトメンバーのスケジュールを含むウェブページをクライアントプロセッサ602に対してリターンさせる。

【0043】

HTTP/HTTPS GET login.htmlリクエストは、サーバプロセッサ604にエディタ(プロジェクトスケジュール、メンバースケジュール、タスク割当て)の1つにプロジェクトメンバー又はマネージャがログオンすることを可能にするウェブページをクライアントプロセッサ602に対してリターンさせる。メンバー又はマネージャは、プロジェクト、メンバー名及びエディタセッションタイプに関する情報を入力する。図4は、エディタの1つにログインするためのウェブページを示す。

【0044】

HTTP/HTTPS GET TaskAssignEditor.htmlリクエストは、サーバプロセッサ604に現在のプロジェクトについてプロジェクトメンバーに各タスクを割り当てるのに利用されるタスク割当てエディタのウェブページをクライアントプロセッサ602に対してリターンさせる。プロジェクトマネージャは、サーバプロセッサ604がタスク割当てエディタウェブページをリターンする前に、プロジェクトメンバーに各タスクを割り当てるためのアクセス権限を要求する。この権限は、マネージャがログインウェブページ(図4)により情報を送信すると検証される。一実施例によると、TaskAssignEditor.htmlは、タスク割当てエディタにイベントを表示、管理及び処理するためのJavaScriptを有する。一実施例によると、TaskAssignEditor.htmlは、データベース506、536から情報を取得し、当該情報をJavaScriptにわたすためのPHPスクリプトを有し、これにより、当該情報は図1Aに示されるようなタスク割当てエディタにより表示される。

【0045】

HTTP/HTTPS GET ProjScheduleEditor.htmlリクエストは、サーバプロセッサ604に現在のプロジェクトのプロジェクトスケジュールを作成又は更新するのに利用されるプロジェクトスケジュールエディタのウェブページをクライアントプロセッサ602に対してリターンさせる。プロジェクトマネージャは、サーバプロセッサ604がプロジェクトスケジュールエディタをリターンする前に、プロジェクトスケジュールを作成するためアクセス権限を有する必要がある。この権限は、マネージャがログインウェブページ(図4)により情報を送信すると検証される。一実施例によると、ProjScheduleEditor.htmlは、プロジェクトスケジュールエディタのウェブページによりイベントを表示、管理及び処理するためのJavaScriptを含む。一実施例によると、ProjScheduleEditor.htmlは、データベース506、536から情報を取得し、当該情報をJavaScriptにわたすためのPHPスクリプトを有し、これにより、当該情報は、図2Aに示されるようなプロジェクトスケジュールエディタに表示される。

【0046】

HTTP/HTTPS GET MembScheduleEditor.htmlリクエストは、サーバプロセッサ604に現在のプロジェクトのプロジェクトメンバーのスケジュールを作成又は更新するのに使用されるメンバースケジュールエディタのウェブページをクライアントプロセッサ602に対してリターンさせる。一実施例によると、スケジュールエディタは、プロジェクトメンバーのスケジュールが以前に作成されていた場合、終了していないタスクのみを表示する。プロジェクトメンバーは、サーバプロセッサ604がこのウェブページをリターンする前に、スケジュールを作成又は編集するための権限を有している必要がある。この権限は、メンバーがログインウェブページ(図4)に情報

10

20

30

40

50

を送信すると検証される。一実施例によると、MembScheduleEditor.htmlは、プロジェクトメンバーのスケジュールエディタにイベントを表示、管理及び処理するためのJavaScriptを有する。一実施例によると、MembScheduleEditor.htmlは、データベース506、536から情報を取得し、当該情報をJavaScriptにわたすためのPHPスクリプトを有し、これにより、情報は、図3Aに示されるようなメンバースケジュールエディタに表示される。

【0047】

HTTP/HTTPS POST login.htmlインタフェースは、クライアントプロセッサ602が各種エディタ（プロジェクトスケジュール、メンバースケジュール、タスク割当て）にアクセス及び表示することを可能にする。このインタフェースは、“送信”ボタンがlogin.htmlに対応するウェブページから選択されると呼び出される。login.htmlに入力される情報は、サーバプロセッサ604のPostLogin.htmlにわたされる。PostLogin.htmlは、当該情報を利用してプロジェクトのメンバーを検証し、当該メンバーがリクエストされたエディタへのアクセス権限を有しているか判断する。当該情報が有効でないか、又は当該メンバーがエディタへのアクセス権限を有していない場合、PostLogin.htmlは、プロジェクトメンバーがリクエストされたエディタにアクセス不可であるというメッセージをクライアントプロセッサ6092にリターンする。そうでない場合、PostLogin.htmlは、エディタの1つに対応するウェブページをリターンする。すなわち、ウェブブラウザは、リクエストされたエディタに対応するウェブページにリダイレクトされる。

10

20

【0048】

HTTP/HTTPS POST PostTaskAssign.htmlは、クライアントプロセッサ602がタスク割当てエディタ（図1A）に入力されたすべての情報をサーバプロセッサ604に送信することを可能にする。このインタフェースは、“終了”ボタンがTaskAssignEditor.htmlに対応するウェブページから選択されると呼び出される。TaskAssignEditor.htmlのエディタに入力された情報は、サーバプロセッサ604のPostTaskAssign.htmlにわたされる。PostTaskAssign.htmlは、適切なデータベース506、536のタスク割当て情報を追加及び更新する。入力された情報の何れかが有効でない場合、又は当該プロセスが適切なデータベースにアクセス又はクエリするのに失敗した場合、適切なメッセージが表示される。PostTaskAssign.htmlはまた、図1Bに示されるようなタスク割当てウェブページを作成する。

30

【0049】

HTTP/HTTPS POST PostProjSchedule.htmlは、クライアントプロセッサ602がプロジェクトスケジュールエディタ（図2A）に入力されたすべての情報をサーバプロセッサ604に送信することを可能にする。このインタフェースは、“終了”ボタンがProjScheduleEditor.htmlに対応するウェブページから選択されると呼び出される。PostProjSchedule.htmlのエディタに入力される情報は、サーバプロセッサ604のPostProjSchedule.htmlにわたされる。PostProjSchedule.htmlは、適切なデータベース506、536のタスクスケジュール情報を追加及び更新する。入力された情報の何れかが有効でない場合、又は当該プロセスが適切なデータベースにアクセス又はクエリすることに失敗した場合、適切なメッセージが表示される。PostProjSchedule.htmlはまた、図2Bに示されるようなプロジェクトスケジュールウェブページを作成する。

40

【0050】

HTTP/HTTPS POST PostMembSchedule.htmlは、クライアントプロセッサ602がプロジェクトメンバーのスケジュールエディタ（図3A）に入力されるすべての情報をサーバプロセッサ604に送信することを可能にする。このインタフェースは、“終了”ボタンがMembScheduleEditor.htmlに

50

対応するウェブページから選択されると呼び出される。MembScheduleEditor.htmに入力された情報は、サーバプロセッサ604のPostMembSchedule.htmにわたされる。PostMembSchedule.htmは、適切なデータベース506、536のタスクスケジュール情報を追加及び更新する。入力された情報の何れかが有効でない場合、又は当該プロセスがデータベースにアクセス又はクエリすることに失敗した場合、適切なメッセージが表示される。PostMembSchedule.htmはまた、図3Bに示されるようなメンバーのウェブページを作成する。

【0051】

各種エディタのウェブページ(TaskAssignEditor.htm、ProjScheduleEditor.htm及びMembScheduleEditor.htm)は、非限定的な一実施例によるJavaScript又はPHPスクリプトを含むファイルを有する。ここに記載される各種機能を実行するのに使用されるスクリプト言語は、実現形態に応じて可変的なものであるかもしれない。ウェブブラウザ(クライアントプロセッサ602など)がエディタのウェブページをリクエストすると、エディタウェブページ及びJavaScriptに対応するすべてのファイルがウェブブラウザにわたされ、これにより、ウェブブラウザはJavaScriptを処理する。しかしながら、PHPスクリプトのファイルはウェブブラウザにはわたされない。PHPスクリプトは、図5のウェブサーバ507、530などのウェブサーバにおいて処理され、PHPスクリプトがウェブページに書き込んだもののみがウェブブラウザにわたされる。

【0052】

図8は、本発明の実施例によるログインウェブページを利用してプロジェクトメンバー又はマネージャがエディタの1つにログオンするためのシーケンス図を示す。この図は、エディタがメンバー又はマネージャに表示される前に、システムの各コンポーネントの間でわたされる情報を示す。ログインウェブページ(図4)上で発生するすべてのイベントを処理するため、クライアントプロセッサ602の内部において処理が実行される。ログインページに入力された情報を検証し、エディタのメンバーのアクセス権限を検証するため、サーバプロセッサ604の内部において処理が実行される。サーバプロセッサ604は、アクセス権限を検証するため、適切なデータベース506又は536から情報を取得する。プロジェクトメンバー又はマネージャは、図9~11に示されるシーケンスを有するエディタの何れかに入る前に、このプロセスを実行する。

【0053】

[各エディタのシーケンス図]

図9(タスク割当てエディタ)、図10(プロジェクトスケジュールエディタ)及び図11(メンバースケジュールエディタ)は、ウェブブラウザにおいて各エディタを表示し、セッションが終了するとエディタに情報を掲示するためのシーケンスを示す。すべてのエディタは、同様のシーケンスに従う。クライアントプロセッサのウェブブラウザのエディタの初期表示を取得するため、適切なタスク割当て/スケジュール情報は、サーバプロセッサのデータベースから取得される(PHPスクリプトを使用して)。サーバプロセッサは、サーバプロセッサが(PHPスクリプトを使用して)生成し、クライアントプロセッサの初期エディタを表示するコードと共に、クライアントプロセッサがエディタを管理及び維持するため実行可能なコード(Javascript)を含むウェブページをわたす。サーバプロセッサは、それがデータベースから取得したタスク割当て/スケジュール情報をクライアントプロセッサにわたすためのコードを生成する。

【0054】

図9は、本発明の実施例によるタスク割当てエディタによるセッションにおけるプロジェクトマネージャのシーケンス図を示す。クライアントプロセッサ602がTaskAssignEditor.htmをリクエストすると、TaskAssignEditor.htmファイルとJavaScriptを有する含まれるすべてのファイル(“.js"の拡張子により示される)が、サーバプロセッサ604からクライアントプロセッサ602にわたされる。PHPスクリプトを有する含まれるファイル(“.php"拡張子に

より示される)は、サーバプロセッサ604において処理される。PHPスクリプトは、適切なデータベース506又は536からタスク割当て情報を取得し、情報をクライアントプロセッサ602にわたすため、TaskAssignEditor.htmのウェブページにJavaScriptを書き込む。クライアントプロセッサ602は、対応するタスク割当てエディタを表示するため、それが受信したすべてのファイルのJavaScriptを処理する。プロジェクトマネージャとタスク割当てエディタとの間のすべてのやりとりは、タスク割当てエディタを管理、維持及び更新するため、JavaScriptにより処理される。プロジェクトマネージャがセッションを終了すると(“終了”を選択するなど)、タスク割当てエディタのすべてのタスク割当て情報は、PostTaskAssign.htmを介しクライアントプロセッサ602からサーバプロセッサ604にわたされる。サーバプロセッサ604は、適切なデータベースの情報を追加又は更新することによって、情報を処理する。データベースのタスク割当て情報を利用して、サーバプロセッサ604は、図1Bに示されるようなプロジェクトタスク割当てのためのウェブページを自動生成する。

【0055】

図10は、本発明の実施例によるプロジェクトスケジュールエディタによるセッションにおけるプロジェクトマネージャのためのシーケンス図を示す。クライアントプロセッサ602がProjScheduleEditor.htmをリクエストすると、ProjScheduleEditor.htmファイルとJavaScriptを有する含まるすべてのファイルが、サーバプロセッサ604からクライアントプロセッサ602にわたされる。PHPスクリプトを有する含まれるファイルは、サーバプロセッサ604において処理される。PHPスクリプトは、適切なデータベースからプロジェクトタスクスケジュール情報を取得し、情報をクライアントプロセッサ602にわたすため、ProjScheduleEditor.htmのウェブページにJavaScriptを書き込む。クライアントプロセッサ602は、プロジェクトスケジュールエディタを表示するため、それが受信したファイルのJavaScriptを処理する。プロジェクトマネージャとプロジェクトスケジュールエディタとの間のすべてのやりとりは、エディタを管理、維持及び更新するため、JavaScriptにより処理される。マネージャがセッションを終了すると(“終了”を選択などして)、プロジェクトスケジュールエディタのすべてのプロジェクトタスクスケジュール情報は、PostProjSchedule.htmインタフェースを介しクライアントプロセッサ602からサーバプロセッサ604にわたされる。サーバプロセッサ604は、適切なデータベースの情報を追加又は更新することによって、情報を処理する。サーバプロセッサ604はまた、プロジェクトスケジュールによりプロジェクトメンバーのスケジュールを自動集計し、データベースのプロジェクトスケジュールを追加又は更新する。データベースのプロジェクトタスクスケジュール情報を利用して、サーバプロセッサ604は、図2Bに示されるようなプロジェクトスケジュールのウェブページを自動生成する。

【0056】

“併合”ボタンの選択に応答したシステムの動作は、“終了”ボタンの選択に対するものと同一である。両方のボタンは、(a)プロジェクトスケジュールエディタからの情報による適切なデータベースの追加及び更新、(b)プロジェクトスケジュールによるメンバーの各スケジュールの集計、(c)データベースのプロジェクトスケジュールの追加及び更新、及び(d)プロジェクトスケジュールのウェブページの作成を引き起こす。さらに、“併合”は、ProjScheduleEditor.htmを再びリクエストすることによって、更新されたプロジェクトスケジュールによるプロジェクトスケジュールエディタを再表示する。

【0057】

図11は、本発明の実施例によるプロジェクトメンバースケジュールエディタ(すなわち、メンバースケジュールエディタ)によるセッションにおけるプロジェクトメンバーのためのシーケンス図を示す。クライアントプロセッサ602がMemSchedule

Editor.htmをリクエストすると、MemberScheduleEditor.htmファイルとJavaScriptを有する含まれるすべてのファイルが、サーバプロセッサ604からクライアントプロセッサ602にわたされる。PHPスクリプトを有する含まれるファイルは、サーバプロセッサ604において処理される。PHPスクリプトは、適切なデータベースからメンバータスクスケジュール情報を取得し、情報をクライアントプロセッサ602にわたすため、MemberScheduleEditor.htmのウェブページにJavaScriptを書き込む。クライアントプロセッサ602は、メンバースケジュールエディタを表示するため、それが受信したファイルのJavaScriptを処理する。プロジェクトメンバーとメンバースケジュールエディタとの間のやりとりは、メンバースケジュールエディタを管理、維持及び更新するため、JavaScriptにより処理される。メンバーがセッションを終了すると(“終了”を選択などして)、メンバースケジュールエディタのメンバータスクスケジュール情報は、PostMemberSchedule.htmインタフェースを介しクライアントプロセッサ602からサーバプロセッサ604にわたされる。サーバプロセッサ604は、適切なデータベースの情報を追加又は更新することによって、情報を処理する。データベースのメンバータスクスケジュール情報を利用して、サーバプロセッサ604は、図3Bに示されるようなメンバースケジュールのウェブページを自動生成する。

10

【0058】

[データベーススキーマ]

図12は、本発明の実施例によるプロジェクト及びプロジェクトメンバーのタスク割当て及びタスクスケジュール情報を格納及び管理するのに使用されるデータベーステーブルのスキーマを示す。当該テーブルは、タスク割当て、プロジェクトタスクのスケジュール、及び各プロジェクトメンバーのスケジュールに関する情報を維持する。これらのテーブルは、タスク割当て、プロジェクトスケジュール及びメンバーのスケジュールのすべてが関連するように、構成及びリンクされる。

20

【0059】

TaskAssignmentテーブル1202は、プロジェクトのプロジェクトタスクと対応するサブタスクとを格納する。TopLevelProjectTaskテーブル1204は、TaskAssignmentテーブル1202にあるプロジェクトタスクのスケジュールを格納する。Level1MemberTaskテーブル1206は、TaskAssignmentテーブル1202に割り当てられているメンバータスクのスケジュールを格納し、TopLevelProjectTaskテーブル1204の対応するプロジェクトタスクのスケジュールにリンクする。これらのテーブルの間のリンクは、プロジェクトスケジュールによるメンバースケジュールの自動集計を可能にする。Level1MemberTaskテーブル1206はまた、何れのプロジェクトタスクにも関連しないメンバータスクのスケジュールを格納する。LevelXMemberTaskテーブル(ただし、 $X = 1, 2, 3, 4$)及びMemberTasksテーブル1208は、メンバーの各種タスクレベルの間のリンクを格納及び管理する。下位レベルのタスクは、上位レベルのタスクのより詳細なタスクである。これらのテーブルの構成は、メンバーのスケジュールを維持する。ProjectTeamテーブル1210は、プロジェクトメンバーに関する情報を含む。プロジェクトメンバーのプロジェクトメンバー情報は、(a)各種エディタへのアクセス権限を決定するための役割、(b)メンバースケジュールウェブページが格納される場所を決定するためのディレクトリ、及び(c)各種レベルにおけるメンバータスクの識別子を決定するのに使用されるIDを含む。

30

40

【0060】

ログインプロセスは、ProjectTeamテーブル1210の情報を利用して、エディタを表示する前にリクエストされたエディタへのアクセス権限を決定する。タスク割当てエディタは、DefaultTasks1212、TaskAssignment1202、TaskAssignmentHistory1218、TopLevelProjectTask1204及びMemberTasks1208の各テーブルの情報を

50

利用及び／又は更新する。プロジェクトスケジュールエディタは、DefaultTasks1212、TaskAssignment1202、TopLevelProjectTask1204、MemberTasks1208及びLevel1MemberTask1206の各テーブルの情報を利用及び／又は更新する。メンバースケジュールエディタは、ProjectTeam1210、TaskAssignment1202、TopLevelProjectTask1204、MemberTasks1208及びLevelXMemberTaskの情報を利用及び／又は更新する。

【0061】

図12に示され、ここに記載されるプロジェクトスケジュール管理システムの実施例に使用される各種テーブルが以下に示される。しかしながら、図12を参照して説明されたテーブルの個数及び構成は、実現形態に応じて可変的なものであるかもしれない。

【0062】

DefaultTasksテーブル1212：このテーブルは、典型的にはすべてのプロジェクトのタスクであるタスクの名称を含む。ソフトウェア開発プロジェクトに関して、デフォルトタスクの一部の例として、ProjectPlans、Requirements及びTopLevelDesignがあげられる。

【0063】

ProjectTeamテーブル1210：このテーブルは、プロジェクトのプロジェクトメンバーに関する情報を含む。sMemberLabelは、プロジェクトスケジュールを表示する際にプロジェクト番号を特定するのに使用される2～4文字の文字列であり、図1A及び1Bに示されるようなプロジェクトタスクと関連するメンバータスクを示す。一実施例では、プロジェクトメンバーのイニシャルが、sMemberLabelに使用される。nMemberRoleは、プロジェクトメンバーの役割を示すのに使用される番号である。例えば、プロジェクトマネージャは1であり、プロジェクトリーダーは2であり、プロジェクトアドミニストレータは3であり、プロジェクトメンバーは4である。その役割は、誰がエディタにアクセスできるか示すことである。例えば、役割番号が1であるプロジェクトメンバーは、すべてのエディタにアクセス可能である。しかしながら、役割番号が4であるプロジェクトメンバーは、メンバーのスケジュールエディタにのみアクセス可能である。システムは、何れの役割番号が各エディタにアクセス可能であるか決定するよう構成可能である。sMemberDirectoryは、プロジェクトチームがメンバーのスケジュールを閲覧できるように、メンバースケジュールのHTMLファイルが格納される場所を決定するのに使用される。

【0064】

nMemberTaskIDは、当該メンバーのタスクのIDを決定するため使用されるプロジェクトメンバーに割り当てられた番号である。一実施例によると、nMemberTaskIDは、タスクのスタートIDとして使用される。プロジェクトチームのサイズに応じて、IDは小さなチームにはMOD10(1, 2, . . . , 9)とされ、大きなチームにはMOD100(1, 2, . . . , 99)又はそれ以上とすることができる。タスクIDは、MODのインクリメントである。例えば、プロジェクトメンバー“test1”のnMemberTaskIDが1である場合、test1のタスクのタスクIDは11, 21, 31, . . . (大きなチームについては、101, 201, 301, . . .)となるであろう。タスクIDは、タスクの一部の名前が同じであったとしても、プロジェクトメンバーのタスクを一意的に特定する。タスクIDはまた、すべてのレベルにおいてタスクを一意的に特定する。nLevelXMaxTaskIDは、プロジェクトメンバーの異なるレベルのタスクについてこれまで使用されてきた最も高いタスクIDを維持するため使用される番号である。これらの番号は、メンバーのスケジュールエディタセッションにおいて追加されるタスクのタスクIDを決定するのに使用されるスタートIDを提供する。これらの値は、各エディタセッションの後に抽出及び更新される。nLevelXMaxTaskIDの値を除いて、他のエントリの値はプロジェクトの開始前に設定される必要がある。

10

20

30

40

50

【 0 0 6 5 】

TaskAssignmentテーブル1202：このテーブルは、プロジェクトのプロジェクトメンバーに割り当てられるプロジェクトタスクとそれのサブタスクに関する情報を含む。sTaskNameは、タスクの名称について使用され、nProjectTaskIDは、タスクに係るIDである。プロジェクトスタートタスクIDは0であり、このため、そのタスクのIDはMOD（小さなチームについては、10, 20, 30, . . .）のインクリメントとなる。sLevelTaskNameは、プロジェクトタスクに係るサブタスク（メンバータスク）の名称に使用され、nLevel1TaskIDは、サブタスクに係るIDについて使用される。sMemberLabelは、サブタスクに割り当てられたプロジェクトメンバーを特定するのに使用される。bIsObsoleteは、タスクがプロジェクトから取り除かれたか示すのに使用される。タスクがスケジュールから削除されても、当該タスクに関する情報はデータベースに維持される。sTaskName、nProjectTaskID、sLevel1TaskName及びsMemberLabelの各値が、タスク割当てエディタセッションを通じてTaskAssignmentテーブル1202に追加可能である。プロジェクトスケジュールエディタセッションは、sTaskName及びnProjectTaskIDの値を追加可能である。メンバースケジュールエディタセッションのみが、nLevel1TaskIDの値を追加可能である。

10

【 0 0 6 6 】

TopLevelProjectTaskテーブル1204：このテーブルは、プロジェクトタスクのスケジューリングに関する情報を有する。sTaskNameは、タスクの名称について使用され、nProjectTaskIDは、タスクに係るIDについて使用される。planStart及びplanEndは、タスクの予想される開始日及び終了日について使用される。actualStart及びactualEndは、タスクが実際の開始及び終了した日について使用される。setDateは、プランされた日付が設定又は変更された日付について使用される。nScheduleRevNumberは、タスクスケジュールの修正回数について使用される。プロジェクトタスクの直近のスケジュールは、最も大きな修正回数に対応する。プロジェクトタスクのスケジュールの変更を追跡するため、プロジェクトタスクのすべての修正がTopLevelProjectTaskテーブル1204に維持される。プランされた日付が異なる日にプロジェクトスケジュールエディタにおいて変更されるときに限って、この修正はインクリメントされる。nProjectTaskID、sTaskName、日付及びnScheduleRevNumberのすべての値は、プロジェクトスケジュールエディタセッション又はタスク割当てエディタセッションを通じてTopLevelProjectTaskテーブル1204において追加又は更新される。

20

30

【 0 0 6 7 】

MemberTasksテーブル1208：このテーブルは、すべてのプロジェクトメンバーのすべてのタスク（すべてのレベルにおけるタスク）に関する情報を有する。プロジェクトの各メンバー（sMemberName）には、タスクIDとnLevelXTaskIDが関連付けられ、すべてのタスクと互いの関係とを特定する。TaskAssignmentテーブルに関して、bIsObsoleteは、タスクがプロジェクトメンバーのスケジュールから削除されているか示す。bIsCompletedは、タスクが終了したか示す。nLevelXTaskIDは、MemberTasksテーブル1208に追加され、新たなタスクがメンバーのスケジュールエディタセッションに追加されるとき、ProjectTeamテーブル1210のnLevelXMaxTaskIDから決定されるタスクについて使用される。このテーブルの各値は、3つのエディタセッション（メンバースケジュール、プロジェクトスケジュール、タスク割当て）の何れかの結果から更新又は変更（bIsObsolete又はbIsCompleted）することが可能である。MemberTasksテーブル1208は、下位レベルのタスクスケジュールの間のリンクを上位レベルのタスクスケジュールに提供するため重要であ

40

50

る。

【0068】

LevelXMemberTaskテーブル(Level1MemberTaskテーブル1206、Level2MemberTaskテーブル1214、Level3MemberTaskテーブル1216)：このテーブルは、メンバータスクのスケジューリングに関する情報を有する。sLevelXTaskNameは、タスクの名称について使用され、nLevelXTaskIDは、タスクに係るIDについて使用される。テーブルに追加されるタスクのnLevelXTaskIDは、新たなタスクがメンバーのスケジュールエディタセッションに追加されると、ProjectTeamテーブル1210のnLevelXMaxTaskIDから決定される。planStart及びplanEndは、タスクの予想される開始日及び終了日について使用される。actualStart及びactualEndは、タスクの実際の開始日及び終了日について使用される。setDateは、プランされた日付が設定又は変更された日付について使用される。メンバータスクの直近のスケジュールは、最も大きな修正番号に対応する。メンバータスクのすべての修正は、メンバータスクのスケジュールの変更を追跡するため、テーブルにおいて維持される。一実施例によると、プランされた日付がメンバースケジュールエディタにおいて変更されるときに限って、修正はインクリメントされる。各LevelXMemberTaskテーブルは、上位レベルタスクのタスクIDを含む(タスクがそのペアレントとしてのプロジェクトタスクを有するか、又はペアレントタスクを有しないレベル1を除いて)。これは、タスクに対してそのペアレントタスクとそのチャイルドタスクへのリンクを提供する。ペアレントタスクID、sLevelXTaskName、nLevelXTaskID、日付及びnScheduleRevNumberのすべての値は、メンバースケジュールエディタセッションを通じてテーブルにおいて追加又は更新される。Level1MemberTaskテーブル1206のみが、TaskAssignmentテーブル1202とのリンクを提供するため、sMemberLabelを有する。

10

20

【0069】

データベースは、レベル4までの下位レベルしか表示しない。しかしながら、データベースは、タスクスケジュールのさらなる詳細のため、下位レベルを含むよう変更可能である。

30

【0070】

TaskAssignmentHistoryテーブル1218：このテーブルは、プロジェクトタスクに係るタスクのプロジェクトメンバーに対する割当ての履歴に関する情報を有する。このテーブルは、タスクが他のプロジェクトメンバーに再割り当てされる前に、当該タスクに以前に割り当てられたプロジェクトメンバーに関する情報を維持する。nProjectTaskIDは、タスクに係るIDである。sLevel1TaskNameは、プロジェクトに係るサブタスク(メンバータスク)の名称である。sMemberLabelは、サブタスクに割り当てられるプロジェクトメンバーである。nRevNumberは、タスクのプロジェクトメンバーへの割当ての修正回数である。nRevNumberは、プロジェクトにおけるタスクの再割当てを示す。タスク割当てエディタ102(図1A)は、TaskAssignmentHistoryテーブル1218の情報を利用及び/又は更新する。

40

【0071】

[サーバのためのプログラミングパッケージ図]

図13は、本発明の実施例による図6のサーバプロセッサ604のプログラミングパッケージ図を示す。サーバプロセッサ1300は4つのパッケージを有し、各パッケージは、クライアントプロセッサ602上でユーザに表示されるウェブページ/エディタに対応し、それを介して、ユーザにより入力された情報が、ユーザがログイン又はエディタセッションを終了すると処理される。

【0072】

50

Log inProcessor1302パッケージは、プロジェクトメンバーがエディタの1つにログインすることを可能にするフォームを表示するためのウェブページを提供する。メンバーが当該フォームを送信すると、Log inProcessor1302パッケージは、当該情報を検証するため、メンバーにより入力された情報を処理する。情報が有効であり、メンバーが適切なアクセス権限を有する場合、Log inProcessor1302パッケージは、システムをエディタに対応するパッケージの1つにリダイレクトする。

【0073】

TaskAssignmentProcessor1304パッケージは、プロジェクトメンバーへのプロジェクトタスクの割当てを追加又は変更するのに使用されるタスク割当てエディタ102（図1A）を表示するため、ウェブページを提供する。タスク割当てエディタ102が送信されると、TaskAssignmentProcessor1304パッケージは、タスク割当てエディタ102からの情報を処理及び格納し、直近のタスク割当てのウェブページを作成する。

【0074】

ProjectScheduleProcessor1306パッケージは、プロジェクトタスクのスケジュールを追加又は変更するのに使用されるプロジェクトスケジュールエディタ202（図2A）を表示するため、ウェブページを提供する。プロジェクトスケジュールエディタ202が送信されると、ProjectScheduleProcessor1306パッケージは、プロジェクトスケジュールエディタ202からの情報を処理及び格納し、直近のプロジェクトスケジュールのウェブページを作成する。

【0075】

MemberScheduleProcessor1308パッケージは、メンバータスクのスケジュールを追加又は変更するのに使用されるメンバースケジュールエディタ302（図3A）を表示するため、ウェブページを提供する。メンバースケジュールエディタ302が送信されると、MemberScheduleProcessor1308パッケージは、メンバースケジュールエディタ302からの情報を処理及び格納し、直近のメンバースケジュールのウェブページを作成する。

【0076】

Log inProcessor1302パッケージのエディタパッケージへのリダイレクトを除いて、プロセッサパッケージは互いに独立し、一般には、エディタパッケージの間のやりとりはない。各プロセッサパッケージ1302～1308は、情報を取得、追加又は更新するため、データベースとやりとりする。Log inProcessor1302パッケージは、メンバーがアクセス権限を有しているか判断するため、データベース1310にアクセスする。その他のプロセッサパッケージ1304～1308のそれぞれは、対応するエディタ及びそれが生成する対応するウェブページに表示すべきタスク情報を取得し、対応するタスク情報を追加又は更新するため、データベース1310にアクセスする。非限定的な具体例として、データベース1310は、MySQLを利用して実現されてもよいが、データベース1310はMySQLを利用した実現形態に限定されるものでない。

【0077】

実施例によると、各エディタプロセッサ1304～1308パッケージは、PHPスクリプトファイル、JavaScriptファイル及びHTMLファイルから構成される。PHPスクリプトファイルは、データベース1310からプロジェクト及びタスク情報を取得し、クライアントプロセッサ602（図6）上にエディタを表示するJavaScriptを生成する。これは、PHPスクリプトがJavaScriptとやりとりすることを可能にする。JavaScriptは、エディタを生成し、エディタとプロジェクトメンバーとの間のすべてのやりとりを管理する。エディタが送信されると、PHPスクリプトファイルは、エディタの情報を処理し、当該情報をデータベース1310において追加又は更新し、エディタに対応するウェブページを生成する。

10

20

30

40

50

【 0 0 7 8 】

図 1 4 は、本発明の実施例によるエディタプロセッサ 1 3 0 4 ~ 1 3 0 8 のプログラミングパッケージ図を示す。実施例によると、Task Assignment Processor 1 3 0 4、Project Schedule Processor 1 3 0 6 及び Member Schedule Processor 1 3 0 8 パッケージのすべては、図 1 4 に示されるパッケージ図を利用する。パッケージは主要な 2 つの部分に分割され、表示エディタ 1 4 0 2 は、エディタ 1 4 0 4 からの提示情報及びエディタの表示及び管理を行い、エディタ 1 4 0 4 は、エディタに情報を提示し、ウェブページを生成する。

【 0 0 7 9 】

XXX 1 4 0 6 (ただし、“XXX”は、Task Assignment、Project Schedule 又は Member Schedule を表す)は、エディタを表示するため、以下のパッケージを一体化する。ウェブページ 1 4 0 6 は、エディタを表示及び管理するための XXXX PHP Processor 1 4 0 8 パッケージのすべての PHP スクリプトファイルと、XXX JavaScript Processor 1 4 1 0 パッケージのすべての JavaScript ファイルとを有する。データベースからタスク情報を所得し、XXX JavaScript Processor 1 4 1 0 パッケージとやりとりする JavaScript を生成するため、ウェブサーバ (図 5 のウェブサーバ 5 0 7、5 3 0 など) 上で処理される。エディタの初期表示を提供するため、すべての JavaScript がクライアントプロセッサ 6 0 2 (図 6) のウェブブラウザにおいて実行される。すべての JavaScript ファイルは、エディタを管理するため、すなわち、対応するすべての編集イベントを処理するため、クライアントプロセッサ 6 0 2 のウェブブラウザにわたされる。

【 0 0 8 0 】

Post XXX 1 4 1 2 のウェブページは、情報を提示し、提示ウェブページを生成する以下のパッケージを一体化する。Post XXX 1 4 1 2 のウェブページは、エディタからの情報を提示するための Post XXX PHP Processor 1 4 1 4 パッケージのすべての PHP スクリプトファイルと、ウェブページを生成するための XXX Web Page Generator 1 4 1 6 パッケージのすべての PHP スクリプトファイルとを有する。Post XXX PHP Processor 1 4 1 4 パッケージは、エディタからすべてのタスク情報を取得し、データベースのタスク情報を追加又は更新する。XXX Web Page Generator 1 4 1 6 パッケージは、適切なウェブページを生成するため、データベースからタスク情報を取得する。

【 0 0 8 1 】

図 1 4 の各パッケージは、パッケージのインタフェースを提供し、パッケージ内部のクラスを管理するクラスを提供する。これは、パッケージ内のデザインがその他のパッケージに影響を与えることなく容易に変更されることを可能にする。

【 0 0 8 2 】

[メンバースケジュールプロセッサパッケージ]

図 1 5 ~ 1 8 は、メンバースケジュールエディタ 3 0 2 (図 3) に対応する図 1 3 の Member Schedule Processor 1 3 0 8 パッケージに対応する図 1 4 のパッケージのクラス図を示す。これらの図は、表示エディタ 1 4 0 2 の 4 つのパッケージとエディタ 1 4 0 4 からの掲示情報に対応するクラス構成を示す。XXX PHP Processor 1 4 0 8 (図 1 4) パッケージは、データベースからタスク割当て / スケジュール情報を取得し、サーバプロセッサ 6 0 4 (図 6) におけるエディタの初期表示のためのコードを生成する。XXX JavaScript Processor 1 4 1 0 (図 1 4) パッケージは、クライアントプロセッサ 6 0 2 (図 6) のエディタを表示、管理及び維持する。Post XXX PHP Processor 1 4 1 4 (図 1 4) パッケージは、クライアントプロセッサ 6 0 2 のエディタセッションからのすべてのタスク割当て / スケジュール情報をサーバプロセッサ 6 0 4 のデータベースに提供する。XXX Web Page Generator 1 4 1 6 (図 1 4) パッケージは、タスク情報を表示する適切なウェブ

ブページを生成するため、サーバプロセッサ604のデータベースからタスク割当てノスケジュール情報を取得する。これらの図は、3つのエディタの間の4つのパッケージのデザインの類似性を示している。これらのエディタは異なるタスクを実行するが、すべてのエディタが類似したデザインパターンに従っている。

【0083】

図15は、本発明の実施例によるMemberSchedulePHPProcessorパッケージ1500(図14のXXXPHPProcessor1408など)のクラス図を示す。MemberSchedulePHPProcessorパッケージ1500は、クライアントプロセッサ602(図6)のウェブブラウザに初期メンバースケジュールエディタ302(図3A)を表示するJavaScriptインタフェースを生成する。

10

【0084】

CMemberScheduleManager1502クラスは、MemberSchedulePHPProcessorパッケージ1500のためのインタフェースを提供し、JavaScriptを生成するため、MemberSchedulePHPProcessorパッケージ1500のクラスを管理する。CMemberScheduleInitialData1504クラスは、エディタにおいて初期データを設定するためのJavaScriptを生成する。初期データは、スケジュールに追加可能なプロジェクトメンバーに割り当てられるメンバータスクである。CMemberScheduleTaskRowData1506クラスは、以前のエディタセッションにおいてメンバーのスケジュールに追加されたメンバータスクの各行を表示するためのJavaScriptを生成する。CJavaScriptInterface1508クラスは、ウェブブラウザに初期エディタを生成し、図16のMemberScheduleJavaScriptProcessor1600パッケージとインタフェースをとるJavaScriptのシーケンスを生成する。CEditorDBInterface1510クラスは、エディタに表示されるデータベースからの情報にアクセスする。CEditorDBInterface1510は、所望される表示用の情報を取得するため、適切なデータベースクエリを生成する。CEditorDBInterface1510は、データベースにアクセスするため、CScheduleDB1512とインタフェースをとる。CMemberScheduleInitialData1504及びCMemberScheduleTaskRowData1506は、CEditorDBInterface1510を介しデータベースからタスク情報を取得する。一実施例によると、MemberSchedulePHPProcessorパッケージの上記クラスは、PHPスクリプトにより実現される。

20

30

【0085】

図16は、本発明の実施例によるMemberScheduleJavaScriptProcessorパッケージ1600(図4のXXXJavaScriptProcessor1410など)のクラス図を示す。MemberScheduleJavaScriptProcessorパッケージ1600は、クライアントプロセッサ602(図6)のウェブブラウザにおいてメンバースケジュールエディタ302(図3A)を管理する。

40

【0086】

CEditorManager1602クラスは、当該パッケージのインタフェースを提供し、メンバースケジュールエディタ302のウェブページ及びフォームを生成する。CTableManager1604クラスは、メンバースケジュールエディタ302のテーブルを生成し、当該テーブルに影響を与えるすべてのイベントを管理する。CMemberScheduleTable1606クラスは、メンバースケジュールエディタ302のテーブルを初期化及び管理し、テーブルの各行を生成及び管理する。CMemberScheduleRow1608クラスは、メンバースケジュールエディタ302のテーブルの行を初期化及び管理し、当該行に影響を与えるすべてのイベントを管理し、当該

50

行のセルを生成及び管理する。CTaskCell1610クラスは、行のタスクセルを初期化及び管理し、タスクに関する情報を維持する。CDateCell1612クラスは、行の日付セルを初期化及び管理し、タスクのスケジュールに関する情報を維持する。SmemberTaskInfo1614の構成は、エディタにおいてタスクを表示するため、メンバータスク情報がMemberSchedulePHPProcessor1500パッケージからMemberScheduleJavaScriptProcessor1600パッケージにわたされることを可能にする。CDetailTaskInfo1616クラスは、タスクの詳細タスクに関する情報を格納及び維持し、そのサブタスクによりタスクのスケジュールを更新するのに使用される。CDateCell1612は、データセルにより月、日及び年メニュー選択を表示するため、CDateSelector1618を有する。一実施例によると、MemberScheduleJavaScriptProcessor1600パッケージの上記クラス及び構成のすべてが、JavaScriptにより実現される。

10

20

30

40

50

【0087】

図17は、本発明の実施例によるPostMemberSchedulePHPProcessorパッケージ1700(図14のPostXXXPHPProcessor1414など)のクラス図を示す。CMemberTaskManager1702クラスは、当該パッケージのためのインタフェースを提供し、パッケージの他のすべてのクラスを管理する。CMemberTaskManager1702は、エディタからの各タスクに対して実行されるアクションを決定する。CMemberTaskUpdater1704クラスは、データベースのタスクのスケジュールを更新する。この更新は、予定日の編集、実際の日付の更新、タスクの削除及び新たなタスクの追加を含む。CPostDBInterface1706クラスは、情報を取得し、データベースの情報を更新するため、クラスに対するインタフェースを提供する。CPostDBQueryGenerator1708クラスは、CPostDBInterface1706についてSQLデータベースクエリを生成する。CPostDBInterface1706は、データベースにアクセスするため、CScheduleDB1710とのインタフェースをとる。CMemberTaskUpdater1704は、CPostDBInterface1706を介しデータベースのタスク情報を更新する。実施例によると、PostMemberSchedulePHPProcessorパッケージ1700の上記クラスは、PHPスクリプトにより実現される。

【0088】

図18は、本発明の実施例によるMemberScheduleWebPageGeneratorパッケージ1800(図14のXXXWebPageProcessor1416など)のクラス図を示す。CMemberScheduleWebPageManager1802クラスは、当該パッケージがメンバースケジュールウェブページを生成するためのインタフェースを提供する。CMemberScheduleTable1804は、メンバースケジュールウェブページのテーブルを生成する。CProjectTaskRow1806は、テーブル内にタスク行を生成する。CWebPageDBInterface1808クラスは、クラスがデータベースにおいて情報を取得するためのインタフェースを提供する。CWebPageDBQueryGenerator1810クラスは、CWebPageDBInterface1808についてSQLデータベースクエリを生成する。CWebPageDBInterface1808は、データベースにアクセスするため、CScheduleDB1812とインタフェースをとる。CMemberScheduleTable1804及びCProjectTaskRow1806は、CWebPageDBInterface1808を介しデータベースからタスク情報を取得する。実施例によると、MemberScheduleWebPageGenerator1800パッケージの上記クラスは、PHPスクリプトにより実現される。

【0089】

テーブル 1 は、本発明の実施例によるメンバースケジュールエディタ 302 (図 3) の文書オブジェクトモデル表現を示す。テーブル 1 は、メンバースケジュールエディタ 302 を構成する各要素と対応する要素名及び id プロパティとを示す。これらの要素の一部は、テーブル要素、行要素、セル要素、チェックボックス入力要素、入力テキスト要素及び選択要素など、エディタに表示されるエディタの各部分に対応する。隠れた入力要素など、これらの要素の一部は、情報を表示するのではなく、格納するのに使用される。情報を格納し、又はユーザから情報を受け取る要素は、エディタセッションからのタスク情報を掲示するため、情報をサーバプロセッサにわたすのに重要である。ドキュメント・オブジェクト・モデル (DOM) については、O'Reilly & Associates, Inc. により刊行されている David Flanagan による "JavaScript: the Definitive Guide" (Fourth Edition) に記載されている。

10

【0090】

エディタを構成する各要素は、その id を介しアクセス可能であり、これらの要素のプロパティは当該要素の値及び / 又は表示を変更するよう設定可能である。実施例によると、メンバースケジュールエディタ 302 の各要素について、当該要素は図 16 の MemberScheduleJavaScriptProcessor 1600 パッケージのクラスの 1 つの内部にラップされる。これらの要素はクラスの属性である。このため、クラスのメンバー関数は、要素への直接的なアクセスを有し、必要に応じてそれらのプロパティを変更する。クラスが要素への直接的なアクセスを有することによって、それらの id を使用して要素を取得する必要はない。

20

【表 1】

テーブル1

| | | | | | |
|---|--|--|--|---|---|
| Form Element id = "MemberScheduleFormID" Table Element id = "MemberScheduleTableID" | | | | | |
| Row Element id = row_id + "_RowID" | | | | | |
| Task Cell Element id = row_id + "_TaskCellID" | Set Date Cell Element id = row_id + "_SetDateCellID" | Planned Start Date Cell Element id = row_id + "_PlanStartDateCellID" | Planned End Date Cell Element id = row_id + "_PlanEndDateCellID" | Actual Start Date Cell Element id = row_id + "_ActualStartDateCellID" | Actual End Date Cell Element id = row_id + "_ActualEndDateCellID" |
| CheckBox Element id = row_id + "_CheckBoxID" name = row_id + "_CheckBox" | Set Date Hidden Input Element id = row_id + "_HID_SetDateID" name = row_id + "_HID_SetDate" | Planned Start Date Hidden Input Element id = row_id + "_HID_PlanStartDateID" name = row_id + "_HID_PlanStartDate" | Planned End Date Hidden Input Element id = row_id + "_HID_PlanEndDateID" name = row_id + "_HID_PlanEndDate" | Actual Start Date Hidden Input Element id = row_id + "_HID_ActualStartDateID" name = row_id + "_HID_ActualStartDate" | Actual End Date Hidden Input Element id = row_id + "_HID_ActualEndDateID" name = row_id + "_HID_ActualEndDate" |
| Project Task Selection Element id = row_id + "_ProjectTaskSelectID" name = row_id + "_ProjectTaskSelect" | | Selection Element id = row_id + "_PlanStartMonthID" name = row_id + "_PlanStartMonth" | Selection Element id = row_id + "_PlanEndMonthID" name = row_id + "_PlanEndMonth" | Selection Element id = row_id + "_ActualStartMonthID" name = row_id + "_ActualStartMonth" | Selection Element id = row_id + "_ActualEndMonthID" name = row_id + "_ActualEndMonth" |
| Task Name Input Text Element id = row_id + "_TaskInputBoxID" name = row_id + "_TaskInputBox" | | Selection Element id = row_id + "_PlanStartDayID" name = row_id + "_PlanStartDay" | Selection Element id = row_id + "_PlanEndDayID" name = row_id + "_PlanEndDay" | Selection Element id = row_id + "_ActualStartDayID" name = row_id + "_ActualStartDay" | Selection Element id = row_id + "_ActualEndDayID" name = row_id + "_ActualEndDay" |
| Action On Task Hidden Input Element id = row_id + "_HID_ActionOnTaskID" name = row_id + "_HID_ActionOnTask" | | Selection Element id = row_id + "_PlanStartYearID" name = row_id + "_PlanStartYear" | Selection Element id = row_id + "_PlanEndYearID" name = row_id + "_PlanEndYear" | Selection Element id = row_id + "_ActualStartYearID" name = row_id + "_ActualStartYear" | Selection Element id = row_id + "_ActualEndYearID" name = row_id + "_ActualEndYear" |
| ID of Task Hidden Input Element id = row_id + "_HID_IDofTaskID" name = row_id + "_HID_IDofTask" | | | | | |
| ID of Parent Task Hidden Input Element id = row_id + "_HID_IDofParentTaskID" name = row_id + "_HID_IDofParentTask" | | | | | |
| Revision Number of Task Hidden Element id = row_id + "_HID_RevNumberID" name = row_id + "_HID_RevNumber" | | | | | |
| Name of Task Hidden Input Element id = row_id + "_HID_TaskNameID" name = row_id + "_HID_TaskName" | | | | | |
| Level of Task Hidden Input Element id = row_id + "_HID_TaskLevelID" name = row_id + "_HID_TaskLevel" | | | | | |
| Number of Detailed Task Hidden Input Element id = row_id + "_HID_NumOfDetailedTaskID" name = row_id + "_HID_NumOfDetailedTask" | | | | | |
| Number of Rows Menu Selection Element id = "AddRowSelectID" | | | | | |

【0091】

以下のテーブル2は、本発明の実施例による図16に示されるMemberScheduleJavaScriptProcessor1600パッケージのCTaskCell11610クラスの属性メンバーを示す。CTaskCell11610は、それが有する

10

20

30

40

50

すべての要素のプロパティの値を取得及び設定することが可能である。

【表 2】

テーブル 2

| Type | Attribute Name | Description |
|--------------------|-------------------------------|---|
| HTML.CellElement | m_TaskCellElement | This attribute member is an object for the cell element that contains task information |
| HTML.InputElement | m_TaskNameHiddenElement | This attribute member is an object for the hidden input element containing information about the task name. |
| HTML.InputElement | m_LevelOfTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the level of the task. |
| HTML.InputElement | m_NumOfDetailsHiddenElement | This attribute member is an object for the hidden input element containing information about the highest possible number of detail tasks the task currently has. A task can have from 0 to the value of the hidden element of detailed tasks. |
| HTML.InputElement | m_ActionOnTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the action taken on the task. |
| HTML.InputElement | m_IDOfTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the ID of the task. |
| HTML.InputElement | m_IDOfParentTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the task ID of its parent task. |
| HTML.InputElement | m_SelectedIndexHiddenElement | This attribute member is an object for the hidden input element containing information about the index of the selected task in the task select element. |
| HTML.InputElement | m_TaskNameInputElement | This attribute member is an object for the input element corresponding to an input text box that lets the project member input a task. |
| HTML.SelectElement | m_TaskNameSelectElement | This attribute member is an object for the select element that lets the project member select a project task to schedule. This element is initialized with unscheduled project tasks obtained from the database during the setup of the editor. |
| TextNode | m_TaskNameTextNode | This attribute member is an object for the text node that will display the task name in the task cell. |
| String | m_sRowID | This attribute member is a string for the row id of the row. |

10

20

30

40

50

【0092】

[プロジェクトスケジュールプロセッサパッケージ]

図19～22は、プロジェクトスケジュールエディタ202（図2A）に対応する図13のProjectScheduleProcessor1310パッケージに対応する図14のパッケージのクラス図を示す。これらの図は、表示エディタ1402の4つのパッケージとエディタ1404からの掲示情報に対応するクラス構成を示す。XXXPHPProcessor1408（図14）パッケージは、データベースからタスク情報/スケジュール情報を取得し、サーバプロセッサ604（図6）のエディタの初期表示のためのコードを生成する。XXXJavaScriptProcessor1410（図14）パッケージは、クライアントプロセッサ602（図6）のエディタを表示、管理及び維持する。PostXXXPHPProcessor1414（図14）パッケージは、クライアントプロセッサ602のエディタセッションからのすべてのタスク情報/スケジュール情報をサーバプロセッサ604のデータベースに提供する。XXXWebPageGenerator1416（図14）パッケージは、タスク情報を表示する適切なウェブページを生成するため、サーバプロセッサ604のデータベースからタスク割当て/スケジュール情報を取得する。これらの図は、3つのエディタの間の4つのパッケージのデザインに関する類似性を示している。エディタは異なるタスクを実行するが、それらはすべて類似したデザインパターンに従う。

【0093】

図19は、本発明の実施例によるProjectSchedulePHPProcessorパッケージ1900（図14のXXXPHPProcessor1408など）のクラス図を示す。ProjectSchedulePHPProcessorパッケージ1900は、クライアントプロセッサ602（図6）のウェブブラウザにおいて初期的なプロジェクトスケジュールエディタ202（図2A）を表示するJavaScriptパッケージを生成する。

【0094】

CProjectScheduleManager1902クラスは、ProjectSchedulePHPProcessorパッケージ1900のためのインタフェースを提供し、JavaScriptを生成するため、ProjectSchedulePHPProcessorパッケージ1900のクラスを管理する。CProjectScheduleInitialData1904クラスは、エディタの初期データを設定するためのJavaScriptを生成する。初期データは、プロジェクトスケジュールに追加可能なプロジェクトタスクである。CProjectScheduleTaskRowData1906クラスは、以前のエディタセッションにおいてメンバーのスケジュールに追加された対応するメンバータスクと共に、プロジェクトタスクの各行を表示するためのJavaScriptを生成する。CJavaScriptInterface1912クラスは、ウェブブラウザにおいて初期エディタを生成し、ProjectScheduleJavaScriptProcessor2000パッケージとインタフェースをとるJavaScriptのシーケンスを生成する。CProjectEditorDBInterface1908クラスは、エディタに表示されるデータベースからの情報にアクセスする。CProjectEditorDBQueryGenerator1910クラスは、CProjectEditorDBInterface1908についてSQLデータベースクエリを生成する。CProjectEditorDBInterface1908は、データベースにアクセスするためCScheduleDB1914とインタフェースをとる。CProjectScheduleInitialData1904及びCProjectScheduleTaskRowData1906は、CProjectEditorDBInterface1908を介しデータベースからタスク情報を取得する。実施例によると、ProjectSchedulePHPProcessor1900の上記クラスは、PHPスクリプトにより実現される。

10

20

【0095】

図20は、本発明の実施例によるProjectScheduleJavaScriptProcessorパッケージ2000(図14のPostXXXJavaScriptProcessor1410など)のクラス図を示す。ProjectScheduleJavaScriptProcessorパッケージ2000は、ブラウザにおいてプロジェクトスケジュールエディタ202(図2A)を管理する。CProjectEditorManager2002クラスは、当該パッケージのインタフェースを提供し、プロジェクトスケジュールエディタ202のウェブページ及びフォームを生成する。CProjectTableManager2004クラスは、プロジェクトスケジュールエディタ202のテーブルを生成、初期化及び管理し、当該テーブルに影響を与えるすべてのイベントを管理する。CProjectTableManager2004はまた、テーブルの各行を生成及び管理する。CProjectScheduleRow2006クラスは、プロジェクトスケジュールエディタ202のテーブルの行を初期化及び管理し、当該行に影響を与えるすべてのイベントを管理し、当該行のセルを生成及び管理する。CTaskCell2008クラスは、行のタスクセルを初期化及び生成する。CMemberCell2010クラスは、行のメンバーセルを初期化及び管理する。SProjectTaskInfo2014の構成は、プロジェクトスケジュールエディタ2092においてプロジェクトタスクとそれのメンバータスクスケジュールを表示するため、プロジェクト/メンバータスク情報をProjectSchedulePHPProcessor1900パッケージからProjectScheduleJavaScriptProcessor2000パッケージにわたすことを可能にする。CDateCell2012は、予定/実際の日付セルの月、日及び年メニュー選択を表示するため、CDateSelector2016を有する。実施例によると、ProjectScheduleJavaScriptProcessorパッケージ2000の上記クラス及び構成は、JavaScriptにより実現される。

30

40

【0096】

50

図21は、本発明の実施例によるPostProjectSchedulePHPProcessor2100パッケージ(図14のPostXXXPHPProcessor1414など)のクラス図を示す。CProjectTaskManager2102クラスは、当該パッケージのインタフェースを提供し、パッケージの他のすべてのクラスを管理する。CProjectTaskManager2102は、プロジェクトスケジュールエディタ202からの各プロジェクトタスクに対して実行されるアクションを決定する。CProjectTaskUpdater2104クラスは、データベースのプロジェクトタスクのスケジュールを更新する。この更新は、プロジェクトタスクのスケジュールの追加又は更新を含む。CProjectTaskUpdater2104クラスは、メンバーのタスクによりプロジェクトタスクを併合し、データベースにおいてプロジェクトタスクを更新する。CPostProjectDBInterface2106は、クラスがデータベースにおける情報を取得及び更新するためのインタフェースを提供する。CPostProjectDBQueryGenerator2108クラスは、CPostProjectDBInterface2106についてSQLデータベースクエリを生成する。CPostProjectDBInterface2106は、データベースにアクセスするため、CScheduleDB2110とインタフェースをとる。CProjectTaskUpdater2104は、CPostProjectDBInterface2106を介しデータベースのタスク情報を更新する。実施例によると、PostProjectSchedulePHPProcessor2100パッケージの上記クラスは、PHPスクリプトにより実現される。

10

20

【0097】

図22は、本発明の実施例によるProjectScheduleWebPageGenerator2200パッケージ(図14のXXXWebPageProcessor1416など)のクラス図を示す。CProjectScheduleWebPageManager2202クラスは、当該パッケージがプロジェクトスケジュールウェブページを生成するためのインタフェースを提供する。CProjectScheduleTable2204は、プロジェクトスケジュールウェブページのテーブルを生成する。CProjectTaskRow2206クラスは、テーブル内にプロジェクト及びメンバータスク行を生成する。CProjectWebPageDBInterface2208クラスは、クラスがデータベースから情報を取得するためのインタフェースを提供する。CProjectWebPageDBQueryGenerator2210クラスは、CProjectWebPageDBInterface2208についてSQLデータベースクエリを生成する。CProjectWebPageDBInterface2208は、データベースにアクセスするため、CScheduleDB2212とインタフェースをとる。CProjectScheduleTable2204及びCProjectTaskRow2206は、CProjectWebPageDBInterface2208を介しデータベースからタスク情報を取得する。実施例によると、ProjectScheduleWebPageGenerator2200パッケージの上記クラスは、PHPスクリプトにより実現される。

30

40

【0098】

テーブル3は、本発明の実施例によるプロジェクトスケジュールエディタ202(図2A)のドキュメント・オブジェクト・モデル表現を示す。テーブル3は、プロジェクトスケジュールエディタ202を構成する各要素と、対応する要素名及びidプロパティとについて記載する。各要素は、そのidを介しアクセス可能であり、要素のプロパティは要素の値及び/又は表示を変更するよう設定可能である。実施例によると、プロジェクトスケジュールエディタ202の各要素について、当該要素は、図20のProjectScheduleJavaScriptProcessor2000パッケージのクラスの1つの内部にラップされる。これらの要素はクラスの属性である。このため、クラスのメンバーファンクションは、要素への直接的なアクセスを有し、必要に応じてそれらのプロパティを変更する。クラスが要素への直接的なアクセスを有することによって、それらの

50

id を利用して要素を取得する必要はない。

【表 3】

テーブル3

| | | | | | |
|---|--|--|--|---|---|
| Form Element id = "ProjectScheduleFormID" | | | | | |
| Table Element id = "ProjectScheduleTableID" | | | | | |
| Row Element id = row_id + "_RowID" | | | | | |
| Task Cell Element id = row_id + "_TaskCellID" | Set Date Cell Element id = row_id + "_SetDateCellID" | Planned Start Date Cell Element id = row_id + "_PlanStartDateCellID" | Planned End Date Cell Element id = row_id + "_PlanEndDateCellID" | Actual Start Date Cell Element id = row_id + "_ActualStartDateCellID" | Actual End Date Cell Element id = row_id + "_ActualEndDateCellID" |
| CheckBox Element id = row_id + "_CheckBoxID" name = row_id + "_CheckBox" | Set Date Hidden Input Element id = row_id + "_HID_SetDateID" name = row_id + "_HID_SetDate" | Planned Start Date Hidden Input Element id = row_id + "_HID_PlanStartDateID" name = row_id + "_HID_PlanStartDate" | Planned End Date Hidden Input Element id = row_id + "_HID_PlanEndDateID" name = row_id + "_HID_PlanEndDate" | Actual Start Date Hidden Input Element id = row_id + "_HID_ActualStartDateID" name = row_id + "_HID_ActualStartDate" | Actual End Date Hidden Input Element id = row_id + "_HID_ActualEndDateID" name = row_id + "_HID_ActualEndDate" |
| Project Task Selection Element id = row_id + "_ProjectTaskSelectID" name = row_id + "_ProjectTaskSelect" | | Selection Element id = row_id + "_PlanStartMonthID" name = row_id + "_PlanStartMonth" | Selection Element id = row_id + "_PlanEndMonthID" name = row_id + "_PlanEndMonth" | | |
| Task Name Input Text Element id = row_id + "_TaskInputBoxID" name = row_id + "_TaskInputBox" | | Selection Element id = row_id + "_PlanStartDayID" name = row_id + "_PlanStartDay" | Selection Element id = row_id + "_PlanEndDayID" name = row_id + "_PlanEndDay" | | |
| Action On Task Hidden Input Element id = row_id + "_HID_ActionOnTaskID" name = row_id + "_HID_ActionOnTask" | | Selection Element id = row_id + "_PlanStartYearID" name = row_id + "_PlanStartYear" | Selection Element id = row_id + "_PlanEndYearID" name = row_id + "_PlanEndYear" | | |
| ID of Task Hidden Input Element id = row_id + "_HID_IDofTaskID" name = row_id + "_HID_IDofTask" | | | | | |
| Name of Task Hidden Input Element id = row_id + "_HID_TaskNameID" name = row_id + "_HID_TaskName" | | | | | |
| Number of Detailed Task Hidden Input Element id = row_id + "_HID_NumOfDetailedTaskID" name = row_id + "_HID_NumOfDetailedTask" | | | | | |
| Member Label Cell Element id = row_id + "_MemberLabelCellID" | | | | | |
| Member Label Hidden Input Element id = row_id + "_HID_MemberLabelCellID" name = row_id + "_HID_MemberLabelCell" | | | | | |
| Number of Rows Menu Selection Element id = "AddRowSelectID" | | | | | |

10

20

30

40

【0099】

テーブル 4 は、本発明の実施例による図 20 に示される ProjectScheduleJavaScriptProcessor2000 パッケージの CTaskCell12008 クラスの属性メンバーを示す。CTaskCell12008 は、それが有するすべての要素のプロパティの値を取得及び設定可能である。

【表 4】

テーブル 4

| Type | Attribute Name | Description |
|-------------------|------------------------------|--|
| HTMLCellElement | m_TaskCellElement | This attribute member is an object for the cell element that contains task information |
| HTMLInputElement | m_TaskNameHiddenElement | This attribute member is an object for the hidden input element containing information about the task name. |
| HTMLInputElement | m_NumOfDetailsHiddenElement | This attribute member is an object for the hidden input element containing information about the number of member tasks the project task has. |
| HTMLInputElement | m_ActionOnTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the action taken on the task. |
| HTMLInputElement | m_IDOfTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the ID of the task. |
| HTMLInputElement | m_SelectedIndexHiddenElement | This attribute member is an object for the hidden input element containing information about the index of the selected task in the task select element. |
| HTMLInputElement | m_TaskNameInputElement | This attribute member is an object for the input element corresponding to an input text box that lets the project member input a task. |
| HTMLSelectElement | m_TaskNameSelectElement | This attribute member is an object for the select element that lets the project member select a project task to schedule. This element is initialized with unassigned project tasks obtained from the database during the setup of the editor. |
| String | m_sRowID | This attribute member is a string for the row id of the row. |

10

【 0 1 0 0 】

20

[タスク割当てプロセッサパッケージ]

図 2 3 ~ 2 6 は、タスク割当てエディタ 1 0 2 (図 1 A) に対応する図 1 3 の Task Assignment Processor パッケージに対応する図 1 4 のパッケージのクラス図を示す。これらの図は、表示エディタ 1 4 0 2 の 4 つのパッケージとエディタ 1 4 0 4 からの掲示情報に対応するクラス図を示す。XXX PHP Processor 1 4 0 8 (図 1 4) パッケージは、データベースからタスク割当て / スケジュール情報を取得し、サーバプロセッサ 6 0 4 (図 6) においてエディタの初期表示のためのコードを生成する。XXX JavaScript Processor 1 4 1 0 (図 1 4) パッケージは、クライアントプロセッサ 6 0 2 (図 6) においてエディタを表示、管理及び維持する。Post XXX PHP Processor 1 4 1 4 (図 1 4) パッケージは、クライアントプロセッサ 6 0 2 のエディタセッションからのすべてのタスク割当て / スケジュール情報をサーバプロセッサ 6 0 4 のデータベースに提供する。XXX Web Page Generator 1 4 1 6 (図 1 4) パッケージは、タスク情報を表示する適切なウェブページを生成するため、サーバプロセッサ 6 0 4 のデータベースからタスク割当て / スケジュール情報を取得する。これらの図は、3 つのエディタの間の 4 つのパッケージの構成の類似性を示している。エディタは異なるタスクを実行するが、それらはすべて同様のデザインパターンに従う。

30

【 0 1 0 1 】

図 2 3 は、本発明の実施例による Task Assignment PHP Processor 2 3 0 0 パッケージ (図 1 4 の XXX PHP Processor 1 4 0 8 など) のクラス図を示す。Task Assignment PHP Processor 2 3 0 0 パッケージは、クライアントプロセッサ 6 0 2 (図 6) のウェブブラウザに初期的なタスク割当てエディタ 1 0 2 (図 1 A) を表示する JavaScript インタフェースを生成する。

40

【 0 1 0 2 】

C Task Assignment Manager 2 3 0 2 クラスは、Task Assignment PHP Processor 2 3 0 0 パッケージのためのインタフェースを提供し、JavaScript を生成するため、パッケージのすべてのクラスを管理する。C Task Assignment Initial Data 2 3 0 4 クラスは、タスク割当てエディタ 1 0 2 (図 1 A) において初期データを設定するための JavaScript

50

を生成する。この初期データは、プロジェクトスケジュールに追加され、メンバーに割当て可能なプロジェクトタスクである。CTaskAssignmentTaskRowData2306クラスは、プロジェクトタスクの各行を、以前のエディタセッションにおいて割り当てられたタスクに割り当てられたメンバーとそれのメンバータスクと共に表示するためのJavaScriptを生成する。CTaskAssignmentJavaScriptInterface2310クラスは、ウェブブラウザにおいて初期的なタスク割当てエディタ102を生成し、TaskAssignmentJavaScriptProcessor2400パッケージとインタフェースをとるJavaScriptシーケンスを生成する。CTaskAssignmentEditorDBInterface2308は、エディタに表示されるデータベースからの情報にアクセスする。CTaskAssignmentEditorDBInterface2308は、所望される表示用の情報を取得するため、適切なクエリを生成する。CTaskAssignmentEditorDBInterface2308は、データベースにアクセスするため、CScheduleDB2314とインタフェースをとる。CTaskAssignmentInitialData2304及びCTaskAssignmentTaskRowData2306は、CTaskAssignmentEditorDBInterface2308を介しデータベースからタスク情報を取得する。実施例によると、TaskAssignmentPHPProcessor2300パッケージの上記クラスは、PHPスクリプトにより実現される。

10

20

【0103】

図24は、本発明の実施例によるTaskAssignmentJavaScriptProcessor2400パッケージ(図14のPostXXXJavaScriptProcessor1410など)のクラス図を示す。TaskAssignmentJavaScriptProcessor2400パッケージは、ブラウザにおけるタスク割当てエディタ102(図1A)を管理する。CTaskAssignmentEditorManager2402クラスは、当該パッケージのためのインタフェースを提供し、タスク割当てエディタ102についてウェブページ及びフォームを生成する。CTaskAssignmentTable2404クラスは、タスク割当てエディタ102のテーブルを生成、初期化及び管理し、当該テーブルに影響を与えるすべてのイベントを管理する。CTaskAssignmentTable2404はまた、テーブルの各行を生成及び管理する。CTaskAssignmentRow2406クラスは、タスク割当てエディタ102のテーブルの行を初期化及び管理し、当該行に影響を与えるすべてのイベントを管理し、当該行のセルを生成及び管理する。CTaskCell2408クラスは、行のタスクセルを初期化及び管理する。CAssignmentCell2410クラスは、行の割当てセルを初期化及び管理する。実施例によると、TaskAssignmentJavaScriptProcessor2400パッケージの上記クラス及び構成は、JavaScriptにより実現される。

30

【0104】

図25は、本発明の実施例によるPostTaskAssignmentPHPProcessor2500パッケージ(図14のPostXXXPHPProcessor1414など)のクラス図を示す。CTaskAssignmentManager2502クラスは、当該パッケージのためのインタフェースを提供し、パッケージの他のすべてのクラスを管理する。CTaskAssignmentManager2502は、タスク割当てエディタ102(図1A)からの各プロジェクトタスクに対して実行されるアクションを決定する。CTaskAssignmentUpdater2504クラスは、データベースにおけるプロジェクトタスクの割当てを更新する。この更新は、プロジェクトタスクの割当ての追加又は削除を含む。CPostTaskAssignmentDBInterface2508クラスは、クラスがデータベースにおいて情報を取得及び更新するためのインタフェースを提供する。CPostTaskAssignmentDBQueryGenerator2506クラスは、CPostTaskAssignment

40

50

ntDBInterface2508についてSQLデータベースクエリを生成する。CPostTaskAssignmentDBInterface2508は、データベースにアクセスするため、CScheduleDB2510とインタフェースをとる。CTaskAssignmentUpdater2504は、CPostTaskAssignmentDBInterface2508を介しデータベースにおけるタスク情報を更新する。実施例によると、PostTaskAssignmentPHPProcessor2500パッケージの上記クラスは、PHPスクリプトにより実現される。

【0105】

図26は、本発明の実施例によるTaskAssignmentWebPageGenerator2600パッケージ(図14のXXXWebPageProcessor1416など)のクラス図を示す。CTaskAssignmentWebPageManager2602クラスは、当該パッケージがタスク割当てウェブページを生成するためのインタフェースを提供する。CTaskAssignmentTable2604クラスは、タスク割当てウェブページについてテーブルを生成する。CTaskAssignmentWebPageGeneratorDBInterface2606クラスは、クラスがデータベースから情報を取得するためのインタフェースを提供する。CTaskAssignmentWebPageGeneratorDBInterface2606は、所望の情報を取得するため、適切なクエリを生成する。CTaskAssignmentWebPageGeneratorDBInterface2606は、データベースにアクセスするため、CScheduleDB2608とインタフェースをとる。CTaskAssignmentTable2604は、CTaskAssignmentWebPageGeneratorDBInterface2606を介しデータベースからタスク情報を取得する。実施例によると、TaskAssignmentWebPageGenerator2600パッケージの上記クラスは、PHPスクリプトにより実現される。

10

20

【0106】

テーブル5は、タスク割当てエディタ102(図1A)のドキュメント・オブジェクト・モデル表現を示す。テーブル5は、タスク割当てエディタ102を構成する各要素と、対応する要素名及びidプロパティとを記述する。各要素はそのidを介しアクセス可能であり、要素のプロパティは、当該要素の値及び/又は表示を変更するよう設定可能である。実施例によると、タスク割当てエディタ102の各要素について、当該要素は図24のTaskAssignmentJavaScriptProcessor2400パッケージのクラスの1つの内部にラップされる。各要素はクラスの属性である。このため、クラスのメンバー関数は、要素への直接的なアクセスを有し、必要に応じてそのプロパティを変更する。クラスが要素への直接的なアクセスを有することによって、それらのidを使用して各要素を取得する必要がなくなる。

30

【表 5】

テーブル5

| | |
|---|---|
| Form Element id = "TaskAssignmentFormID" Table Element id = "TaskAssignmentTableID" | |
| Row Element id = row_id + "_RowID" | |
| Task Cell Element id = row_id + "_TaskCellID" | Member Assignment Cell Element id = row_id + "_MemberAssignmentCellID" |
| CheckBox Element id = row_id + "_CheckBoxID" name = row_id + "_CheckBox" | Member Assignment Hidden Input Element id = row_id + "_HID_MemberAssignmentID" name = row_id + "_HID_MemberAssignment" |
| Project Task Selection Element id = row_id + "_ProjectTaskSelectID" name = row_id + "_ProjectTaskSelect" | Member Assignment Selection Element id = row_id + "_MemberAssignmentID" name = row_id + "_MemberAssignment" |
| Task Name Input Text Element id = row_id + "_TaskInputBoxID" name = row_id + "_TaskInputBox" | |
| Action On Task Hidden Input Element id = row_id + "_HID_ActionOnTaskID" name = row_id + "_HID_ActionOnTask" | |
| ID of Task Hidden Input Element id = row_id + "_HID_IDofTaskID" name = row_id + "_HID_IDofTask" | |
| ID of Parent Task Hidden Input Element id = row_id + "_HID_IDofParentTaskID" name = row_id + "_HID_IDofParentTask" | |
| Name of Task Hidden Input Element id = row_id + "_HID_TaskNameID" name = row_id + "_HID_TaskName" | |
| Number of Detailed Task Hidden Element id = row_id + "_HID_NumOfDetailedTaskID" name = row_id + "_HID_NumOfDetailedTask" | |
| Number of Rows Menu Selection Element id = "AddRowSelectID" | |

10

20

30

【 0 1 0 7 】

以下のテーブル6は、本発明の実施例による図24に示されるTaskAssignmentJavaScriptProcessorパッケージのCTaskCell12408クラスの属性メンバーを示す。CTaskCell12408は、それが有するすべての要素のプロパティの値を取得及び設定することが可能である。

【表 6】

テーブル 6

| Type | Attribute Name | Description |
|-------------------|-------------------------------|---|
| HTMLCellElement | m_TaskCellElement | This attribute member is an object for the cell element that contains task information |
| HTMLInputElement | m_TaskNameHiddenElement | This attribute member is an object for the hidden input element containing information about the task name. |
| HTMLInputElement | m_LevelOfTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the level of the task. |
| HTMLInputElement | m_NumOfDetailsHiddenElement | This attribute member is an object for the hidden input element containing information about the highest possible number of detail tasks the task currently has. A task can have from 0 to the value of the hidden element of detailed tasks. |
| HTMLInputElement | m_ActionOnTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the action taken on the task. |
| HTMLInputElement | m_IDOfTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the ID of the task. |
| HTMLInputElement | m_IDOfParentTaskHiddenElement | This attribute member is an object for the hidden input element containing information about the task ID of its parent task. |
| HTMLInputElement | m_SelectedIndexHiddenElement | This attribute member is an object for the hidden input element containing information about the index of the selected task in the task select element. |
| HTMLInputElement | m_TaskNameInputElement | This attribute member is an object for the input element corresponding to an input text box that lets the project member input a task. |
| HTMLSelectElement | m_TaskNameSelectElement | This attribute member is an object for the select element that lets the project member select a project task to schedule. This element is initialized with unscheduled project tasks obtained from the database during the setup of the editor. |
| TextNode | m_TaskNameTextNode | This attribute member is an object for the text node that will display the task name in the task cell. |
| String | m_sRowID | This attribute member is a string for the row id of the row. |

10

20

【0108】

メンバースケジュールエディタ302（図3A）、プロジェクトスケジュールエディタ202（図2A）及びタスク割当てエディタ102（図1A）のそれぞれについてPHP Processor、JavaScript Processor、PostPHP Processor及びWebPage Generatorを記載した図15～26から理解されるように、各エディタのデザインは同様のパターンに従っている。このため、システムに追加される新たなエディタは、同様のデザインパターンに従うことができる。

30

【0109】

テーブル7は、デザインの類似性を確認するため、ここに記載されるオブジェクト指向スケジューリングシステムのソフトウェア設計仕様のインデックス化を示す。テーブル7は、各パッケージとパッケージ内の各クラスを列記し、3つのエディタの構成の類似性を示す。

【表 7】

テーブル 7

| | | |
|----------------------------|---|--|
| Common | PHP Common | CScheduleDB, DateUtility, DebugUtility, phpSystemConstants |
| | JavaScript Common | CDateSelector, EditorUtility, DateUtility, CalendarUtility, DebugUtility, JavaScriptSystemConstants |
| Login Processor | login.htm, PostLogin.htm, CPostLoginPHP, CProjectTeamAccess | |
| Task Assignment Processor | TaskAssignEditor.htm, PostTaskAssignment.htm | |
| | TaskAssignmentPHPProcessor | CTaskAssignmentManager, CTaskAssignmentInitialData, CTaskAssignmentTaskRowData, CTaskAssignmentJavaScriptInterface, CTaskAssignmentEditorDBInterface, TaskAssignmentPHPProcessorConstants |
| | TaskAssignmentJavaScriptProcessor | CTaskAssignmentEditorManager, CTaskAssignmentTable, CTaskAssignmentRow, CTaskCell, CAssignmentCell |
| | PostTaskAssignmentPHPProcessor | CTaskAssignmentManager, CTaskAssignmentUpdater, CPostTaskAssignmentDBInterface, CPostTaskAssignmentDBQueryGenerator, PostTaskAssignmentPHPProcessorConstants |
| | TaskAssignmentWebPageGenerator | CTaskAssignmentWebPageManager, CTaskAssignmentTable, CTaskAssignmentWebPageGeneratorDBInterface, AssignmentWebPageGeneratorConstants |
| Project Schedule Processor | ProjScheduleEditor.htm, PostProjSchedule.htm | |
| | ProjectSchedulePHPProcessor | CProjectScheduleManager, CProjectScheduleInitialData, CProjectScheduleTaskRowData, CProjectEditorDBInterface, CProjectEditorDBQueryGenerator, CJavaScriptInterface, ProjectSchedulePHPProcessorConstants |
| | ProjectScheduleJavaScriptProcessor | CProjectEditorManager, CProjectTableManager, CProjectScheduleRow, CTaskCell, CMemberCell, CDateCell, SProjectTaskInfo |
| | PostProjectSchedulePHPProcessor | CProjectTaskManager, CProjectTaskUpdater, CPostProjectDBInterface, CPostProjectDBQueryGenerator, PostProjectSchedulePHPProcessorConstants |
| | ProjectScheduleWebPageGenerator | CProjectScheduleWebPageManager, CProjectScheduleTable, CProjectTaskRow, CProjectWebPageDBInterface, CProjectWebPageDBQueryGenerator, ProjectScheduleWebPageGeneratorConstants |
| Member Schedule Processor | MembScheduleEditor.htm, PostMembSchedule.htm | |
| | MemberSchedulePHPProcessor | CMemberScheduleManager, CMemberScheduleInitialData, CMemberScheduleTaskRowData, CEditorDBInterface, CJavaScriptInterface, MemberSchedulePHPProcessorConstants |
| | MemberScheduleJavaScriptProcessor | CEditorManager, CTableManager, CMemberScheduleTable, CMemberScheduleRow, CTaskCell, CDateCell, CDetailTaskInfo, SMemberTaskInfo |
| | PostMemberSchedulePHPProcessor | CMemberTaskManager, CMemberTaskUpdater, CPostDBInterface, CPostDBQueryGenerator, PostMemberSchedulePHPProcessorConstants |
| | MemberScheduleWebPageGenerator | CMemberScheduleWebPageManager, CMemberScheduleTable, CProjectTaskRow, CWebPageDBInterface, CWebPageDBQueryGenerator, MemberScheduleWebPageGeneratorConstants |

10

20

30

【 0 1 1 0 】

[プレースホルダー文字列によるコンスタント文字列からのデータベースクエリの生成]

図 2 7 は、本発明の実施例によるデータベースクエリを生成するのに使用される一例となるコンスタント文字列を示す。2つのタイプのコンスタント文字列が使用される。“コンスタントクエリ文字列”は、所与のクエリの値と置換されるプレースホルダー文字列を有するクエリ文字列全体を含む。コンスタントクエリ文字列はクエリ全体を示し、プレースホルダー文字列は、何れの値がクエリに入力される必要があるかを示す。“プレースホルダー文字列に対するコンスタント”とは、コンスタントクエリ文字列のプレースホルダー文字列を検索し、実際の値と置換するのに使用される。クエリのプレースホルダー文字列は、クエリの結果を限定するための制限を適用する。図 2 7 に示される例は、PHPスクリプトに対応しているが、任意の言語により利用可能である。

40

【 0 1 1 1 】

プレースホルダー文字列を有するコンスタントクエリ文字列を使用することは、読解することが困難な文字列の連結の系列を介し文字列を構成することからの改良を提供する。データベースにアクセスするパッケージの各クラス図は、図 1 5、1 7、1 8、1 9、2 1、2 2、2 3、2 5 及び 2 6 に示されるように、パッケージ内で使用されるパッケージコンスタントを含む。コンスタントクエリ文字列は、それらが容易に特定できるように、パッケージ内で定義される。コンスタントクエリ文字列の他の効果は、Navicat MySQLなどのデータベースツールにおいてそれらをテストすることである。コンスタントクエリ文字列は、クエリ文字列が有効な文字列であるかテストするため、各値と置換

50

されたプレースホルダー文字列を有するこのようなツールにコピーすることができる。

【0112】

図28は、本発明の実施例による図27のコンスタント文字列からデータベースクエリを生成するのに使用される一例となるスクリプトを示す。図28に示される例は、PHPスクリプトに対応するが、当該シーケンスを実現するため任意の言語が利用可能である。上記例は、スクリプトの各ステートメントが実行された後のクエリ文字列の値を示す。第1ステートメントの実行では、コンスタント文字列は変数文字列\$loc_sQueryに割り当てられる。変数\$loc_sQueryは、データベースクエリに使用されるクエリを含む。第2、第3及び第4ステートメントの実行では、プレースホルダー文字列“%%ProjectNumber%%”、“%%MemberLabel%%”及び“%%ProjectTaskID%%”はそれぞれ、“J17”、“T1”及び“40”と置換される。第4ステップの実行は、結果として得られるクエリ文字列を示す。この例は、プロジェクト番号、メンバーラベル及びプロジェクトタスクidなどのシンプルな値によるプレースホルダーの置換を示す。エディタによるセッション中に変動しないプロジェクト番号やメンバーラベルなど、プレースホルダー文字列を置換するいくつかの値はスタティックなものとなる。例示されるクエリは、指定されたプロジェクト番号、メンバーラベル及びプロジェクトタスクidによるデータベースのテーブルの記録に制限される。

【0113】

図29は、本発明の実施例によるコンスタント文字列からクエリ文字列を生成するプロセスを示すフロー図である。ブロック2902において、コンスタントクエリ文字列が変数文字列に割り当てられる。変数文字列は、プレースホルダー文字列を各値に置換することを可能にするため必要とされ、コンスタント文字列の各値は変化しない。ブロック2904において、変数文字列は、それがプレースホルダー文字列を有するか確認するためチェックされる。変数文字列がさらなるプレースホルダー文字列を有しない場合、クエリ文字列はオリジナルのコンスタントクエリ文字列に対応し、本プロセスはブロック2906において終了する。変数文字列がさらなるプレースホルダー文字列を有する場合、ブロック2908において、変数文字列のプレースホルダー文字列はある値と置換される。ブロック2908の置換後、ブロック2904に移行し、変数文字列がさらなるプレースホルダー文字列を有しているか判断される。変数のすべてのプレースホルダー文字列が各値と置換されると、クエリ文字列が生成され、データベースへの送信準備が整う。クエリがデータベースに送信されると、データベースは、要求元に返され、他のプロセスにわたされ、又は必要に応じて処理することが可能な結果を生成する。

【0114】

CXXXDBInterfaceクラス(図17のCPostDBInterface1706クラス、図18のCWebPageDBInterface1808クラスなど)及びCYYYDBQueryGeneratorクラス(図17のCPostDBQueryGenerator1708クラス、図18のCWebPageDBQueryGenerator1810クラスなど)が、クエリを生成及び使用する。いくつかのケースでは、CXXXDBInterfaceクラスは、ユーザから、エディタを介し及びデータベースから取得した定数及び値からクエリ文字列を生成するプライベートファンクションを有する。一例として、図15のCEditorDBInterface1510があげられる。ほとんどの場合、CXXXDBInterfaceクラスは、クエリ文字列を生成するため、CYYYDBQueryGeneratorのパブリックファンクションを使用する。一例として、図17のCPostDBInterface1706及びCPostDBQueryGenerator1708があげられる。

【0115】

[エディタウェブページコンポーネント]

図30は、本発明の実施例によるエディタ(メンバースケジュールエディタ302、プロジェクトスケジュールエディタ202、タスク割当てエディタ102など)のウェブページの各コンポーネントを示す。ウェブページは、ウェブサーバなどのサーバプロセッサ

604 (図6) に格納されているファイルである。ウェブページは、JavaScriptコンポーネントとPHPコンポーネントとを有する。JavaScriptコンポーネントは、エディタにおいて発生するイベントを処理するJavaScriptファンクションを含む。JavaScriptコンポーネントは、エディタを表示、管理及び維持するためのクラス、ユーティリティ及びコンスタントに対応する他のJavaScriptファイルを含む。ウェブページのPHPコンポーネントは、データベースから取得されるタスク割当て/スケジュール情報をエディタに表示するJavaScriptコードの生成を開始するためのPHPスクリプトを含む。PHPコンポーネントは、データベースからタスク割当て/スケジュール情報を取得し、エディタに対するJavaScriptコードを生成するためのクラス、ユーティリティ及びコンスタントに対応するPHPスクリプトを有するファイルを含む。

10

【0116】

ウェブページがエディタウェブページなどをリクエストするウェブブラウザなどのクライアントプロセッサ602 (図6) によりリクエストされると、ウェブページのPHPコンポーネントのみがサーバプロセッサ604 により処理される。例えば、PHPスクリプトは、ウェブサーバ507 及び530 (図5) などのウェブサーバにおいて実行される。PHPスクリプトは、データベースからのタスク割当て/スケジュール情報にアクセス及び取得する。PHPスクリプトは、タスク情報を格納し、JavaScriptに転送するため、JavaScriptコードによる構成を生成する。PHPスクリプトは、エディタを生成、管理及び維持するJavaScriptクラスのオブジェクトを生成するJavaScriptコードを生成し、タスク情報を有するエディタの初期表示を生成するため、オブジェクトのメンバーファンクションを呼び出す。PHPスクリプトにより生成されるJavaScriptコードが、ウェブページがクライアントプロセッサ602 にわたされると、エディタウェブページに追加される。PHPコードは、それがクライアントプロセッサ602 にわたされるとき、ウェブページにはない。クライアントプロセッサは、初期的なエディタを表示し、ユーザがエディタとやりとりするとき、エディタを管理及び維持するため、ウェブページにおけるすべてのJavaScriptコードを実行する。PHPスクリプトは、クライアントプロセッサ602 にはわたされないが、サーバ側コードである。

20

【0117】

図31は、本発明の実施例によるエディタ (メンバースケジュールエディタ302、プロジェクトスケジュールエディタ202、タスク割当てエディタ102 など) のウェブページの各コンポーネントを示す。PHPプロセッサはサーバ側で実行され、JavaScriptプロセッサはクライアント側で実行される。サーバ上のPHPプロセッサは、クライアント上でJavaScriptプロセッサにより実行されるJavaScriptコードを生成するため、PHPコンポーネントを実行する。

30

【0118】

[クライアント・サーバベースプロジェクトスケジュールシステムによるプロジェクトスケジュール管理方法]

図32は、本発明の実施例によるクライアント・サーバベースプロジェクトスケジュール管理システムによるプロジェクトスケジュール管理方法を示すフロー図である。図32に示される方法の実施例は、1以上のプロセッサが命令を実行することによってなど、コンピュータ又はマシンが方法の各ステップを実行するコンピュータ及び/又はマシンにより実現される方法として実現される。例えば、本方法は、図35のコンピュータシステム3500などのコンピュータシステムにより実行可能である。

40

【0119】

ブロック3202において、クライアント・サーバベースプロジェクトスケジュールシステムに係るエディタを閲覧するリクエストに回答して、サーバはデータベースの第1のスケジュールに関する情報にアクセスする。例えば、クライアントプロセッサ602 (図6) のユーザは、タスク割当てエディタ102 (図1A)、メンバースケジュールエディ

50

タ 3 0 2 (図 3 A) 又はプロジェクトスケジュールエディタ 2 0 2 (図 2 A) の 1 つをリクエストするため、ユーザインタフェースとやりとりする。当該リクエストに回答して、サーバプロセッサ 6 0 4 (図 6) は、データベース 5 0 6 (図 5) からのデータ 5 0 8 (図 5) 及び / 又はデータベース 5 3 8 (図 5) からのデータ 5 3 6 (図 5) など、データベースからデータにアクセスする。例えば、サーバプロセッサ 6 0 4 は、要求元のユーザと対応するプロジェクトに対して要求されたエディタを提供するため、データベースから適切なデータを抽出するよう PHP スクリプトコードを実行する。データベースからサーバが抽出するデータは、ユーザがリクエストしたエディタに固有のものであり、またユーザ i d やプロジェクト i d など、当該リクエストに関連してユーザにより入力された各種情報に固有なものである。エディタに対するリクエストに回答してサーバがデータベースから抽出するデータは、必要に応じて、リクエストされたエディタに各フィールドを提供するための初期情報を含む。

10

【 0 1 2 0 】

ブロック 3 2 0 4 において、サーバは、要求元のクライアントによる実行のためのクライアントにより実行可能なコードを生成する。サーバにより生成されるこのクライアントにより実行可能なコードは、リクエストされたエディタをクライアントに表示し、エディタの適当なフィールドに抽出した情報を表示し、クライアントにおいてエディタを管理するためのものである。例えば、サーバプロセッサ 6 0 4 (図 6) は、抽出したデータをクライアントプロセッサ 6 0 2 (図 6) が利用可能なフォーマットに変換するため、PHP スクリプトコードを実行する。例えば、クライアントプロセッサ 6 0 2 は、サーバスクリプトコードを理解しないため、クライアントプロセッサ 6 0 2 により実行可能な JavaScript コードに抽出した情報をラップするなど、サーバは抽出した情報をクライアントが理解及び利用可能なフォーマットに処理する必要がある。ブロック 3 2 0 6 において、サーバはクライアントにより実行可能なコードと第 1 のスケジュールに関する情報を実行のためクライアントにわたす。

20

【 0 1 2 1 】

付録 A、C 及び E は、各エディタのための一例となるコードリストを提供する。これらのコードリスト例は、< s c r i p t > タグにより示される JavaScript と、< ? p h p a n d ? > タグ内に囲まれた PHP スクリプトを示す。エディタページは、ウェブサーバ 5 0 7、5 3 0 (図 5) などのサーバプロセッサ 6 0 4 に格納される。ウェブブラウザなどのクライアントプロセッサ 6 0 2 がエディタページにアクセスすると、サーバプロセッサ 6 0 4 において PHP スクリプトが実行され、PHP スクリプト全体が、PHP スクリプトが生成する JavaScript コードに置換される。PHP スクリプトにより生成されるものを含むすべての JavaScript コードが、実行のためクライアントプロセッサ 6 0 2 にわたされる。

30

【 0 1 2 2 】

ブロック 3 2 0 8 において、クライアントは、リクエストされたエディタに第 1 のスケジュールに関する情報を表示し、データ及びエディタを管理するため、一般にクライアントにより実行可能なコード又は当該コードの少なくとも一部を実行する。このため、リクエストされたエディタの初期表示は、クライアントプロセッサとサーバプロセッサのそれぞれに係る上記アクションに基づき完了する。

40

【 0 1 2 3 】

エディタページがサーバにより生成されるクライアントにより実行可能なコード (JavaScript など) を実行することによってクライアントにロードされると、ユーザは、エディタに係る情報を編集及び / 又は追加することを開始できる。ブロック 3 2 1 0 において、クライアントは第 2 スケジュールに関する情報をエディタを介しユーザから受信する。例えば、エディタに応じて、クライアントプロセッサ 6 0 2 (図 6) は、エディタを介し入力されたユーザからのプロジェクトスケジュール情報、メンバースケジュール情報又はタスク割当て情報を受信する。

【 0 1 2 4 】

50

ブロック 3 2 1 2 において、クライアントは、クライアント側において第 2 スケジュールに関する情報をエディタにおいて管理及び / 又は維持するため、クライアントにより実行可能なコードの少なくとも一部を実行する。例えば、コードの実行は、サーバへの送信前にクライアントにおいて新規の又は更新されたデータを管理するためのデータ構造及び関連付けを生成し、エディタページオブジェクト (HTML ボタン、テキストエントリオブジェクトなど) に実現される各機能を提供する。

【 0 1 2 5 】

ブロック 3 2 1 4 において、クライアントは第 2 スケジュールに関する情報をエディタからサーバに転送する。ブロック 3 2 1 6 において、サーバは第 2 スケジュールに関する情報をデータベースに格納し、そこからリクエストに回答してクライアントに返送するため、当該情報は以降においてアクセス可能となる。例えば、各エディタページ (図 1 A、2 A、3 A など) に対するリクエスト又は各エディタに係るウェブページ (図 1 B、2 B、3 B など) に対するリクエストに回答して、スケジュール関連情報がサーバからクライアントにわたされるかもしれない。

【 0 1 2 6 】

[ネットワークベースプロジェクトスケジュールシステムにおけるデータベースクエリの自動生成方法]

図 3 3 は、本発明の実施例によるネットワークベースプロジェクトスケジュール管理システムにおけるデータベースクエリ自動生成方法を示すフロー図である。図 3 3 に示される方法の実施例は、1 以上のプロセッサが命令を実行などすることによって、コンピュータ又はマシンが方法の各ステップを実行するコンピュータ及び / 又はマシンにより実現される方法として実現される。例えば、本方法は、図 3 5 のコンピュータシステム 3 5 0 0 などのコンピュータシステムにより実行可能である。

【 0 1 2 7 】

ブロック 3 3 0 2 において、ネットワークベースプロジェクトスケジュールシステムのエディタに係るリクエストに回答して、当該エディタに係るクエリ文字列が特定される。“コンスタントクエリ文字列” (図 2 7 ~ 2 9 など) としても記載されるクエリ文字列は、1 以上のプレースホルダー文字列を有する。プレースホルダー文字列は、当該プレースホルダー文字列の置換としてわたされる各値について、コンスタントクエリ文字列内においてプレースホルダーとして機能する。このため、各プレースホルダー文字列は、データベース 5 0 6 及び / 又はデータベース 5 3 6 などのデータベースに送信するためのクエリを生成するため、プレースホルダー文字列が置換される値のタイプを指定する。この“値のタイプ”は、データタイプを表すのではなく、対応するプレースホルダー文字列を置換するため値が使用される対応する変数名である。例えば、図 2 7 及び 2 8 を参照するに、これらの図に示されるように、プレースホルダー文字列 “%% Project Number %%” はプロジェクト番号の値 (“J 1 7” など) と置換され、プレースホルダー文字列 “%% Member Label %%” は、プロジェクトメンバーのラベルの値 (“T 1” など) と置換され、プレースホルダー文字列 “%% Project Task ID %%” は、プロジェクトタスクの id の値 (“4 0” など) と置換される。C__Project Number Key、C__Member Label Key 及び C__Project Task ID Key などのプレースホルダー文字列の各定数は、プレースホルダー文字列を適切な値と置換するため、コンスタントクエリ文字列のプレースホルダー文字列を特定するのにストリングファンクション (PHP の str__replace () など) により使用される。

【 0 1 2 8 】

ブロック 3 3 0 4 において、クエリ文字列の 1 以上のプレースホルダー文字列を対応する値と自動的に置換することによって、データベースクエリが生成される。例えば、プレースホルダー文字列 “%% Project Number %%” は “J 1 7” と置換され、プレースホルダー文字列 “%% Member Label %%” は “T 1” と置換され、プレースホルダー文字列 “%% Project Task ID %%” は “4 0” と置換される

。

【0129】

図27を参照して説明されたように、実施例によると、何れかのプレースホルダー文字列について“コンスタントクエリ文字列”を検索し、プレースホルダー文字列をある値と置換するため、“プレースホルダー文字列のための定数”が使用される。

【0130】

図29を参照して説明されたように、実施例によると、サーバプロセッサ604(図6)に係るCXXxDBInterfaceクラスとCYYYDBQueryGeneratorクラスは、ブロック3304においてデータベースクエリを生成するのに使用される。ここで、クエリ生成プロセスは、CXXxDBInterfaceクラスのプライベートファンクションに基づくものであってもよいし、又はCYYYDBQueryGeneratorクラスのパブリックファンクションに基づくものであってもよい。実施例によると、クエリ文字列は、データベースのデータにアクセスするための類似したデータベースクエリを自動生成するための再利用可能なクエリテンプレートとして機能する基礎となるコンスタントクエリ文字列を変更せずに、プレースホルダー文字列の置換を可能にするため変数文字列(図28の“\$loc_sQuery”など)に割り当てられる。

【0131】

ブロック3306において、自動生成されたデータベースクエリがデータベースに送信され、ブロック3308において、データベースクエリの結果が、リクエストに回答して返される。

【0132】

[プロジェクトスケジュールシステムにおけるタスク管理方法]

図34は、本発明の実施例によるプロジェクトスケジュール管理システムにおけるタスク管理方法を示すフロー図である。図34に示される方法の実施例は、1以上のプロセッサが命令を実行などすることによって、コンピュータ又はマシンが方法の各ステップを実行するコンピュータ及び/又はマシンにより実現される方法として実現される。例えば、本方法は、図35のコンピュータシステム3500などのコンピュータシステムにより実行可能である。

【0133】

ブロック3402において、エディタの表示テーブルの行に影響を与えるイベントに回答して、影響を受けた当該行に対応するクラスオブジェクトは、プロジェクトスケジュールシステムに係るエディタの要素に対応するクラスオブジェクトの1以上の属性に直接アクセスする。表示テーブルの各行は、エディタの各要素に対応する表示値とプロジェクトスケジュールに係るスケジュールタスクとに係るスケジュールタスクに対応する。エディタの各要素はクラスオブジェクトの属性として構成されるため、クラスオブジェクトはこれらの属性に直接的にアクセス可能であることが重要である。このため、クラスオブジェクトは、影響を受けた行の影響を受けた各要素の要素idを構成する必要はなく、またこのような要素を取得する必要もない。

【0134】

例えば、ユーザは、メンバースケジュールエディタ302(図3A)を介しタスクのスケジュールデータを編集する。ユーザの編集は、メンバースケジュールエディタのテーブルの行に影響を与えるイベントを有する。メンバースケジュールエディタ302のXXXJavaScriptProcessor1410(図14)のクラス(図15のCMemberScheduleTaskRowData1506など)のメンバーファンクションは、必要に応じてイベントに基づき要素のプロパティを変更するため、クラスのオブジェクトの属性として各要素への直接的なアクセスを有する。各要素は、エディタセッションの終了時に、サーバプロセッサにわたすことが可能なタスクに関する情報を当該行に維持する。

【0135】

ブロック3404において、影響を受けた行に対応するクラスオブジェクトは、イベン

トに基づきクラスオブジェクトの1以上の属性のそれぞれについて値を直接操作する。この例から続けて、メンバースケジュールエディタ302のXXXJavaScriptProcessor1410のCMemberScheduleTaskRowData1506クラスのオブジェクトのメンバーファンクションは、オブジェクトの属性値を設定し、メンバースケジュールエディタ302の要素の各値を操作する。

【0136】

ブロック3406において、クライアントは、ブロック3404において操作された属性の各値を含む1以上の属性のそれぞれの値をサーバに送信する。例えば、メンバースケジュールエディタ302のXXXJavaScriptProcessor1410を有するクライアントプロセッサ602（図6）は、操作されたデータをサーバプロセッサ604（図6）に提供する。ブロック3408において、サーバは、各属性の値をデータベースに格納する。例えば、サーバプロセッサ604は、データベース506、536（図5）などのデータベースにデータを格納する。エディタセッションの終了時、テーブルの各行のイベントがタスクに関する情報を変更又は追加したタスクが、データベースにおいて更新又は追加される。

【0137】

[デザインパターン]

“デザインパターン”は、オブジェクト指向システムにおいて繰り返し発生するデザイン問題を解決する一般的なデザインを意味する。メンバースケジュールエディタの一般的なデザインは、タスク割当てエディタ及びプロジェクトスケジュールエディタに適用される。デザインパターンは、Addison-Wesleyにより刊行されているEric Gammaらによる“Design Patterns: Elements of Reusable Object-Oriented Software”に記載されている。図15～18は、メンバースケジュールエディタの各種パッケージのクラスのデザインを示している。当該デザインは、図19～22に示されるようなプロジェクトスケジュールエディタと、図23～26に示されるようなタスク割当てエディタに同様に使用される。これらのエディタの特性及び動作は異なっているが、デザインパターンはシステムのすべてのエディタにより利用可能である。さらなるエディタがプロジェクトスケジュール管理システムに追加される場合、新たなエディタの設計及び実装に関する労力及び作業は、既存のエディタのデザインパターンに従うことによって大きく低減することができる。

【0138】

図36は、本発明の実施例によるエディタに対応するシステムの3つの主要なパッケージに着目したテーブル7のインデックス化部分を示す図である。各エディタは、図14に記載されるような4つのサブパッケージを有する。各サブパッケージは、それらのプロセスを実行するための同様のクラス構成を有する。各パッケージからのクラスの記載は、エディタのデザインパターンを示すのに役立つ。

【0139】

CTaskAssignmentTaskRowData3602、CProjectScheduleTaskRowData3612及びCMemberScheduleTaskRowData3622の各クラスは、データベースからタスク情報を取得し、行のタスク情報をその対応するエディタに表示するためのクライアントコードを生成する各自のXXXPHPProcessorパッケージの一部である。CTaskAssignmentTaskRowData3602は、プロジェクトタスク、対応するメンバータスク及びメンバータスクのメンバーへの割当てに関する情報を取得する。CTaskAssignmentTaskRowData3602は、プロジェクトタスク行とメンバー割当てによるメンバータスク行とをタスク割当てエディタに表示するためのクライアントコードを生成する。CProjectScheduleTaskRowData3612は、プロジェクトタスク、対応するメンバータスク及びタスクのスケジュールに関する情報を取得する。CProjectScheduleTaskRowData3612

は、プロジェクトタスクスケジュールと対応するメンバータスクスケジュールの行をプロジェクトスケジュールエディタに表示するためのクライアントコードを生成する。CMemberScheduleTaskRowData3622は、メンバータスク、すべての詳細タスク（レベル4までのタスク）及びタスクのスケジュールに関する情報を取得する。CMemberScheduleTaskRowData3622は、メンバータスクスケジュールと対応する詳細タスクスケジュールの行をメンバースケジュールエディタに表示するためのクライアントコードを生成する。各エディタのXXXPHPProcessorパッケージは、情報が異なるものであっても、タスク行をクライアントプロセッサのエディタに表示するためのコードを生成するためのクラスを利用する。

【0140】

CTaskAssignmentTable3604及びCProjectTableManager3614の各クラス並びにCTableManager及びCMemberScheduleTable3624の組み合わせは、対応するエディタのテーブル及び各行を生成、管理及び維持する各自のXXXJavaScriptProcessorパッケージの一部である。メンバースケジュールエディタはタスク割当てエディタ及びプロジェクトスケジュールエディタより相対的に複雑であるため（すなわち、異なるレベルでのタスクの追加及び削除、実際の日付の設定、上位レベルのタスクスケジュールにより下位レベルタスクスケジュールの更新）、2つのクラスはテーブル及び行を管理するのに利用される。エディタのテーブルにおいて発生しうるイベントのタイプ及びテーブルの各コンポーネントは異なるが、それらはすべてパッケージのデザインにおいて1又は2のクラスにより表すことができる。XXXJavaScriptProcessorパッケージは、テーブル、行及びセルなどのエディタの各部分に対応するクラスを含む。

【0141】

CTaskAssignmentUpdater3606、CProjectTaskUpdater3616及びCMemberTaskUpdater3626の各クラスは、クライアントプロセッサ上の対応するエディタセッションから転送される情報によって、データベースのタスク情報を更新する各自のPostXXXPHPProcessorパッケージの一部である。エディタにおいてタスクに対して実行されるアクションに応じて、データベースのタスクに関する情報を更新するため、適切なアクションがとられる。アクションのタイプは各エディタ間で異なり、当該アクションの詳細はクラスのデザインの内部で処理され、クラスの全体ファンクションは、データベースのタスク情報を更新するためのものである。従って、すべてのエディタに対して情報をエディタセッションからデータベースに更新するため、デザインパターンが利用可能である。

【0142】

CTaskAssignmentWebPageManager3608、CProjectScheduleWebPageManager3618及びCMemberScheduleWebPageManager3628の各クラスはそれぞれ、タスク割当て、プロジェクトスケジュール及びメンバースケジュールのウェブページを生成するクラスを管理する各自のXXXWebPageGeneratorパッケージの一部である。CTaskAssignmentWebPageManager3608は、各種クラスを利用してプロジェクトタスクとメンバータスクを示すテーブルによりウェブページを生成する。ここで、メンバータスクはタスクに割り当てられたメンバー及びタスクの履歴を表示する。

【0143】

CProjectScheduleWebPageManager3618は、各種クラスを利用して、スケジュールの履歴と共にプロジェクトタスクスケジュール及びそのメンバータスクスケジュールを示すテーブルによるウェブページを生成する。CMemberScheduleWebPageManager3628は、各種クラスを利用して、スケジュールの履歴と共にタスクスケジュール及びその詳細タスクを示すテーブルによるウェブページを生成する。同一のデザインパターンが、異なる情報を有するウェブペ

10

20

30

40

50

ージを生成するすべてのエディタにより使用される。

【0144】

CTaskAssignmentWebPageGeneratorDBInterface3610、CProjectWebPageDBInterfaceとCProjectWebPageDBQueryGenerator3620の組み合わせ、及びCWebPageDBInterface及びCWebPageDBQueryGenerator3630の組み合わせの各クラスはそれぞれ、タスク割当て、プロジェクトスケジュール及びメンバースケジュールの各ウェブページを生成するのに必要とされるタスク情報にアクセスするため、データベースとのインタフェースを処理する各自のXXXWebPageGeneratorパッケージの一部である。エディタの各クラス及びクラスの各組み合わせは、データベースクエリを生成し、クエリに応答して情報を取得するデータベースインタフェースを表す。

10

【0145】

図36のパッケージの各クラスの記載では、メンバースケジュールエディタの各クラスは、その他のエディタに類似するクラスを有する。このため、メンバースケジュールに使用されるデザインパターンは、その他のエディタにおいて利用可能である。エディタの各パッケージは異なる動作をするが、同一のデザインパターンが依然として使用可能である。

【0146】

[付録]

20

付録Aは、本発明の実施例によるプロジェクトスケジュールエディタのウェブページの一例となるコードリストを示す。このコードリスト例は、<script>タグにより示されるJavaScriptと、<?php and?>タグ内に囲まれたPHPスクリプトとを示す。ウェブページは、ウェブサーバ507、530(図5)などのサーバプロセッサ604(図6)に格納される。ウェブブラウザなどのクライアントプロセッサ602(図6)がウェブページにアクセスすると、サーバプロセッサ604においてPHPスクリプトが実行され、PHPスクリプト全体がPHPスクリプトが生成するJavaScriptコードと置換される。PHPスクリプトにより生成されたものを含むすべてのJavaScriptコードが、実行のためクライアントプロセッサ602にわたされる。

【表 8】

付録A

```

<!--////////////////////////////////////
//      Copyright 2006 by Ricoh Corporation
//      All Rights Reserved
//      Confidential and Proprietary
////////////////////////////////////
// File:      ProjScheduleEditor.htm
//
// Description: This file is the web page for the project schedule
//              editor.
// Author:
// History:
////////////////////////////////////-->
<html>
<head>
  <title>Project Schedule Editor</title>
</head>

<script>
const C_DEBUG = true;
</script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectEditorManager.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectTableManager.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectScheduleTable.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectScheduleRow.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectTaskCell.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/cProjectDateCell.js"></script>
<script src="ProjectScheduleJavaScriptProcessor/sProjectTaskInfo.js"></script>
<script src="../Common/JavaScriptCommon/debugUtility.js"></script>
<script src="../Common/JavaScriptCommon/dateUtility.js"></script>
<script src="../Common/JavaScriptCommon/editorUtility.js"></script>
<script src="../Common/JavaScriptCommon/javascriptSystemConstants.js"></script>
<script src="../Common/JavaScriptCommon/cDateSelector.js"></script>
<script src="../Common/JavaScriptCommon/calendarUtility.js"></script>

<script>
////////////////////////////////////
// Global Function: fglo_deleteTask()
//-----
// Description:   This function is the event handler for Delete button
//               that will delete the selected project task and its
//               member subtasks.
// Input:         None
// Output:        None
// Preconditions: glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1 if preconditions are not met, return.
//               2 call glo_EditorManager.deleteSelectedTask().
////////////////////////////////////
function fglo_deleteTask() {
  fglo_PrintDebug("fglo_deleteTasks");

  if (glo_EditorManager == null) {
    return;
  }

  glo_EditorManager.deleteSelectedTask();
}

```

10

20

30

【表 9】

付録A (続き)

```

////////////////////////////////////
// Global Function: fglo_addTasks()
//-----
// Description:   This function is the event handler for the AddXXX button.
//               This function adds empty task rows to the editor for
//               the project to add project tasks to the schedule.
// Input:         None
// Output:        None
// Preconditions:  glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1  if preconditions are not met, return.
//               2  if this.name is empty, return
//               3  call glo_EditorManager.addTasks(this.name).
////////////////////////////////////
function fglo_addTasks() {
    fglo_PrintDebug("fglo_addTasks()");

    if (glo_EditorManager == null) {
        return;
    }

    if (this.name.length == 0) {
        return;
    }

    glo_EditorManager.addTasks(this.name);
}

////////////////////////////////////
// Global Function: fglo_consolidateSchedule()
//-----
// Description:   This function is the event handler for Consolidate
//               button which consolidate the project schedule with the
//               member schedules and redisplay the project schedule
//               editor after posting the project schedule.
// Input:         None
// Output:        None
// Preconditions:  glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1  if preconditions are not met, return.
//               2  call glo_EditorManager.consolidateSchedule().
////////////////////////////////////
function fglo_consolidateSchedule() {
    fglo_PrintDebug("fglo_consolidateSchedule()");

    if (glo_EditorManager == null) {
        return;
    }

    glo_EditorManager.consolidateSchedule();
}

////////////////////////////////////
// Global Function: fglo_submitSchedule()
//-----
// Description:   This function is the onsubmit event handler when the
//               Finish or Consolidate button is selected for posting
//               the project schedule. This function validates the
//               schedule.
// Input:         None
// Output:        bool indicating if the task schedule is valid.
// Preconditions:  glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1  if preconditions are not met, return false.
//               2  return glo_EditorManager.submitSchedule().
////////////////////////////////////
function fglo_submitSchedule() {
    fglo_PrintDebug("fglo_submitSchedule()");

    if (glo_EditorManager == null) {
        return false;
    }

    return glo_EditorManager.submitSchedule();
}
</script>

```

10

20

30

40

【表 10】

付録A (続き)

```

<style type="text/css">
<!--
span.label {color:black;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold 13px
Arial}
span.c1 {cursor:hand;color:black;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold
13px Arial}
span.c2 {cursor:hand;color:red;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold
13px Arial}
span.c3
{cursor:hand;color:#b0b0b0;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold 12px
Arial}
-->
</style>

<body id="ProjSchedBodyID">
  <p id="WorkingID" align="center"><font size="7">Working ...</font><br>
    <br>
  </p>
  <?php
    include_once("ProjectSchedulePHPProcessor/cProjectScheduleManager.php");
    include_once("ProjectSchedulePHPProcessor/cProjectScheduleInitialData.php");
    include_once("ProjectSchedulePHPProcessor/cProjectScheduleTaskRowData.php");
    include_once("ProjectSchedulePHPProcessor/cProjectEditorDBInterface.php");
    include_once("ProjectSchedulePHPProcessor/cProjectEditorDBQueryGenerator.php");
    include_once("ProjectSchedulePHPProcessor/cJavaScriptInterface.php");
    include_once("ProjectSchedulePHPProcessor/projectSchedulePHPProcessorConstants.php");
    include_once("../Common/PHPCommon/debugUtility.php");
    include_once("../Common/PHPCommon/phpSystemConstants.php");
    include_once("../Common/PHPCommon/dateUtility.php");
    include_once("../Common/PHPCommon/cScheduleDB.php");

    //////////////////////////////////////
    // Main
    //-----
    // Description:   This function will create the object that will generate
    //               the initial display of the project schedule editor.
    // Input:         None
    // Output:        None
    // Preconditions: None
    // Postconditions: None
    // Security:      None
    // Algorithm:
    //               1 create and assign a CProjectScheduleManager object
    //               to $glo_ProjectScheduleManager.
    //               2 call createProjectScheduleEditor() of
    //               $glo_ProjectScheduleManager.
    //////////////////////////////////////

    fglo_debugPrint("ProjScheduleEditor.htm Main");
    fglo_debugPrintVar("$_GET", $_GET);
    $glo_ProjectScheduleManager = new CProjectScheduleManager();
    $glo_ProjectScheduleManager->createProjectScheduleEditor();
    ?>

  <p align="center">
    <script type="text/javascript">

    //////////////////////////////////////
    // Main
    //-----
    // Description:   This function removes the working display and displays
    //               the calendar at the bottom of the web page.
    // Input:         None
    // Output:        None
    // Preconditions: None
    // Postconditions: None
    // Security:      None
    // Algorithm:
    //               1 get the body element and assign it to a local
    //               element loc_BodyElement.
    //               2 get the paragraph element containing the working
    //               display and assign it to a local element
    //               loc_ParagraphElement.
    //               3 call loc_BodyElement.removeChild() with
    //               loc_ParagraphElement passed in.
    //               4 call writeCalendar().
    //////////////////////////////////////
    var loc_BodyElement = document.getElementById("ProjSchedBodyID");
    var loc_ParagraphElement = document.getElementById("WorkingID");
    loc_BodyElement.removeChild(loc_ParagraphElement);
    writeCalendar();

    </script>
  </p>
</body>
</html>

```

10

20

30

40

付録Bは、本発明の実施例による付録AのPHPスクリプトによって生成される一例となるJavaScriptコードを示す。このJavaScriptコードは、ウェブページのPHPコードを置換する。JavaScriptコードは、データベースから取得されるタスクスケジューリング情報を含む。タスク情報は、情報を処理のためJavaScriptにわたすためのデータ構成に割り当てられる(例えば、`var glo__ProjectTaskInfo=new SProjectTaskInfo()`及び`glo__ProjectTaskInfo.xxx="value"`など)。また、オブジェクトを生成し、プロジェクトスケジュールエディタの初期表示を提供するためにオブジェクトのメンバーファンクションを呼び出すため、JavaScriptコードが生成される(`var glo__EditorManager=new CProjectEditorManager()`、`glo__EditorManager.setup_createEditor("J99")`及び`glo__EditorManager.setup_addTaskToEditor(glo__ProjectTaskInfo)`など)。

【表 1 1】

付録B

```
<body id="ProjSchedBodyID">
  <p id="WorkingID" align="center"><font size="7">Working ....</font><br>
    <br>
  </p>

  <script>

var glo_TaskOptionList = null;
var glo_EditorManager = new CProjectEditorManager();
glo_EditorManager.setup_createEditor("J99");
var loc_UnassignedProjectTaskList = new Array();
loc_UnassignedProjectTaskList.push("Class Specification");
loc_UnassignedProjectTaskList.push("Implementation and Unit Test Plan");
loc_UnassignedProjectTaskList.push("Iteration 1");
loc_UnassignedProjectTaskList.push("Iteration 2");
loc_UnassignedProjectTaskList.push("Iteration 3");
loc_UnassignedProjectTaskList.push("Planning");
loc_UnassignedProjectTaskList.push("Project Closing Documents");
loc_UnassignedProjectTaskList.push("System Test");
loc_UnassignedProjectTaskList.push("Post Documents");
glo_EditorManager.setup_addUnassignedProjectTasks(loc_UnassignedProjectTaskList);
var glo_ProjectTaskInfo = new SProjectTaskInfo();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Project Preparation";
glo_ProjectTaskInfo.m_sTaskID = "10";
glo_ProjectTaskInfo.m_SetDate = "2006-08-18";
glo_ProjectTaskInfo.m_PlanStart = "2006-08-25";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-01";
glo_ProjectTaskInfo.m_ActualStart = "2006-08-25";
glo_ProjectTaskInfo.m_ActualEnd = "2006-09-02";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = false;
glo_ProjectTaskInfo.m_sMemberLabel = "T1";
glo_ProjectTaskInfo.m_sTaskName = "Project Plan";
glo_ProjectTaskInfo.m_sTaskID = "12";
glo_ProjectTaskInfo.m_SetDate = "2006-09-07";
glo_ProjectTaskInfo.m_PlanStart = "2006-08-25";
glo_ProjectTaskInfo.m_PlanEnd = "2006-08-27";
glo_ProjectTaskInfo.m_ActualStart = "2006-08-25";
glo_ProjectTaskInfo.m_ActualEnd = "2006-08-29";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Requirements";
glo_ProjectTaskInfo.m_sTaskID = "20";
glo_ProjectTaskInfo.m_SetDate = "2007-01-18";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-01";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-15";
glo_ProjectTaskInfo.m_ActualStart = "2006-08-31";
glo_ProjectTaskInfo.m_ActualEnd = "2006-09-15";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = false;
glo_ProjectTaskInfo.m_sMemberLabel = "T1";
glo_ProjectTaskInfo.m_sTaskName = "Req. Doc";
glo_ProjectTaskInfo.m_sTaskID = "22";
glo_ProjectTaskInfo.m_SetDate = "2006-09-08";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-01";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-09";
glo_ProjectTaskInfo.m_ActualStart = "2006-08-31";
glo_ProjectTaskInfo.m_ActualEnd = "2006-09-08";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = false;
glo_ProjectTaskInfo.m_sMemberLabel = "T1";
glo_ProjectTaskInfo.m_sTaskName = "Req. Matrix";
glo_ProjectTaskInfo.m_sTaskID = "32";
glo_ProjectTaskInfo.m_SetDate = "2006-09-11";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-11";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-15";
glo_ProjectTaskInfo.m_ActualStart = "2006-09-11";
glo_ProjectTaskInfo.m_ActualEnd = "2006-09-15";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Document Guidelines";
glo_ProjectTaskInfo.m_sTaskID = "30";
glo_ProjectTaskInfo.m_SetDate = "2007-01-22";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-08";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-11";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
```

10

20

30

40

【表 1 2】

付録B (続き)

```

glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = false;
glo_ProjectTaskInfo.m_sMemberLabel = "T1";
glo_ProjectTaskInfo.m_sTaskName = "Code Conv";
glo_ProjectTaskInfo.m_sTaskID = "42";
glo_ProjectTaskInfo.m_SetDate = "2006-09-08";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-08";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-11";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Top Level Design";
glo_ProjectTaskInfo.m_sTaskID = "40";
glo_ProjectTaskInfo.m_SetDate = "2007-01-22";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-08";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-11";
glo_ProjectTaskInfo.m_ActualStart = "2006-09-07";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = false;
glo_ProjectTaskInfo.m_sMemberLabel = "T1";
glo_ProjectTaskInfo.m_sTaskName = "Major Packages";
glo_ProjectTaskInfo.m_sTaskID = "62";
glo_ProjectTaskInfo.m_SetDate = "2006-09-08";
glo_ProjectTaskInfo.m_PlanStart = "2006-09-08";
glo_ProjectTaskInfo.m_PlanEnd = "2006-09-11";
glo_ProjectTaskInfo.m_ActualStart = "2006-09-07";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Package Design";
glo_ProjectTaskInfo.m_sTaskID = "50";
glo_ProjectTaskInfo.m_SetDate = "2007-01-09";
glo_ProjectTaskInfo.m_PlanStart = "2007-01-10";
glo_ProjectTaskInfo.m_PlanEnd = "";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Class Design";
glo_ProjectTaskInfo.m_sTaskID = "60";
glo_ProjectTaskInfo.m_SetDate = "2007-01-18";
glo_ProjectTaskInfo.m_PlanStart = "2007-01-17";
glo_ProjectTaskInfo.m_PlanEnd = "2007-01-20";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Unit Test Plan";
glo_ProjectTaskInfo.m_sTaskID = "70";
glo_ProjectTaskInfo.m_SetDate = "2007-01-18";
glo_ProjectTaskInfo.m_PlanStart = "2007-01-18";
glo_ProjectTaskInfo.m_PlanEnd = "";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();
glo_ProjectTaskInfo.m_bIsProjectTask = true;
glo_ProjectTaskInfo.m_sMemberLabel = "";
glo_ProjectTaskInfo.m_sTaskName = "Implementation";
glo_ProjectTaskInfo.m_sTaskID = "80";
glo_ProjectTaskInfo.m_SetDate = "2007-01-18";
glo_ProjectTaskInfo.m_PlanStart = "2007-02-01";
glo_ProjectTaskInfo.m_PlanEnd = "2007-03-16";
glo_ProjectTaskInfo.m_ActualStart = "";
glo_ProjectTaskInfo.m_ActualEnd = "";
glo_EditorManager.setup_addTaskToEditor(glo_ProjectTaskInfo);
glo_ProjectTaskInfo.reset();

</script>

<p align="center">

```

10

20

30

40

タスク割当てエディタ（付録C及びD）とメンバースケジュールエディタ（付録E及びF）は、プロジェクトスケジュールエディタの付録A及びBに示されるように、そのウェブページがエディタを生成するための類似したフォーマットに従う。

【0149】

付録Cは、本発明の実施例によるタスク割当てエディタのウェブページの一例となるコードリストを示す。このコードリスト例は、`<script>`タグにより示されるJavaScriptと、`<?php and ?>`タグ内に囲まれたPHPスクリプトとを示す。ウェブページは、ウェブサーバ507、530（図5）などのサーバプロセッサ604（図6）に格納される。ウェブブラウザなどのクライアントプロセッサ602（図6）がウェブページにアクセスすると、サーバプロセッサ604においてPHPスクリプトが実行され、PHPスクリプト全体がPHPスクリプトが生成するJavaScriptコードと置換される。PHPスクリプトにより生成されたものを含むすべてのJavaScriptコードが、実行のためクライアントプロセッサ602にわたされる。

【表 1 3】

付録C

```

<!--////////////////////////////////////
//          Copyright 2006 by Ricoh Corporation
//          All Rights Reserved
//          Confidential and Proprietary
////////////////////////////////////
// File:      TaskAssignEditor.htm
//
// Description: This file is the web page for the task assignment
//              editor.
// Author:
// History:
////////////////////////////////////-->
<html>
<head>
  <title>Task Assignment Editor</title>
</head>
<script>
const C_DEBUG = false;
</script>
<script src="TaskAssignmentJavaScriptProcessor/cTaskAssignmentEditorManager.js"></script>
<script src="TaskAssignmentJavaScriptProcessor/cTaskAssignmentTable.js"></script>
<script src="TaskAssignmentJavaScriptProcessor/cTaskAssignmentRow.js"></script>
<script src="TaskAssignmentJavaScriptProcessor/cTaskCell.js"></script>
<script src="TaskAssignmentJavaScriptProcessor/cAssignmentCell.js"></script>
<script src="../../Common/JavaScriptCommon/debugUtility.js"></script>
<script src="../../Common/JavaScriptCommon/editorUtility.js"></script>
<script src="../../Common/JavaScriptCommon/javascriptSystemConstants.js"></script>

<script>
////////////////////////////////////
// Global Function: fglo_deleteTask()
//-----
// Description: This function is the event handler for Delete button
//              that will delete the selected task and its subtasks.
// Input:      None
// Output:     None
// Preconditions: glo_EditorManager cannot be null.
// Postconditions: None
// Security:   None
// Algorithm:
//              1 if preconditions are not met, return.
//              2 call glo_EditorManager.deleteSelectedTask().
////////////////////////////////////
function fglo_deleteTask() {
  fglo_PrintDebug("fglo_deleteTask()");

  if (glo_EditorManager == null) {
    return;
  }

  glo_EditorManager.deleteSelectedTask();
}

////////////////////////////////////
// Global Function: fglo_addTasks()
//-----
// Description: This function is the event handler for the AddXXX button.
//              This function adds empty task rows to the editor for
//              the member to add tasks to the schedule.
// Input:      None
// Output:     None
// Preconditions: glo_EditorManager cannot be null.
// Postconditions: None
// Security:   None
// Algorithm:
//              1 if preconditions are not met, return.
//              2 if this.name is empty, return
//              3 call glo_EditorManager.addTasks(this.name).
////////////////////////////////////
function fglo_addTasks() {
  fglo_PrintDebug("fglo_addTasks()");

  if (glo_EditorManager == null) {
    return;
  }

  if (this.name.length == 0) {
    return;
  }

  glo_EditorManager.addTasks(this.name);
}

```

10

20

30

40

【表 1 4】

付録C (続き)

```

////////////////////////////////////
// Global Function: fglo_addDetailTasks()
//-----
// Description:      This function is the event handler for Add Details
//                   button which adds empty task rows to the editor
//                   corresponding to detailed tasks of the selected task.
// Input:           None
// Output:          None
// Preconditions:    glo_EditorManager cannot be null.
// Postconditions:   None
// Security:         None
// Algorithm:
//
//                   1 if preconditions are not met, return.
//                   2 call glo_EditorManager.addDetailTasks().
////////////////////////////////////
function fglo_addDetailTasks() {
    fglo_PrintDebug("fglo_addDetailTasks()");

    if (glo_EditorManager == null) {
        return;
    }

    glo_EditorManager.addDetailTasks();
}

////////////////////////////////////
// Global Function: fglo_submitAssignment()
//-----
// Description:      This function is the onsubmit event handler when the
//                   Finish button is selected for posting the task assignment.
// Input:           None
// Output:          bool indicating if the task assignment is valid.
// Preconditions:    glo_EditorManager cannot be null.
// Postconditions:   None
// Security:         None
// Algorithm:
//
//                   1 if preconditions are not met, return false.
//                   2 return glo_EditorManager.submitAssignment().
////////////////////////////////////
function fglo_submitAssignment() {
    fglo_PrintDebug("fglo_submitAssignment()");

    if (glo_EditorManager == null) {
        return false;
    }

    return glo_EditorManager.submitTaskAssignment();
}

</script>

<body id="AssignmentBodyID">
    <p id="WorkingID" align="center"><font size="7">Working ...</font><br>
        <br>
    </p>
<?php
    include_once("TaskAssignmentPHPPProcessor/cTaskAssignmentManager.php");
    include_once("TaskAssignmentPHPPProcessor/cTaskAssignmentInitialData.php");
    include_once("TaskAssignmentPHPPProcessor/cTaskAssignmentTaskRowData.php");
    include_once("TaskAssignmentPHPPProcessor/cTaskAssignmentEditorDBInterface.php");
    include_once("TaskAssignmentPHPPProcessor/cTaskAssignmentJavaScriptInterface.php");
    include_once("TaskAssignmentPHPPProcessor/taskAssignmentPHPPProcessorConstants.php");
    include_once("../Common/PHPCCommon/debugUtility.php");
    include_once("../Common/PHPCCommon/phpSystemConstants.php");
    include_once("../Common/PHPCCommon/cScheduleDB.php");

////////////////////////////////////
// Main
//-----
// Description:      This function will create the object that will generate
//                   the initial display of the task assignment editor.
// Input:           None
// Output:          None
// Preconditions:    None
// Postconditions:   None
// Security:         None
// Algorithm:
//
//                   1 create and assign a CTaskAssignmentManager object
//                     to $glo_TaskAssignmentManager.
//                   2 call createTaskAssignmentEditor() of
//                     $glo_TaskAssignmentManager.
////////////////////////////////////

```

10

20

30

40

【表 1 5】

付録C (続き)

```

fglo_debugPrint("TaskAssignEditor.htm PHP Main");
fglo_debugPrintVar("$_GET", $_GET);
$glo_TaskAssignmentManager = new CTaskAssignmentManager();
$glo_TaskAssignmentManager->createTaskAssignmentEditor();
?>
<p align="center">
<script type="text/javascript">
////////////////////////////////////
// Main
//-----
// Description:      This function removes the working display.
// Input:            None
// Output:           None
// Preconditions:    None
// Postconditions:   None
// Security:         None
// Algorithm:
//
//      1 get the body element and assign it to a local
//        element loc_BodyElement.
//
//      2 get the paragraph element containing the working
//        display and assign it to a local element
//        loc_ParagraphElement.
//
//      3 call loc_BodyElement.removeChild() with
//        loc_ParagraphElement passed in.
////////////////////////////////////

fglo_PrintDebug("TaskAssignEditor.htm JavaScript Main");
var loc_BodyElement = document.getElementById("AssignmentBodyID");
var loc_ParagraphElement = document.getElementById("WorkingID");
loc_BodyElement.removeChild(loc_ParagraphElement);
</script>
</p>
</body>
</html>

```

10

20

【0 1 5 0】

付録Dは、本発明の実施例による付録CのPHPスクリプトによって生成される一例となるJavaScriptコードを示す。このJavaScriptコードは、ウェブページのPHPコードを置換する。JavaScriptコードは、データベースから取得されるタスクスケジューリング情報を含む。タスク情報は、情報を処理のためJavaScriptにわたされる。また、オブジェクトを生成し、タスク割当てエディタの初期表示を提供するためにオブジェクトのメンバーファンクションを呼び出すため、JavaScriptコードが生成される(var glo__EditorManager=new CTaskAssignmentManager(), glo__EditorManager.setup_createEditor("J99")及びglo__EditorManager.setup_addTopLevelTaskToEditor("10", "Project Preparation")など)。

30

【表 16】

付録D

```

<body id="AssignmentBodyID">
  <p id="WorkingID" align="center"><font size="7">Working ....</font><br>
    <br>
  </p>
  <h2 align="center">J99 Task Assignment Editor</h2>

  <script>

    var glo_TaskList = new Array();
    var glo_TaskOptionList = new Array();
    glo_TaskList.push("Class Specification");
    glo_TaskList.push("Implementation and Unit Test Plan");
    glo_TaskList.push("Iteration 1");
    glo_TaskList.push("Iteration 2");
    glo_TaskList.push("Iteration 3");
    glo_TaskList.push("Planning");
    glo_TaskList.push("Project Closing Documents");
    glo_TaskList.push("System Test");
    glo_TaskList.push("Post Documents");
    var glo_MemberList = new Array();
    glo_MemberList.push("T1");
    glo_MemberList.push("MGR");
    var glo_EditorManager = new CTaskAssignmentEditorManager();
    glo_EditorManager.setup_createEditor("J99");
    glo_EditorManager.setup_addTopLevelTaskToEditor("10", "Project Preparation");
    glo_EditorManager.setup_addLevelOneTaskToEditor("10", "Project Initiation", "MGR");
    glo_EditorManager.setup_addLevelOneTaskToEditor("10", "Project Plan", "MGR");
    glo_EditorManager.setup_addLevelOneTaskToEditor("10", "Resource Plan", "MGR");
    glo_EditorManager.setup_addTopLevelTaskToEditor("20", "Requirements");
    glo_EditorManager.setup_addLevelOneTaskToEditor("20", "Reqt Doc", "T1");
    glo_EditorManager.setup_addLevelOneTaskToEditor("20", "Reqt Matrix", "T1");
    glo_EditorManager.setup_addTopLevelTaskToEditor("30", "Document Guidelines");
    glo_EditorManager.setup_addLevelOneTaskToEditor("30", "Des Doc Guide", "MGR");
    glo_EditorManager.setup_addLevelOneTaskToEditor("30", "Code Conv", "T1");
    glo_EditorManager.setup_addLevelOneTaskToEditor("30", "Impl Plan Guide", "T1");
    glo_EditorManager.setup_addTopLevelTaskToEditor("40", "Top Level Design");
    glo_EditorManager.setup_addLevelOneTaskToEditor("40", "Database", "MGR");
    glo_EditorManager.setup_addLevelOneTaskToEditor("40", "Major Interfaces", "T1");
    glo_EditorManager.setup_addLevelOneTaskToEditor("40", "Major Packages", "T1");
    glo_EditorManager.setup_addTopLevelTaskToEditor("50", "Package Design");
    glo_EditorManager.setup_addLevelOneTaskToEditor("50", "Task Assignment", "");
    glo_EditorManager.setup_addTopLevelTaskToEditor("60", "Class Design");
    glo_EditorManager.setup_addLevelOneTaskToEditor("60", "Task Assignment", "MGR");
    glo_EditorManager.setup_addTopLevelTaskToEditor("70", "Unit Test Plan");
    glo_EditorManager.setup_addLevelOneTaskToEditor("70", "MemberSchedule Package", "");
    glo_EditorManager.setup_addLevelOneTaskToEditor("70", "ProjectSchedule Package", "MGR");
    glo_EditorManager.setup_addTopLevelTaskToEditor("80", "Implementation");
    glo_EditorManager.setup_addLevelOneTaskToEditor("80", "Project Schedule", "");
    glo_EditorManager.setup_addLevelOneTaskToEditor("80", "Task Assignment", "MGR");
    glo_EditorManager.setup_addLevelOneTaskToEditor("80", "Member Schedule", "T1");
    glo_EditorManager.setup_addTopLevelTaskToEditor("90", "2nd Iteration");

  </script>

```

10

20

30

【0151】

付録 E は、本発明の実施例によるメンバースケジュールエディタのウェブページの一例となるコードリストを示す。このコードリスト例は、<script>タグにより示される JavaScript と、<?php and?>タグ内に囲まれた PHP スクリプトとを示す。ウェブページは、ウェブサーバ 507、530 (図 5) などのサーバプロセッサ 604 (図 6) に格納される。ウェブブラウザなどのクライアントプロセッサ 602 (図 6) がウェブページにアクセスすると、サーバプロセッサ 604 において PHP スクリプトが実行され、PHP スクリプト全体が PHP スクリプトが生成する JavaScript コードと置換される。PHP スクリプトにより生成されたものを含むすべての JavaScript コードが、実行のためクライアントプロセッサ 602 にわたされる。

40

【表 17】

付録E

```

<!--////////////////////////////////////
//          Copyright 2006 by Ricoh Corporation
//          All Rights Reserved
//          Confidential and Proprietary
////////////////////////////////////
// File:      MembScheduleEditor.htm
//
// Description: This file is the web page for the member schedule
//              editor.
// Author:
// History:
////////////////////////////////////-->
<html>
<head>
  <title>Member Schedule Editor</title>
</head>
<script>
  const C_DEBUG = false;
</script>
<script src="MemberScheduleJavaScriptProcessor/cEditorManager.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cTableManager.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cMemberScheduleTable.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cMemberScheduleRow.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cTaskCell.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cDateCell.js"></script>
<script src="MemberScheduleJavaScriptProcessor/sMemberTaskInfo.js"></script>
<script src="MemberScheduleJavaScriptProcessor/cDetailTaskInfo.js"></script>
<script src="../Common/JavaScriptCommon/debugUtility.js"></script>
<script src="../Common/JavaScriptCommon/dateUtility.js"></script>
<script src="../Common/JavaScriptCommon/editorUtility.js"></script>
<script src="../Common/JavaScriptCommon/javascriptSystemConstants.js"></script>
<script src="../Common/JavaScriptCommon/cDateSelector.js"></script>
<script src="../Common/JavaScriptCommon/calendarUtility.js"></script>

<script>
////////////////////////////////////
// Global Function: fglo_deleteTask()
//-----
// Description:   This function is the event handler for Delete button
//               that will delete the selected task and its subtasks.
// Input:         None
// Output:        None
// Preconditions: glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1 if preconditions are not met, return.
//               2 call glo_EditorManager.deleteSelectedTask().
//////////////////////////////////// function fglo_deleteTask() {
  fglo_PrintDebug("fglo_deleteTask()");

  if (glo_EditorManager == null) {
    return;
  }

  glo_EditorManager.deleteSelectedTask();
}

////////////////////////////////////
// Global Function: fglo_addTasks()
//-----
// Description:   This function is the event handler for the AddXXX button.
//               This function adds empty task rows to the editor for
//               the member to add tasks to the schedule.
// Input:         None
// Output:        None
// Preconditions: glo_EditorManager cannot be null.
// Postconditions: None
// Security:      None
// Algorithm:
//               1 if preconditions are not met, return.
//               2 if this.name is empty, return
//               3 call glo_EditorManager.addTasks(this.name).
//////////////////////////////////// function fglo_addTasks() {

```

10

20

30

40

【表 18】

付録E (続き)

```

fglo_PrintDebug("fglo_addTasks()");

if (glo_EditorManager == null) {
    return;
}

if (this.name.length == 0) {
    return;
}

glo_EditorManager.addTasks(this.name);
}

////////////////////////////////////
// Global Function: fglo_addDetailTasks()
//-----
// Description:      This function is the event handler for Add Details
//                   button which adds empty task rows to the editor
//                   corresponding to detailed tasks of the selected task.
// Input:            None
// Output:           None
// Preconditions:    glo_EditorManager cannot be null.
// Postconditions:   None
// Security:         None
// Algorithm:        1 if preconditions are not met, return.
//                   2 call glo_EditorManager.addDetailTasks().
//////////////////////////////////// function fglo_addDetailTasks() {
fglo_PrintDebug("fglo_addDetailTasks()");

if (glo_EditorManager == null) {
    return;
}

glo_EditorManager.addDetailTasks();
}

////////////////////////////////////
// Global Function: fglo_updateTasks()
//-----
// Description:      This function is the event handler for Update button
//                   which updates all the task rows of the editor such
//                   that the schedules of the tasks are consolidated
//                   with the schedules of its subtasks.
// Input:            None
// Output:           None
// Preconditions:    glo_EditorManager cannot be null.
// Postconditions:   None
// Security:         None
// Algorithm:        1 if preconditions are not met, return.
//                   2 call glo_EditorManager.updateTasks().
//////////////////////////////////// function fglo_updateTasks() {
fglo_PrintDebug("fglo_updateTasks()");

if (glo_EditorManager == null) {
    return;
}

glo_EditorManager.updateTasks();
}

////////////////////////////////////
// Global Function: fglo_submitSchedule()
//-----
// Description:      This function is the onsubmit event handler when the
//                   Finish button is selected for posting the task schedule.
//                   This function updates and validates the schedule.
// Input:            None
// Output:           bool indicating if the task schedule is valid and
//                   could be updated.
// Preconditions:    glo_EditorManager cannot be null.
// Postconditions:   None
// Security:         None
// Algorithm:        1 if preconditions are not met, return false.
//                   2 return glo_EditorManager.submitSchedule().
//////////////////////////////////// function fglo_submitSchedule() {

```

10

20

30

40

【表 19】

付録E (続き)

```

fglo_PrintDebug("fglo_submitSchedule()");

if (glo_EditorManager == null) {
    return false;
}

return glo_EditorManager.submitSchedule();
}

</script>

<style type="text/css">
<!--
span.label {color:black;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold 13px
Arial}
span.c1 {cursor:hand;color:black;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold
13px Arial}
span.c2 {cursor:hand;color:red;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold
13px Arial}
span.c3
{cursor:hand;color:#b0b0b0;width:30;height:16;text-align:center;margin-top:0;background:#fff;font:bold 12px
Arial}
-->
</style>

<body id="MembSchedBodyID">
    <p id="WorkingID" align="center"><font size="7">Working ...</font><br>
        <br>
    </p>
    <?php
        include_once('MemberSchedulePHPPProcessor/cMemberScheduleManager.php');
        include_once('MemberSchedulePHPPProcessor/cMemberScheduleInitialData.php');
        include_once('MemberSchedulePHPPProcessor/cMemberScheduleTaskRowData.php');
        include_once('MemberSchedulePHPPProcessor/cEditorDBInterface.php');
        include_once('MemberSchedulePHPPProcessor/cJavaScriptInterface.php');
        include_once('MemberSchedulePHPPProcessor/memberSchedulePHPPProcessorConstants.php');
        include_once("../Common/PHPCCommon/debugUtility.php");
        include_once("../Common/PHPCCommon/phpSystemConstants.php");
        include_once("../Common/PHPCCommon/dateUtility.php");
        include_once("../Common/PHPCCommon/cScheduleDB.php");

        //////////////////////////////////////
        // Main
        //-----
        // Description:      This function will create the object that will generate
        //                    the initial display of the member schedule editor.
        // Input:            None
        // Output:           None
        // Preconditions:    None
        // Postconditions:    None
        // Security:         None
        // Algorithm:
        //
        //      1 create and assign a CMemberScheduleManager object
        //      to $glo_MemberScheduleManager.
        //      2 call createMemberScheduleEditor() of
        //      $glo_MemberScheduleManager.
        //////////////////////////////////////

        fglo_debugPrint("MembScheduleEditor.htm Main");
        fglo_debugPrintVar("$_GET", $_GET);
        $glo_MemberScheduleManager = new CMemberScheduleManager();
        $glo_MemberScheduleManager->createMemberScheduleEditor();
    ?>

    <p align="center">
        <script type="text/javascript">

        //////////////////////////////////////
        // Main
        //-----
        // Description:      This function removes the working display and displays
        //                    the calendar at the bottom of the web page.
        // Input:            None
        // Output:           None
        // Preconditions:    None
        // Postconditions:    None
        // Security:         None
        // Algorithm:
        //
        //      1 get the body element and assign it to a local
        //      element loc_BodyElement.
        //      2 get the paragraph element containing the working
        //      display and assign it to a local element
        //      loc_ParagraphElement.
        //      3 call loc_BodyElement.removeChild() with
        //      loc_ParagraphElement passed in.
        //      4 call writeCalendar().
        //////////////////////////////////////

```

10

20

30

40

【表 2 0】

付録 E (続き)

```
var loc_BodyElement = document.getElementById("MembSchedBodyID");
var loc_ParagraphElement = document.getElementById("WorkingID");
loc_BodyElement.removeChild(loc_ParagraphElement);
writeCalendar();

</script>
</p>
</body>
</html>
```

10

【 0 1 5 2】

付録 F は、本発明の実施例による付録 E の P H P スクリプトによって生成される一例となる J a v a S c r i p t コードを示す。この J a v a S c r i p t コードは、ウェブページの P H P コードを置換する。J a v a S c r i p t コードは、データベースから取得されるタスクスケジューリング情報を含む。タスク情報は、情報を処理のため J a v a S c r i p t にわたすためのデータ構成に割り当てられる。また、オブジェクトを生成し、メンバースケジュールエディタの初期表示を提供するためにオブジェクトのメンバーファンクションを呼び出すため、J a v a S c r i p t コードが生成される (v a r g l o _ E d i t o r M a n a g e r = n e w C E d i t o r M a n a g e r () 、 g l o _ E d i t o r M a n a g e r . s e t u p _ c r e a t e E d i t o r (" J 9 9 " , " t e s t 1 ") 及び g l o _ E d i t o r M a n a g e r . s e t u p _ a d d T a s k T o E d i t o r (g l o _ M e m b e r T a s k I n f o) など) 。

20

【表 2 1】

付録 F

```

<body id="MembSchedBodyID">
  <p id="WorkingID" align="center"><font size="7">Working ...</font><br>
    <br>
  </p>
  <h2 align="center">test1's J99 Schedule</h2>

  <script>
    var glo_TaskOptionList = null;
    var glo_EditorManager = new CEditorManager();
    glo_EditorManager.setup_createEditor("J99", "test1");
    var loc_MemberTaskNameList = new Array();
    loc_MemberTaskNameList.push("Impl Plan Guide");
    glo_EditorManager.setup_addUnscheduledTasks(30, "Document Guidelines", loc_MemberTaskNameList);
    loc_MemberTaskNameList.splice(0, 1);
    loc_MemberTaskNameList.push("Major Interfaces");
    glo_EditorManager.setup_addUnscheduledTasks(40, "Top Level Design", loc_MemberTaskNameList);
    loc_MemberTaskNameList.splice(0, 1);
    loc_MemberTaskNameList.push("Member Schedule");
    glo_EditorManager.setup_addUnscheduledTasks(80, "Implementation", loc_MemberTaskNameList);
    loc_MemberTaskNameList.splice(0, 1);
    var glo_MemberTaskInfo = new SMemberTaskInfo();
    glo_MemberTaskInfo.m_nTaskLevel = 1;
    glo_MemberTaskInfo.m_nParentTaskID = 30;
    glo_MemberTaskInfo.m_nTaskID = 42;
    glo_MemberTaskInfo.m_sTaskName = "Code Conv";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
    glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
    glo_MemberTaskInfo.m_PlanEnd = "2006-09-11";
    glo_MemberTaskInfo.m_ActualStart = "2006-09-08";
    glo_MemberTaskInfo.m_ActualEnd = "";
    glo_MemberTaskInfo.m_nRev = 1;
    glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
    glo_MemberTaskInfo.reset();
    glo_MemberTaskInfo.m_nTaskLevel = 2;
    glo_MemberTaskInfo.m_nParentTaskID = 42;
    glo_MemberTaskInfo.m_nTaskID = 32;
    glo_MemberTaskInfo.m_sTaskName = "draft";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
    glo_MemberTaskInfo.m_PlanStart = "2006-09-11";
    glo_MemberTaskInfo.m_PlanEnd = "2006-09-13";
    glo_MemberTaskInfo.m_ActualStart = "";
    glo_MemberTaskInfo.m_ActualEnd = "";
    glo_MemberTaskInfo.m_nRev = 1;
    glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
    glo_MemberTaskInfo.reset();
    glo_MemberTaskInfo.m_nTaskLevel = 2;
    glo_MemberTaskInfo.m_nParentTaskID = 42;
    glo_MemberTaskInfo.m_nTaskID = 42;
    glo_MemberTaskInfo.m_sTaskName = "review/inspection";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
    glo_MemberTaskInfo.m_PlanStart = "2006-09-14";
    glo_MemberTaskInfo.m_PlanEnd = "2006-09-15";
    glo_MemberTaskInfo.m_ActualStart = "";
    glo_MemberTaskInfo.m_ActualEnd = "";
    glo_MemberTaskInfo.m_nRev = 1;
    glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
    glo_MemberTaskInfo.reset();
    glo_MemberTaskInfo.m_nTaskLevel = 1;
    glo_MemberTaskInfo.m_nParentTaskID = 40;
    glo_MemberTaskInfo.m_nTaskID = 62;
    glo_MemberTaskInfo.m_sTaskName = "Major Packages";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
    glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
    glo_MemberTaskInfo.m_PlanEnd = "2006-09-11";
    glo_MemberTaskInfo.m_ActualStart = "2006-09-07";
    glo_MemberTaskInfo.m_ActualEnd = "";
    glo_MemberTaskInfo.m_nRev = 2;
    glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
    glo_MemberTaskInfo.reset();
    glo_MemberTaskInfo.m_nTaskLevel = 2;
    glo_MemberTaskInfo.m_nParentTaskID = 62;
    glo_MemberTaskInfo.m_nTaskID = 92;
    glo_MemberTaskInfo.m_sTaskName = "Component";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
    glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
    glo_MemberTaskInfo.m_PlanEnd = "2006-09-11";
    glo_MemberTaskInfo.m_ActualStart = "2006-09-07";
    glo_MemberTaskInfo.m_ActualEnd = "";
    glo_MemberTaskInfo.m_nRev = 2;
    glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
    glo_MemberTaskInfo.reset();
    glo_MemberTaskInfo.m_nTaskLevel = 3;
    glo_MemberTaskInfo.m_nParentTaskID = 92;
    glo_MemberTaskInfo.m_nTaskID = 12;
    glo_MemberTaskInfo.m_sTaskName = "Interfaces";
    glo_MemberTaskInfo.m_SetDate = "2006-09-08";
  </script>

```

10

20

30

40

【表 2 2】

付録 F (続き)

```

glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
glo_MemberTaskInfo.m_PlanEnd = "2006-09-11";
glo_MemberTaskInfo.m_ActualStart = "2006-09-07";
glo_MemberTaskInfo.m_ActualEnd = "";
glo_MemberTaskInfo.m_nRev = 2;
glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
glo_MemberTaskInfo.reset();
glo_MemberTaskInfo.m_nTaskLevel = 4;
glo_MemberTaskInfo.m_nParentTaskID = 12;
glo_MemberTaskInfo.m_nTaskID = 12;
glo_MemberTaskInfo.m_sTaskName = "Structures";
glo_MemberTaskInfo.m_SetDate = "2006-09-07";
glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
glo_MemberTaskInfo.m_PlanEnd = "2006-09-10";
glo_MemberTaskInfo.m_ActualStart = "2006-09-08";
glo_MemberTaskInfo.m_ActualEnd = "";
glo_MemberTaskInfo.m_nRev = 1;
glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
glo_MemberTaskInfo.reset();
glo_MemberTaskInfo.m_nTaskLevel = 4;
glo_MemberTaskInfo.m_nParentTaskID = 12;
glo_MemberTaskInfo.m_nTaskID = 22;
glo_MemberTaskInfo.m_sTaskName = "Drawings";
glo_MemberTaskInfo.m_SetDate = "2006-09-08";
glo_MemberTaskInfo.m_PlanStart = "2006-09-08";
glo_MemberTaskInfo.m_PlanEnd = "2006-09-11";
glo_MemberTaskInfo.m_ActualStart = "2006-09-07";
glo_MemberTaskInfo.m_ActualEnd = "";
glo_MemberTaskInfo.m_nRev = 1;
glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
glo_MemberTaskInfo.reset();
glo_MemberTaskInfo.m_nTaskLevel = 1;
glo_MemberTaskInfo.m_nParentTaskID = 20;
glo_MemberTaskInfo.m_nTaskID = 32;
glo_MemberTaskInfo.m_sTaskName = "Regt Matrix";
glo_MemberTaskInfo.m_SetDate = "2006-09-11";
glo_MemberTaskInfo.m_PlanStart = "2006-09-11";
glo_MemberTaskInfo.m_PlanEnd = "2006-09-15";
glo_MemberTaskInfo.m_ActualStart = "2006-09-11";
glo_MemberTaskInfo.m_ActualEnd = "2006-09-15";
glo_MemberTaskInfo.m_nRev = 3;
glo_EditorManager.setup_addTaskToEditor(glo_MemberTaskInfo);
glo_MemberTaskInfo.reset();
</script>

```

10

20

【0153】

[ハードウェア概略]

図35は、本発明の実施例が実現可能なコンピュータシステム3500を示すブロック図である。コンピュータシステム3500はさらに、ワークステーション102(図1)及びウェブサーバ104(図1)のシステム構成の非限定的な具体例を示す。コンピュータシステム3500は、情報を通信するためのバス3502又は他の通信機構と、情報を処理するためのバス3502に接続されたプロセッサ3504とを有する。コンピュータシステム3500はまた、プロセッサ3504により実行される命令及び情報を格納するため、バス3502に接続されたRAM(Random Access Memory)又は他のダイナミックストレージデバイスなどのメインメモリを有する。メインメモリ3506はまた、プロセッサ3504により実行される命令の実行中に一時的な変数又は他の中間情報を格納するため利用可能である。コンピュータシステム3500はさらに、プロセッサ3504の静的情報及び命令を格納するため、バス3502に接続されるROM(Read Only Memory)3508又は他のスタティックストレージデバイスを有する。磁気ディスク、光ディスク又は光磁気ディスクなどの記憶装置3510が、情報及び命令を格納するため設けられ、バス3502に接続される。

30

40

【0154】

コンピュータシステム3500が、情報をコンピュータユーザに表示するため、CRT(Cathode Ray Tube)又はLCD(Liquid Crystal Display)などのディスプレイ3512にバス3502を介し接続されてもよい。英数字及び他のキーを含む入力装置3514が、情報及びコマンド選択をプロセッサ3504に通信するためバス3502に接続される。他のタイプのユーザ入力装置は、指示情報及びコマンド選択をプロセッサ3504に通信し、ディスプレイ3512上のカーソルの

50

移動を制御するため、マウス、トラックボール又はカーソル指示キーなどのカーソルコントロール 3 5 1 6 がある。この入力装置は、典型的には、当該装置が平面上でイチを指定することを可能にする第 1 軸 (x など) と第 2 軸 (y など) の 2 つの軸による 2 つの自由度を有する。

【 0 1 5 5 】

本発明の実施例は、ここに記載された技術を実現するためのコンピュータシステム 3 5 0 0 の利用に関する。本発明の一実施例によると、これらの技術は、メインメモリ 3 5 0 6 に含まれる 1 以上の命令の 1 以上のシーケンスをプロセッサ 3 5 0 4 が実行することに対応して、コンピュータシステム 3 5 0 0 により実行される。このような命令は、記憶装置 3 5 1 0 などの他のコンピュータ可読媒体からメインメモリ 3 5 0 6 に読み込まれるかもしれない。メインメモリ 3 5 0 6 に含まれる命令シーケンスの実行は、プロセッサ 3 5 0 4 にここに記載されたプロセスステップを実行させる。他の実施例では、配線回路が本発明を実現するためのソフトウェア命令の代わりに又はそれと共に利用されるかもしれない。このため、本発明の実施例は、ハードウェア回路とソフトウェアの何れか特定の組み合わせに限定されるものでない。

10

【 0 1 5 6 】

ここで使用される“コンピュータ可読媒体”という用語は、実行のためプロセッサ 3 5 0 4 に命令を提供するのに関係する何れかの媒体を表す。このような媒体は、以下に限定されるものではないが、不揮発性媒体、揮発性媒体及び伝送媒体を含む多数の形態をとる。限定されることなく、不揮発性媒体の具体例として、記憶装置 3 5 1 0 などの光ディスク、磁気ディスク又は光磁気ディスクがあげられる。揮発性媒体は、メインメモリ 3 5 0 6 などの動的メモリを含む。伝送媒体は、バス 3 5 0 2 を構成する配線を含む同軸ケーブル、銅線及び光ファイバを含む。電装媒体はまた、無線及び赤外線データ通信中に生成されるものなど、音響波又は光波の形態をとる。

20

【 0 1 5 7 】

コンピュータ可読媒体の一般的な形態として、以下に限定されることなく、フロッピー（登録商標）ディスク、フレキシブルディスク、ハードディスク、磁気テープ、他の何れかの磁気媒体、CD-ROM、DVD、他の何れかの光若しくは光磁気媒体、パンチカード、紙テープ、ホールパターンを有する他の何れかの物理媒体、RAM、PROM、EPROM、FLASH-EPROM、他の何れかのメモリチップ若しくはカートリッジ、後述されるような搬送波又はコンピュータにより読み込み可能な他の何れかの媒体があげられる。

30

【 0 1 5 8 】

各種形態のコンピュータ可読媒体は、実行のため 1 以上の命令の 1 以上のシーケンスをプロセッサ 3 5 0 4 に搬送することに関するものであるかもしれない。例えば、命令は初期的にはリモートコンピュータの磁気ディスクに備えられるかもしれない。リモートコンピュータは、命令をその動的メモリにロードし、モデムを利用して電話線を介し送信することができる。コンピュータシステム 3 5 0 0 のローカルモデムは、電話線を介し当該データを受信し、赤外線送信機を利用して、データを赤外線信号に変換する。赤外線検出装置は、赤外線信号により搬送されたデータを受信し、適切な回路がバス 3 5 0 2 上に当該データを配設することが可能である。バス 3 5 0 2 は、データをメインメモリ 3 5 0 6 に搬送し、そこからプロセッサ 3 5 0 4 が命令を抽出及び実行する。メインメモリ 3 5 0 6 により受信された命令は、任意的にプロセッサ 3 5 0 4 による実行前後に記憶装置 3 5 1 0 に格納されてもよい。

40

【 0 1 5 9 】

コンピュータシステム 3 5 0 0 はまた、バス 3 5 0 2 に接続される通信インタフェース 3 5 1 8 を有する。通信インタフェース 3 5 1 8 は、ローカルネットワーク 3 5 2 2 に接続されるネットワークリンク 3 5 2 0 に接続した双方向データ通信を提供する。例えば、通信インタフェース 3 5 1 8 は、対応するタイプの電話線とのデータ通信接続を提供する ISDN (Integrated Services Digital Network

50

）カード又はモデムであってもよい。他の例として、通信インタフェース 3518 は、互換的な LAN (Local Area Network) とのデータ通信接続を提供する LAN カードであってもよい。ワイヤレスリンクがまた実装されてもよい。このような何れかの実現形態において、通信インタフェース 3518 は、各種タイプの情報を表すデジタルデータストリームを搬送する電気、電磁又は光信号を送受信する。

【0160】

ネットワークリンク 3520 は、典型的には、他のデータ装置との 1 以上のネットワークを介したデータ通信を提供する。例えば、ネットワークリンク 3520 は、インターネット・サービス・プロバイダ (ISP) 3526 により操作されるデータ装置又はホストコンピュータ 3524 とのローカルネットワーク 3522 を介した接続を提供するかもしれない。さらに、ISP 3526 は、現在では一般に“インターネット”3528 と呼ばれるワールドワイドパケットデータ通信ネットワークを介したデータ通信サービスを提供する。ローカルネットワーク 3522 及びインターネット 3528 は共に、デジタルデータストリームを搬送する電気、電磁又は光信号を利用する。コンピュータシステム 3500 との間でデジタルデータを搬送する各種ネットワークを介した信号及びネットワークリンク 3520 及び通信インタフェース 3518 を介した信号は、情報を伝送する搬送波の例示的な形態である。

10

【0161】

コンピュータシステム 3500 は、ネットワーク、ネットワークリンク 3520 及び通信インタフェース 3518 を介しプログラムコードを含むデータ及びメッセージを送受信することができる。インターネットの例では、サーバ 3530 は、インターネット 3528、ISP 3526、ローカルネットワーク 3522 及び通信インタフェース 3518 を介しアプリケーションプログラムのリクエストされたコードを送信するかもしれない。

20

【0162】

受信したコードは、実行のため記憶装置 3510 又は他の不揮発性ストレージに受信及び/又は格納されると、プロセッサ 3504 により実行されるかもしれない。このようにして、コンピュータシステム 3500 は、搬送波の形態によりアプリケーションコードを取得してもよい。

【0163】

[拡張及び代替]

本発明の代替的な実施例が、上記記載を通じて記載され、また実施例の文脈の理解を最も容易にする場所に記載されている。さらに、本発明は、その具体的な実施例を参照して説明された。しかしながら、本発明の実施例のより広範な範囲から逸脱することなく、各種改良及び変更が可能であることは明らかであろう。従って、本明細書及び図面は、限定的でなく例示的なものとしてみなされるべきである。

30

【0164】

さらに本記載では、特定のプロセスステップが特定の順序により与えられ、ステップを特定するため、英字及び英数字ラベルが利用可能である。本記載において特段の断りが無い場合、本発明の実施例はこのようなステップを実行する何れか特定の順序に限定される必要はない。特に、これらのラベルは、便宜的なステップの識別のためだけに使用され、当該ステップを実行する特定の順序を指定又は要求することを意図するものでない。

40

【図面の簡単な説明】

【0165】

【図 1 A】図 1 A は、本発明の実施例によるタスク割当てエディタの画面図である。

【図 1 B】図 1 B は、本発明の実施例によるタスク割当てウェブページのサンプルの画面図である。

【図 2 A】図 2 A は、本発明の実施例によるプロジェクトスケジュールエディタの画面図である。

【図 2 B】図 2 B は、本発明の実施例によるプロジェクトスケジュールウェブページのサンプルの画面図である。

50

【図 3 A】図 3 A は、本発明の実施例によるメンバースケジュールエディタの画面図である。

【図 3 B】図 3 B は、本発明の実施例によるメンバースケジュールウェブページのサンプルの画面図である。

【図 4】図 4 は、本発明の実施例によるプロジェクトメンバーがエディタの 1 つにログオンするためのログインウェブページの画面図である。

【図 5】図 5 は、本発明の実施例が実現可能な動作環境を示す図である。

【図 6】図 6 は、自動スケジューリングシステムのソフトウェアコンポーネントを含む本発明の実施例が実現可能な通信アーキテクチャを示す図である。

【図 7】図 7 は、本発明の実施例によるシステムのクライアントプロセッサとサーバプロセッサとの間のインタフェースを示す図である。

【図 8】図 8 は、本発明の実施例によるログインウェブページを使用してプロジェクトメンバー又はマネージャがエディタの 1 つにログオンするためのシーケンス図を示す。

【図 9】図 9 は、本発明の実施例によるタスク割当てエディタによるセッションにおけるプロジェクトマネージャのシーケンス図を示す。

【図 10】図 10 は、本発明の実施例によるプロジェクトスケジュールエディタによるセッションにおけるプロジェクトマネージャのシーケンス図を示す。

【図 11】図 11 は、本発明の実施例によるプロジェクトメンバースケジュールエディタによるセッションにおけるプロジェクトメンバーのシーケンス図を示す。

【図 12】図 12 は、本発明の実施例によるプロジェクト及びプロジェクトメンバーのタスク割当て及びタスクスケジュール情報を格納及び管理するのに使用されるデータベーステーブルのスキーマを示す。

【図 13】図 13 は、本発明の実施例による図 6 のサーバプロセッサのプログラミングパッケージ図を説明する図である。

【図 14】図 14 は、本発明の実施例によるエディタプロセッサパッケージのプログラミングパッケージ図を説明する図である。

【図 15】図 15 は、本発明の実施例による Member Schedule PHP Processor パッケージのクラス図を示す。

【図 16】図 16 は、本発明の実施例による Member Schedule JavaScript Processor パッケージのクラス図を示す。

【図 17】図 17 は、本発明の実施例による Post Member Schedule PHP Processor パッケージのクラス図を示す。

【図 18】図 18 は、本発明の実施例による Member Schedule Web Page Generator パッケージのクラス図を示す。

【図 19】図 19 は、本発明の実施例による Project Schedule PHP Processor パッケージのクラス図を示す。

【図 20】図 20 は、本発明の実施例による Project Schedule JavaScript Processor パッケージのクラス図を示す。

【図 21】図 21 は、本発明の実施例による Post Project Schedule PHP Processor パッケージのクラス図を示す。

【図 22】図 22 は、本発明の実施例による Project Schedule Web Page Generator パッケージのクラス図を示す。

【図 23】図 23 は、本発明の実施例による Task Assignment PHP Processor パッケージのクラス図を示す。

【図 24】図 24 は、本発明の実施例による Task Assignment JavaScript Processor パッケージのクラス図を示す。

【図 25】図 25 は、本発明の実施例による Post Task Assignment PHP Processor パッケージのクラス図を示す。

【図 26】図 26 は、本発明の実施例による Task Assignment Web Page Generator パッケージのクラス図を示す。

10

20

30

40

50

【図 27】図 27 は、本発明の実施例によるデータベースクエリを生成するのに使用される一例となるコンスタント文字列を示す。

【図 28】図 28 は、本発明の実施例による図 27 のコンスタント文字列からデータベースクエリを生成するのに使用される一例となるスクリプトを示す。

【図 29】図 29 は、本発明の実施例によるコンスタント文字列からクエリ文字列を生成するプロセスを示すフロー図である。

【図 30】図 30 は、本発明の実施例によるエディタのウェブページの各コンポーネントを示す。

【図 31】図 31 は、本発明の実施例によるウェブページの各コンポーネントを処理するプロセッサと共にエディタのウェブページの各コンポーネントを示す。

【図 32】図 32 は、本発明の実施例によるクライアント・サーバベースプロジェクトスケジュール管理システムによるプロジェクト管理方法を示すフロー図である。

【図 33】図 33 は、本発明の実施例によるネットワークベースプロジェクトスケジュール管理システムにおけるデータベースクエリを自動生成する方法を示すフロー図である。

【図 34】図 34 は、本発明の実施例によるプロジェクトスケジュール管理システムにおけるタスク管理方法を示すフロー図である。

【図 35】図 35 は、本発明の実施例が実現可能なコンピュータシステム 3500 を示すブロック図である。

【図 36】図 36 は、本発明の実施例によるエディタに対応するシステムの 3 つの主要なパッケージに着目したテーブル 7 のインデックス化部分を示す図である。

【符号の説明】

【0166】

- 102 タスク割当てエディタ
- 202 プロジェクトスケジュールエディタ
- 302 メンバースケジュールエディタ
- 500 ネットワーク
- 501、502 コンピュータワークステーション
- 506、536、606 データベース
- 507、530 ウェブサーバ
- 602 クライアントプロセッサ
- 604 サーバプロセッサ
- 3500 コンピュータシステム
- 3502 バス
- 3504 プロセッサ
- 3506 メインメモリ
- 3508 ROM
- 3510 記憶装置
- 3512 ディスプレイ
- 3514 入力装置
- 3516 カーソルコントロール
- 3518 通信インタフェース
- 3520 ネットワークリンク
- 3522 ローカルネットワーク
- 3524 ホストコンピュータ
- 3526 インターネット・サービス・プロバイダ (ISP)
- 3528 インターネット
- 3530 サーバ

10

20

30

40








































【 図 1 A 】

[illegible]

【 図 2 A 】

[illegible]

【 ㄨ 1 B 】

【 ㄨ 2 B 】

204

File Edit View Window Help

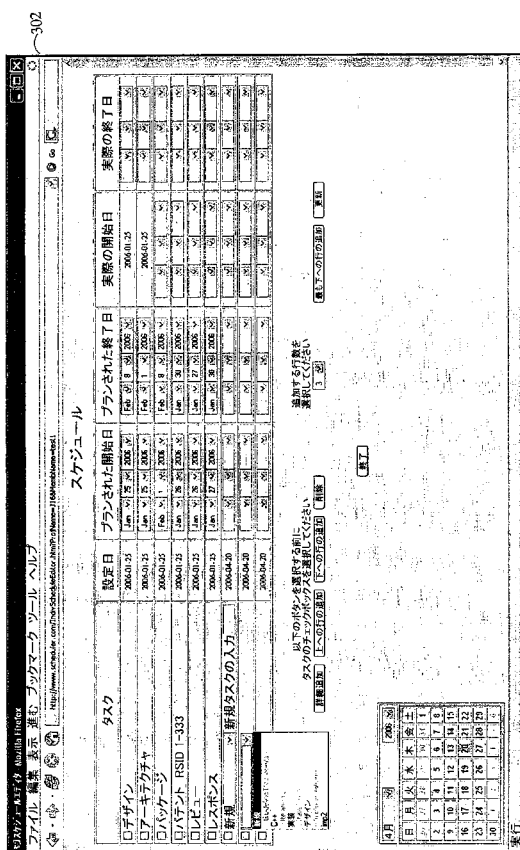
Schedule - Mozilla Firefox
 ファイル 編集 表示 進む プログマーク ツール ヘルプ

http://www.schedule.com/ProjectSchedule.htm

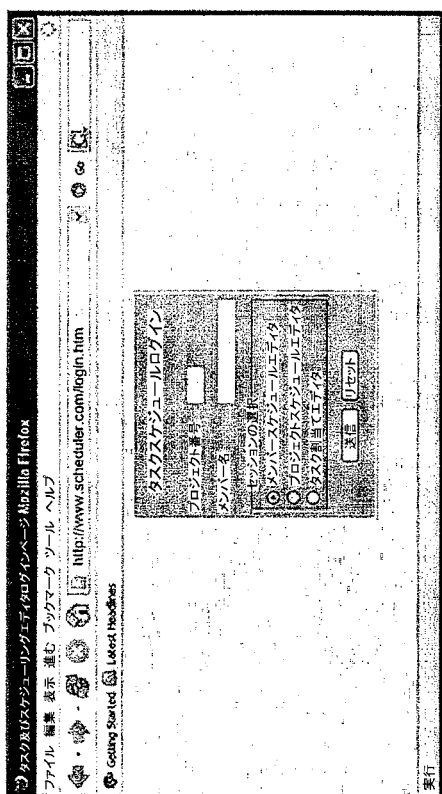
プロジェクトスケジュール

| タスク | 設定日 | プランされた 開始日 | プランされた 終了日 | 実際の開始日 | 実際の終了日 |
|-----|------------|---------------|---------------|------------|--------|
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| T1 | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 2005-09-18 | 2005-10-14 | | |
| | 2005-09-20 | 2005-09-20 | 2005-09-20 | | |
| MGR | 2005-04-21 | 2005-04-20 | 2005-10-13 | 2005-09-20 | |
| | 2005-09-19 | 200 | | | |

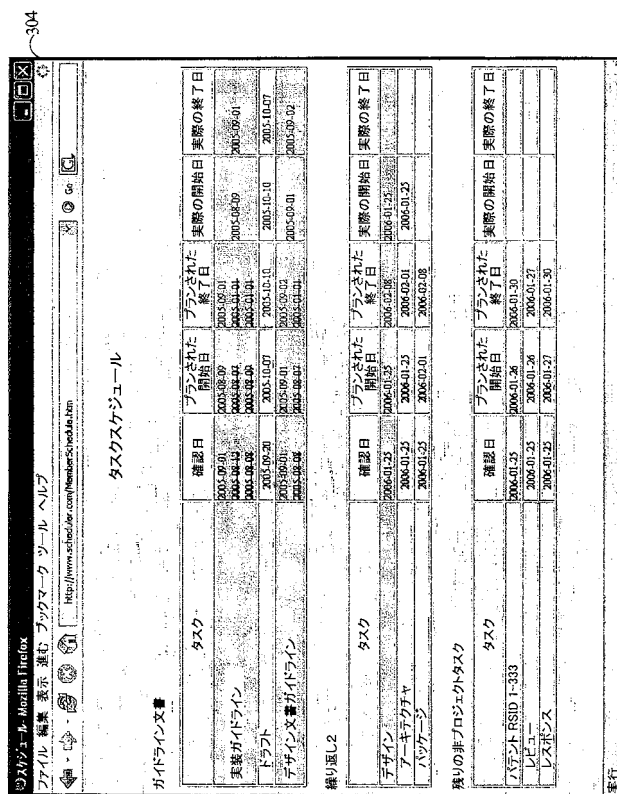
【 図 3 A 】



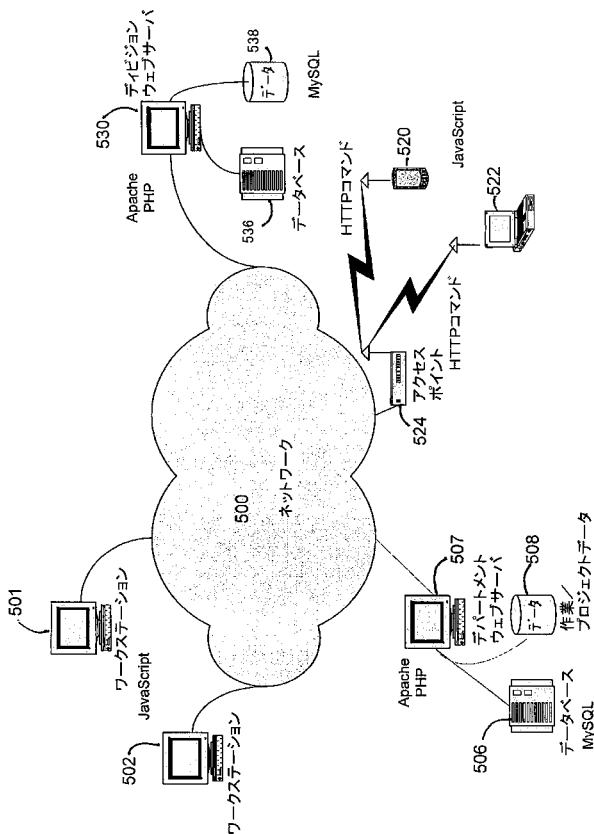
【 図 4 】



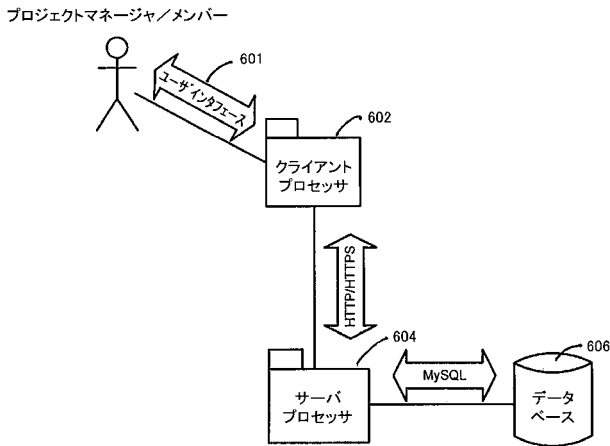
【 図 3 B 】



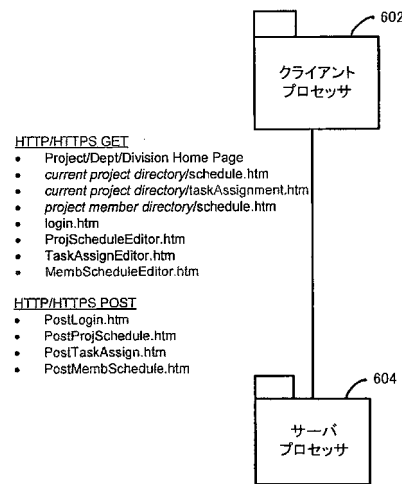
【 図 5 】



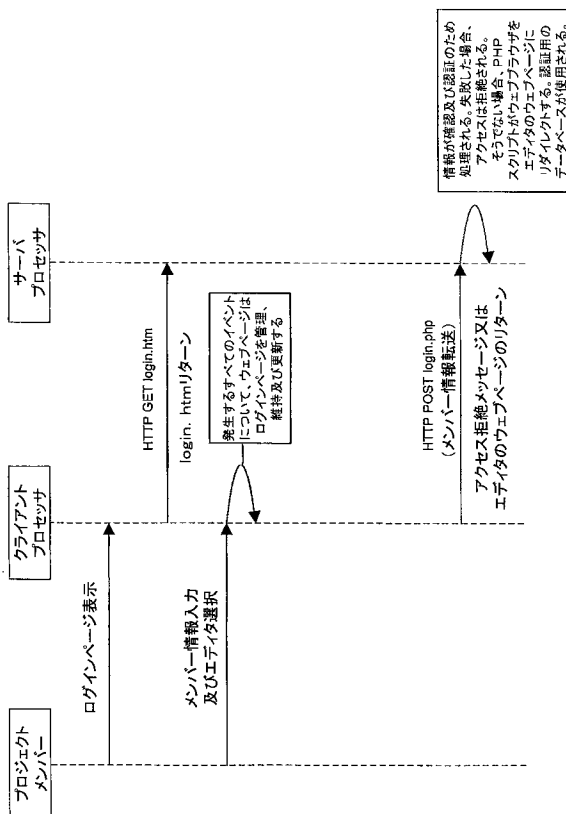
【 図 6 】



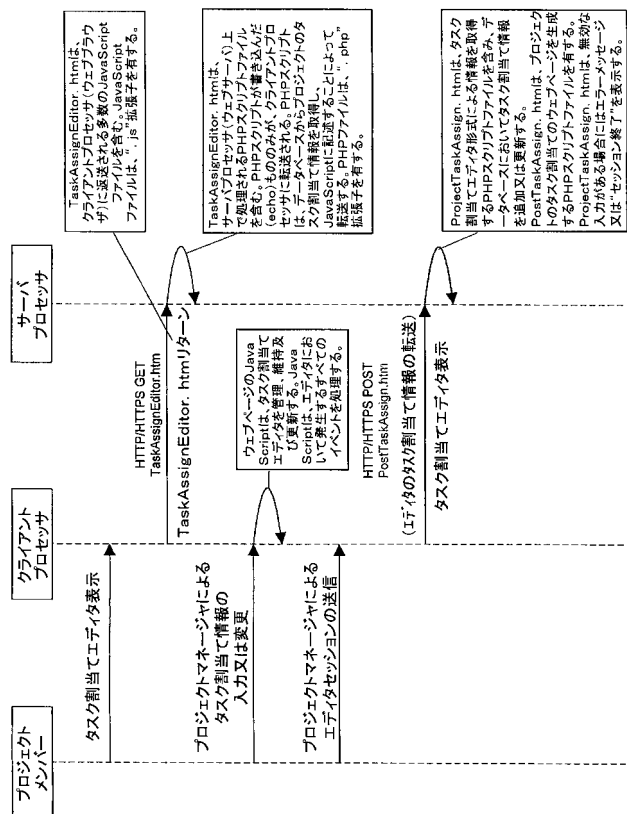
【 圖 7 】



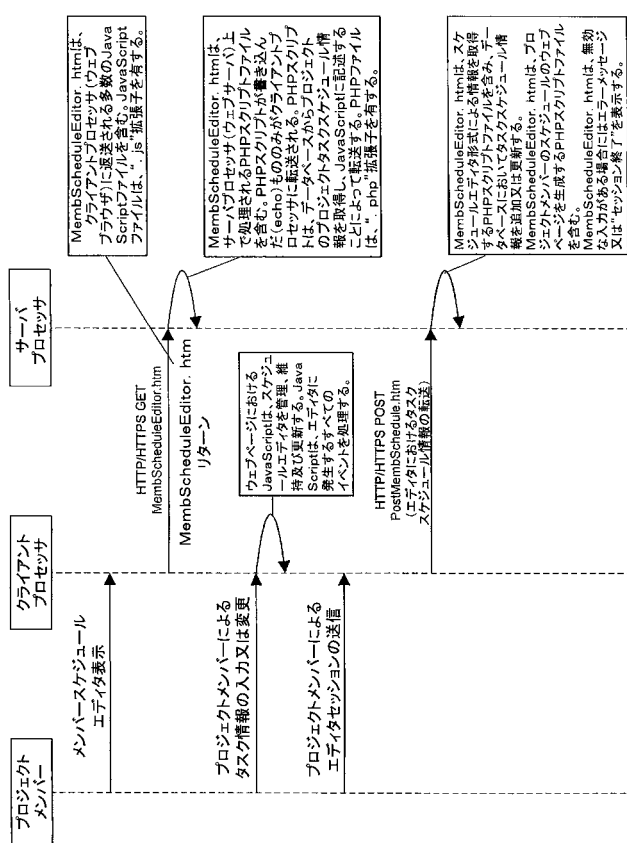
【 図 8 】



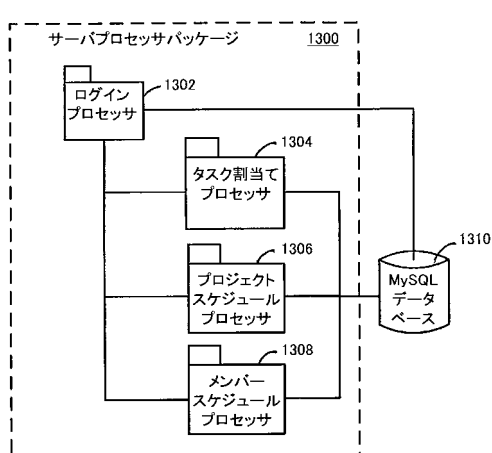
【圖 9】



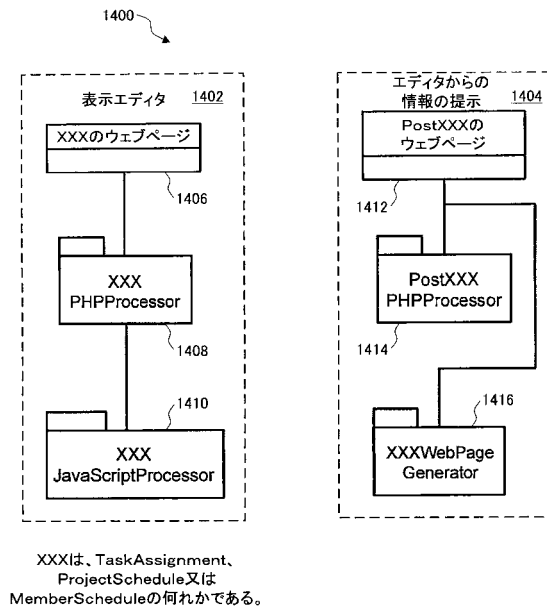
【 ㊦ 1 1 】



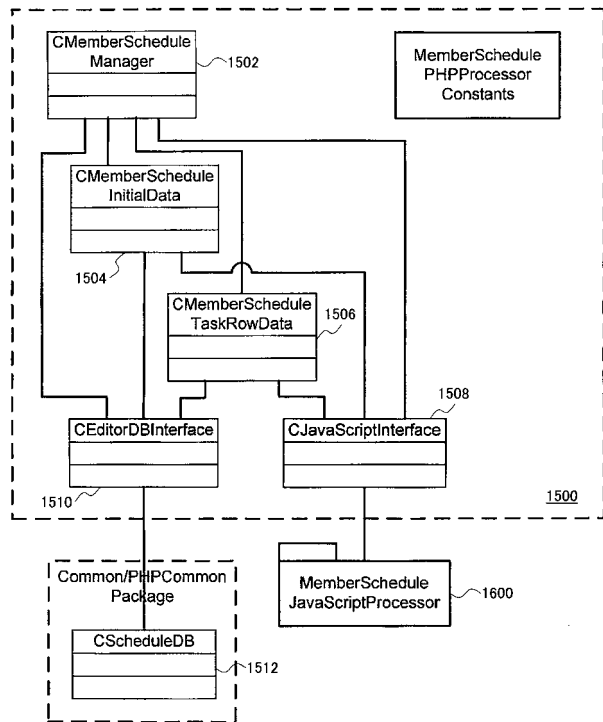
【 図 1 3 】



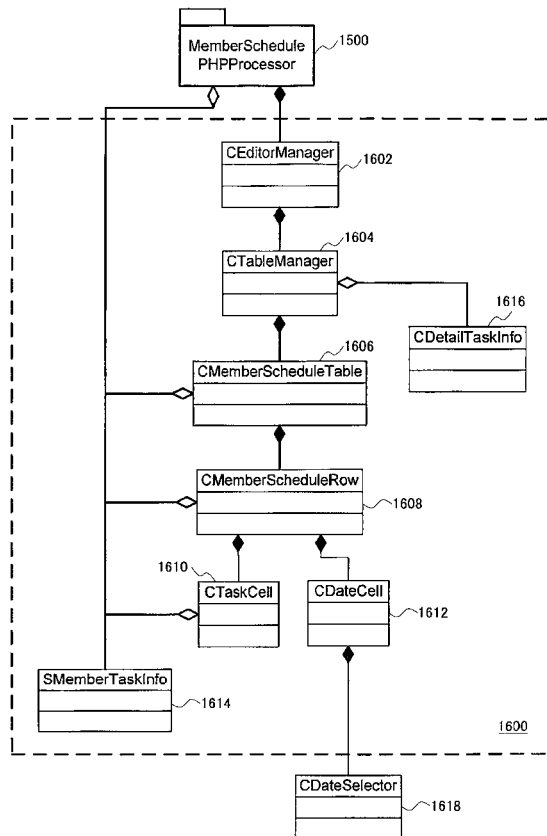
【図 14】



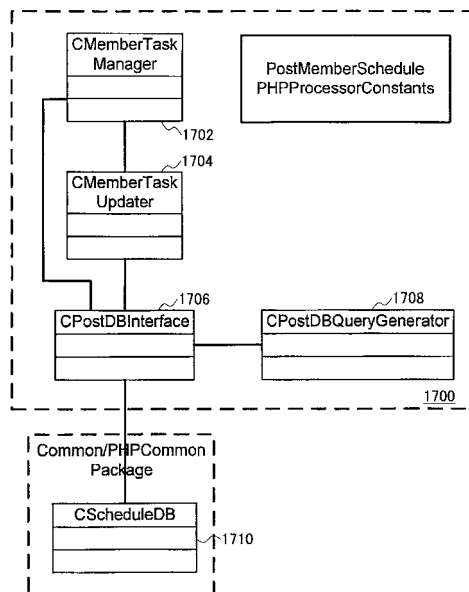
【図 15】



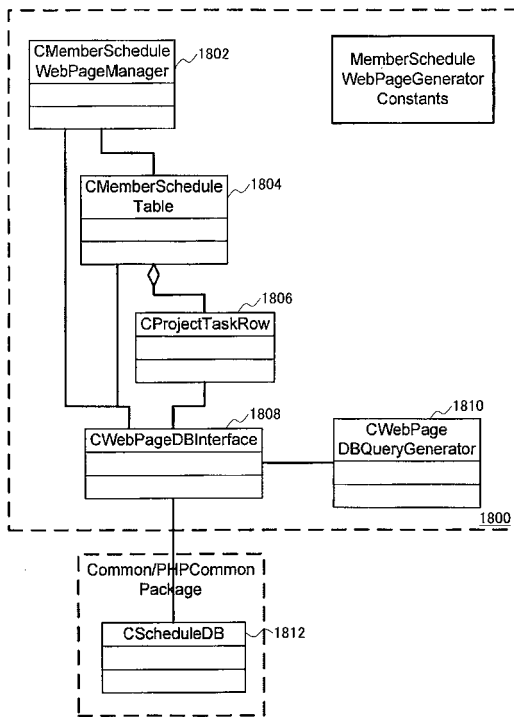
【図 16】



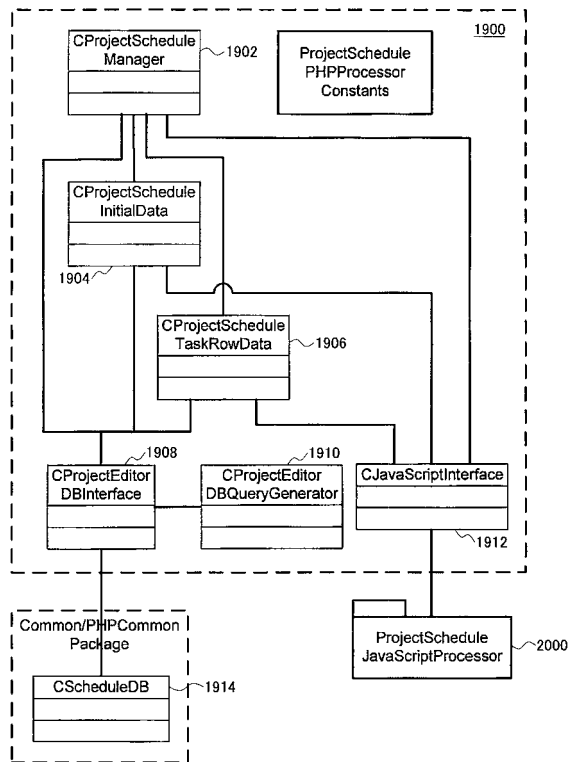
【図 17】



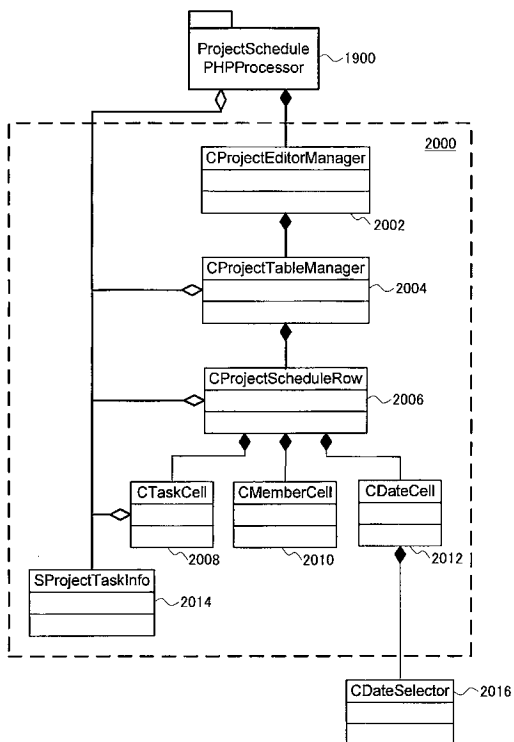
【図 18】



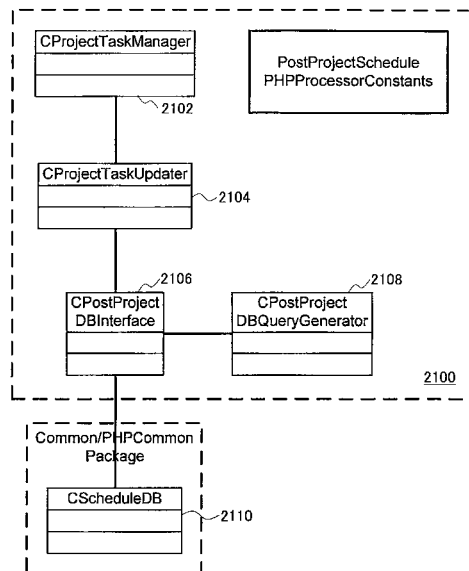
【図 19】



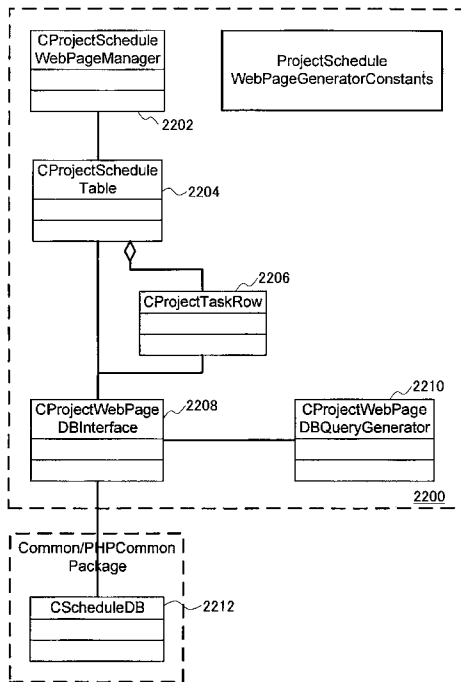
【図 20】



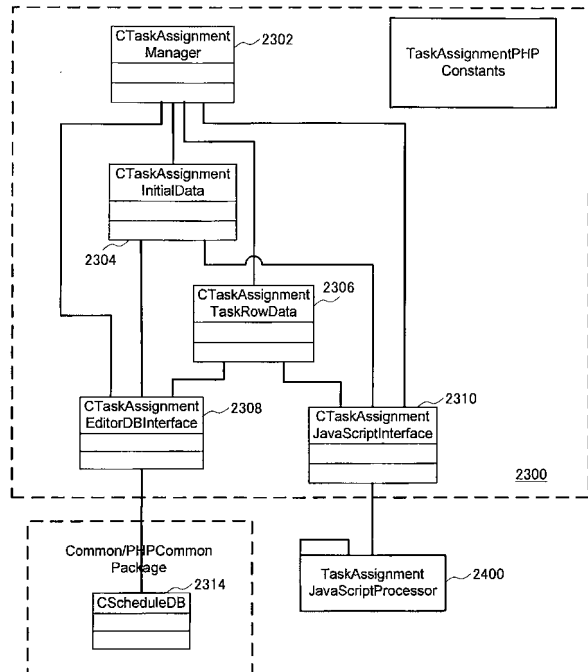
【図 21】



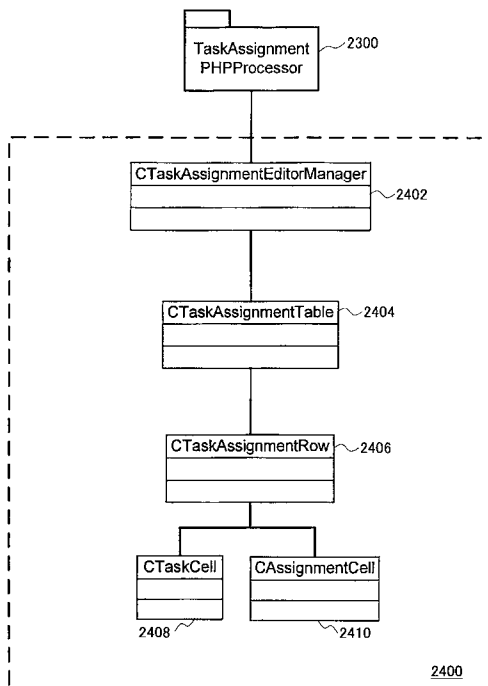
【 図 2 2 】



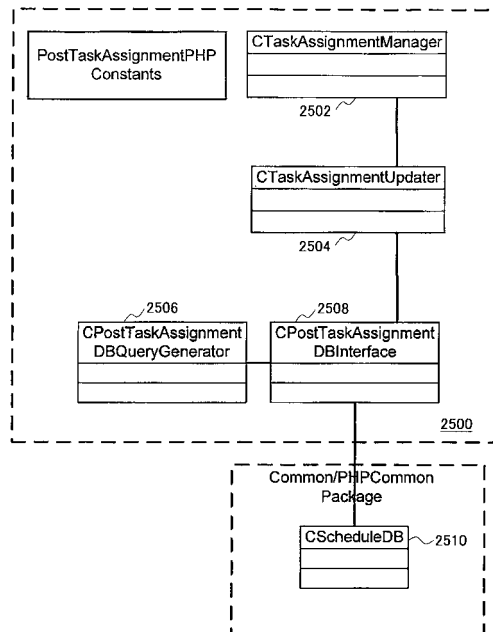
【 図 2 3 】



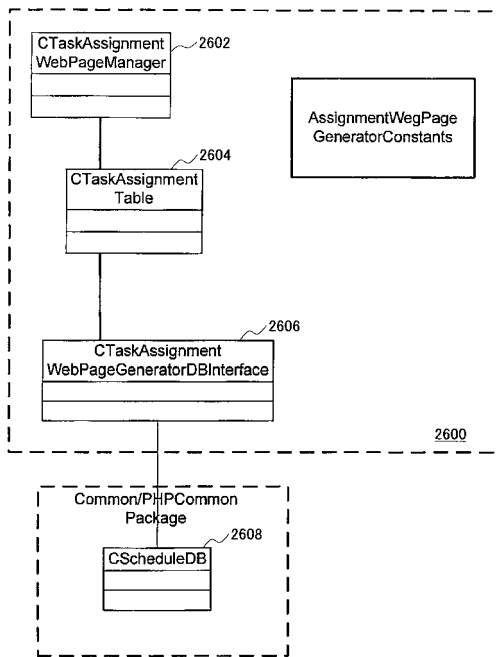
【 図 2 4 】



【 図 2 5 】



【図 26】



【図 27】

プレースホルダー文字列を有するコンスタントクエリ文字列

```
define('C_SelectHighestRevForMemberTaskQuery', "SELECT nLevelID, max(nScheduleRevNumber) FROM LevelMemberTask WHERE sProjectNumber=%% AND sMemberLabel=%% AND nProjectTaskID=%% AND nProjectTaskID=%% GROUP BY nLevelID");
```

プレースホルダー文字列のコンスタント

```
define('C_SelectAllForMemberTaskQuery', "SELECT * FROM LevelMemberTask WHERE sProjectNumber=%% AND nLevelID=%% AND nLevelID=%% ORDER BY nScheduleRevNumber DESC");
```

【図 28】

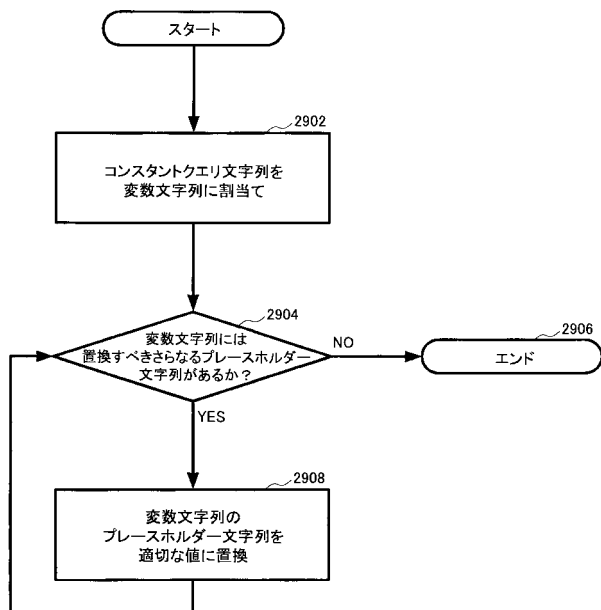
クエリ文字列を生成するためのスクリプト

```
$loc_Query = C_SelectHighestRevForMemberTaskQuery; (1)
$loc_Query = str_replace(C_ProjectNumberKey, "117", $loc_Query); (2)
$loc_Query = str_replace(C_MemberLabel, "T1", $loc_Query); (3)
$loc_Query = str_replace(C_ProjectTaskIDKey, "40", $loc_Query); (4)
```

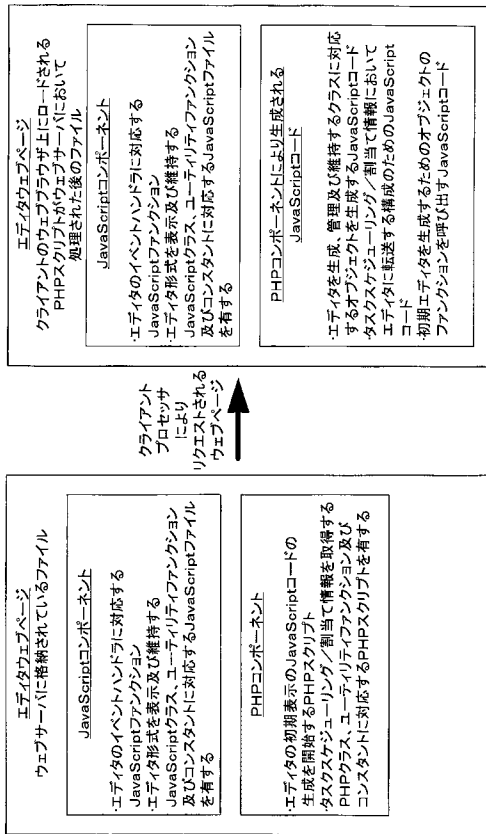
置換から生じる文字列

- 1 SELECT nLevelID, max(nScheduleRevNumber) FROM LevelMemberTask WHERE sProjectNumber=%% AND sMemberLabel=%% AND nProjectTaskID=%% AND nProjectTaskID=%% GROUP BY nLevelID
- 2 SELECT nLevelID, max(nScheduleRevNumber) FROM LevelMemberTask WHERE sProjectNumber=117 AND sMemberLabel=T1 AND nProjectTaskID=40 GROUP BY nLevelID
- 3 SELECT nLevelID, max(nScheduleRevNumber) FROM LevelMemberTask WHERE sProjectNumber=117 AND sMemberLabel=T1 AND nProjectTaskID=40 GROUP BY nLevelID
- 4 SELECT nLevelID, max(nScheduleRevNumber) FROM LevelMemberTask WHERE sProjectNumber=117 AND sMemberLabel=T1 AND nProjectTaskID=40 GROUP BY nLevelID

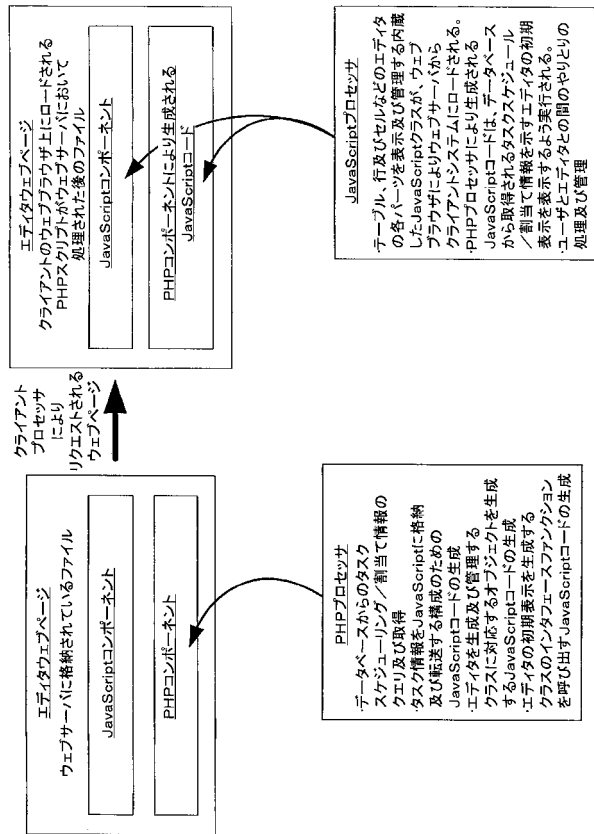
【図 29】



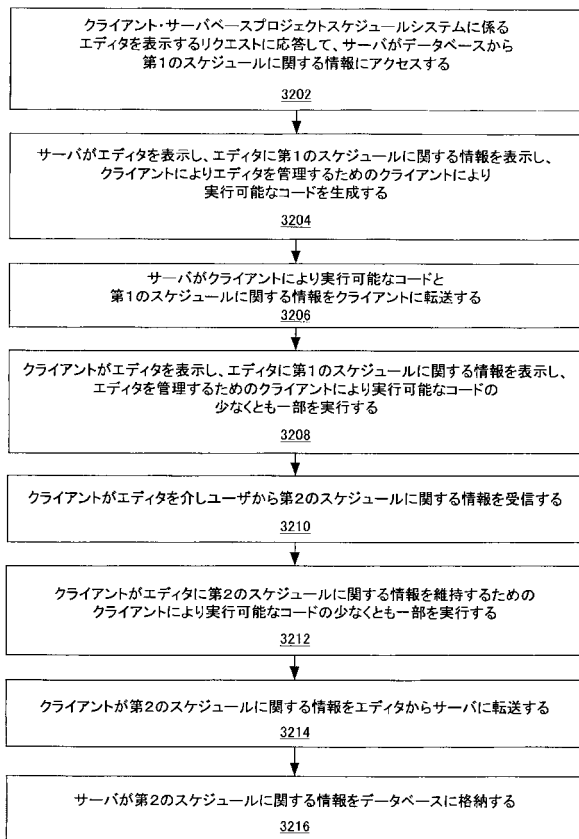
【図 3 0】



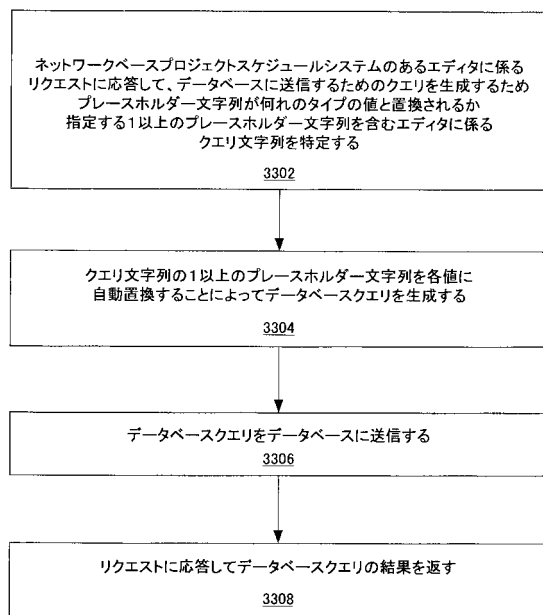
【図 3 1】



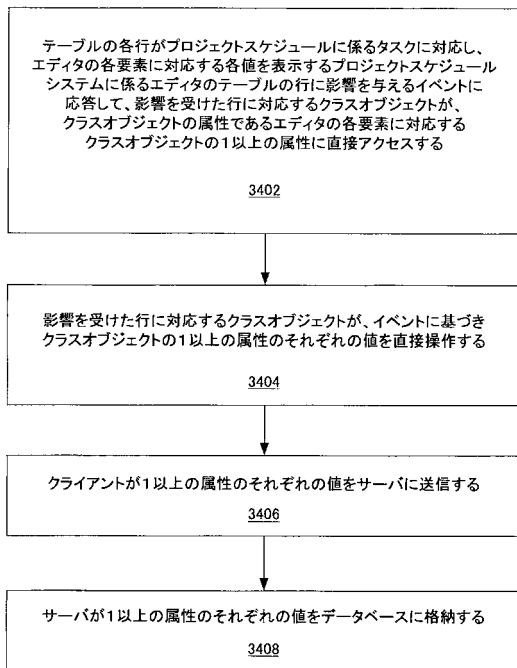
【図 3 2】



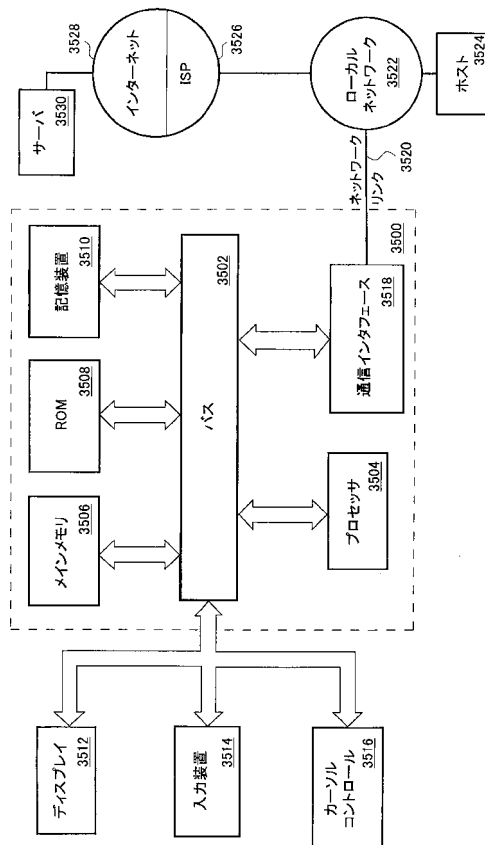
【図 3 3】



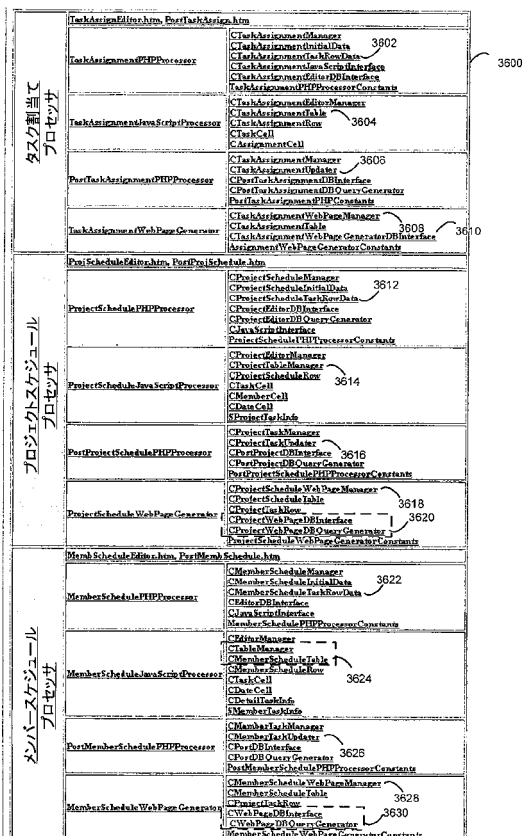
【図 3 4】



【図 3 5】



【図 3 6】



フロントページの続き

F ターム(参考) 5B019 KA04

5B185 BA06 BG07