

(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2009-15816
(P2009-15816A)

(43) 公開日 平成21年1月22日(2009.1.22)

(51) Int.Cl.
G06F 21/20 (2006.01)

F I
G06F 15/00 330A

テーマコード(参考)
5B285

審査請求 未請求 請求項の数 20 O L (全 21 頁)

(21) 出願番号 特願2008-41479 (P2008-41479)
(22) 出願日 平成20年2月22日(2008.2.22)
(31) 優先権主張番号 特願2007-149341 (P2007-149341)
(32) 優先日 平成19年6月5日(2007.6.5)
(33) 優先権主張国 日本国(JP)

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区丸の内一丁目6番6号
(74) 代理人 110000350
ポレール特許業務法人
(72) 発明者 小島 剛
神奈川県川崎市麻生区王禅寺1099番地
株式会社日立製作所システム開発研究所内
(72) 発明者 植田 良一
神奈川県川崎市麻生区王禅寺1099番地
株式会社日立製作所システム開発研究所内
Fターム(参考) 5B285 AA01 AA04 BA07 DA05

(54) 【発明の名称】 Webアプリケーションの構成方法とそのシステム

(57) 【要約】

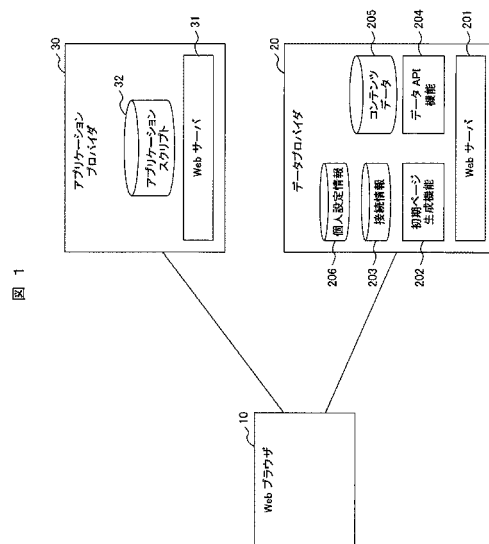
【課題】

Webアプリケーションの提供者に直接データを受け渡すことなくWebアプリケーションを利用可能とする。

【解決手段】

ユーザインタフェース機能を提供するアプリケーションプロバイダ30は、Webブラウザ10上で動作するアプリケーションスクリプト32を、ユーザが操作するWebブラウザ10に送信する。接続情報を用いてアプリケーションプロバイダ30と連携し、データ操作に関する機能を提供するデータプロバイダ20は、Webブラウザ10上でのユーザの操作に伴ってデータ操作処理が必要になった場合、データの追加・修正・削除・参照などの要求に応じて保持するデータを操作し、その結果の情報のみを要求元のWebブラウザ10に返信する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

Webブラウザを利用するユーザからの要求に基づき、Web上でアプリケーションのサービスを提供するWebアプリケーション構成方法であって、
前記Webブラウザに対し、ユーザインタフェース機能を提供するユーザインタフェースサーバと、データを保持し、当該データの操作に関するデータ操作機能を提供するデータ操作サーバとを用い、前記Webブラウザは、前記ユーザインタフェースサーバの前記ユーザインタフェース機能に基づき画面表示を行い、
前記データ操作サーバは、前記画面表示された前記Webブラウザから入力される要求に応じて、前記データ操作機能を実行し、操作結果を前記Webブラウザに送信する
Webアプリケーション構成方法。

10

【請求項 2】

請求項 1 記載のWebアプリケーション構成方法であって、
前記データ操作サーバは、前記データ操作サーバに接続するために必要な接続情報を保持し、当該接続情報に従って前記ユーザインタフェースサーバとの接続方法を設定し、前記データ操作機能と前記ユーザインタフェース機能とを連携して動作させる
Webアプリケーション構成方法。

【請求項 3】

請求項 2 記載のWebアプリケーション構成方法において、
前記データ操作サーバは、前記接続情報に従って前記ユーザインタフェースサーバとの接続方法を設定する処理を行う画面情報を生成し前記Webブラウザに提供する
Webアプリケーション構成方法。

20

【請求項 4】

請求項 2 記載のWebアプリケーション構成方法において、
前記データ操作サーバは、前記ユーザインタフェース機能の動作設定をユーザ毎に行うユーザ設定情報を更に保持し、該ユーザ設定情報と前記接続情報に従って、前記ユーザインタフェースサーバとの接続方法を設定する処理を行う画面情報を生成し前記Webブラウザに提供する
Webアプリケーション構成方法。

【請求項 5】

請求項 2 記載のWebアプリケーション構成方法において、
前記ユーザインタフェースサーバは、前記ユーザインタフェース機能の動作設定をユーザ毎に行うユーザ設定情報を保持し、該ユーザ設定情報に従って動作する
Webアプリケーション構成方法。

30

【請求項 6】

請求項 1 記載のWebアプリケーション構成方法において、
前記データ操作サーバは、保持する前記データを分析するデータ分析機能を更に具備する
Webアプリケーション構成方法。

【請求項 7】

請求項 6 記載のWebアプリケーション構成方法において、
前記データ操作サーバは、前記データ分析機能の分析結果を前記データ操作機能として提供する
Webアプリケーション構成方法。

40

【請求項 8】

請求項 6 記載のWebアプリケーション構成方法において、
前記ユーザインタフェースサーバの前記ユーザインタフェース機能は、前記データ操作サーバが提供する前記分析結果を利用する
Webアプリケーション構成方法。

【請求項 9】

ユーザからの要求に基づき、Web上でアプリケーションのサービスを提供するためのW

50

e bアプリケーション構成システムであって、
前記ユーザが要求を入力するW e bブラウザと、
前記ユーザの要求に対応するユーザインタフェース機能を提供するユーザインタフェースサーバと、
データを記憶し、データ操作に関する機能を構成するデータ操作サーバとを有し、
前記W e bブラウザは、前記ユーザインタフェースサーバから、前記ユーザインタフェース機能のためのアプリケーションスクリプトを受信し、受信した前記アプリケーションスクリプトを用いて画面表示し、
前記データ操作サーバは、前記W e bブラウザから入力される要求に応じて、前記データ操作機能を実行し、操作結果を前記W e bブラウザに送信する
W e bアプリケーション構成システム。

10

【請求項10】

請求項9記載のW e bアプリケーション構成システムであって、
前記データ操作サーバは、前記データ操作サーバに接続するために必要な情報を登録した接続情報を記憶し、前記接続情報に従って、前記ユーザインタフェースサーバとの接続方法を設定し、前記データ接続機能と前記ユーザインタフェース機能とを連携する
W e bアプリケーション構成システム。

【請求項11】

請求項10記載のW e bアプリケーション構成システムであって、前記データ操作サーバは、前記接続情報に従って前記ユーザインタフェースサーバとの接続方法を設定する処理を行う画面情報を生成し、生成した前記画面情報を前記W e bブラウザに提供する
W e bアプリケーション構成システム。

20

【請求項12】

請求項11記載のW e bアプリケーション構成システムであって、前記データ操作サーバは、前記ユーザインタフェース機能の動作設定をユーザ毎に行うユーザ設定情報を記憶し、前記ユーザ設定情報に従って、前記ユーザ毎に前記画面情報を生成する
W e bアプリケーション構成システム。

【請求項13】

請求項10記載のW e bアプリケーション構成システムであって、前記ユーザインタフェースサーバは、前記ユーザインタフェース機能の動作設定をユーザ毎に行うユーザ設定情報を記憶し、前記ユーザ設定情報に従って動作する
W e bアプリケーション構成システム。

30

【請求項14】

請求項9記載のW e bアプリケーション構成システムであって、
前記データ操作サーバは、前記データとしてコンテンツデータを記憶しており、
前記データ操作サーバは記憶する前記コンテンツデータを分析する分析機能を更に具備し、前記分析機能の分析結果を前記コンテンツデータの一部として記憶する
W e bアプリケーション構成システム。

【請求項15】

請求項14記載のW e bアプリケーション構成システムであって、
前記データ操作サーバは、前記分析結果を前記データ操作に関する機能として提供する
W e bアプリケーション構成システム。

40

【請求項16】

請求項14記載のW e bアプリケーション構成システムであって、
前記ユーザインタフェースサーバは、広告情報を記憶し、前記広告情報を配信する配信機能を有し、
前記データ操作サーバは、前記W e bブラウザからのコンテンツの特徴情報の取得要求に応答し、前記分析結果を前記特徴情報として前記W e bブラウザに提供し、
前記W e bブラウザは、前記特徴情報に適した広告情報の取得要求を前記ユーザインタフェースサーバに送信し、

50

前記ユーザインタフェースサーバの前記配信機能は、記憶する前記広告情報中前記特徴情報に関連するものを前記Webブラウザに配信するWebアプリケーション構成システム。

【請求項17】

請求項9記載のWebアプリケーション構成システムであって、前記ユーザインタフェースサーバと前記データ操作サーバが、データ管理企業内に設置されているWebアプリケーション構成システム。

【請求項18】

ユーザ企業からの要求に基づき、Web上でアプリケーションのサービスを提供するためのWebアプリケーション構成システムであって、

前記ユーザ企業が要求を入力するWebブラウザと、

前記ユーザ企業とネットワークを介して接続されるデータ管理企業内に設置され、前記ユーザ企業の要求に対応するユーザインタフェース機能を提供するユーザインタフェースサーバと、

前記データ管理企業内に設置され、データを記憶し、データ操作に関する機能を構成するデータ操作サーバとを有し、

前記Webブラウザは、前記ユーザインタフェースサーバから、前記ユーザインタフェース機能のためのアプリケーションスクリプトを受信し、受信した前記アプリケーションスクリプトを用いて画面表示し、

前記データ操作サーバは、前記Webブラウザから入力される要求に応じて、前記データ操作機能を実行し、操作結果を前記Webブラウザに送信する

Webアプリケーション構成システム。

【請求項19】

請求項18記載のWebアプリケーション構成システムであって、

前記データ操作サーバは、前記データ操作サーバに接続するために必要な情報を登録した接続情報を記憶し、前記接続情報に従って、前記ユーザインタフェースサーバとの接続方法を設定し、前記データ接続機能と前記ユーザインタフェース機能とを連携する

Webアプリケーション構成システム。

【請求項20】

請求項18記載のWebアプリケーション構成システムであって、前記データ操作サーバは、前記接続情報に従って前記ユーザインタフェースサーバとの接続方法を設定する処理を行う画面情報を生成し、生成した前記画面情報を前記Webブラウザに提供する

Webアプリケーション構成システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、Webアプリケーションのシステム構成方法に関し、ユーザ（利用者）の扱うデータの格納先を所望の場所に制限することのできる技術に関する。

【背景技術】

【0002】

これまでクライアント装置上で実行するデスクトップアプリケーションとして実装されてきた企業内の業務システムの多くが、Web上のサービス（Webアプリケーション）として提供されはじめている。このようなWeb上のサービスの利用が広まるに従い、ユーザ個人のプライバシー情報や社内の営業情報などをWebサーバ側に漏洩してしまう問題があった。

【0003】

この問題に対処すべく、特許文献1にサービスを提供するWEBサービス提供装置及び認証機関へのそれぞれの情報を送信する場合に行う暗号化、復号に必要なユーザの秘密鍵を格納する耐タンパ性記憶装置と、当該耐タンパ性記憶装置に格納されているユーザの秘密鍵により暗号化の処理を行うCPUを具備する装置によりユーザのプライバシー情

10

20

30

40

50

報を保護する方法が記載されている。

【0004】

この方法によれば、ユーザにとって煩雑な処理操作を低減させ、秘密鍵等のユーザのプライバシー情報が第三者に獲得される危険性を低減することができる。

【0005】

また、特許文献2に、選択されたキーワード及び選択された要素をそれぞれ秘密にして、送信された購入希望情報を受信すると、購入希望情報及び秘密鍵から送付鍵を計算し、利用者端末に送信し、利用者端末に、送付鍵、選択された要素、複数の情報から、選択されたキーワードに対応する情報のみを復号化させる装置により、利用者が選択した情報(そのアブストラクト)を情報提供者に秘匿した状態で、選択した情報の詳細を利用者に提供する方法が記載されている。

10

【0006】

この方法によれば、ユーザは情報提供者にどの情報を必要としているのかを秘匿したまま、情報提供者から取得したい情報を獲得することができる。

【0007】

【特許文献1】特開2003-304234号公報

【特許文献2】特開2006-39674号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

従来技術では、利用者側の装置や情報の提供者側の装置に情報秘匿のための特別な装置やプログラムを設置しなければならないという課題がある。また、情報秘匿の対象が認証処理の際に通信される情報であったり、取得したい情報を示す情報であったり、限定された場面や情報を対象としており、汎用的に活用できないという課題がある。

20

【0009】

本発明はこれらの課題に鑑みてなされたものであり、利用者側には特別な機能や装置を新たに導入することなく、且つ利用者のデータをアプリケーション提供者に対し渡すことなく、提供者の提供するアプリケーションを利用できるシステムを提供することにある。

【課題を解決するための手段】

【0010】

上記の目的を達成するため、本発明のシステムは、通常単一のサーバ(組織)で提供されるWeb上でサービスされるアプリケーション(以下、Webアプリケーション)の機能を二つに分割し、ユーザインタフェースの機能を提供するユーザインタフェースサーバ(組織)(以下、アプリケーションプロバイダと呼ぶ)とデータ操作に関する機能を提供するデータ操作サーバ(組織)(以下、データプロバイダと呼ぶ)の二つのサーバ(組織)によりWebアプリケーションの機能を提供するように構成する。

30

【0011】

旧来のWebアプリケーションでは、ユーザインタフェースの機能はWebアプリケーションを提供するサーバにより生成されるHTMLにより構成され、ユーザが情報を入力するとその情報がWebアプリケーションを提供するサーバに送信され、Webアプリケーションサーバは入力された情報を反映したHTMLを生成しそれをWebブラウザに送信して新たな画面を表示させる、という処理を繰り返すものである。データ操作に関する機能についても同一のサーバにより提供され、上記のユーザの入力を反映したHTMLを生成する際にデータ操作に関する処理も実行する。

40

【0012】

本発明のシステムによる、アプリケーションプロバイダはWebブラウザ上で動作するプログラム(以下、アプリプログラム)をユーザが操作するWebブラウザに提供する。このアプリプログラムはWebブラウザ上で動作するプログラムであるため、画面上の表示を変更する際にアプリケーションプロバイダのサーバへのアクセスの必要をなくすることができる。

50

【0013】

本発明のシステムによる、データプロバイダはユーザインタフェースを提供するためのHTMLの生成処理は行わず、データの追加・修正・削除・参照といった要求に応じて、データプロバイダ内のサーバ内で保持するデータを操作し、その結果の情報のみを要求元に返信する。

【0014】

アプリケーションにおいてユーザの操作に伴いデータ操作処理が必要な部分は、データプロバイダの通信の仕様に従ってデータプロバイダに対し通信を行うようにすることで、データプロバイダにデータ操作処理を行わせることができる。

【0015】

すなわち、本発明においては、Webブラウザを利用するユーザからの要求に基き、Web上でアプリケーションのサービスを提供するWebアプリケーション構成方法として、Webブラウザに対し、ユーザインタフェース機能を提供するユーザインタフェースサーバと、データを保持し、このデータの操作に関するデータ操作機能を提供するデータ操作サーバとを備え、Webブラウザがユーザインタフェースサーバからユーザインタフェース機能のためのアプリケーションプログラム（アプリケーションスクリプト）を受信し、受信したアプリケーションプログラム（アプリケーションスクリプト）に基づき画面を表示し、この画面表示されたWebブラウザから入力される要求に応じて、データ操作サーバがデータ操作機能を実行し、その操作結果をWebブラウザが受信することにより画面表示を更新するWebアプリケーション構成方法、及びその構成システムを提供する。

【発明の効果】

【0016】

この本発明によれば、Webアプリケーションの機能を、ユーザインタフェースの機能とデータ操作に関する機能との二つに分割し、それぞれ別のサーバに配備することができ、Webアプリケーションが扱うデータをユーザインタフェースの機能を提供するユーザインタフェースサーバに直接渡さずに、Webアプリケーションを利用することができるシステムをユーザに提供することができる。

【0017】

更に、好適にはデータ操作に関する機能を提供するデータ操作サーバをユーザの組織内やユーザの信頼する組織内に配備することで、ユーザの扱う情報を外部に漏洩することなく、ユーザの組織およびユーザの信頼する組織の外部の組織が提供するユーザインタフェースの機能を利用したWebアプリケーションを利用することができる。

【0018】

また更に、異なる複数のユーザインタフェース機能により共通のデータに対する操作を行うシステムを容易に提供することができる。更にまた、複数のユーザインタフェース機能を提供するサーバを利用対象として登録し選択可能にすることができるため、データを外部に送信するような不正なプログラムを排除することが可能である。

【発明を実施するための最良の形態】

【0019】

以下、添付の図面を参照しながら、本発明にかかわる実施の形態について説明する。

【実施例1】

【0020】

まず、図1により、第一の実施例に係るネットワークシステムの全体構成について説明する。

【0021】

実施例1に係るネットワークシステムには、ネットワークに接続された複数の情報処理装置10、20、30が含まれている。これらの情報処理装置の中には、(1)Webブラウザがインストールされた1台以上の情報処理装置(以下、Webブラウザ)10、(2)本システムで提供するWebアプリケーションの機能のうち、データの検索・追加・削除・編集・参照といったデータ操作に関する機能を提供する情報処理装置(以下、デー

10

20

30

40

50

タブロバイダ) 20、(3)本システムで提供するWebアプリケーションの機能のうちWebブラウザ上で提供するユーザインタフェースの機能を提供する情報処理装置(以下、アプリケーションプロバイダ)30が含まれている。

なお、図1ではWebブラウザ10はユーザの数だけ存在しうるが、ここでは説明のためひとつだけ示している。また、複数の情報処理装置はそれぞれ、通常のコンピュータシステムで構成され、処理部(CPU)、メモリやディスク装置などの記憶部、外部ネットワークに接続される通信部等を有することは言うまでもない。

【0022】

以下、実施例1の構成要素の機能および処理手順について説明する。データプロバイダ20は、ネットワークを介してWebブラウザ10からの処理要求を受け付け、処理要求に応じたWebページを返信するWebサーバ201と、本システムによるWebアプリケーションを実行するための初期ページを生成する初期ページ生成機能202と、アプリケーションプロバイダ30が提供するアプリケーションとしてのアプリケーションスクリプト32とデータプロバイダ20が提供する機能を接続するために必要な情報を保持する接続情報203と、本システムによるWebアプリケーションで扱うデータを操作する処理を行うためのユーザインタフェース機能を提供するデータAPI(Application Program Interface)機能204と、本システムによるWebアプリケーションで扱うコンテンツデータ205と、ユーザの個人設定情報206とから構成される。接続情報203、コンテンツデータ205、個人設定情報206は記憶部に記憶されており、初期ページ生成機能202とデータAPI機能204はプログラムであり、通常、上述の記憶部に記憶され、処理部で実行される。

10

20

【0023】

一方、アプリケーションプロバイダ30は、ネットワークを介してWebブラウザ10からの処理要求を受け付け、処理要求に応じたアプリケーションスクリプトを返信するWebサーバ31と、本システムによるWebアプリケーションの機能のうちユーザインタフェースの機能を提供するアプリケーションであるアプリケーションスクリプト32とから構成される。このアプリケーションスクリプト32等のプログラムは、通常、プロバイダ30を構成する情報処理装置の記憶部に記憶されている。

【0024】

実施例1のシステムはWebブラウザ10とデータプロバイダ20およびアプリケーションプロバイダ30との間でネットワークを介して通信を行いながら、データの操作処理(データの作成、編集、参照、削除)に関してはデータプロバイダ20のデータAPI機能を利用し、ユーザインタフェース機能についてはアプリケーションプロバイダ30の提供する機能を利用するアプリケーション実行環境をユーザに提供する。

30

【0025】

以下、実施例1のシステムの処理手順を具体的に説明する。

【0026】

ここでは本システムの適用例として予定管理アプリケーションを例にとって説明する。図11に予定管理アプリケーションが提供する画面の一例を示す。同図において、1101の部分は2007年2月21日から始まる予定を新規に追加するためのもので、ユーザが予定の情報を入力した状態であり、1102のaddボタンを押すことで予定情報を追加することができる。1103の部分はすでに登録されている予定情報を表示している部分で、2007年2月23日に「web meeting」の予定があることを示している。例にあげる予定管理アプリケーションは画面上にカレンダーを表示するとともにユーザの予定情報をそのカレンダー上に表示するものであり、ユーザは予定情報の追加・修正・削除・閲覧の操作を行うことができるものとする。

40

【0027】

予定管理アプリケーションを実施例1のシステムに適用する場合、アプリケーションプロバイダ30はWebブラウザ10の画面上にカレンダーや予定情報を表示したり、予定情報の操作処理をしたりする機能を提供するアプリケーションスクリプト32を提供し、デ

50

ータプロバイダ 20 は予定情報をコンテンツデータ 205 に保持し、予定情報の追加・修正・削除・参照の機能をデータ API 機能 204 により提供する。

【0028】

アプリケーションプロバイダ 30 が提供するアプリケーションスクリプト 32 は、そのデータ操作処理に関してはデータプロバイダ 20 が提供するデータ API 機能 204 にそってデータプロバイダ 20 に対し通信を行い、データプロバイダ 20 にデータ操作処理を行わせるように記述される。言い換えれば、本システムにおいては、データ API が仲介となってアプリケーションプロバイダ 30 とデータプロバイダ 20 が連携し動作する。

【0029】

図 2 はデータプロバイダ 20 の初期ページ生成機能 202 が生成する初期ページの HTML (HyperText Markup Language) の一例を示した図である。ただし、説明のため左端には行番号を付記している。

10

【0030】

図 3 は実施例 1 のシステムの処理手順を示すフロー図である。以下、実施例 1 の処理手順を図 3 のフローに従って説明する。なお、図 3 のうち、ステップ 304 以降の点線で囲われている部分は第二の実施例以降でも共通に用いる本システムにおける基本処理フローの部分である。

【0031】

はじめに、本システムを利用する際、ユーザは Web ブラウザ 10 に対しデータプロバイダ 20 が提供する初期ページにアクセスする操作を行う(ステップ 301)。Web ブラウザ 20 はユーザ操作により初期ページの取得要求をデータプロバイダ 20 に送信する(ステップ 302)。データプロバイダ 20 は Web サーバ 201 により初期ページの取得要求を受信すると、図 2 にその一例を示す初期ページを生成し、生成した初期ページを要求元の Web ブラウザ 10 に返信する(ステップ 303)。

20

【0032】

初期ページは、接続情報 203 に設定されている情報に従って、初期ページ生成機能 202 が生成する。図 4、図 5 はその接続情報 203 の一具体例であり、図 4 はアプリケーションプロバイダ 30 が提供するアプリケーションスクリプトの URL の情報(以下、アプリスクリプト情報)の一例であり、図 5 はデータプロバイダ 20 が提供するデータ API 機能 204 と URL との対応情報(以下、データ API 情報)の一例である。また、図 6 に個人設定情報 206 の一例を示す。

30

【0033】

初期ページは基本的に図 2 の一例のように、4 行目の破線で囲まれたアプリケーションプロバイダ 30 が提供するアプリケーションスクリプト 32 の参照記述部分と、5 行目から 14 行目までの破線で囲まれた、アプリケーションスクリプトに対しデータプロバイダ 20 が提供するデータ API 機能 204 と URL の対応情報の設定と、ユーザの個人設定を反映させるためのスクリプト記述の二つの部分からなる。

【0034】

一つ目のアプリケーションスクリプト 32 の参照記述部分は接続情報 203 に設定されるアプリスクリプト情報を元に生成する。図 4 のアプリスクリプト情報の一例の場合、例えば、ユーザが利用するアプリケーションスクリプトの名前が calendar app A であれば、http://appproviderA.com/script/calendar.js をアプリケーションスクリプトの参照記述に用いる。

40

【0035】

二つ目のアプリケーションスクリプトに対しデータプロバイダ 20 が提供するデータ API 機能と URL の対応情報の設定と、ユーザの個人設定を反映させるスクリプト記述の部分は、接続情報 203 に設定されるデータ API 情報と、個人設定情報 206 の情報を元に生成する。図 5 のデータ API 情報の一例の場合、提供するデータ API の名前が「GET」のものは「http://dataprovder.com/calendar/api/get/\$user\$」を、「ADD」のものは「http://data

50

provider.com/calendar/api/add/\$user\$」を、「EDIT」のものは「http://dataproducer.com/calendar/api/edit/\$user\$/ \$id\$」を、アプリケーションスクリプトに対し設定するURLとして用いる。

【0036】

なお、上記の例におけるURLのうち、「\$user\$」や「\$id\$」の部分はスクリプトの処理中に動的に変化する部分であることを意味しており、それぞれ操作対象のデータに対するユーザID、予定情報IDを当該部分に埋め込んで利用することを意味している。例えば、データAPI「GET」のURLは、ユーザID「kojima」のデータに対して処理を行いたい場合であれば「http://dataproducer.com/calendar/api/get/kojima」となる。また、図5中、「ANALYZE」は実施例1には関係しないため、ここでは説明を省略する。

10

【0037】

個人設定情報206が図6の例のように設定されていた場合、ユーザが「kojima」である場合、アプリケーションスクリプトに対する個人設定情報として、カレンダーの初期表示方法の設定情報として情報種別initViewの値はmonthly(レコード601)を、最初に表示する曜日の設定情報として情報種別initDayの値はsunday(レコード602)を、表示に利用する言語の設定情報として情報種別langの値はja(レコード603)を用いる。

20

【0038】

以上の手順によりデータプロバイダ20の初期ページ生成機能202は初期ページを生成する。Webブラウザ10は、このように生成された初期ページを受信すると画面上に初期ページとして受信したHTMLを表示する処理を行う(図3のステップ304)。受信した初期ページには図2の例のようにアプリケーションスクリプト32への参照記述が含まれているため、当該参照記述に記載されているURLのスクリプトの取得要求をアプリケーションプロバイダ30に送信する。アプリケーションプロバイダ30はWebサーバ31によりスクリプトの取得要求を受信すると、アプリケーションプロバイダ30が保持しているスクリプトのうち取得要求にそったアプリケーションスクリプト32を要求元のWebブラウザ10に返信する(ステップ305)。

30

【0039】

アプリケーションスクリプト32を受信したWebブラウザ10は、受信したアプリケーションスクリプト32の実行を開始する(ステップ306)。次に初期ページ内に記述されるスクリプトに従い、データプロバイダ20の提供するデータAPIのURL設定処理を行う(ステップ307)。当該設定処理に用いられる関数はアプリケーションプロバイダ30からのスクリプト32内で定義される。

40

【0040】

図2の例では8行目から10行目で定義されているデータAPIとURLの関係情報と個人設定情報206で設定される情報を引数とし、11行目の2001の箇所でexecuteCalendar関数を実行しているが、このexecuteCalendar関数はアプリケーションプロバイダ30が提供するスクリプト32内で定義され、2002の箇所で示す引数として渡されたデータAPIとURLの関係情報を取り込み、データ操作処理の際に利用するよう動作する。また2003の箇所で示す二つ目の引数の値に従い、ユーザインタフェースの動作を制御する。ここではユーザの個人設定の値として個人設定情報206に図6の一例のように設定されていた例となっており、アプリケーションスクリプトはカレンダーの初期表示時に月間表示をし、最初に表示する曜日を日曜日にし、表示に用いる言語に日本語を用いる。

40

【0041】

ステップ306でスクリプトの実行が開始されるが、ユーザの操作に伴いデータの操作を実行する処理はステップ307のデータAPIのURLの設定処理の後から開始するよう処理する。

50

【 0 0 4 2 】

ステップ 3 0 6 の後、アプリケーションスクリプト 3 2 の実行により、Web ブラウザ 1 0 は画面上に情報を表示したり、データを操作するためのユーザインタフェースを表示したりする。予定管理アプリケーションの場合、カレンダーを表示し、表示されている日付に対する予定情報を追加したり、表示されている日付に設定されている予定情報を編集したり削除したりするためのインターフェースを表示する。

【 0 0 4 3 】

次に、図 3 の処理フローに示すように、ユーザが Web ブラウザ 1 0 に対し、表示されている日付に対する予定情報を追加する操作を行うと（ステップ 3 0 8 ）、Web ブラウザ 1 0 は当該操作に対するイベントを発生させる。アプリケーションスクリプト 3 2 は当該イベントの発生を契機に、ユーザが入力した情報に基づき予定情報の追加処理を行う。ユーザが入力した情報に従い、データプロバイダが提供するデータ API の形式にそったリクエスト情報を生成し、当該リクエスト情報をデータプロバイダ 2 0 に非同期通信により送信する（ステップ 3 0 9 ）。

10

【 0 0 4 4 】

リクエスト情報を Web サーバ 2 0 1 により受信したデータプロバイダ 2 0 は、当該リクエスト情報に従いデータ API 機能 2 0 4 によりコンテンツデータ 2 0 5 に対し追加処理を行い、追加処理結果を要求元の Web ブラウザ 1 0 に返信する（ステップ 3 1 0 ）。

【 0 0 4 5 】

追加処理結果を受信した Web ブラウザ 1 0 は追加処理結果の情報を、追加要求送信処理を行ったアプリケーションスクリプト 3 2 の該当処理部に受け渡す。アプリケーションスクリプト 3 2 は追加処理結果の情報に従ってその旨を画面上に表示するなどの追加処理の結果に対する処理を行う（ステップ 3 1 1 ）。

20

【 0 0 4 6 】

図 1 0 は実施例 1 におけるデータ API の通信形式の一例を示した表である。アプリケーションスクリプト 3 2 はステップ 3 0 9 において図 1 0 の通信形式に従い、データプロバイダ 2 0 にリクエスト情報を送信する。また、ステップ 3 1 1 において、リクエスト情報の送信の結果得られる情報を解析し、結果の画面表示処理を行う。図 1 0 の 1 0 0 1 では、予定情報の追加処理の際、予定の開始日を 2 0 0 7 年 2 月 2 1 日、終了日を 2 0 0 7 年 2 月 2 2 日、タイトルを「web conference」、場所を「tokyo」とする予定を追加したい場合に「startdate = 2 0 0 7 0 2 2 1 & enddate = 2 0 0 7 0 2 2 2 & title = web + conference & place = tokyo」というリクエストパラメタを生成することを意味しており、その結果、追加処理が完了した場合「({ " status " : " ok " })」をデータプロバイダ 2 0 が返信することを意味している。また、1 0 0 2 は予定情報の検索処理の際、開始日が 2 0 0 7 年 1 月 1 日以降で「web」という文字列がタイトルや場所に含まれる、という条件で検索をする場合に「startdate = 2 0 0 7 0 1 0 1 & keyword = web」というリクエストパラメタを生成することを意味しており、その結果、条件に沿った予定情報として「({ " schedules " : [{ " startdate " : " 2 0 0 7 0 2 2 0 " , " enddate " : " 2 0 0 7 0 2 2 2 " , " title " : " web conference " , " place " : " tokyo " } , { " startdate " : " 2 0 0 7 0 2 2 3 " , " enddate " : " 2 0 0 7 0 2 2 3 " , " title " : " web meeting " , " place " : " meeting room " } ,] })」をデータプロバイダ 2 0 が返信することを意味している。

30

40

【 0 0 4 7 】

同様に、図 3 の処理フローにおいて、ユーザが Web ブラウザ 1 0 に対し、表示されている予定情報に対し修正する操作を行うと（ステップ 3 1 2 ）、Web ブラウザ 1 0 は当該操作に対するイベントを発生させる。アプリケーションスクリプト 3 2 は当該イベントの発生を契機に、ユーザが入力した情報に基づき予定情報の修正処理を行う。ユーザが入力した情報に従い、データプロバイダが提供するデータ API の形式にそったリクエスト

50

情報を生成し、当該リクエスト情報をデータプロバイダ 2 0 に送信する（ステップ 3 1 3）。

【 0 0 4 8 】

リクエスト情報を Webサーバ 2 0 1 により受信したデータプロバイダ 2 0 は、当該リクエスト情報に従いデータ API 機能 2 0 4 によりコンテンツデータ 2 0 5 に対し修正処理を行い、修正処理結果を要求元の Webブラウザ 1 0 に返信する（ステップ 3 1 4）。

【 0 0 4 9 】

修正処理結果を受信した Webブラウザ 1 0 は修正処理結果の情報を、修正要求送信処理を行ったアプリケーションスクリプト 3 2 の該当処理部に受け渡す。アプリケーションスクリプト 3 2 は修正処理結果の情報に従ってその旨を画面上に表示するなどの修正処理の結果に対する処理を行う（ステップ 3 1 5）。

10

【 0 0 5 0 】

その他の予定情報の削除や参照のユーザ操作に関しても同様に、Webブラウザ 1 0 がデータプロバイダ 2 0 に対し対応するリクエスト情報を送信し、当該リクエスト情報を受信したデータプロバイダ 2 0 がデータ操作処理を実行する。

【 0 0 5 1 】

なお、ステップ 3 0 8 ~ 3 1 1 とステップ 3 1 2 ~ ステップ 3 1 5 は説明上その順で述べたが、ユーザの操作を契機に開始されるステップであり、順序に関係なく処理可能である。

【 0 0 5 2 】

以上説明した実施例 1 のシステムによれば、そのユーザは、Webアプリケーションを利用する際、そこで扱うデータをWebアプリケーションのうちユーザインタフェース機能を提供するアプリケーションプロバイダ 3 0 に直接渡すことなく利用することができる。また、Webアプリケーションの開発者はデータ API の形式を仲介させることにより、Webアプリケーションのユーザインタフェース機能とデータ操作処理機能とを分離し、それぞれを別々のドメインのサーバ上に配備させることができる。

20

【 0 0 5 3 】

これにより、企業内システムの開発者は、インターネット上で公開されているアプリケーション機能に対応したデータプロバイダ 2 0 を自社内に設置することで、データを自社内にとどめたままインターネット上で公開されているアプリケーション機能を利用することができる。逆に、インターネット上のアプリケーション開発者は、企業内に設置されているデータプロバイダに対応するデータ API に対応したアプリケーション機能を開発することで、企業内のデータを直接扱うことなく、インターネット上にアプリケーションを配備しサービスすることができる。なお、個人設定情報 2 0 6 についてはアプリケーションプロバイダ 2 0 に配備することも可能である。

30

【 0 0 5 4 】

また、本実施例によるシステム構成にあつては、データプロバイダ 2 0 を、Webブラウザ 1 0 を利用する企業とは別のネットワークに設置して稼働させることも可能である。したがって、より安全にデータを保護することが可能なデータセンタ等にデータプロバイダ 2 0 を配備して利用する、という形態をとることも可能である。そのため、図 1 8 に示すように、顧客企業のデータを安全に保護しながら、外部のアプリケーションを活用可能とするといったデータ管理ビジネスにも応用できる。

40

【 0 0 5 5 】

図 1 8 において、3 0 A、3 0 B、3 0 C、3 0 D はいずれも図 1 のアプリケーションプロバイダ 3 0 と同様 Webサーバ 3 1 とアプリケーションスクリプト 3 2 を有しており、それぞれ別の企業が提供している様子を表している。アプリケーションプロバイダ 3 0 D とデータプロバイダ 2 0 はデータ管理企業が提供しており、そのデータ管理企業はアプリケーションの提供サービスとデータ管理のサービスを同時に行うビジネスを行っていることを示している。データプロバイダ 2 0 は図 1 のデータプロバイダと同様の構成を有する。これにより、Webブラウザ 1 0 のユーザである企業は、データ管理企業が提供する

50

アプリケーションの利用を図ることができると同時に、データの安全管理も保証される。

【実施例 2】

【0056】

上述した実施例 1 で Web アプリケーションの機能のうち、ユーザインタフェースの機能とデータ操作の処理機能とを分離する仕組みを説明したが、データプロバイダ 20 から提供される初期ページによりアクセスされるアプリケーションスクリプトが一種類に対応したものであった。データプロバイダが提供するデータ操作機能に複数のアプリケーションプロバイダが対応した機能を提供する場合、データプロバイダが提供するデータ操作機能を扱うユーザインタフェース機能をユーザが選択して利用可能である。実施例 1 で説明した初期ページ生成機能 202 にアプリケーションプロバイダの選択ページ生成機能を具備させることにより、この機能を提供することができる。

10

【0057】

以下、実施例 1 で説明したシステムに対し、更にアプリケーションプロバイダの選択処理を具備した第二の実施例のシステムの処理手順を図 7 のフロー図を用いて説明する。

【0058】

はじめに、ユーザは Web ブラウザ 10 に対しデータプロバイダ 20 が提供する、アプリケーションスクリプト選択ページ（以下、アプリ選択ページ）にアクセスする操作を行う（ステップ 701）。なお、アプリケーションスクリプト選択ページは、データプロバイダ 20 が提供するデータ操作機能に対応したアプリケーションスクリプトのうち、どのアプリケーションスクリプトを利用するかを選択するユーザインタフェースを備えた構成になっているものである。図 12 はアプリケーションスクリプト選択ページの一例であり、ボタンを押すことでそのボタンに記載されている名前のアプリケーションスクリプトを選択できる構成をとっている。

20

【0059】

Web ブラウザ 20 はユーザ操作により、アプリ選択ページの取得要求をデータプロバイダ 20 に送信する（ステップ 702）。データプロバイダ 20 は Web サーバ 201 によりアプリ選択ページの取得要求を受信するとアプリ選択ページを生成し、生成したアプリ選択ページを要求元の Web ブラウザ 10 に返信する（ステップ 703）。

【0060】

アプリ選択ページは接続情報 203 で保持するデータプロバイダ 20 が提供するデータ API に対応したアプリケーションスクリプトの情報（以下、アプリスクリプト対応情報）を元に初期ページ生成機能 202 が生成する。図 4 はアプリスクリプト情報の一例である。データプロバイダ 20 が提供するデータ API が「calendar API」という名前である場合、「calendar app A」という名前のアプリケーションスクリプトと「Bcalendar」というアプリケーションスクリプトがあることを意味している。そこで初期ページ生成機能 202 は選択可能なアプリケーションスクリプトとして「calendar app A」と「Bcalendar」があり、いずれかをユーザに選択させるアプリ選択ページを生成する。

30

【0061】

アプリ選択ページを受信した Web ブラウザ 10 はアプリ選択ページを画面上に表示する（ステップ 704）。

40

【0062】

ユーザが画面上に表示されたアプリケーションスクリプトの情報を閲覧し、いずれかのアプリケーションスクリプトを選択する操作を Web ブラウザ 10 に対して行う（705）。ユーザによるアプリケーションスクリプトの選択操作が行われると、Web ブラウザ 10 は選択されたアプリケーションスクリプトの情報とともに、選択されたアプリケーションスクリプトへの参照記述を含む初期ページの取得要求をデータプロバイダ 20 に送信する（ステップ 706）。データプロバイダ 20 は Web サーバ 201 により選択されたアプリケーションスクリプトに対する初期ページ取得要求を受信すると、初期ページ生成処理機能 202 により選択されたアプリケーションスクリプトに対する初期ページを生成

50

し、生成した初期ページを要求元のWebブラウザ10に返信する(ステップ706)。このときの初期ページの生成処理は実施例1で説明した図3のステップ303と同様である。

【0063】

初期ページを受信したWebブラウザ10は初期ページの表示処理を行う(ステップ304)。以降の処理フローは実施例1で説明した図3のステップ304以降の基本フローと同様である。

【0064】

本実施例によれば、同一のデータプロバイダ20が提供するデータAPIに対応した複数のアプリケーションスクリプトを選択して実行できるシステムをユーザに提供することができる。またこのときどのアプリケーションスクリプトを利用しても、扱われるデータはデータプロバイダ20にあるため、複数のアプリケーションスクリプトで共通のデータを操作することができる。

10

【実施例3】

【0065】

実施例2で複数のアプリケーションスクリプト32を選択して利用する機能について説明したが、複数のデータプロバイダが提供する情報を扱うことのできるシステムを構成することもできる。予定管理システムの例であれば、仕事に関する予定情報を自社内のデータプロバイダに管理し、祝日の情報や天気予報の情報を外部のデータプロバイダから取得して仕事に関する予定情報と並べて表示する、といったことができる。

20

【0066】

実施例1では接続情報203の一例として図5を示し、この情報に従ってデータプロバイダが提供するデータAPIの情報をアプリケーションプロバイダ30が提供するアプリケーションスクリプト32に設定する処理を含むよう初期ページを生成することで、アプリケーションスクリプト32がデータ操作処理を行う際にデータAPIの情報に従って通信処理を行えるようにする方法を述べた。これと同様に、他に取得したい情報を持つデータプロバイダの情報を接続情報203に保持させ、この情報をアプリケーションスクリプト32に設定する処理を含むよう初期ページを生成することで、アプリケーションスクリプト32が他のデータプロバイダが提供する情報を取得する処理を行う際にこの情報に従って通信処理を行うようにすることができる。

30

【0067】

この場合を第三の実施例として説明する。このとき、アプリケーションスクリプト32は実施例1で説明した予定情報管理アプリケーションのユーザインタフェース機能の他に、他のデータプロバイダが提供する情報を取得し、その情報をその情報の種類に応じた形式で予定情報と並べて表示する処理が行う機能を持っており、他のデータプロバイダとの通信に必要な情報を設定する手段を備えているものとする。

【0068】

図13は接続情報203に設定する、他に取得したい情報を持つデータプロバイダの情報の一例である。実施例1で説明した処理フローの中で、図3のステップ303において初期ページを生成する際に、図13のような情報に従い、他に取得したい情報を持つデータプロバイダの情報を設定するスクリプト記述(以下、追加データプロバイダ設定)を追加する。図13の1301は情報の種類が"holiday"でURLが"http://www.holidaycalendar.com/api/jsonp"であるデータプロバイダの情報が取得候補であることを示しており、1302は情報の種類が"weather"でURLが"http://www.weatherinformation.com/jsonpapi"であるデータプロバイダことを示している。

40

【0069】

アプリケーションスクリプト32が持つ他のデータプロバイダとの通信に必要な情報を設定する関数の名前が「setOptionalInformationURL」である場合、「setOptionalInformationURL({ "type":

50

" holiday ", " url " : " http : / / www . holiday calendar . com / api / jsonp " } , { " type " : " weather " , " url " : " http : / / www . weather information . com / jsonp api " }) ; 」という記述を追加生成することとなる。図2の初期ページの一例にこの処理が追加される場合は当該記述が11行目と12行目の間に挿入されることとなる。

【0070】

本実施例3によれば、複数のデータプロバイダが提供する情報をアプリケーションスクリプト32が利用することができ、多様な情報を表示するアプリケーションを構築することができる。なお、ここでは他のデータプロバイダに対してはデータ取得の操作のみを例としてあげたが、データ取得の操作に限定するものではなく、データの追加や編集、削除などの更新処理も可能である。

10

【実施例4】

【0071】

上述した各実施例ではユーザは本システムを利用するときに、はじめにWebブラウザ10を操作してデータプロバイダ20にアクセスするところから始めていたが、利用したいアプリケーションスクリプトを探しだしてから利用することも可能である。この場合を第四の実施例として説明する。

【0072】

アプリケーションスクリプトへの接続情報を設定する機能をデータプロバイダ20の初期ページ生成機能202に配備し、この機能を利用して接続情報203を設定するページ（以下アプリ利用開始ページ）をアプリケーションプロバイダが提供することで、アプリケーションスクリプトを探索し、探索した結果利用したいアプリケーションスクリプトが発見できたときにそれをデータプロバイダに設定し利用する、という利用の流れをユーザに提供することができる。

20

【0073】

以下、実施例1で説明したシステムに対し、アプリケーションスクリプトへの接続情報を設定する機能を具備したシステムの処理手順を図9のフロー図を用いて説明する。なお、図9は上記のうち、ユーザが利用したいアプリケーションスクリプトを発見し、アプリ利用開始ページを表示させる操作を開始するところから開始するフロー図である。

30

【0074】

はじめに、ユーザはWebブラウザ10に対しアプリケーションプロバイダ20が提供するアプリ利用開始ページにアクセスする操作を行う（ステップ901）。

【0075】

Webブラウザ20はユーザ操作により、アプリ利用開始ページの取得要求をアプリケーションプロバイダ30に送信する（ステップ902）。アプリケーションプロバイダ30はWebサーバ31によりアプリ利用開始ページの取得要求を受信するとアプリ利用開始ページを要求元のWebブラウザ10に返信する（ステップ903）。

【0076】

アプリ利用開始ページはユーザに、データプロバイダ20のアプリケーションスクリプトの接続情報を設定する機能にアクセスするためのURL（以下、接続情報設定用URL）を入力させるユーザインタフェースを備え、当該アプリケーションスクリプトの接続情報を入力されたURLに対し送信し、アプリケーションスクリプトをデータプロバイダに設定させ、アプリケーションスクリプトの利用を開始するための初期画面を生成させる機能を含んだページである。

40

【0077】

アプリ利用開始ページを受信したWebブラウザ10はアプリ利用開始ページを画面上に表示する（ステップ904）。

【0078】

ユーザが接続情報設定用URLを入力すると（ステップ905）、Webブラウザ10

50

は入力された接続情報設定用URLに対し、アプリケーションスクリプトの情報とともにアプリケーションスクリプトの接続設定および当該アプリケーションスクリプトに対する初期ページ生成要求をデータプロバイダ20（ユーザが入力する接続情報設定用URLに対応するデータプロバイダ20）に送信する（ステップ906）。データプロバイダ20はWebサーバ201により接続設定および初期ページ取得要求を受信すると、初期ページ生成処理機能202により接続情報203に対しアプリケーションスクリプトの情報を追加設定し、追加設定したアプリケーションスクリプトに対する初期ページを生成し、生成した初期ページを要求元のWebブラウザ10に返信する（ステップ907）。このときの初期ページの生成処理は実施例1で説明した図3のステップ303と同様である。

【0079】

初期ページを受信したWebブラウザ10は初期ページの表示処理を行う（ステップ304）。以降の処理フローは実施例1で説明した図3のステップ304以降の基本フローと同様である。

【0080】

本実施の形態によれば、ユーザがWebブラウザ10を操作してアプリケーションスクリプトをネットワーク内で探索し、探索した結果利用したいアプリケーションスクリプトが発見できたときにそれをデータプロバイダに設定し利用する、という利用の流れをユーザに提供することができる。

【実施例5】

【0081】

実施例1から実施例4ではアプリケーションの機能のうちデータ操作に関する処理はデータプロバイダに、ユーザインタフェースに関する処理はアプリケーションプロバイダに配備するものとして説明したが、データ操作に関する処理のうち、データAPI機能以外の、検索処理の高速化のためのインデックス作成処理や、コンテンツ内容に即した広告配信のためのコンテンツ内容の特徴抽出処理などのデータ分析処理機能もデータプロバイダ20に配備する必要がある。

【0082】

以下、実施例1で示した予定情報管理アプリケーションのシステム例に対し、コンテンツ内容に即した広告を画面上に表示する処理も行うシステムを第五の実施例として説明する。

【0083】

図14は実施例5におけるシステム構成図である。図1と異なる構成として、データプロバイダ20に対しデータ分析機能207が、アプリケーションプロバイダ30に対し広告配信機能33ならびに広告情報34が追加されている。なお、データAPI機能204はデータ分析機能207により抽出されたコンテンツデータに対する特徴となる情報を提供する機能を新たに具備する。この新たに具備する機能は図5のデータAPI情報の最後の4行目で記録されているAPIメソッド、ANALYZEの機能として提供されるものとし、そのURLはhttp://dataproducer.com/calendar/api/analyze/\$user\$であるとする。アプリケーションスクリプト32はこのURLに対しアクセスすることで、データ分析機能207により抽出されたコンテンツデータに対する特徴となる情報を取得できる。

【0084】

またアプリケーションスクリプト32はデータAPI機能204からコンテンツデータに対する特徴となる情報を取得し、取得した特徴情報を広告配信機能33に送信し、関連する広告情報を取得し、画面上に表示する処理機能を新たに具備する。

【0085】

このデータ分析機能207は上記のようなデータAPI機能以外の、検索処理の高速化のためのインデックス作成処理や、コンテンツ内容に即した広告配信のためのコンテンツ内容の特徴抽出処理などのデータ分析処理を行う機能である。ここではデータ分析機能207はあらかじめコンテンツデータ205が保持する内容を分析し、各コンテンツ内容の

10

20

30

40

50

特徴情報をコンテンツデータの一部として記録しておくものとする。

【 0 0 8 6 】

一方、広告配信機能 3 3 はコンテンツ内容に即した広告を配信する機能であり、コンテンツ内容の特徴となるキーワードを Web サーバ 3 1 で受信すると、そのキーワードに適した、関連する広告を広告情報 3 4 から抽出し要求元に広告情報を返信する機能である。

【 0 0 8 7 】

図 1 5 はアプリケーションプロバイダ 3 0 の広告配信機能 3 3 により送信された広告情報を Web ブラウザ 1 0 の画面上に表示した一具体例であり、1 5 0 1 がその広告の一例である。

【 0 0 8 8 】

図 1 7 は広告情報 3 4 の一例であり、特徴情報と配信すべき広告画像の URL の対応関係を示している。図 1 7 の 1 7 0 1 では特徴情報が web であった場合は `http://www.add.com/image/001.png` を、1 7 0 2 では特徴情報が network であった場合は `http://www.add.com/image/002.png` を、1 7 0 3 では特徴情報が application であった場合は `http://www.add.com/image/003.png` を、特徴情報に関連する、配信すべき広告画像として設定していることを示している。広告配信を行う予定情報管理アプリケーションの処理フローは、広告配信に関する部分以外、実施例 1 で説明した図 3 の予定情報管理アプリケーションの処理フローと大きな差はない。

【 0 0 8 9 】

図 1 6 は、実施例 5 のシステムにおいて、図 3 のステップ 3 0 7 に続いて行われる広告配信のための処理フローである。以下、図 1 6 を用いて広告配信のための処理フローを説明する。

【 0 0 9 0 】

アプリケーションスクリプト 3 2 の実行が開始され (ステップ 3 0 6)、データ API 情報の設定処理が完了 (ステップ 3 0 7) した後、表示中のコンテンツの特徴情報を取得するため、Web ブラウザ 1 0 はユーザの指示に基づき、データプロバイダ 2 0 に対しコンテンツの特徴情報の取得要求を送信する (ステップ 1 6 0 1)。データ API 機能 2 0 4 は Web サーバ 2 0 1 により特徴情報の取得要求を受信すると、取得要求により指定されたコンテンツに対する特徴情報をコンテンツデータ 2 0 5 から取得し、取得した特徴情報を要求元の Web ブラウザ 1 0 に返信する (ステップ 1 6 0 2)。特徴情報を受信した Web ブラウザ 1 0 はその情報を、特徴情報の取得要求送信処理を行ったアプリケーションスクリプト 3 2 の該当処理部に受け渡す。アプリケーションスクリプト 3 2 は特徴情報に適した広告情報の取得要求をアプリケーションプロバイダ 3 0 に送信する (ステップ 1 6 0 3)。

【 0 0 9 1 】

アプリケーションプロバイダ 3 0 の広告配信機能 3 3 は Web サーバ 3 1 により広告情報の取得要求を受信すると受信した特徴情報に即した情報を広告情報 3 4 から取得し、要求元の Web ブラウザ 1 0 に返信する (ステップ 1 6 0 4)。広告情報を受信した Web ブラウザ 1 0 はその情報を、広告情報の取得要求送信処理を行ったアプリケーションスクリプト 3 2 の該当処理部に受け渡す。アプリケーションスクリプト 3 2 は受信した広告情報に従い画面上に広告情報を表示する (ステップ 1 6 0 5)。

【 0 0 9 2 】

例えば、ステップ 1 6 0 2 でコンテンツの特徴情報として "web" をデータプロバイダ 2 0 が返信した場合、ステップ 1 6 0 3 で "web" に適した広告情報の取得要求をアプリケーションスクリプト 3 2 はアプリケーションプロバイダ 3 0 に送信する。そしてステップ 1 6 0 4 で "web" に適した広告情報として "http://www.add.com/image/001.png" (図 1 7 の 1 7 0 1 参照) をアプリケーションスクリプト 3 2 に返信し、アプリケーションスクリプト 3 2 は "http://www.add.com/image/001.png" で示される URL の画像を Web ブラ

10

20

30

40

50

ウザ 10 上に表示する。

【0093】

本実施例によればデータプロバイダ 20 が保持するデータと密接に関わる、データ A P I 機能として外部に公開しないデータ分析処理のような機能が必要となるシステムを提供可能となり、コンテンツに即した広告配信の機能や高速な検索処理の機能を提供することができる。

【0094】

これまでデスクトップアプリケーションとして提供されてきたアプリケーションが W e b アプリケーションとしてインターネット上で公開されはじめている。しかし情報漏洩の観点から企業ユーザがこうした W e b アプリケーションを使うことは少なかった。

10

【0095】

それに対し、本発明を用いることで、W e b アプリケーションの機能をデータ操作に関する機能とユーザインタフェースに関する機能の二つにそれぞれ別のドメイン（別のサーバ）上のサービスとして分割することができるため、データ操作に関する機能を提供するサーバを自社内（もしくは信頼できる社外の組織）に配備することにより、ユーザインタフェースに関する機能を提供するサーバ（を提供する組織）に直接データを渡さず、自社内などに情報を置いたままアプリケーションの機能を利用することができる。

【0096】

以上詳述した本発明により、企業内ユーザによるインターネット上の W e b アプリケーション利用が拡大すれば、本発明を用いたユーザインタフェースに関する機能のみを提供するサービスや、信頼性の高いデータ操作機能サーバを提供もしくは構築するサービスなどの事業が広まると考えられる。

20

【図面の簡単な説明】

【0097】

【図 1】実施例 1 のシステムの機能構成およびネットワーク構成を示す図。

【図 2】実施例 1 の初期ページの一例を示す図。

【図 3】実施例 1 によるシステムの処理フローを示す図。

【図 4】実施例 1 における接続情報が保持するアプリケーションスクリプト情報を示す図

。

【図 5】実施例 1 における接続情報が保持するデータ A P I 情報の一例を示す図。

30

【図 6】実施例 1 における個人設定情報 206 の一例を示す図。

【図 7】実施例 2 における本発明によるシステムの処理フローを示す図。

【図 8】実施例 3 における接続情報が保持するデータ A P I 情報の一例を示す図。

【図 9】実施例 4 におけるシステムの処理フローを示す図。

【図 10】実施例 1 におけるデータ A P I の通信形式を示す図。

【図 11】実施例 1 における予定管理アプリケーションが提供する画面の一例を示す図。

【図 12】実施例 2 におけるアプリケーションスクリプト選択ページの一例を示す図。

【図 13】実施例 1 における追加データプロバイダの情報の一例を示す図。

【図 14】実施例 5 におけるシステムの機能構成およびネットワーク構成を示す図。

【図 15】実施例 5 における予定管理アプリケーションが提供する画面の一例を示す図。

40

【図 16】実施例 5 におけるシステムの処理フローを示す図。

【図 17】実施例 5 における広告情報 34 の一例を示す図。

【図 18】実施例 1 における応用例のネットワーク構成を示す図。

【符号の説明】

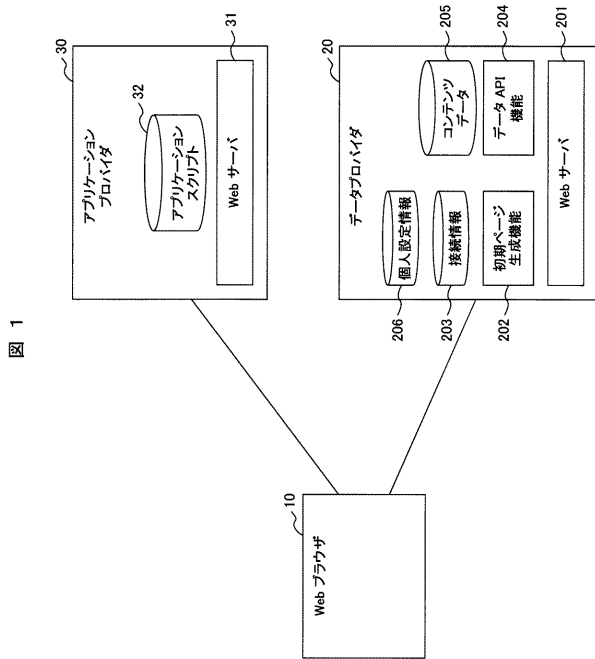
【0098】

10 ... W e b ブラウザ、20 ... データプロバイダ、30、30A、30B、30C、30D ... アプリケーションプロバイダ、31 ... W e b サーバ、32 ... アプリケーションスクリプト、33 ... 広告配信機能、34 ... 広告情報、201 ... W e b サーバ、202 ... 初期ページ生成機能、203 ... 接続情報、204 ... データ A P I 機能、205 ... コンテンツデータ、206 ... 個人設定情報、207 ... データ分析機能、601 ~ 603 ... 個人設定情報のレ

50

コード、1701~1703...特徴情報と配信すべき広告画像のURLの対応関係。

【 図 1 】



【 図 2 】

```

1: <html>
2: <head>
3: <title>カレンダーアプリ</title>
4: <script src = "http://appproviderA.com/script/calendar.js"></script>
5: </script>
6: <!--//
7: function init () {
8:   var urls = ["get": "http://datapvider.com/calendar/api/get/$user$",
9:               "add": "http://datapvider.com/calendar/api/add/$user$",
10:              "edit": "http://datapvider.com/calendar/api/edit/$user$/id$"];
11:   ;executeCalendar({urls: ["initView", "monthly", "initDay", "Sunday", "lang", "ja"]});
12: }
13: //-->
14: </script>
15: </head>
16: <body onload = "init 0;">
17: </body>
18: </html>

```

図 2

【 図 3 】

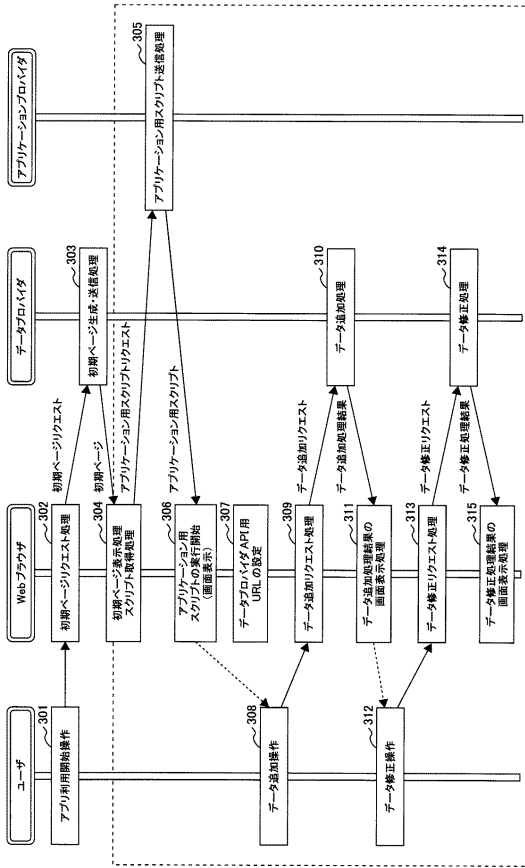


図 3

【 図 4 】

API名	アプリケーション名	アプリケーションURL
calendar API	calendar app A	http://appproviderA.com/script/calendar.js
	Bcalendar	http://app.b.net/cal.js
blog API	blog app C	http://blogc.com/blog.js
	Dblog	http://www.d.net/app/dblog.js

図 4

【 図 5 】

APIメソッド名	API URL
GET	http://dataprovder.com/calendar/api/get/\$user\$
ADD	http://dataprovder.com/calendar/api/add/\$user\$
EDIT	http://dataprovder.com/calendar/api/edit/\$user\$/\$id\$
ANALYZE	http://dataprovder.com/calendar/api/analyze/\$user\$

図 5

【 図 6 】

ユーザ名	情報種別	設定値
Kojima	initialView	monthly
Kojima	initialDay	sunday
Kojima	language	japanese

図 6

【 図 7 】

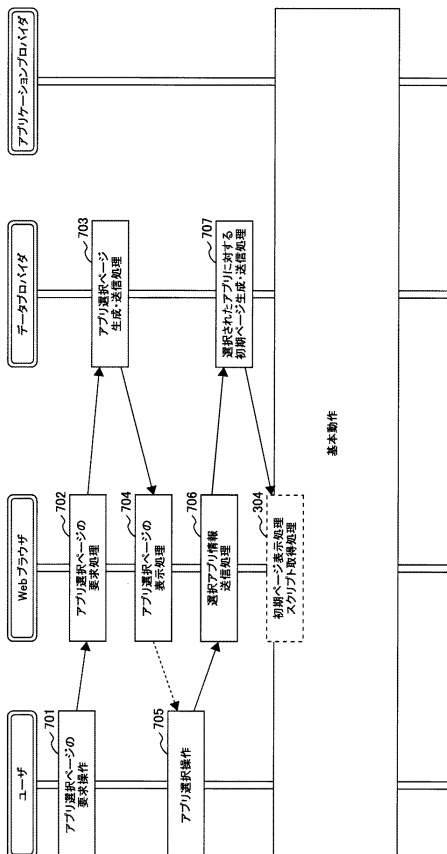


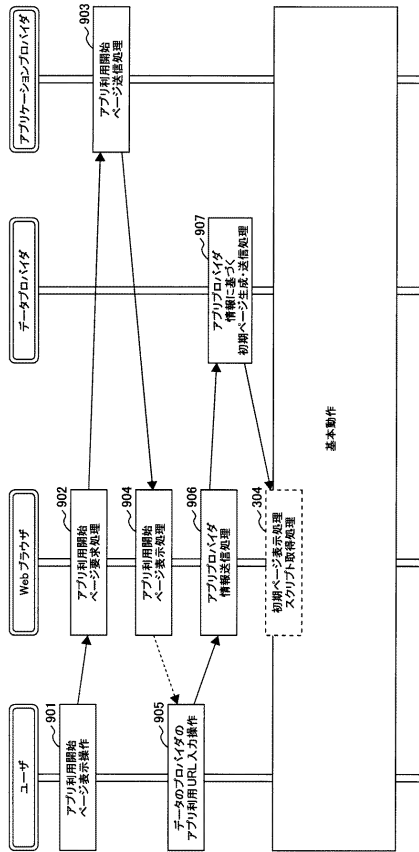
図 7

【 図 8 】

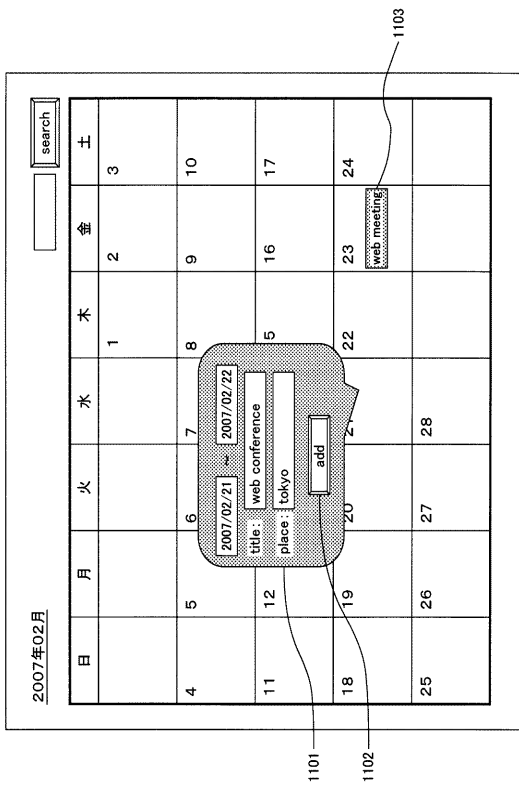
API名	APIメソッド名	API URL
calendar API	GET	http://dataprovder.com/calendar/api/get/\$user\$
	ADD	http://dataprovder.com/calendar/api/add/\$user\$
	EDIT	http://dataprovder.com/calendar/api/edit/\$user\$/\$id\$
blog API	GET	http://dataprovder.com/blog/api/get/\$user\$
	ADD	http://dataprovder.com/blog/api/add/\$user\$
	EDIT	http://dataprovder.com/blog/api/edit/\$user\$/\$id\$

図 8

【 図 9 】



【 図 1 1 】



【 図 1 0 】

APIメソッド名	リクエストパラメタ	レスポンス
ADD	startdate = 20070221 &enddate = 20070222 &title = web+conference &place = tokyo	({"status": "ok"})
GET	startdate = 20070101 &keyword = web	({"schedules": [{"startdate": "20070220", "enddate": "20070222", "title": "web conference", "place": "tokyo"}, {"startdate": "20070223", "enddate": "20070223", "title": "web meeting", "place": "meeting room"}, ...]})

図 10

【 図 1 2 】

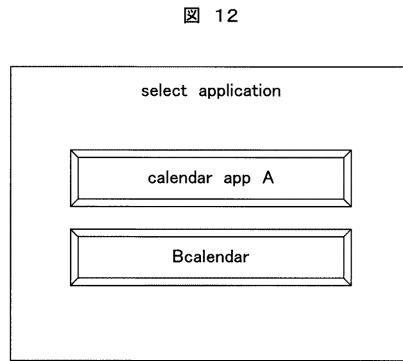


図 12

【 図 1 3 】

情報の種類	URL
holiday	http://www.holidaycalendar.com/api/jsonp/
weather	http://www.weatherinformation.com/jsonpapi/

図 13

【 図 1 4 】

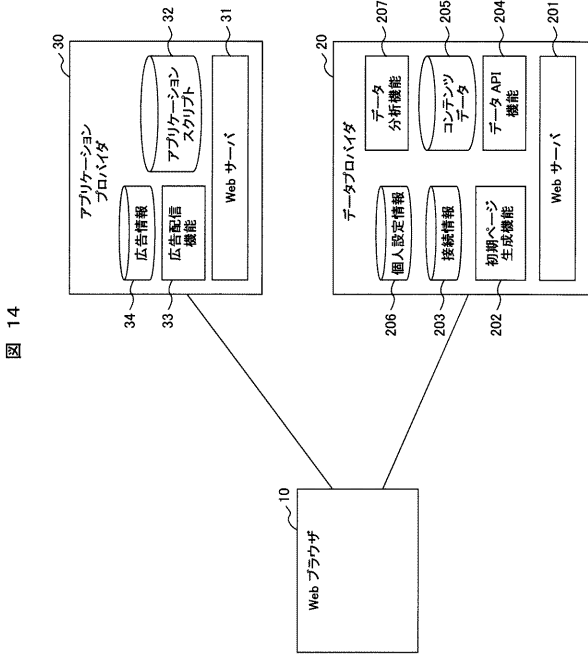


図 14

【 図 1 5 】

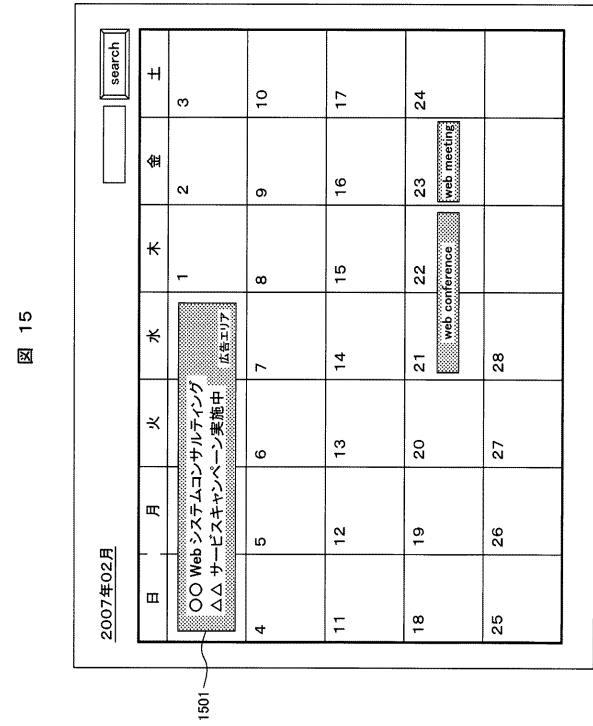


図 15

【 図 1 6 】

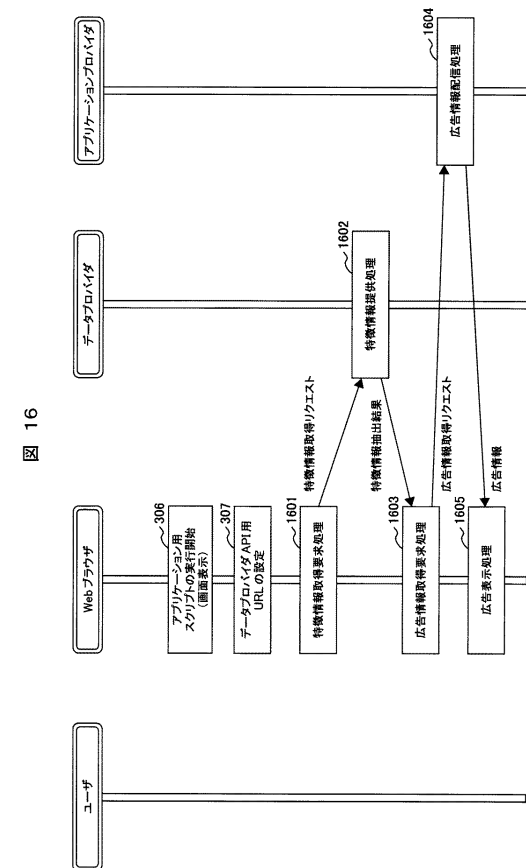


図 16

【 図 1 7 】

図 17

特徴情報	広告 URL
web	http://www.add.com/image/001.png
network	http://www.add.com/image/002.png
application	http://www.add.com/image/003.png

【 図 1 8 】

