

(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2010-191908

(P2010-191908A)

(43) 公開日 平成22年9月2日(2010.9.2)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 21/24 (2006.01)</b>	G06F 12/14 560A	5B017
<b>G06F 13/00 (2006.01)</b>	G06F 13/00 550A	5B084

審査請求 未請求 請求項の数 14 O L (全 34 頁)

(21) 出願番号 特願2009-38386 (P2009-38386)  
 (22) 出願日 平成21年2月20日 (2009.2.20)

(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 110000198  
 特許業務法人湘洋内外特許事務所  
 (72) 発明者 乾 敦行  
 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 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  
 Fターム(参考) 5B017 AA08 CA16  
 5B084 AA01 AB16 BB16

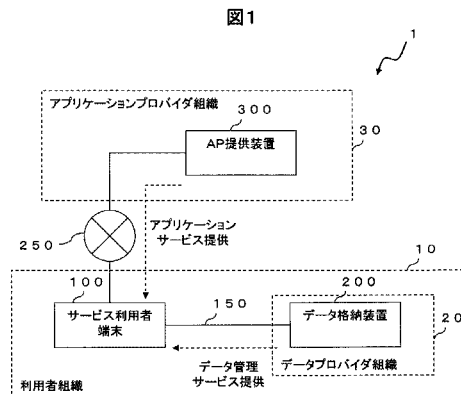
(54) 【発明の名称】 分散処理システム、分散処理方法、分散処理システムの開発支援装置および開発支援方法

(57) 【要約】 (修正有)

【課題】従来、アプリケーションサービス提供サーバが、要求された電子データと、電子データを操作するプログラムと、を利用端末に送信し、電子データと共に送信されたプログラムを介して利用端末に電子データの操作を可能とさせる技術がある。しかし、電子データを操作するプログラムおよび電子データは、重要な技術やノウハウを含んでいる場合が多いため、当該プログラムを提供する者も、当該サービスを利用する者も、どちらもノウハウ流出リスクを負う。そこで、重要な技術やノウハウを含むプログラムまたは電子データを他に流出させずに、サービスの利用を可能とする技術を提供する。

【解決手段】ノウハウ等を含む処理はアプリケーションサービス提供サーバ上で実行し、利用者側の機密情報等を含むデータはアプリケーションサービス提供サーバから分離可能とする。

【選択図】 図1



**【特許請求の範囲】****【請求項 1】**

サービス利用者端末と、ネットワークを介して前記サービス利用者端末に所定のサービスを提供するサービス提供装置と、データ格納装置と、を備える分散処理システムであって、

前記サービス提供装置は、

前記サービス利用者端末を、前記サービス提供装置に所定の処理を前記ネットワークを介して開始させ、前記サービス提供装置からの指示を受け付けて所定の処理を実施し、前記データ格納装置に所定のデータ操作を指示する制御部として動作させるプログラムを記憶する記憶部と、

前記サービス利用者端末からのサービス開始要求を受け付けるサービス開始要求受付部と、

前記サービス開始要求受付部によりサービス開始要求を受け付けると、前記プログラムを前記サービス利用者端末に送信するプログラム送信部と、

前記プログラム送信部が送信した前記プログラムを実行した前記サービス利用者端末により前記所定の処理を開始するサービス実行部と、

前記サービス実行部による前記所定の処理が終了すると、前記ネットワークを介して前記サービス利用者端末に所定の処理を実施するよう指示するコールバック処理部と、を備え、

前記データ格納装置は、

前記プログラムを実行した前記サービス利用者端末によるデータ操作を受け付けるデータ操作受付部と、

前記データ操作受付部により受け付けたデータ操作を実行するデータ操作部と、

前記データ操作部により実行されたデータ操作の結果を前記サービス利用者端末に出力する操作結果出力部と、を備える、

ことを特徴とする分散処理システム。

**【請求項 2】**

請求項 1 に記載の分散処理システムであって、

前記プログラム送信部は、画面表示情報に関連付けて前記プログラムを送信する、ことを特徴とする分散処理システム。

**【請求項 3】**

請求項 2 に記載の分散処理システムであって、

前記画面表示情報は、HTML で記述されている、ことを特徴とする分散処理システム。

**【請求項 4】**

請求項 1 ～ 3 のいずれか一項に記載の分散処理システムであって、

前記プログラムは、JavaScript で記述されたプログラムである、ことを特徴とする分散処理システム。

**【請求項 5】**

請求項 1 ～ 4 のいずれか一項に記載の分散処理システムであって、

前記サービス実行部による前記所定の処理は、前記プログラムと同じプログラミング言語で記述されている、

ことを特徴とする分散処理システム。

**【請求項 6】**

分散処理システムによる分散処理方法であって、

前記分散処理システムは、

サービス利用者端末と、ネットワークを介して前記サービス利用者端末に所定のサービスを提供するサービス提供装置と、データ格納装置と、を備える分散処理システムであって、

前記サービス提供装置は、

10

20

30

40

50

前記サービス利用者端末を、前記サービス提供装置に所定の処理を前記ネットワークを介して開始させ、前記サービス提供装置からの指示を受け付けて所定の処理を実施し、前記データ格納装置に所定のデータ操作を指示する制御部として動作させるプログラムを記憶する記憶部を備え、

前記サービス利用者端末からのサービス開始要求を受け付けるサービス開始要求受付ステップと、

前記サービス開始要求受付ステップによりサービス開始要求を受け付けると、前記プログラムを前記サービス利用者端末に送信するプログラム送信ステップと、

前記プログラム送信ステップにおいて送信した前記プログラムを実行した前記サービス利用者端末により前記所定の処理を開始するサービス実行ステップと、

前記サービス実行ステップにおける前記所定の処理が終了すると、前記ネットワークを介して前記サービス利用者端末に所定の処理を実施するよう指示するコールバック処理ステップと、を実行し、

前記データ格納装置は、

前記プログラムを実行した前記サービス利用者端末によるデータ操作を受け付けるデータ操作受付ステップと、

前記データ操作受付ステップにおいて受け付けたデータ操作を実行するデータ操作ステップと、

前記データ操作ステップにより実行されたデータ操作の結果を前記サービス利用者端末に出力する操作結果出力ステップと、を実行する、

ことを特徴とする分散処理方法。

#### 【請求項 7】

サービス利用者端末と、ネットワークを介して前記サービス利用者端末に所定のサービスを提供するサービス提供装置と、を備える分散処理システムの開発支援装置であって、

基礎となる所定のプログラムと、前記所定のプログラムに含まれる特定の処理について当該処理を実行する実行装置を定義する処理実行装置定義情報と、データの配置場所を定義するデータ配置場所定義情報と、を記憶する記憶部と、制御部と、を備え、

前記制御部は、

前記所定のプログラムに含まれる各処理について、前記実行装置を前記処理実行場所定義情報から特定し、

特定した前記実行装置が前記サービス利用者端末である場合に、当該処理においてアクセスするデータの配置場所を前記データ配置場所定義情報により特定し、前記データの配置場所が前記サービス提供装置上であれば、前記サービス利用者端末において実行される端末側プログラムと、前記サービス提供装置からコールバックされる端末側プログラムと、に分けて当該処理を実現するプログラムを生成し、前記データの配置場所が前記サービス提供装置上でなければ、前記サービス利用者端末において実行される端末側プログラムが当該処理を実現するよう端末側プログラムを生成する、

ことを特徴とする分散処理システムの開発支援装置。

#### 【請求項 8】

請求項 7 に記載の開発支援装置であって、

前記制御部は、さらに、

特定した前記実行装置が前記サービス提供装置である場合には、前記サービス提供装置において実行されて当該処理を実現するサービス提供プログラムを生成するとともに、当該処理を開始させるためのプログラムを前記サービス利用者端末において実行される端末側プログラムとして生成する、

ことを特徴とする開発支援装置。

#### 【請求項 9】

請求項 7 または 8 に記載の開発支援装置であって、

前記制御部は、

特定した前記実行装置が前記サービス利用者端末である場合において、前記データの配

10

20

30

40

50

置場所が前記サービス提供装置上であれば、前記データにアクセスする処理ごとに前記コールバックされる端末側プログラムを生成する、  
ことを特徴とする開発支援装置。

【請求項 10】

請求項 7～9 のいずれか一項に記載の開発支援装置であって、  
前記制御部は、

特定した前記実行装置が前記サービス利用者端末である場合において、前記データの配置場所が前記サービス提供装置上であれば、当該処理が繰り返し処理を含む場合には、当該繰り返し処理を再帰呼び出しにより実現する端末側プログラムを生成する、  
ことを特徴とする開発支援装置。

10

【請求項 11】

サービス利用者端末と、ネットワークを介して前記サービス利用者端末に所定のサービスを提供するサービス提供装置と、を備える分散処理システムの開発支援装置による開発支援方法であって、

前記開発支援装置は、

基礎となる所定のプログラムと、前記所定のプログラムに含まれる特定の処理について当該処理を実行する実行装置を定義する処理実行装置定義情報と、データの配置場所を定義するデータ配置場所定義情報と、を記憶する記憶部を備え、

前記開発支援装置は、

前記所定のプログラムに含まれる各処理について、前記実行装置を前記処理実行場所定義情報から特定するステップと、

20

特定した前記実行装置が前記サービス利用者端末である場合において、当該処理からアクセスするデータの配置場所を前記データ配置場所定義情報により特定し、前記データの配置場所が前記サービス提供装置上であれば、前記サービス利用者端末において実行される端末側プログラムと、前記サービス提供装置からコールバックされる端末側プログラムと、に分けて当該処理を実現するプログラムを生成し、前記データの配置場所が前記サービス提供装置上でなければ、前記サービス利用者端末において実行される当該処理を実現する端末側プログラムを生成するステップと、

を実行することを特徴とする開発支援方法。

【請求項 12】

30

請求項 11 に記載の開発支援方法であって、

前記開発支援装置は、さらに、

特定した前記実行装置が前記サービス提供装置である場合には、前記サービス提供装置において実行される当該処理を実現するサービス提供プログラムを生成するとともに、当該処理を開始させるためのプログラムを前記サービス利用者端末において実行される端末側プログラムとして生成するステップ、

を実行することを特徴とする開発支援方法。

【請求項 13】

請求項 11 または 12 に記載の開発支援方法であって、

前記開発支援装置は、

40

特定した前記実行装置が前記サービス利用者端末である場合において、前記データの配置場所が前記サービス提供装置上であれば、前記データにアクセスする処理ごとに前記コールバックされる端末側プログラムを生成するステップ、

を実行することを特徴とする開発支援方法。

【請求項 14】

請求項 11～13 のいずれか一項に記載の開発支援方法であって、

前記開発支援装置は、

特定した前記実行装置が前記サービス利用者端末である場合において、前記データの配置場所が前記サービス提供装置上であれば、当該処理が繰り返し処理を含む場合には、当該繰り返し処理を再帰呼び出しにより実現する端末側プログラムを生成するステップ、

50

を実行することを特徴とする開発支援方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、分散処理システム、分散処理方法、分散処理システムの開発支援装置および開発支援方法に関する。

【背景技術】

【0002】

従来、アプリケーションサービス提供サーバが、要求された電子データと、電子データを操作するためのプログラムと、を利用端末に送信し、電子データと共に送信されたプログラムを介して利用端末に電子データの操作を可能とさせる技術がある（特許文献1）。

10

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2005-353058号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

上記特許文献1に記載の技術では、電子データを操作するためのプログラムおよび電子データは、重要な技術やノウハウを含んでいる場合があるにもかかわらず、利用端末またはアプリケーションサービス提供サーバに流出してしまうおそれがある。つまり、電子データを操作するためのプログラムを提供する者も、アプリケーションサービスを利用する者も、どちらもプログラムの提供又は利用に際してノウハウ流出リスクを負わざるを得ない。

20

【0005】

そこで、本発明は、重要な技術やノウハウを含むアプリケーションプログラムまたは電子データを他に流出させるリスクを低減できるアプリケーションサービスの利用を可能とする技術の提供を目的とする。

【課題を解決するための手段】

【0006】

本願に係る分散処理の技術は、上記課題を解決するため、ノウハウ等を含む処理はアプリケーションサービス提供サーバ上で実行し、利用者側の機密情報等を含むデータはアプリケーションサービス提供サーバから分離可能とする。

30

【0007】

例えば、サービス利用者端末と、ネットワークを介して前記サービス利用者端末に所定のサービスを提供するサービス提供装置と、データ格納装置と、を備える分散処理システムであって、前記サービス提供装置は、前記サービス利用者端末を、前記サービス提供装置に所定の処理を前記ネットワークを介して開始させ、前記サービス提供装置からの指示を受け付けて所定の処理を実施し、前記データ格納装置に所定のデータ操作を指示する制御部として動作させるプログラムを記憶する記憶部と、前記サービス利用者端末からのサービス開始要求を受け付けるサービス開始要求受付部と、前記サービス開始要求受付部によりサービス開始要求を受け付けると、前記プログラムを前記サービス利用者端末に送信するプログラム送信部と、前記プログラム送信部が送信した前記プログラムを実行した前記サービス利用者端末により前記所定の処理を開始するサービス実行部と、前記サービス実行部による前記所定の処理が終了すると、前記ネットワークを介して前記サービス利用者端末に所定の処理を実施するよう指示するコールバック処理部と、を備え、前記データ格納装置は、前記プログラムを実行した前記サービス利用者端末によるデータ操作を受け付けるデータ操作受付部と、前記データ操作受付部により受け付けたデータ操作を実行するデータ操作部と、前記データ操作部により実行されたデータ操作の結果を前記サービス利用者端末に出力する操作結果出力部と、を備える、ことを特徴とする。

40

50

## 【発明の効果】

## 【0008】

重要な技術やノウハウを含むアプリケーションプログラムまたは電子データを他に流出させるリスクを低減できるアプリケーションサービスの利用を可能とする技術を提供することができる。

## 【図面の簡単な説明】

## 【0009】

【図1】分散情報処理システムの概略図である。

【図2】サービス利用者端末の概略図である。

【図3】APスクリプト（呼出）の例を示す図である。

10

【図4】APスクリプト（チケット）の例を示す図である。

【図5】APスクリプト（プロジェクト）の例を示す図である。

【図6】スクリプトライブラリの例を示す図である。

【図7】DSDAモジュールの例を示す図である。

【図8】APDAモジュールの例を示す図である。

【図9】データ格納装置の概略図である。

【図10】AP提供装置の概略図である。

【図11】APの例を示す図である。

【図12】DAモジュール（プロジェクト）の例を示す図である。

【図13】DAモジュール（Wiki）の例を示す図である。

20

【図14】サービス利用者端末のハードウェア構成を示す図である。

【図15】プロジェクト管理処理の処理内容を示す図である。

【図16】プロジェクト管理処理の画面例を示す図である。

【図17】プロジェクト管理処理の画面例を示す図である。

【図18】開発支援装置の概略図である。

【図19】基準APの例を示す図である。

【図20】AP実行場所表のデータ構造の例を示す図である。

【図21】データ場所表の例を示す図である。

【図22】モジュールリストのデータ構造の例を示す図である。

【図23】AP生成処理の処理フローを示す図である。

30

【図24】AP生成処理のループ変換処理の処理フローを示す図である。

【図25】DAモジュール生成処理の処理フローを示す図である。

## 【発明を実施するための形態】

## 【0010】

図1は、本発明の実施形態における分散情報処理システム1を示す図である。

## 【0011】

本実施形態における分散情報処理システム1は、サービス利用者端末100と、組織内ネットワーク150を介してサービス利用者端末100と接続可能なデータ格納装置200と、組織間ネットワーク250を介してサービス利用者端末100と接続可能なアプリケーション提供装置（以下、「AP提供装置」と記載）300と、を含む。

40

## 【0012】

なお、組織間ネットワーク250は、通常はインターネット等の公衆通信網、WAN（Wide Area Network）またはVPN（Virtual Private Network）等の異組織間の通信網である。

## 【0013】

組織内ネットワーク150は、通常はLAN（Local Area Network）等の利用者組織が管理する通信網である。ただし、これに限らず、組織内ネットワーク150は、インターネット等の公衆通信網、WAN（Wide Area Network）またはVPN（Virtual Private Network）等の異組織間の通信網であってもよい。

## 【0014】

50

サービス利用者端末 100 は、利用者組織 10 に属する。

【0015】

データ格納装置 200 は、データプロバイダ組織 20 に属しており、本実施形態においては、データプロバイダ組織 20 は利用者組織 10 に属する。

【0016】

データ格納装置 200 は、データを保持し、保持するデータの参照や変更を要求に応じて実施するデータ管理サービスをサービス利用者端末 100 に提供する。

【0017】

AP 提供装置 300 は、アプリケーションプロバイダ組織 30 に属する。

【0018】

AP 提供装置 300 は、例えばプロジェクト管理や表計算等のアプリケーションサービスを、サービス利用者端末 100 からの要求に応じて提供する。

【0019】

図 2 は、サービス利用者端末 100 の概略図である。図示するように、サービス利用者端末 100 は、記憶部 120 と、制御部 130 と、入力部 141 と、出力部 142 と、通信部 143 と、を有する。

【0020】

記憶部 120 は、AP スクリプト記憶領域 121 と、DSDA モジュール記憶領域 122 と、APDA モジュール記憶領域 123 と、を備える。

【0021】

AP スクリプト記憶領域 121 には、AP 提供装置 300 から提供されるアプリケーションサービスを実現するアプリケーションプログラム（以下、「AP スクリプト」と表記）が記憶される。

【0022】

本実施形態においては、AP スクリプトは、JavaScript（登録商標）により記述されるが、これに限らず、サービス利用者端末 100 に搭載された図示しない実行エンジン上で実行されるクライアントサイドスクリプトであればよい。

【0023】

本実施形態においては、AP スクリプト記憶領域 121 には、AP スクリプト（呼出）160 と、AP スクリプト（チケット）165 と、AP スクリプト（プロジェクト）170 と、ライブラリスクリプト 180 と、が格納される。

【0024】

AP スクリプト（呼出）160 は、図 3 に示すように、ブラウザ部 133 により実行される JavaScript の関数である。

【0025】

AP スクリプト（呼出）160 の処理内容である Wiki 表示処理 161 を説明する。Wiki 表示処理 161 は、AP スクリプト（呼出）160 を含む HTML（Hyper Text Markup Language）ページに、「script」要素を追加するものである。なお、当該 script 要素は、実行されると、AP 提供装置 300 に対して、引数に与えられた情報に該当する Wiki ページを要求し、受け取った Wiki ページを当該 HTML ページに表示させるものである。

【0026】

AP スクリプト（チケット）165 は、図 4 に示すように、ブラウザ部 133 により実行される JavaScript の関数である。

【0027】

AP スクリプト（チケット）165 の第一の処理内容であるチケット一覧処理 166 を説明する。AP スクリプト（チケット）165 のチケット一覧処理 166 は、AP スクリプト（チケット）165 を含む HTML ページに既に生成されたチケット（本実施形態におけるチケットとは、バグ情報等のレポート一件をいう。）の一覧情報を追加して表示させる処理である。なお、当該処理においては、データ格納装置 200 のデータベース領域

10

20

30

40

50

2 2 1 に格納されたチケット情報を取得して一覧情報を生成する処理が含まれる。

【 0 0 2 8 】

A P スクリプト ( チケット ) 1 6 5 の第二の処理内容であるチケット生成処理 1 6 7 を説明する。A P スクリプト ( チケット ) 1 6 5 のチケット生成処理 1 6 7 は、チケット情報を一件生成して、A P スクリプト ( チケット ) 1 6 5 を含む H T M L ページに、既に生成されたチケットの一覧情報を追加して表示させるものである。なお、当該処理においては、データ格納装置 2 0 0 のデータベース領域 2 2 1 にチケット情報を新たに追加する処理と、データベース領域 2 2 1 に格納されたチケット情報を取得して一覧情報を生成する処理が含まれる。

【 0 0 2 9 】

A P スクリプト ( プロジェクト ) 1 7 0 は、図 5 に示すように、ブラウザ部 1 3 3 により実行される JavaScript の関数である。

【 0 0 3 0 】

A P スクリプト ( プロジェクト ) 1 7 0 の第一の処理内容であるプロジェクト一覧処理 1 7 1 を説明する。プロジェクト一覧処理 1 7 1 は、実行されると、A P 提供装置 3 0 0 に対して、既に生成されたプロジェクト管理情報の一覧情報を要求するものである。なお、当該処理においては、A P 提供装置 3 0 0 のデータベース領域 3 2 4 に格納されたプロジェクト情報を要求する処理が含まれる。

【 0 0 3 1 】

次に、A P スクリプト ( プロジェクト ) 1 7 0 の第二の処理内容であるプロジェクト一覧コールバック処理 1 7 2 を説明する。プロジェクト一覧コールバック処理 1 7 2 は、プロジェクト一覧処理 1 7 1 が実行された後に、A P 提供装置 3 0 0 から呼び出されるコールバック処理である。プロジェクト一覧コールバック処理 1 7 2 は、A P スクリプト ( プロジェクト ) 1 7 0 を含む H T M L ページに、引数に渡されたプロジェクト管理情報一覧のデータ ( 例えば J S O N ( JavaScript Object Notation ) 形式のデータ ) を追加して表示させるものである。なお、当該処理においては、A P 提供装置 3 0 0 のデータベース領域 3 2 4 に格納されたプロジェクト管理情報を受信して一覧情報を生成する処理が含まれる。

【 0 0 3 2 】

次に、A P スクリプト ( プロジェクト ) 1 7 0 の第三の処理内容であるプロジェクト表示処理 1 7 3 を説明する。プロジェクト表示処理 1 7 3 は、実行されると、A P 提供装置 3 0 0 に対して、特定のプロジェクト管理情報を要求する処理である。なお、当該処理においては、A P 提供装置 3 0 0 のデータベース領域 3 2 4 に格納された特定のプロジェクト管理情報を要求する処理が含まれる。

【 0 0 3 3 】

次に、A P スクリプト ( プロジェクト ) 1 7 0 の第四の処理内容であるプロジェクト表示コールバック処理 1 7 4 を説明する。プロジェクト表示コールバック処理 1 7 4 は、プロジェクト表示処理 1 7 3 が実行された後に、A P 提供装置 3 0 0 から呼び出されるコールバック処理である。プロジェクト表示コールバック処理 1 7 3 は、A P スクリプト ( プロジェクト ) 1 7 0 を含む H T M L ページに、引数に渡された特定のプロジェクト管理情報のデータ ( 例えば J S O N ( JavaScript Object Notation ) 形式のデータ ) を追加して表示させる処理である。なお、当該処理においては、A P 提供装置 3 0 0 のデータベース領域 3 2 4 に格納された特定のプロジェクト管理情報を受信して表示情報を生成する処理が含まれる。

【 0 0 3 4 】

次に、A P スクリプト ( プロジェクト ) 1 7 0 の第五の処理内容であるプロジェクト生成処理 1 7 5 を説明する。プロジェクト生成処理 1 7 5 は、実行されると、A P 提供装置 3 0 0 に対して、新規のプロジェクト管理情報を生成して格納するよう要求する処理である。なお、当該処理においては、A P 提供装置 3 0 0 のデータベース領域 3 2 4 に新規のプロジェクト管理情報を格納させ、その結果を表示する処理が含まれる。

10

20

30

40

50

## 【 0 0 3 5 】

次に、A P スクリプト ( プロジェクト ) 1 7 0 の第六の処理内容であるプロジェクト生成コールバック処理 1 7 6 を説明する。プロジェクト生成コールバック処理 1 7 6 は、プロジェクト生成処理 1 7 5 が実行された後に、A P 提供装置 3 0 0 から呼び出されるコールバック処理である。プロジェクト生成コールバック処理 1 7 6 は、実行されると、上記プロジェクト一覧処理 1 7 1 に処理を委譲する処理である。

## 【 0 0 3 6 】

スクリプトライブラリ 1 8 0 は、図 6 に示すように、ブラウザ部 1 3 3 により実行される JavaScript の関数である。

## 【 0 0 3 7 】

スクリプトライブラリ 1 8 0 には、A P スクリプトの実行をサポートするユーティリティプログラム等を含む処理が含まれる。

## 【 0 0 3 8 】

例えば、ユーティリティプログラムには、画面描画プログラム等種々の A P I ( Application Program Interface ) が含まれる。

## 【 0 0 3 9 】

本実施形態におけるスクリプトライブラリ 1 8 0 に含まれる処理のうち、第一の処理内容である画面構成処理 1 8 1 を説明する。画面構成処理 1 8 1 は、スクリプトライブラリ 1 8 0 を含む H T M L ページに、引数で渡された H T M L を追加して画面を構成する処理である。

## 【 0 0 4 0 】

スクリプトライブラリ 1 8 0 の第二の処理内容である処理委譲処理 1 8 2 を説明する。処理委譲処理 1 8 2 は、スクリプトライブラリ 1 8 0 を含む H T M L ページに含まれる他の関数の名称を引数として受け取り、引数に指定された名称の関数を実行する処理である。

## 【 0 0 4 1 】

D S D A モジュール記憶領域 1 2 2 には、データ提供装置 2 0 0 に格納されたデータベースへアクセスするためのプログラムモジュール ( D S D A モジュール 1 9 0 ) が記憶される。

## 【 0 0 4 2 】

D S D A モジュール 1 9 0 とは、データアクセス用のプログラムモジュールのうち、データプロバイダ組織 2 0 が管理するデータ格納装置 2 0 0 に格納されたデータにアクセスするためのプログラムモジュールをいう。

## 【 0 0 4 3 】

D S D A モジュール 1 9 0 は、図 7 に示すように、ブラウザ部 1 3 3 により実行される JavaScript の関数である。

## 【 0 0 4 4 】

本実施形態においては、D S D A モジュール 1 9 0 は、JavaScript により記述されるが、これに限らず、サービス利用者端末 1 0 0 に搭載されたブラウザ部 1 3 3 ( 後述 ) 上で実行されるクライアントサイドスクリプトであればよい。

## 【 0 0 4 5 】

D S D A モジュール 1 9 0 の第一の処理内容であるチケットオブジェクト生成処理 1 9 1 を説明する。チケットオブジェクト生成処理 1 9 1 は、引数に指定された名称を有するデータベースアクセス用のオブジェクトを生成する処理である。本実施形態においては、データ格納装置 2 0 0 に対して接続を行うデータベースアクセスオブジェクトを生成する処理が含まれる。

## 【 0 0 4 6 】

D S D A モジュール 1 9 0 の第二の処理内容である指定チケット取得処理 1 9 2 を説明する。指定チケット取得処理 1 9 2 は、データベースアクセスオブジェクトを介してチケット情報を格納するデータベースに接続して、引数に指定された情報を識別子として有す

10

20

30

40

50

るレコードを取得する処理である。本実施形態においては、データ格納装置 200 に対して接続を行い、データベース領域 221 に格納されたチケット情報を取得する処理を含む。

【0047】

DSDA モジュール 190 の第三の処理内容である全チケット取得処理 193 を説明する。全チケット取得処理 193 は、データベースアクセスオブジェクトを介してチケット情報を格納するデータベースに接続して、格納された全レコードを取得する処理である。本実施形態においては、データ格納装置 200 に対して接続を行い、データベース領域 221 に格納された全部のチケット情報を取得する処理が含まれる。

【0048】

DSDA モジュール 190 の第四の処理内容であるチケット格納処理 194 を説明する。チケット格納処理 194 は、データベースアクセスオブジェクトを介してチケット情報を格納するデータベースに接続して、レコードを格納する処理である。本実施形態においては、データ格納装置 200 に対して接続を行い、データベース領域 221 にチケット情報を格納し、当該チケット情報を識別する識別子を取得する処理が含まれる。

【0049】

DSDA モジュール 190 の第五の処理内容であるチケット削除処理 195 を説明する。チケット削除処理 195 は、データベースアクセスオブジェクトを介してチケット情報を格納するデータベースに接続して、識別子で識別されるレコードを削除する処理である。本実施形態においては、データ格納装置 200 に対して接続を行い、データベース領域 221 に格納されたチケット情報から識別子で識別されるレコードを削除する処理が含まれる。

【0050】

APDA モジュール記憶領域 123 には、AP 提供装置 300 から提供されるアプリケーションサービスを実行する際に必要となるデータアクセス用のプログラムモジュール (APDA モジュール 196) が記憶される。

【0051】

APDA モジュール 196 とは、データアクセス用のプログラムモジュールのうち、アプリケーションプロバイダ組織 30 が管理する AP 提供装置 300 に格納されたデータにアクセスするためのプログラムモジュールをいう。

【0052】

APDA モジュール 196 は、図 8 に示すように、ブラウザ部 133 により解釈されて実行される JavaScript の関数である。

【0053】

本実施形態においては、APDA モジュール 196 は、JavaScript により記述されるが、これに限らず、サービス利用者端末 100 に搭載されたブラウザ部 133 (後述) 上で実行されるクライアントサイドスクリプトであればよい。

【0054】

APDA モジュール 196 の第一の処理内容である指定プロジェクト取得処理 197 を説明する。指定プロジェクト取得処理 197 は、引数に指定された情報を識別子として有するプロジェクトの情報を取得するスクリプトを、APDA モジュール 196 を含む HTML ページに追加する処理である。本実施形態においては、上記スクリプトは、AP 提供装置 300 に対して接続を行い、データベース領域 324 に格納されたプロジェクト情報のうち引数に指定された情報を識別子として有するプロジェクトの情報を取得する処理が含まれる。

【0055】

APDA モジュール 196 の第二の処理内容である全プロジェクト取得処理 198 を説明する。全プロジェクト取得処理 198 は、全プロジェクトの情報を取得するスクリプトを、APDA モジュール 196 を含む HTML ページに追加する処理である。本実施形態においては、上記スクリプトは、AP 提供装置 300 に対して接続を行い、データベース

10

20

30

40

50

領域 3 2 4 に格納された全部のプロジェクト情報を取得する処理が含まれる。

【 0 0 5 6 】

A P D A モジュール 1 9 6 の第三の処理内容であるプロジェクト格納処理 1 9 9 を説明する。プロジェクト格納処理 1 9 9 は、データベースアクセスオブジェクトを介してチケット情報を格納するデータベースに接続して、レコードを格納する処理である。本実施形態においては、データ格納装置 2 0 0 に対して接続を行い、データベース領域 2 2 1 にチケット情報を格納し、当該チケット情報を識別する識別子を取得する処理が含まれる。

【 0 0 5 7 】

制御部 1 3 0 は、入力情報受付部 1 3 1 と、出力情報生成部 1 3 2 と、ブラウザ部 1 3 3 と、を備える。

10

【 0 0 5 8 】

入力情報受付部 1 3 1 は、後述する入力部 1 4 1 を介して入力された情報を受け付ける。

【 0 0 5 9 】

出力情報生成部 1 3 2 は、出力する情報と、画面レイアウトと、を組み合わせて出力画面を構成し、後述する出力部 1 4 2 に表示させる。

【 0 0 6 0 】

ブラウザ部 1 3 3 は、後述する通信部 1 4 3 を介して組織間ネットワーク 2 5 0 又は組織内ネットワーク 1 5 0 に接続し、各ネットワークに接続された装置と通信を行う。また、利用者の操作を入力情報受付部 1 3 1 を介して受け付け、利用者に提示する情報を出力情報生成部 1 3 2 に依頼して出力する。

20

【 0 0 6 1 】

ブラウザ部 1 3 3 は、さらに、A P スクリプト記憶領域 1 2 1 と、D S D A モジュール記憶領域 1 2 2 と、A P D A モジュール記憶領域 1 2 3 と、に格納されたプログラムモジュールを解釈して実行する実行エンジンの機能を備える。

【 0 0 6 2 】

入力部 1 4 1 は、操作者からの情報の入力を受け付ける。

【 0 0 6 3 】

出力部 1 4 2 は、情報を出力する。

【 0 0 6 4 】

通信部 1 4 3 は、組織間ネットワーク 2 5 0 および組織内ネットワーク 1 5 0 を介して他の機器に接続し、接続した他の機器に対して情報の送信を行い、接続した他の機器から送信された情報の受信を行う。

30

【 0 0 6 5 】

図 9 は、データ格納装置 2 0 0 の概略図である。図示するように、データ格納装置 2 0 0 は、記憶部 2 2 0 と、制御部 2 3 0 と、通信部 2 4 3 と、を有する。

【 0 0 6 6 】

記憶部 2 2 0 は、データベース領域 2 2 1 を含む。

【 0 0 6 7 】

データベース領域 2 2 1 には、A P 提供装置 3 0 0 から提供されるアプリケーションサービスの実行に必要なデータを格納するデータベースが記憶される。

40

【 0 0 6 8 】

制御部 2 3 0 は、初期画面生成部 2 3 1 と、データ操作部 2 3 2 と、通信部 2 4 3 と、を備える。

【 0 0 6 9 】

初期画面生成部 2 3 1 は、サービス利用者端末 1 0 0 から、A P 提供装置 3 0 0 が提供するアプリケーションの利用依頼を受け付ける。利用依頼を受け付けると、初期画面生成部 2 3 1 は、利用依頼のあったアプリケーションを開始するためのキー情報をサービス利用者端末 1 0 0 に送信する。

【 0 0 7 0 】

50

本実施形態においては、初期画面生成部 231 は、A P 提供装置 300 のアプリケーションの利用要求を受け付ける U R I (Uniform Resource Identifier) をキー情報とする。初期画面生成部 231 は、キー情報を含む情報をサービス利用者端末 100 に送信する。

【0071】

データ操作部 232 は、サービス利用者端末 100 から、指定されたデータの入出力要求を受け付ける。データ操作部 232 は、指定されたデータの入出力要求を受け付けると、記憶部 220 のデータベース領域 221 に対して、指定されたデータの入出力を行う。

【0072】

通信部 243 は、組織内ネットワーク 150 を介して他の機器に接続し、接続した他の機器に対して情報の送信を行い、接続した他の機器から送信された情報の受信を行う。

10

【0073】

図 10 は、A P 提供装置 300 の概略図である。図示するように、A P 提供装置 300 は、記憶部 320 と、制御部 330 と、入力部 341 と、出力部 342 と、通信部 343 と、を有する。

【0074】

記憶部 320 は、A P 記憶領域 321 と、スクリプトライブラリ記憶領域 322 と、D A モジュール記憶領域 323 と、データベース領域 324 と、を備える。

【0075】

A P 記憶領域 321 には、サービス利用者端末 100 に対して提供するアプリケーションサービスのうち、一部または全部の処理を実行するためのアプリケーションプログラム (A P 360) が記憶される。

20

【0076】

本実施形態においては、A P 360 は、図 11 に示すように、A P 実行部 332 により実行される JavaScript により記述されるが、これに限らず、サービス利用者端末 100 上で実行される A P スクリプトと同一のプログラム言語で記載されたスクリプトであればよい。

【0077】

A P 360 の処理内容である W i k i 表示処理 361 を説明する。W i k i 表示処理 361 は、A P 実行部 332 により実行されると、A P 提供装置 300 のデータベース領域 324 に格納された W i k i 情報を格納するデータベースに接続して、引数に指定された識別子で識別される W i k i 情報を取得して表示情報を生成するものである。

30

【0078】

また、W i k i 表示処理 361 は、生成した表示情報を画面に表示するようサービス利用者端末 100 に対して指示するものである。なお、当該処理においては、サービス利用者端末 100 に対して表示するよう指示する際に、スクリプトライブラリ 180 の画面構成処理 181 をコールバックする処理が含まれている。

【0079】

スクリプトライブラリ記憶領域 322 には、A P スクリプトの実行をサポートするユーティリティプログラム等を含むスクリプトライブラリ 180 が記憶される。

40

【0080】

スクリプトライブラリ 180 は、上述したサービス利用者端末 100 の A P スクリプト記憶領域 121 に記憶されるスクリプトライブラリ 180 と同一のものである。

【0081】

D A モジュール記憶領域 323 には、A P 提供装置 300 の後述するデータベース領域 324 に格納されたデータに A P 360 がアクセスする際に必要となるデータアクセス用のプログラムモジュール (D A モジュール) が記憶される。なお、D A モジュールには、D A モジュール (プロジェクト) 370 と、D A モジュール (W i k i) 380 と、が含まれる。

【0082】

50

D A モジュールとは、データアクセス用のプログラムモジュールのうち、A P 提供装置 3 0 0 に格納されたデータにアクセスするためのプログラムモジュールをいう。

【 0 0 8 3 】

本実施形態においては、D A モジュール（プロジェクト）3 7 0 は、図 1 2 に示すように、A P 実行部 3 3 2 により実行される JavaScript により記述されるが、これに限らず、A P スクリプトから呼び出し可能に記述されていれば別の言語により記述されてもよい。

【 0 0 8 4 】

D A モジュール（プロジェクト）3 7 0 の第一の処理内容であるプロジェクトオブジェクト生成処理 3 7 1 を説明する。プロジェクトオブジェクト生成処理 3 7 1 は、引数に指定された名称を有するデータベースアクセス用のオブジェクトを生成する処理である。本実施形態においては、データベース領域 3 2 4 の図示しないプロジェクト情報データベースに対して接続を行うデータベースアクセスオブジェクトを生成する処理が含まれる。

10

【 0 0 8 5 】

D A モジュール（プロジェクト）3 7 0 の第二の処理内容である指定プロジェクト取得処理 3 7 2 を説明する。指定プロジェクト取得処理 3 7 2 は、データベースアクセスオブジェクトを介して、プロジェクト情報を格納するデータベースに接続して、引数に指定された情報を識別子として有するレコードを取得する処理である。本実施形態においては、プロジェクト情報データベースに格納されたプロジェクト情報を取得する処理が含まれる。

【 0 0 8 6 】

D A モジュール（プロジェクト）3 7 0 の第三の処理内容である全プロジェクト取得処理 3 7 3 を説明する。全プロジェクト取得処理 3 7 3 は、データベースアクセスオブジェクトを介してプロジェクト情報を格納するデータベースに接続して、格納された全レコードを取得する処理である。本実施形態においては、プロジェクト情報データベースに格納された全部のプロジェクト情報を取得する処理が含まれる。

20

【 0 0 8 7 】

D A モジュール（プロジェクト）3 7 0 の第四の処理内容であるプロジェクト格納処理 3 7 4 を説明する。プロジェクト格納処理 3 7 4 は、データベースアクセスオブジェクトを介してプロジェクト情報を格納するデータベースに接続して、レコードを格納する処理である。本実施形態においては、プロジェクト情報データベースにプロジェクト情報を格納し、当該チケット情報を識別する識別子を取得する処理が含まれる。

30

【 0 0 8 8 】

D A モジュール（プロジェクト）3 7 0 の第五の処理内容であるプロジェクト削除処理 3 7 5 を説明する。プロジェクト削除処理 3 7 5 は、データベースアクセスオブジェクトを介してプロジェクト情報を格納するデータベースに接続して、識別子で識別されるレコードを削除する処理である。本実施形態においては、プロジェクト情報データベースに格納されたプロジェクト情報から識別子で識別されるレコードを削除する処理が含まれる。

【 0 0 8 9 】

D A モジュール（W i k i ）3 8 0 は、図 1 3 に示すように、A P 実行部 3 3 2 により実行される JavaScript により記述されるが、これに限らず、A P スクリプトから呼び出し可能に記述されていれば別の言語により記述されてもよい。

40

【 0 0 9 0 】

D A モジュール（W i k i ）3 8 0 の第一の処理内容である W i k i オブジェクト生成処理 3 8 1 を説明する。W i k i オブジェクト生成処理 3 8 1 は、引数に指定された名称を有するデータベースアクセス用のオブジェクトを生成する処理である。本実施形態においては、データベース領域 3 2 4 の図示しない W i k i 情報データベースに対して接続を行うデータベースアクセスオブジェクトを生成する処理が含まれる。

【 0 0 9 1 】

D A モジュール（W i k i ）3 8 0 の第二の処理内容である指定 W i k i 取得処理 3 8 2 を説明する。指定 W i k i 取得処理 3 8 2 は、データベースアクセスオブジェクトを介

50

してW i k i 情報を格納するデータベースに接続して、引数に指定された情報を識別子として有するレコードを取得する処理である。本実施形態においては、W i k i 情報データベースに格納されたW i k i 情報を取得する処理が含まれる。

【0092】

D Aモジュール(W i k i)380の第三の処理内容である全W i k i取得処理383を説明する。全W i k i取得処理383は、データベースアクセスオブジェクトを介してW i k i情報を格納するデータベースに接続して、格納された全レコードを取得する処理である。本実施形態においては、W i k i情報データベースに格納された全部のW i k i情報を取得する処理が含まれる。

【0093】

D Aモジュール(W i k i)380の第四の処理内容であるW i k i格納処理384を説明する。W i k i格納処理384は、データベースアクセスオブジェクトを介してW i k i情報を格納するデータベースに接続して、レコードを格納する処理である。本実施形態においては、W i k i情報データベースにW i k i情報を格納し、当該W i k i情報を識別する識別子を取得する処理が含まれる。

【0094】

D Aモジュール(W i k i)380の第五の処理内容であるW i k i削除処理385を説明する。W i k i削除処理385は、データベースアクセスオブジェクトを介してW i k i情報を格納するデータベースに接続して、識別子で識別されるレコードを削除する処理である。本実施形態においては、W i k i情報データベースに格納されたW i k i情報から識別子で識別されるレコードを削除する処理が含まれる。

【0095】

データベース領域324には、A P提供装置300が提供するアプリケーションサービスの実行に必要なデータを格納するデータベースが記憶される。

【0096】

制御部330は、A P利用受付部331と、A P実行部332と、データ操作部333と、を備える。

【0097】

A P利用受付部331は、A P記憶領域321に格納されたアプリケーションプログラムを開始するためのキー情報を、サービス利用者端末100から受け付ける。

【0098】

A P利用受付部331は、サービス利用者端末100からキー情報を受け付けると、当該アプリケーションプログラムのうちサービス利用者端末100側で動作するプログラムを、A P記憶領域321と、スクリプトライブラリ記憶領域322と、D Aモジュール記憶領域323と、の中から特定する。そして、A P利用受付部331は、特定したプログラムをサービス利用者端末100に送信する。

【0099】

A P実行部332は、A P記憶領域321と、スクリプトライブラリ記憶領域322と、D Aモジュール記憶領域323と、データベース領域324と、に格納されたアプリケーションプログラムを実行する。

【0100】

データ操作部333は、サービス利用者端末100またはA P提供装置300自身から指定されたデータの入出力要求を受け付ける。データ操作部333は、指定されたデータの入出力要求を受け付けると、記憶部320のデータベース領域324に対して、指定されたデータの入出力を行う。

【0101】

入力部341は、操作者からの情報の入力を受け付ける。

【0102】

出力部342は、情報を出力する。

【0103】

10

20

30

40

50

通信部 3 4 3 は、組織間ネットワーク 2 5 0 を介して他の機器に接続し、接続した他の機器に対して情報の送信を行い、接続した他の機器から送信された情報の受信を行う。

【 0 1 0 4 】

図 1 4 は、本実施形態におけるサービス利用者端末 1 0 0 のハードウェア構成を示す図である。

【 0 1 0 5 】

本実施形態においては、サービス利用者端末 1 0 0 は、例えば、クライアント P C ( パーソナルコンピュータ ) や、ワークステーション、サーバ装置、各種携帯電話端末、 P D A ( Personal Digital Assistant ) などの計算機である。

【 0 1 0 6 】

サービス利用者端末 1 0 0 は、入力装置 1 1 1 と、出力装置 1 1 2 と、演算装置 1 1 3 と、主記憶装置 1 1 4 と、外部記憶装置 1 1 5 と、通信装置 1 1 6 と、それぞれの装置を互いに接続するバス 1 1 7 と、を有する。

【 0 1 0 7 】

入力装置 1 1 1 は、例えばキーボードやマウス、あるいはタッチペン、その他ポインティングデバイスなどの入力を受け付ける装置である。

【 0 1 0 8 】

出力装置 1 1 2 は、例えばディスプレイなどの、表示を行う装置である

演算装置 1 1 3 は、例えば C P U ( Central Processing Unit ) などの演算装置である

。

【 0 1 0 9 】

主記憶装置 1 1 4 は、例えば R A M ( Random Access Memory ) などのメモリ装置である

。

【 0 1 1 0 】

外部記憶装置 1 1 5 は、例えばハードディスク装置やフラッシュメモリなどの不揮発性記憶装置である。

【 0 1 1 1 】

通信装置 1 1 6 は、アンテナを介して無線通信を行う無線通信ユニット等の通信装置である。

【 0 1 1 2 】

サービス利用者端末 1 0 0 の入力情報受付部 1 3 1 と、出力情報生成部 1 3 2 と、ブラウザ部 1 3 3 と、は、サービス利用者端末 1 0 0 の演算装置 1 1 3 に処理を行わせるプログラムによって実現される。

【 0 1 1 3 】

このプログラムは、主記憶装置 1 1 4 または外部記憶装置 1 1 5 内に記憶され、実行にあたって主記憶装置 1 1 4 上にロードされ、演算装置 1 1 3 により実行される。

【 0 1 1 4 】

また、サービス利用者端末 1 0 0 の記憶部 1 2 0 は、サービス利用者端末 1 0 0 の主記憶装置 1 1 4 または外部記憶装置 1 1 5 により実現される。

【 0 1 1 5 】

サービス利用者端末 1 0 0 の入力部 1 4 1 は、サービス利用者端末 1 0 0 の入力装置 1 1 1 によって実現される。

【 0 1 1 6 】

サービス利用者端末 1 0 0 の出力部 1 4 2 は、サービス利用者端末 1 0 0 の出力装置 1 1 2 によって実現される。

【 0 1 1 7 】

サービス利用者端末 1 0 0 の通信部 1 4 3 は、サービス利用者端末 1 0 0 の通信装置 1 1 6 によって実現される。

【 0 1 1 8 】

また、データ格納装置 2 0 0 と、 A P 提供装置 3 0 0 とは、基本的に、サービス利用者

10

20

30

40

50

端末 100 と同様のハードウェア構成を備える。

【0119】

ただし、データ格納装置 200 については、入力装置 111 と出力装置 112 とを備えない構成であってもよい。

【0120】

また、データ格納装置 200 においては、初期画面生成部 231 と、データ操作部 232 と、は、データ格納装置 200 の演算装置 113 に処理を行わせるプログラムによって実現される。

【0121】

また、AP 提供装置 300 においては、AP 利用受付部 331 と、AP 実行部 332 と、データ操作部 333 と、は、AP 提供装置 300 の演算装置 113 に処理を行わせるプログラムによって実現される。

10

【0122】

次に、本実施形態におけるプロジェクト管理処理について、図 15 に基づいて説明する。

【0123】

図 15 は、プロジェクト管理処理の処理フローを示す図である。

【0124】

まずは、プロジェクト管理処理におけるアプリケーションサービスを開始するための開始フェーズ（ステップ S1 ～ステップ S7 まで）について説明する。

20

【0125】

プロジェクト管理処理は、サービス利用者端末 100 のブラウザ部 133 が、入力情報受付部 131 を介して利用者から開始指示を受け付けることで開始される。

【0126】

まず、サービス利用者端末 100 のブラウザ部 133 は、データ格納装置 200 に対して初期ページを要求する（ステップ S1）。

【0127】

具体的には、ブラウザ部 133 に指定された URL（Uniform Resource Locator）に対して、ブラウザ部 133 は HTTP（Hypertext Transfer Protocol）リクエストを送信する。なお、当該 URL には、利用者が利用したいアプリケーションサービスを識別する識別子等の情報が含まれている。

30

【0128】

次に、データ格納装置 200 は、初期ページを生成する（ステップ S2）。

【0129】

具体的には、初期画面生成部 231 は、ステップ S1 で送信された URL に含まれるアプリケーションサービスを識別する識別子に対応する AP 提供装置 300 の URI を含む HTML ページを初期ページとして生成する。

【0130】

次に、データ格納装置 200 は、生成した初期ページをサービス利用者端末 100 に送信する（ステップ S3）。

40

【0131】

具体的には、初期画面生成部 231 は、ステップ S2 で生成した HTML を HTTP レスポンスとしてサービス利用者端末 100 に送信する。

【0132】

次に、ステップ S3 で送信された初期ページを、サービス利用者端末 100 が実行する（ステップ S4）。

【0133】

具体的には、ブラウザ部 133 は、ステップ S3 にて送信された HTTP レスポンスから HTML を取得し、HTML に基づいて画面を構成して出力情報生成部 132 を介して表示する。その際、ブラウザ部 133 は、当該 HTML ページに含まれる AP 提供装置 3

50

00のURIを取得する。

【0134】

次に、サービス利用者端末100は、AP提供装置300に対して、アプリケーションプログラムの開始要求を送信する(ステップS5)。

【0135】

具体的には、ブラウザ部133は、ステップS4にて取得したAP提供装置300のURIに対して、HTTPリクエストを送信する。

【0136】

次に、AP提供装置300は、要求のあったアプリケーションプログラムをサービス利用者端末100へ配信する(ステップS6)。

【0137】

具体的には、AP利用受付部331は、ステップS5において送信されたHTTPリクエストに含まれるURIに対応するアプリケーションサービスを実現するのに必要なAPスクリプト(チケット)160と、APスクリプト(プロジェクト)165と、スクリプトライブラリ180と、DSDAモジュール190と、APDAモジュール196とを特定する。そして、AP利用受付部331は、特定した各モジュールを含むアプリケーションプログラムをサービス利用者端末100へ配信する。

【0138】

次に、サービス利用者端末100は、ステップS6において配信されたアプリケーションプログラムの実行を開始する(ステップS7)。

【0139】

ここまでが、プロジェクト管理処理におけるアプリケーションサービスを開始するための開始フェーズである。

【0140】

次に、プロジェクト管理処理における実行フェーズを説明する。

【0141】

サービス利用者端末100は、アプリケーションプログラムの実行中に、AP提供装置300側で実行する処理の指示を受け付けた場合、AP提供装置300に関数実行要求を送信する(ステップS8)。

【0142】

具体的には、ブラウザ部133は、アプリケーションプログラムの実行中に、AP提供装置300に処理を要求するAPスクリプト(呼出)160またはAPDAモジュール196の実行指示を受け付けると、AP提供装置300に処理を依頼するHTTPリクエストを作成して送信する。その際、各処理において指定されたURIに対してHTTPリクエストを送信する。

【0143】

次に、AP提供装置300は、関数実行要求を受け付けると、実行要求のある関数を実行する(ステップS9)。

【0144】

具体的には、AP実行部332は、受け付けたURIに対応付けられたAP360またはDAモジュール(プロジェクト)370の処理の実行を開始する。AP360の処理を実施する場合は、AP実行部332は、DAモジュール(Wiki)380を介してデータベース領域324にアクセスして所定のデータ操作を行う。

【0145】

次に、AP提供装置300は、サービス利用者端末100に対して所定の関数をコールバックする(ステップS10)。

【0146】

具体的には、AP360の処理を行った場合には、AP実行部332は、構成したHTMLを表示するようにコールバックし、DAモジュール(プロジェクト)370の処理を行った場合には、取得したデータをJSON形式に成形してコールバックする。その際、

10

20

30

40

50

A P 実行部 3 3 2 は、A P 3 6 0 の実行要求 U R I にパラメータとして付加されたコールバック関数名称を、呼び出し元のアプリケーションプログラムの名称と結合させることで、コールバックする関数の名称を特定する。また、A P 実行部 3 3 2 は、A P 提供装置 3 0 0 のスクリプトライブラリ記憶領域 3 2 2 に格納されたスクリプトライブラリ 1 8 0 を利用して処理を行う。

【 0 1 4 7 】

次に、サービス利用者端末 1 0 0 は、コールバックに応じた関数を実行して、結果を表示する（ステップ S 1 1）。

【 0 1 4 8 】

なお、J S O N 形式でデータが受け渡された場合には、ブラウザ 1 3 3 は、J S O N 形式のデータに基づいてオブジェクトを構成して、続く処理を実行する。例えば、A P スクリプト（プロジェクト）1 7 0 の第二の処理内容であるプロジェクト一覧コールバック処理 1 7 2 のように、受け取った J S O N 形式のデータに基づいてオブジェクト化し、オブジェクト化したデータを用いて表示用のビューオブジェクトを生成してレンダ処理する。

10

【 0 1 4 9 】

また、サービス利用者端末 1 0 0 は、アプリケーションプログラムの実行中に、データ格納装置 2 0 0 にあるデータベース（本実施形態においてはチケット情報データベース）にアクセスする処理の指示を受け付けた場合、データ格納装置 2 0 0 にデータアクセス要求を送信する（ステップ S 1 2）。

【 0 1 5 0 】

具体的には、ブラウザ部 1 3 3 は、アプリケーションプログラムの実行中に、データ格納装置 2 0 0 に処理を要求する D S D A モジュール 1 9 6 を実行する A P スクリプト（チケット）1 6 5 の実行指示を受け付けると、データ格納装置 2 0 0 に処理を依頼する X M L （extensible Markup Language）H T T P リクエストを作成して送信する。

20

【 0 1 5 1 】

次に、データ格納装置 2 0 0 は、ステップ S 1 2 で受け付けたデータアクセス要求に従って、データアクセスを行う（ステップ S 1 3）。

【 0 1 5 2 】

具体的には、データ操作部 2 3 2 は、ステップ S 1 2 で受け付けたデータアクセス要求に従って、データベース領域 2 2 1 にアクセスする。

30

【 0 1 5 3 】

次に、データ格納装置 2 0 0 は、ステップ S 1 3 でデータベースにアクセスして得た結果の情報を返却する（ステップ S 1 4）。

【 0 1 5 4 】

次に、サービス利用者端末 1 0 0 は、ステップ S 1 4 で返却されたデータを使用して、アプリケーション処理の続きを実行する（ステップ S 1 5）。

【 0 1 5 5 】

以上が、本実施形態におけるプロジェクト管理処理の流れである。

【 0 1 5 6 】

図 1 6 と、図 1 7 に、プロジェクト管理処理における画面表示の例を示す。

40

【 0 1 5 7 】

図 1 6 は、A P スクリプト（プロジェクト）1 7 0 のうち、第一の処理内容であるプロジェクト一覧処理 1 7 1 を実施した結果のプロジェクト一覧表示画面 4 0 0 を示す。

【 0 1 5 8 】

プロジェクト一覧画面 4 0 0 は、プロジェクト名表示領域 4 1 0 を含む。プロジェクト名表示領域 4 1 0 には、既存のプロジェクトとして A P 提供装置 3 0 0 のデータベース領域 3 2 4 に格納されているプロジェクト情報がハイパーリンクを伴って全て表示される。図 1 6 においては、音声合成サービスプロジェクト 4 0 2 と、予算管理プロジェクト 4 0 3 とが示されている。

【 0 1 5 9 】

50

いずれかのプロジェクト名がクリックされると、ブラウザ部 133 は該当するプロジェクトの詳細を表示する処理、例えば A P スクリプト (プロジェクト) 170 の第三の処理内容であるプロジェクト表示処理 173 を起動する。

【0160】

図 17 は、上記した音声合成サービスプロジェクト 402 の詳細表示画面において、当該プロジェクトにおいて発行されたチケット情報を表示させた結果得られたチケット表示画面 410 の例である。当該チケット表示画面 410 は、A P スクリプト (チケット) 165 のうち、第一の処理内容であるチケット一覧処理 166 の結果表示される。

【0161】

チケット表示画面 410 は、チケット表示領域 411 と、新しいチケットを生成する指示の入力を受け付ける新しいチケット生成リンク表示 414 と、を含む。

10

【0162】

チケット表示領域 411 は、データ格納装置 200 のデータベース領域 221 にあるチケット情報データベースに格納されたチケットの情報を全て表示する領域である。図 17 においては、チケット表示領域 411 には、既存のチケットとして、音声途切れるバグに関するチケット 412 と、音声再生されないバグに関するチケット 413 と、が表示されている。

【0163】

ブラウザ部 133 は、チケット表示領域 411 のチケットに対して入力を受け付けると、入力を受け付けたチケットの詳細情報を表示する画面を表示する。

20

【0164】

また、ブラウザ部 133 は、新しいチケット生成リンク表示 414 に対して入力を受け付けると、新たなチケットを生成する処理、例えば A P スクリプト (チケット) 165 の第二の処理内容であるチケット生成処理 167 を実行する。

【0165】

以下、上記分散情報処理システム 1 を製造する際に用いる開発支援装置 500 について、説明する。

【0166】

開発支援装置 500 は、図 18 に示すように、開発支援装置 500 は、記憶部 520 と、制御部 530 と、入力部 541 と、出力部 542 と、通信部 543 と、を有する。

30

【0167】

記憶部 520 は、基準 A P 記憶領域 521 と、スクリプトライブラリ記憶領域 522 と、D A モジュール記憶領域 523 と、A P 実行場所記憶領域 524 と、モジュールリスト記憶領域 525 と、を備える。

【0168】

基準 A P 記憶領域 521 には、基準 A P が格納される。基準 A P とは、サービス利用者端末 100 のブラウザ部 133 により読み込まれて処理を行うアプリケーションである A P スクリプトおよび A P 360 を生成する元となるアプリケーションをいう。基準 A P は、データアクセス処理を除きサービス利用者端末 100 上において処理が完結するアプリケーションである。例えば、基準 A P は、ブラウザ部 133 により実行される想定で、JavaScript 等により記述されたアプリケーションである。基準 A P に含まれる処理のうち、データベースにアクセスする処理、すなわち使用するデータの C R U D (Create, Read, Update, Delete) に関する処理は、データベースが配置される装置が分散されている場合には他の装置に処理を依頼するが、アプリケーションに含まれる C R U D 以外のその他の処理は、他の装置に処理を依頼しないように記述される。

40

【0169】

図 19 は、基準 A P の例であるプロジェクト管理 A P 550 を示す図である。

【0170】

プロジェクト管理 A P 550 の第一の処理内容である W i k i 表示処理 551 を説明する。W i k i 表示処理 551 は、プロジェクト管理 A P 550 を含む H T M L ページに、

50

Wikiデータベースから取得した情報を一覧するWikiページを表示させるものである。

【0171】

プロジェクト管理AP550の第二の処理内容であるチケット一覧処理552を説明する。プロジェクト管理AP550のチケット一覧処理552は、プロジェクト管理AP550を含むHTMLページに、既に生成されたチケットの一覧情報を追加して表示させるものである。なお、当該処理においては、データ格納装置200のデータベース領域221に格納されたチケット情報を取得して一覧情報を生成する処理が含まれる。

【0172】

プロジェクト管理AP550の第三の処理内容であるチケット生成処理553を説明する。プロジェクト管理AP550のチケット生成処理553は、チケット情報を一件生成して、プロジェクト管理AP550を含むHTMLページに、既に生成されたチケットの一覧情報を追加して表示させるものである。なお、当該処理においては、データ格納装置200のデータベース領域221にチケット情報を新たに追加する処理と、データベース領域221に格納されたチケット情報を取得して一覧情報を生成する処理が含まれる。

10

【0173】

プロジェクト管理AP550の第四の処理内容であるプロジェクト一覧処理554を説明する。プロジェクト一覧処理554は、実行されると、データ格納装置200に対して、既に生成されたプロジェクト管理情報の一覧情報を要求し、プロジェクト管理AP550を含むHTMLページに、要求により取得したプロジェクト管理情報一覧のデータを追加して表示させるものである。

20

【0174】

次に、プロジェクト管理AP550の第五の処理内容であるプロジェクト表示処理555を説明する。プロジェクト表示処理555は、実行されると、引数に指定された識別子を有するプロジェクトのプロジェクト管理情報をデータ格納装置200に対して要求し、プロジェクト管理AP550を含むHTMLページに、要求により取得したプロジェクト管理情報を追加して表示させるものである。

【0175】

次に、プロジェクト管理AP550の第六の処理内容であるプロジェクト生成処理556を説明する。プロジェクト生成処理556は、実行されると、データ提供装置200に対して、新規のプロジェクト管理情報を生成して格納するよう要求し、プロジェクト管理AP550を含むHTMLページに、要求したプロジェクト管理情報が追加された状態のプロジェクト一覧情報を表示させるものである。なお、当該処理においては、データ格納装置200のデータベース領域221に新規のプロジェクト管理情報を格納させ、上記プロジェクト一覧処理554に処理を委譲して結果を表示する処理が含まれる。

30

【0176】

スクリプトライブラリ記憶領域522には、基準APの実行をサポートするユーティリティプログラム等を含むスクリプトライブラリ180が記憶される。

【0177】

スクリプトライブラリ180は、上述したサービス利用者端末100のAPスクリプト記憶領域121に記憶されるスクリプトライブラリ180と同一のものである。

40

【0178】

DAモジュール記憶領域523には、DAモジュールが記憶される。なお、DAモジュール記憶領域523に記憶されるDAモジュールは、後述するDAモジュール生成部532により生成されたDAモジュールである。本実施形態においては、DAモジュールは、AP実行部332により解釈されて実行されるJavaScriptにより記述されるが、これに限らず、サービス利用者端末100で実行されるAPスクリプトまたはAPP提供装置300のAP360から呼び出し可能であれば別の言語により記述されてもよい。

【0179】

なお、DAモジュール記憶領域523に記憶されるDAモジュールには、サービス利用

50

者端末100に格納されるDSDAモジュール190と、APDAモジュール196と、AP提供装置300に格納されるDAモジュール(プロジェクト)370と、DAモジュール(Wiki)380と、が含まれる。

【0180】

AP実行場所記憶領域524には、AP実行場所表560が格納される。

【0181】

AP実行場所表560は、AP生成部531が基準AP550からAPスクリプトおよびAP360を生成する際に用いる表である。AP実行場所表560は、基準AP550の処理ごとに、処理を実行する場所と、処理を実行するために必要なアクセス情報であるURLと、を格納する。

10

【0182】

図20に、AP実行場所表560の構成例を示す。

【0183】

AP実行場所表560は、処理名欄561と、実行場所欄562と、URL欄563と、を備える。

【0184】

処理名欄561には、基準AP550に含まれる処理の名称が格納される。

【0185】

実行場所欄562には、基準AP550の処理に対応する生成された処理が実行される場所が格納される。例えば、実行される場所として、サービス利用者端末100またはAP提供装置300を指定する情報が格納される。

20

【0186】

URL欄563には、基準AP550の処理に対応する生成された処理を実行するURLが格納される。

【0187】

例えば、処理名欄561に「Wiki表示処理」、実行場所欄562に「AP提供装置」、URL欄563に「<http://earth:3000/wikis/show>」が格納されているレコードは、基準AP550の「Wiki表示処理」は、「AP提供装置」にて実施されるAP360として生成され、その処理を実施するためのURLは「<http://earth:3000/wikis/show>」であること、を示す。

30

【0188】

モジュールリスト記憶領域525には、データ場所表600と、モジュールリスト610と、が格納される。

【0189】

データ場所表600は、DAモジュール生成部532がDSDAモジュール190、APDAモジュール196、DAモジュール(プロジェクト)370およびDAモジュール(Wiki)380を生成する際に用いる表である。データ場所表600は、データベースごとに、当該データベースの配置場所を示すURLを格納する。

【0190】

図21に、データ場所表600の構成例を示す。

40

【0191】

データ場所表600は、データ名欄601と、配置場所(URL)欄602と、を備える。

【0192】

データ名欄601には、基準AP550の処理内でアクセスされるデータの名称(データベースの名称)が格納される。

【0193】

配置場所欄602には、基準AP550の処理内でアクセスされるデータが格納されたデータベースの配置場所を示す情報(URL)が格納される。

【0194】

50

モジュールリスト 6 1 0 は、D A モジュール生成部 5 3 2 が D S D A モジュール 1 9 0、A P D A モジュール 1 9 6、D A モジュール ( プロジェクト ) 3 7 0 および D A モジュール ( W i k i ) 3 8 0 を生成する際に用いる表である。モジュールリスト 6 1 0 は、データベースごとに、当該データベースへアクセスするための D A モジュールの名称を格納する。

【 0 1 9 5 】

図 2 2 に、モジュールリスト 6 1 0 の構成例を示す。

【 0 1 9 6 】

モジュールリスト 6 1 0 は、データ名欄 6 1 1 と、モジュール名欄 6 1 2 と、を備える。

10

【 0 1 9 7 】

データ名欄 6 1 1 には、基準 A P 5 5 0 の処理内でアクセスされるデータの名称 ( データベースの名称 ) が格納される。

【 0 1 9 8 】

モジュール名欄 6 1 2 には、D A モジュールを特定する名称等の情報が格納される。

【 0 1 9 9 】

図 1 8 の説明に戻る。

【 0 2 0 0 】

制御部 5 3 0 は、A P 生成部 5 3 1 と、D A モジュール生成部 5 3 2 と、を備える。

【 0 2 0 1 】

A P 生成部 5 3 1 は、基準 A P 記憶領域 5 2 1 に格納された基準 A P 5 5 0 と、A P 実行場所記憶領域 5 2 4 に格納された A P 実行場所表 5 6 0 と、モジュールリスト記憶領域 5 2 5 に格納されたデータ場所表 6 0 0 と、を用いて、サービス利用者端末 1 0 0 上で動作する A P スクリプトと、A P 提供装置 3 0 0 上で動作する A P 3 6 0 と、を生成する。

20

【 0 2 0 2 】

D A モジュール生成部 5 3 2 は、モジュールリスト 6 1 0 と、データ場所表 6 0 0 と、を用いて、D S D A モジュール 1 9 0 と、A P D A モジュール 1 9 6 と、D A モジュール ( プロジェクト ) 3 7 0 と、D A モジュール ( W i k i ) 3 8 0 と、を生成する。

【 0 2 0 3 】

入力部 5 4 1 は、操作者からの情報の入力を受け付ける。

30

【 0 2 0 4 】

出力部 5 4 2 は、情報を出力する。

【 0 2 0 5 】

本実施形態の開発支援装置 5 0 0 のハードウェア構成を説明する。

【 0 2 0 6 】

本実施形態においては、開発支援装置 5 0 0 は、例えば、クライアント P C ( パーソナルコンピュータ ) や、ワークステーション、サーバ装置、各種携帯電話端末、P D A ( Personal Digital Assistant ) などの計算機である。

【 0 2 0 7 】

開発支援装置 5 0 0 は、図 1 4 に示したサービス利用者端末 1 0 0 と基本的に同様の構成を備える。ただし、通信装置 1 1 6 を備えない点において相違する。

40

【 0 2 0 8 】

開発支援装置 5 0 0 の A P 生成部 5 3 1 と、D A モジュール生成部 5 3 2 と、は、開発支援装置 5 0 0 における演算装置 1 1 3 に処理を行わせるプログラムによって実現される。

【 0 2 0 9 】

このプログラムは、主記憶装置 1 1 4 または外部記憶装置 1 1 5 内に記憶され、実行にあたって主記憶装置 1 1 4 上にロードされ、演算装置 1 1 3 により実行される。

【 0 2 1 0 】

また、開発支援装置 5 0 0 の記憶部 5 2 0 は、開発支援装置 5 0 0 における主記憶装置

50

114または外部記憶装置115により実現される。

【0211】

開発支援装置500の入力部541は、開発支援装置500における入力装置111によって実現される。

【0212】

開発支援装置500の出力部542は、開発支援装置500における出力装置112によって実現される。

【0213】

次に、本実施形態における開発支援装置のAP生成処理について、図23に基づいて説明する。

【0214】

図23は、AP生成処理の処理フローを示す図である。

【0215】

AP生成処理は、開発支援装置500の入力部541が、処理の開始指示を受け付けることで実施される。

【0216】

まず、AP生成部531は、基準AP550に記載された最初の関数を対象の関数として特定する(ステップS101)。

【0217】

例えば、AP生成部531は、基準AP記憶領域521に格納された基準AP550を読み出し、記述された順に関数を読み出すと、その最初に記述された関数を対象関数として特定する。

【0218】

次に、AP生成部531は、ステップS101で特定した対象関数の実行場所を特定する(ステップS102)。

【0219】

具体的には、AP生成部531は、ステップS101で特定した対象関数の名称と、AP実行場所記憶領域524に格納されたAP実行場所表560の処理名欄561の値とが一致するレコードを特定して、その実行場所欄562に格納された実行場所を特定する。

【0220】

次に、AP生成部531は、ステップS102にて特定した実行場所が利用者端末であるか否かを判定する(ステップS103)。

【0221】

実行場所が利用者端末でない場合(ステップS103にて「No」)、AP生成部531は、対象関数をAP提供装置300のAP360として生成し、AP360を呼び出すためのスクリプトであるサービス利用者端末100のAPスクリプト(呼出)160を生成する。そして、処理を後述するステップS108に進める(ステップS104)。

【0222】

具体的には、AP生成部531は、対象関数の処理内容をそのまま処理内容とするAP360を基準AP記憶領域521に格納する。そして、AP生成部531は、当該AP360をサービス利用者端末100のブラウザ部133から呼び出すための処理を定型に従って生成する。その際、AP実行場所表560のURL欄563に格納されたURLを用いてAP360を呼び出すように処理を生成する。生成した処理をAPスクリプト(呼出)160として基準AP記憶領域521に格納する。そして、処理を後述するステップS108に進める。

【0223】

実行場所が利用者端末である場合(ステップS103にて「Yes」)、AP生成部531は、対象関数の処理内容を検索して、アクセスするデータがAP提供装置300上に格納されるデータか否かを判定する(ステップS105)。

【0224】

10

20

30

40

50

具体的には、A P生成部531は、対象関数内の処理において、「find」、「find\_all」、「save」、「destroy」等の所定の関数のうちいずれかの関数を実行している処理があれば、その関数実行を行うオブジェクトを特定する。そして、A P生成部531は、特定したオブジェクトの名称と、データ場所表600のデータ名欄601に格納された値と、が対応するレコードをデータ場所表600から特定し、その配置場所(URL)欄602に格納された値がA P提供装置300を指すものであるか否かを判定する。

【0225】

アクセスするデータがA P提供装置300上に格納されるデータではない場合(ステップS105にて「No」)、A P生成部531は、対象関数の処理内容を有するサービス利用者端末100のA Pスクリプトを生成する。そして、処理を後述するステップS108に進める(ステップS106)。

10

【0226】

具体的には、A P生成部531は、対象関数の処理内容をそのまま処理内容とするA Pスクリプトを生成して基準A P記憶領域521に格納する。そして、処理を後述するステップS108に進める。

【0227】

アクセスするデータがA P提供装置300上に格納されるデータである場合(ステップS105にて「Yes」)、A P生成部531は、対象関数に基づいて、コールバック用関数を生成して、サービス利用者端末100のA Pスクリプトに追加する。そしてA P生成部531は、対象関数を、サービス利用者端末100のA Pスクリプトに追加する(ステップS107)。

20

【0228】

具体的には、A P生成部531は、対象関数の処理内容を検索して、データアクセスを行う処理に続く処理以降に記載された処理を切り出して、コールバック用関数として生成する。そしてA P生成部531は、生成したコールバック用関数を利用者端末100のA Pスクリプトに追加する。そして、A P生成部531は、対象関数のデータアクセスを行う処理より前に記載された処理を切り出して、サービス利用者端末100のA Pスクリプトとして生成する。そして、A P生成部531は、生成したA Pスクリプトを基準A P記憶領域521に格納する。

【0229】

なお、対象関数の処理内容にループ制御(forループ制御)が含まれる場合には、図24に示すループ変換処理フローに従って、A P生成部531はA Pスクリプトを生成する。

30

【0230】

ここで、図24のループ変換処理フローを説明する。

【0231】

まず、A P生成部531は、基準A P550の関数に含まれるforループブロックに対応する無名関数をA Pスクリプトに追加する。そして、A P生成部531は、基準A P550の関数に含まれるforループブロックのループ変数を追加した無名関数の変数として追加する(ステップS121)。

40

【0232】

次に、A P生成部531は、基準A P550の関数に含まれるforループブロックに含まれる処理を、詳細ブロックに分割する(ステップS122)。

【0233】

具体的には、A P生成部531は、基準A P550の関数に含まれるforループブロックに含まれる処理の先頭行から、最初にA P提供装置300の処理を呼び出す処理が記載された行までを、第一の詳細ブロックとして特定する。そして、A P生成部531は、当該A P提供装置300の処理を呼び出す処理が記載された行の次の行から、次にA P提供装置300の処理を呼び出す処理が記載された行までを第二の詳細ブロックとして特定する。同様に、A P生成部531は、forループブロックに含まれる処理の最終行に到るま

50

で、forループブロックに含まれる処理を詳細ブロックに分割する。

【0234】

次に、AP生成部531は、ステップS122において、forループブロックが2以上の詳細ブロックに分割されたか否かを判定する(ステップS123)。

【0235】

forループブロックが2以上の詳細ブロックに分割されなかった場合(ステップS123にて「No」)、AP生成部531は、ステップS121で追加した無名関数に内部関数を追加する(ステップS124)。

【0236】

次に、AP生成部531は、内部関数の処理として、基準AP550の関数に含まれるforループブロックのループ継続条件と同様の継続条件を満たさない場合には当該内部関数の処理を終了させる終了判定処理を追加する(ステップS125)。

【0237】

次に、AP生成部531は、内部関数の処理として、基準AP550の関数に含まれるforループブロックの処理をすべて追加する(ステップS126)。

【0238】

次に、AP生成部531は、内部関数の処理の最後に、ループ変数をインクリメントする処理と、内部関数を呼び出す処理と、を追加し、ループ変換処理を終える(ステップS127)。

【0239】

forループブロックが2以上の詳細ブロックに分割された場合(ステップS123にて「Yes」)、AP生成部531は、ステップS121で追加した無名関数に内部関数と、分割された詳細ブロックの数より少ない数のコールバック関数を追加する(ステップS128)。

【0240】

次に、AP生成部531は、内部関数の処理として、基準AP550の関数に含まれるforループブロックのループ継続条件と同様の継続条件を満たさない場合には当該内部関数の処理を終了させる終了判定処理を追加する(ステップS129)。

【0241】

次に、AP生成部531は、内部関数の処理として、基準AP550の関数に含まれるforループブロックの処理のうち、第一の詳細ブロックの処理を追加する(ステップS130)。なお、その際、AP生成部531は、AP提供装置300の処理を呼び出す処理のパラメータに、第一のコールバック関数の関数名を追加する。

【0242】

次に、AP生成部531は、ステップS128にて追加した第一のコールバック関数に、第二の詳細ブロックの処理を追加する(ステップS131)。また、ステップS128にて二つ以上のコールバック関数が追加されている場合、第三、第四、・・・、第N(Nは三以上の自然数)の詳細ブロックの処理を、それぞれ第二、第三、・・・、第(N-1)のコールバック関数に追加する。なお、その際、AP生成部531は、AP提供装置300の処理を呼び出すパラメータに、次に続くコールバック関数の関数名を追加する。

【0243】

次に、AP生成部531は、ステップS131にて追加した最後のコールバック関数の処理の最後に、ループ変数をインクリメントする処理と、内部関数を呼び出す処理と、を追加し、ループ変換処理を終える(ステップS132)。

【0244】

以上が、ループ変換処理のフローである。

【0245】

AP生成処理のフローの説明に戻る。

【0246】

次に、AP生成部531は、基準AP550の関数のうち、対象関数の次に記載された

10

20

30

40

50

関数を新たに対象関数とし、制御をステップ S 1 0 2 に戻す。基準 A P 5 5 0 に、対象関数の次の関数が無かった場合、A P 生成処理を終了する（ステップ S 1 0 8 ）。

【 0 2 4 7 】

以上が、A P 生成処理のフローである。

【 0 2 4 8 】

このような A P 生成処理を行うことで、サービス利用者端末 1 0 0 上で動作するよう記載された基準 A P 5 5 0 から、A P 提供装置 3 0 0 上で動作する処理を分離して A P スクリプトと A P 3 6 0 とを生成することが可能となる。

【 0 2 4 9 】

次に、本実施形態における D A モジュール生成処理のフローについて、図 2 5 に基づき説明する。

【 0 2 5 0 】

図 2 5 は、D A モジュール生成処理の流れを示すフロー図である。

【 0 2 5 1 】

D A モジュール生成処理は、開発支援装置 5 0 0 の入力部 5 4 1 が、処理の開始指示を受け付けることで実施される。

【 0 2 5 2 】

まず、D A モジュール生成部 5 3 2 は、モジュールリスト 6 1 0 の先頭に記載されたモジュールを、生成対象 D A モジュールとして特定する（ステップ S 2 0 1 ）。

【 0 2 5 3 】

具体的には、D A モジュール生成部 5 3 2 は、モジュールリスト 6 1 0 に記載された先頭のデータベースについて、モジュール名欄 6 1 2 に記載された名称を有する D A モジュールを生成対象として特定する。

【 0 2 5 4 】

次に、D A モジュール生成部 5 3 2 は、ステップ S 2 0 1 で特定した生成対象 D A モジュールの配置場所を特定し、サービス利用者端末 1 0 0 からデータ格納装置 2 0 0 に格納されたデータにアクセスする D S D A モジュール 1 9 0 と、サービス利用者端末 1 0 0 から A P 提供装置 3 0 0 に格納されたデータにアクセスする A P D A モジュール 1 9 6 と、A P 提供装置 3 0 0 から A P 提供装置 3 0 0 に格納されたデータにアクセスする D A モジュール 3 7 0 、 3 8 0 と、を生成する（ステップ S 2 0 2 ）。

【 0 2 5 5 】

具体的には、D A モジュール生成部 5 3 2 は、ステップ S 2 0 1 で特定した生成対象 D A モジュールについて生成する際に、生成対象の D A モジュールによってアクセスされるデータベースの配置場所に応じて、異なるアクセス方法を実装する D A モジュールを複数生成する。本実施形態においては、D A モジュール生成部 5 3 2 は、生成対象 D A モジュールについて、サービス利用者端末 1 0 0 からデータ格納装置 2 0 0 に格納されたデータにアクセスする D S D A モジュール 1 9 0 と、サービス利用者端末 1 0 0 から A P 提供装置 3 0 0 に格納されたデータにアクセスする A P D A モジュール 1 9 6 と、A P 提供装置 3 0 0 から A P 提供装置 3 0 0 に格納されたデータにアクセスする D A モジュール 3 7 0 、 3 8 0 と、の 4 パターンのアクセスに対応する D A モジュールを生成する。

【 0 2 5 6 】

次に、D A モジュール生成部 5 3 2 は、モジュールリスト 6 1 0 に次の対象モジュールとなるレコードがあるか否かを判定し、対象モジュールとなるレコードがなければ（「N o 」の場合）D A モジュール生成処理を終了する（ステップ S 2 0 3 ）。

【 0 2 5 7 】

対象モジュールとなるレコードがあれば（ステップ S 2 0 3 で「Y e s 」）、D A モジュール生成部 5 3 2 は、モジュールリスト 6 1 0 から次の対象モジュールを読み出して、新たに生成対象 D A モジュールとして特定し、ステップ S 2 0 2 に制御を戻す。

【 0 2 5 8 】

以上が、D A モジュール生成処理の処理フローである。

10

20

30

40

50

## 【0259】

DAモジュール生成処理を行うことで、一つのデータベースに対して、複数のアクセス方法でアクセスするDAモジュールを複数生成することができる。

## 【0260】

すなわち、AP提供装置300が提供するデータベース（本実施形態においてはプロジェクト情報データベースとWikiデータベース）を使用せず、データ格納装置200に独自のプロジェクト情報データベースとWikiデータベースを使用する利用者組織10が存在する場合であっても、上記DAモジュール生成処理によってプロジェクト情報データベースがデータ格納装置200にある場合のDAモジュールも生成されるため、当該利用者組織10に属するサービス利用者端末100からも問題なくAP提供装置300が提供

10

## 【0261】

以上が、第一の実施形態である。

## 【0262】

第一の実施形態によれば、アプリケーションプロバイダ組織30は、当該アプリケーションサービスを提供する上でのノウハウや管理情報、プログラムといった非公開にしたい情報を含む処理を公開することなく、アプリケーションサービスをサービス利用者端末100に提供することが可能となる。

## 【0263】

また、利用者組織10は、第三者に知られたくない情報をアプリケーションプロバイダ組織30に対して流出させてしまうことを防ぎつつ、アプリケーションプロバイダ組織30が提供するアプリケーションサービスを受けることが可能となる。

20

## 【0264】

なお、データプロバイダ組織20は、上記実施形態においては利用者組織10に属する組織であるとしているが、これに限られない。すなわち、データプロバイダ組織20は、利用者組織10とは異なる独立した組織であってもよい。または、アプリケーションプロバイダ組織30に属する組織であっても良い。

## 【0265】

続いて、本発明の第二の実施形態を説明する。

## 【0266】

本実施形態における分散情報処理システム1は、第一の実施形態における分散情報処理システム1と基本的に同様の構成を有する。

30

## 【0267】

ただし、第二の実施形態における分散情報処理システム1は、第一の実施形態における分散情報処理システム1と、以下の差異を有する。

## 【0268】

第二の実施形態における分散情報処理システム1は、データ提供装置200に相当する装置として、サービス利用者端末100の記憶部120に、データベース領域221に相当するデータベース領域を備え、制御部130に、データ操作部232に相当するデータ操作部を備える。また、AP提供装置300の制御部330に、初期画面生成部231に相当する初期画面生成部を備える。

40

## 【0269】

また、開発支援装置500が生成するDAモジュールは、ローカルデータベースアクセスをサポートするライブラリのAPI呼び出しを行うものとする。

## 【0270】

第二の実施形態におけるプロジェクト管理処理について、第一の実施形態におけるプロジェクト管理処理の処理フローとの差異を説明する。

## 【0271】

第二の実施形態におけるプロジェクト管理処理では、まず、サービス利用者端末100のブラウザ部133は、AP提供装置300に対して初期ページを要求する。そして、A

50

P 提供装置 300 は、初期ページを生成する。次に、AP 提供装置 300 は、生成した初期ページをサービス利用者端末 100 に送信する（ステップ S1～S3 に相当する処理）。

【0272】

以降、ステップ S4 からステップ S11 に相当する処理は、第一の実施形態と同様である。

【0273】

ステップ S12 からステップ S14 に相当する処理においては、サービス利用者端末 100 は、アプリケーションプログラムの実行中に、サービス利用者端末 100 にあるデータベース（本実施形態においてはチケット情報データベース）にアクセスする処理の指示を受け付けた場合、データ操作部にデータアクセス要求を送信し、データ操作部は、データアクセスを行い、アクセスして得た結果の情報を返却する。

10

【0274】

このような第二の実施形態によると、サービス利用者は、極めて機密性を保持すべき情報をアプリケーションプロバイダ組織 30 およびデータプロバイダ組織 20 に対して流出させてしまうことを防ぎつつ、アプリケーションプロバイダ組織 30 が提供するアプリケーションサービスを利用することが可能となる。

【0275】

以上、実施の形態に基づき具体的に説明したが、これに限定されるものではなく、その要旨を逸脱しない範囲で種々の変更が可能である。

20

【0276】

例えば、上記第一の実施形態の開発支援装置 500 は、AP 生成処理及び DA モジュール生成処理において、生成したプログラムを記憶部 520 に格納しているが、これに限られない。すなわち、開発支援装置 500 は組織間ネットワーク 250 に接続可能な通信部 543 を備え、生成したプログラムを、AP 提供装置 300 の記憶部 320 に格納するようにしてもよい。

【0277】

このようにすることによって、生成したプログラムを稼働サーバに配信するのが容易になる。

【0278】

なお、分散情報処理システム 1 は、システムとして取引対象とするだけでなく、各装置や、各装置の動作を実現するプログラム部品単位で取引対象とすることも可能である。

30

【符号の説明】

【0279】

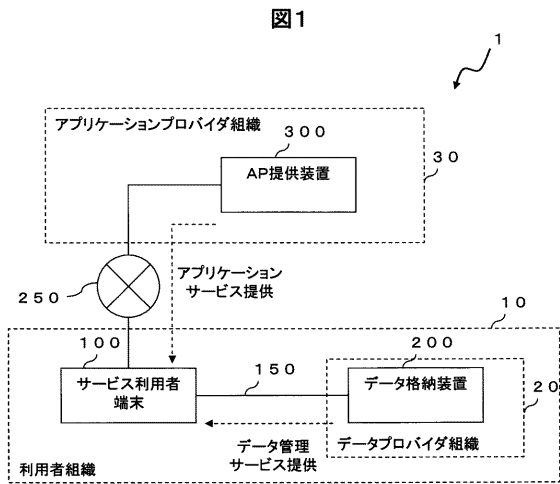
1：分散情報処理システム、10：利用者組織、20：データプロバイダ組織、30：アプリケーションプロバイダ組織、100：サービス利用者端末、111：入力装置、112：出力装置、113：演算装置、114：主記憶装置、115：外部記憶装置、116：通信装置、117：バス、120：記憶部、121：AP スクリプト記憶領域、122：DSDA モジュール記憶領域、123：APDA モジュール記憶領域、130：制御部、131：入力情報受付部、132：出力情報生成部、133：ブラウザ部、141：入力部、142：出力部、143：通信部、150：組織内ネットワーク、160：AP スクリプト（呼出）、165：AP スクリプト（チケット）、170：AP スクリプト（プロジェクト）、180：スクリプトライブラリ、190：DSDA モジュール、195：APDA モジュール、200：データ格納装置、220：記憶部、221：データベース領域、230：制御部、231：初期画面生成部、232：データ操作部、243：通信部、250：組織間ネットワーク、300：AP 提供装置、320：記憶部、321：AP 記憶領域、322：スクリプトライブラリ記憶領域、323：DA モジュール記憶領域、324：データベース領域、330：制御部、331：AP 利用受付部、332：AP 実行部、333：データ操作部、341：入力部、342：出力部、343：通信部、360：AP、370：DA モジュール（プロジェクト）、380：DA モジュール（Wi

40

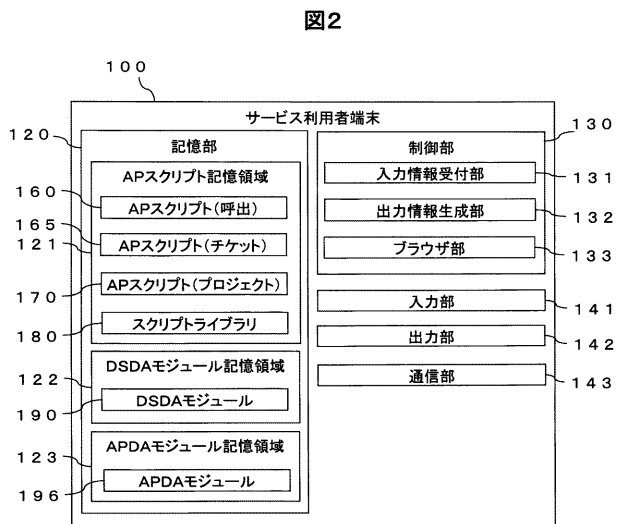
50

k i )、500：開発支援装置、520：記憶部、521：基準AP記憶領域、522：スクリプトライブラリ記憶領域、523：DAモジュール記憶領域、524：AP実行場所記憶領域、525：モジュールリスト記憶領域、530：制御部、531：AP生成部、532：DAモジュール生成部、541：入力部、542：出力部、550：基準AP、560：AP実行場所表、600：データ場所表、610：モジュールリスト

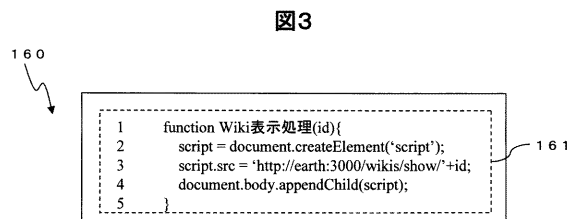
【 図 1 】



【 図 2 】



【 図 3 】



【 図 4 】

図4

```

165 ↘
1  function チケット一覧処理(id){
2     tickets = Ticket.find_all();
3     view = チケット一覧処理のビュー生成処理;
4     render(view);
5  }
6  function チケット生成処理(name){
7     ticket = new Ticket({name:name});
8     ticket.save();
9     redirect_to(チケット一覧処理);
10  }
166
167

```

【 図 5 】

図5

```

170 ↘
1  function プロジェクト一覧処理(){
2     Project.find_all(project_find_all_callback);
3  }
4  function プロジェクト一覧処理 Project_find_all_callback(data){
5     projects = eval(data);
6     view = プロジェクト一覧処理のビュー生成処理;
7     render(view);
8  }
9  function プロジェクト表示処理(){
10     Project.find(id,Project_find_callback);
11  }
12 function Project_find_callback(data){
13     project = eval(data);
14     view = プロジェクト表示処理のビュー生成処理;
15     render(view);
16  }
17 function プロジェクト生成処理(name){
18     project = new Project({name:name});
19     project.save(Project_save_callback);
20     view = プロジェクト生成処理のビュー生成処理;
21     render(view);
22  }
23 function Project_save_callback(){
24     redirect_to(プロジェクト一覧処理);
25  }
171
172
173
174
175
176

```

【 図 6 】

図6

```

180 ↘
1  function render(view){
2     document.body.innerHTML = view;
3  }
4  function redirect_to(action){
5     action;
6  }
181
182

```

【 図 7 】

図7

```

190 ↘
1  function Ticket(params){
2     this.name = params.name;
3     this.db = new DB('tickets');
4  }
5  Ticket.find = function(id){
6     db = new DB('tickets');
7     doc = db.open(id);
8     ticket = new Ticket({name:doc.name});
9     ticket.id = doc._id;
10    return ticket;
11  }
12 Ticket.find_all = function(){
13     db = new DB('tickets');
14     docs = db.allDocs();
15     tickets = new Array(docs.rows.length);
16     for (l = 0; l < docs.rows.length ; l++){
17         doc = db.open(docs.rows[l].id);
18         tickets[l] = new Ticket({name:doc.name});
19         tickets[l].id = doc._id;
20     }
21     return tickets;
22  }
23 Ticket.prototype.save = function(){
24     doc = {name:this.name};
25     this.id = this.db.save(doc).id;
26  }
27 Ticket.prototype.destroy = function(){
28     doc = this.db.open(this.id);
29     this.db.deleteDoc(doc);
30  }
191
192
193
194
195

```

【 図 8 】

図8

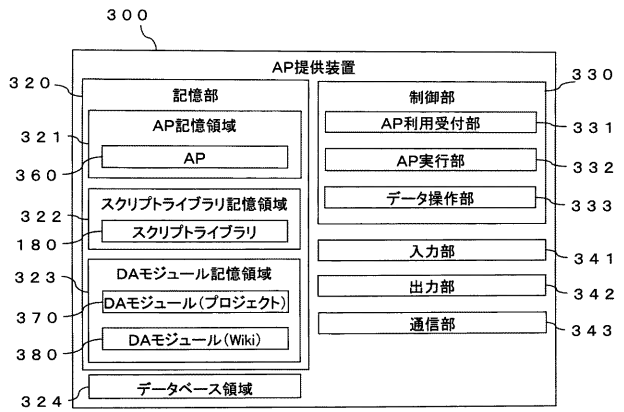
```

196
1 function Project(){
2   Project.find = function(id,callback){
3     script = document.createElement('script');
4     script.src = 'http://earth:3000/projects/show' + id + '?callback=' + callback;
5     document.body.appendChild(script);
6   }
7   Project.find_all = function(callback){
8     script = document.createElement('script');
9     script.src = 'http://earth:3000/projects/list?callback=' + callback;
10    document.body.appendChild(script);
11  }
12  Project.save = function(callback){
13    script = document.createElement('script');
14    script.src = 'http://earth:3000/projects/save?callback=' + callback;
15    document.body.appendChild(script);
16  }
197
198
199

```

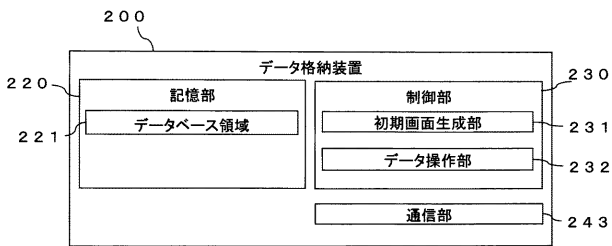
【 図 1 0 】

図10



【 図 9 】

図9



【 図 1 1 】

図11

```

360
1 function Wiki表示処理(id){
2   wiki = Wiki.find(id);
3   view = Wiki表示処理のビュー生成処理;
4   コールバック関数呼び出し(render(view));
5 }
361

```

【 図 1 2 】

図12

```

370
1 function Project(params){
2   this.name = params.name;
3   this.db = new DB('projects');
4 }
5 Project.find = function(id){
6   db = new DB('projects');
7   doc = db.open(id);
8   project = new Project({name:doc.name});
9   project.id = doc._id;
10  return project;
11 }
12 Project.find_all = function(){
13   db = new DB('projects');
14   docs = db.allDocs();
15   projects = new Array(docs.rows.length);
16   for (i = 0; i < docs.rows.length; i++){
17     doc = db.open(docs.rows[i].id);
18     projects[i] = new Project({name:doc.name});
19     projects[i].id = doc._id;
20   }
21  return projects;
22 }
23 Project.prototype.save = function(){
24   doc = {name:this.name};
25   this.id = this.db.save(doc).id;
26 }
27 Project.prototype.destroy = function(){
28   doc = this.db.open(this.id);
29   this.db.deleteDoc(doc);
30 }
371
372
373
374
375

```

【 図 1 3 】

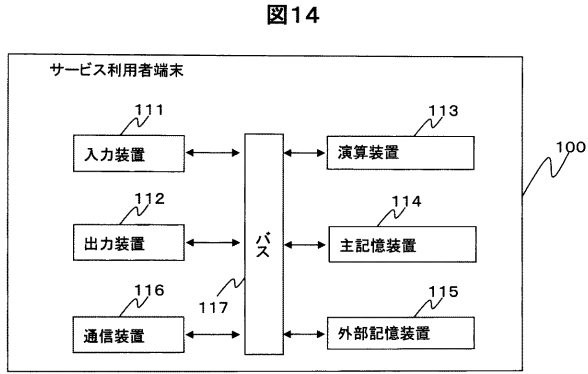
図13

```

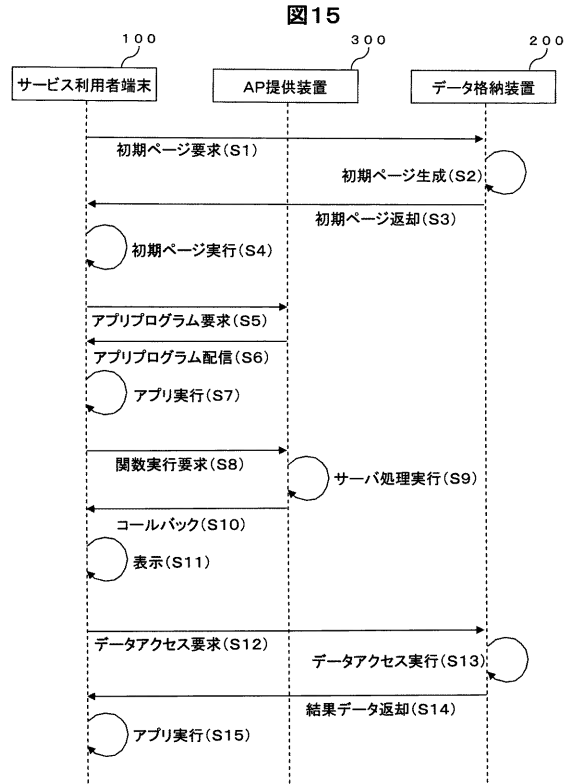
380
1 function Wiki(params){
2   this.name = params.name;
3   this.db = new DB('wikis');
4 }
5 Wiki.find = function(id){
6   db = new DB('wikis');
7   doc = db.open(id);
8   wiki = new Wiki({name:doc.name});
9   wiki.id = doc._id;
10  return wiki;
11 }
12 Wiki.find_all = function(){
13   db = new DB('wikis');
14   docs = db.allDocs();
15   wikis = new Array(docs.rows.length);
16   for (i = 0; i < docs.rows.length; i++){
17     doc = db.open(docs.rows[i].id);
18     wikis[i] = new Wiki({name:doc.name});
19     wikis[i].id = doc._id;
20   }
21  return wikis;
22 }
23 Wiki.prototype.save = function(){
24   doc = {name:this.name};
25   this.id = this.db.save(doc).id;
26 }
27 Wiki.prototype.destroy = function(){
28   doc = this.db.open(this.id);
29   this.db.deleteDoc(doc);
30 }
381
382
383
384
385

```

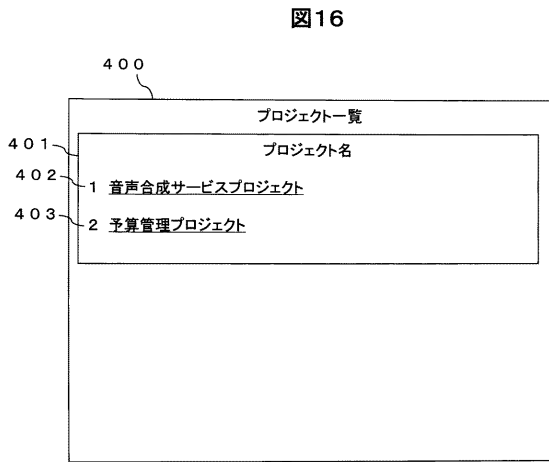
【 図 1 4 】



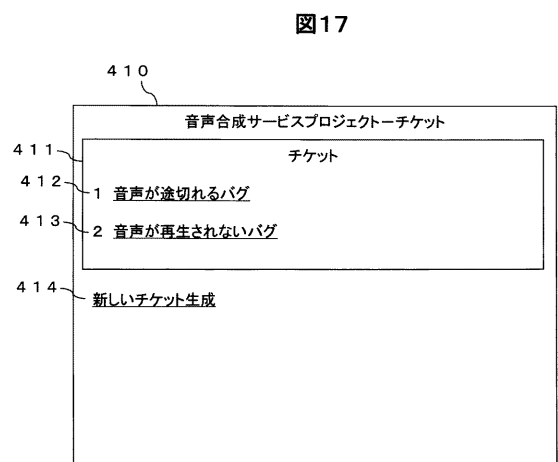
【 図 1 5 】



【 図 1 6 】

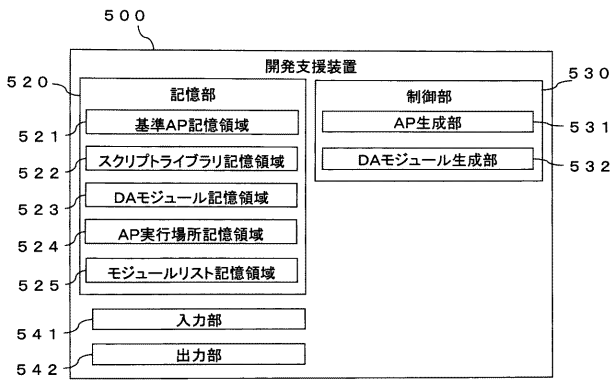


【 図 1 7 】



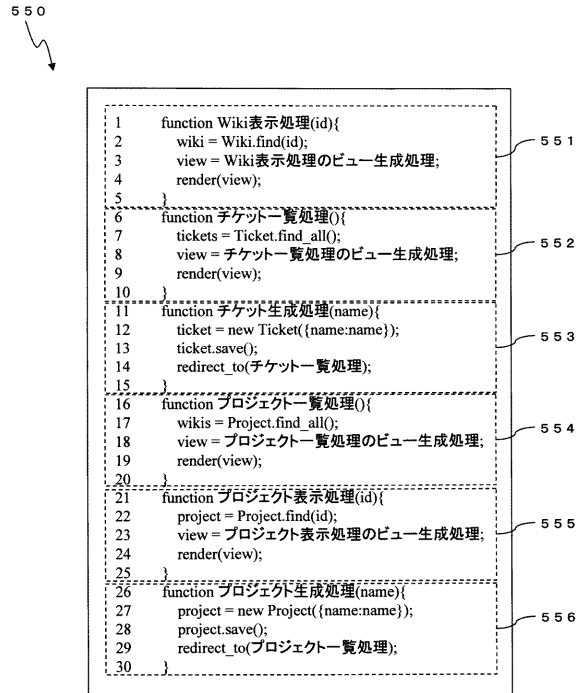
【図18】

図18



【図19】

図19



【図20】

図20

処理名	実行場所	URL
Wiki表示処理	AP提供装置	http://earth:3000/wikis/show
チケット一覧処理	サービス利用者端末	
チケット生成処理	サービス利用者端末	
プロジェクト一覧処理	サービス利用者端末	
プロジェクト表示処理	サービス利用者端末	
プロジェクト生成処理	サービス利用者端末	

【図21】

図21

データ名	配置場所(URL)
wikis	(AP提供装置のURL)
tickets	(データ格納装置内の相対URL)
projects	(AP提供装置のURL)

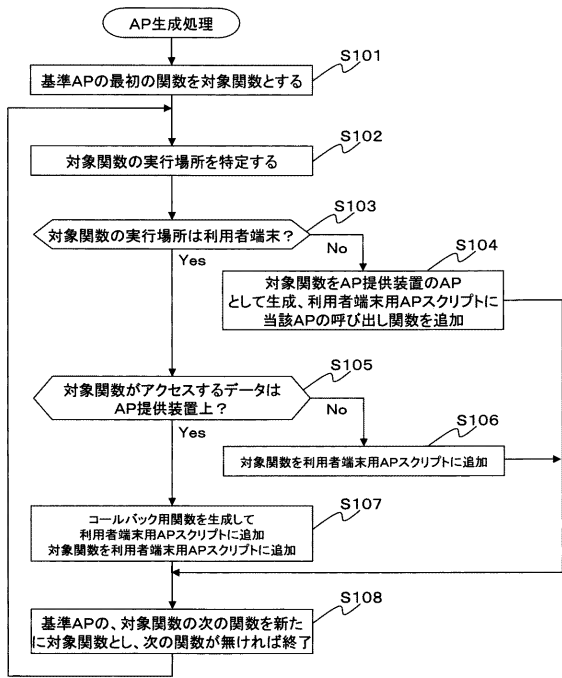
【図22】

図22

データ名	モジュール名
projects	Project
wikis	Wiki
tickets	Ticket

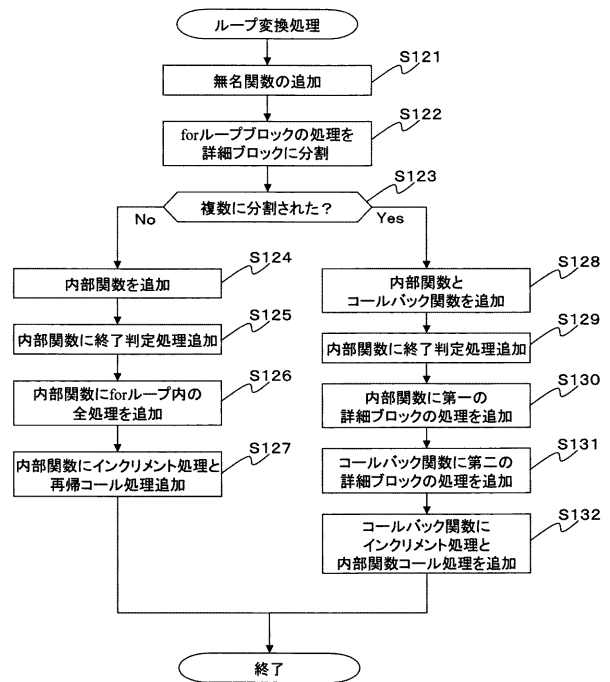
【 図 2 3 】

図23



【 図 2 4 】

図24



【 図 2 5 】

図25

