

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4711922号  
(P4711922)

(45) 発行日 平成23年6月29日(2011.6.29)

(24) 登録日 平成23年4月1日(2011.4.1)

(51) Int. Cl. F I  
**G 0 6 F 15/00 (2006.01)** G O 6 F 15/00 3 1 0 A  
**G 0 6 F 9/50 (2006.01)** G O 6 F 9/46 4 6 5 B

請求項の数 8 (全 18 頁)

<p>(21) 出願番号 特願2006-257914 (P2006-257914)                  (22) 出願日 平成18年9月22日 (2006.9.22)                  (65) 公開番号 特開2008-77511 (P2008-77511A)                  (43) 公開日 平成20年4月3日 (2008.4.3)                  審査請求日 平成20年12月24日 (2008.12.24)</p>	<p>(73) 特許権者 301015956                  キヤノンソフトウェア株式会社                  東京都港区三田3丁目9番6号                  (74) 代理人 100145827                  弁理士 水垣 親房                  (72) 発明者 上田 勲                  東京都港区三田3丁目9番7号 キヤノン                  ソフトウェア株式会社内                  審査官 漆原 孝治                  (56) 参考文献 特開2005-267618 (JP, A )</p>
--	---

最終頁に続く

(54) 【発明の名称】 サーバおよびサーバ連携システムおよびサーバ連携方法およびプログラムおよび記録媒体および情報処理装置

(57) 【特許請求の範囲】

【請求項1】

クライアント端末を含み、当該クライアント端末から入力された入力データの検証を第1のWebアプリケーションが行って、当該検証の結果得られる結果データに応じて第2のWebアプリケーションへ処理を連携させるWebアプリケーション連携システムであって、

前記Webアプリケーション連携システムは、

前記クライアント端末からの要求に従って、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて、結果データが妥当である旨の結果の場合に実行するコードと、妥当でない旨の結果である場合に実行するコードとを含む連携処理実行プログラムにより、前記第2のWebアプリケーションへ処理を連携するための操作指示とを可能とする入力画面を当該クライアント端末のWebブラウザに表示可能な画面データであって、当該連携処理実行プログラムを含む画面データを、当該クライアントへ送信する画面データ送信手段と、

前記クライアント端末から送信された前記第1のWebアプリケーションに対する入力データを受信する第一の入力データ受信手段と、

前記第一の入力データ受信手段により受信された入力データに対する前記第1のWebアプリケーションによる検証の結果得られる結果データを、前記クライアント端末へ送信する結果データ送信手段と、

前記クライアント端末から送信された前記第2のWebアプリケーションに対する入力

データを受信する第二の入力データ受信手段とを有し、

前記クライアント端末は、

前記画面データ送信手段により送信された入力画面の画面データを受信する画面データ受信手段と、

前記画面データ受信手段により受信された画面データを用いてWebブラウザに表示された入力画面を介して、ユーザにより、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて前記第2のWebアプリケーションへ処理を連携するための操作指示とがなされた場合、前記連携処理実行プログラムにより、当該入力データを前記第1のWebアプリケーションへ送信する第一の入力データ送信手段と、

10

前記結果データ送信手段により送信された結果データを受信する結果データ受信手段と、

前記結果データ受信手段により受信した結果データを用いて、前記連携処理実行プログラムにより、当該結果データが妥当である旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当である旨の結果の場合に実行するコードを実行することで、ユーザによる入力を介することなく、前記第2のWebアプリケーションに対して、前記入力データを送信する第二の入力データ送信手段と

を有することを特徴とするWebアプリケーション連携システム。

【請求項2】

前記結果データ受信手段により受信した結果データを用いて、前記連携処理実行プログラムにより、当該データが妥当でない旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当でない旨の結果の場合に実行するコードを実行することで、前記第2のWebアプリケーションに対して、前記入力データを送信することなく、所定の画面をWebブラウザに表示させることを特徴とする請求項1に記載のWebアプリケーション連携システム。

20

【請求項3】

前記連携処理実行プログラムは、前記Webアプリケーション連携システムとの通信を非同期通信により行うことを特徴とする請求項1または2に記載のWebアプリケーション連携システム。

【請求項4】

前記連携処理実行プログラムは、前記画面データ内に含まれるスクリプト、またはアプレット形式のプログラムであることを特徴とする請求項1乃至3のいずれか1項に記載のWebアプリケーション連携システム。

30

【請求項5】

クライアント端末を含み、当該クライアント端末から入力された入力データの検証を第1のWebアプリケーションが行って、当該検証の結果得られる結果データに応じて第2のWebアプリケーションへ処理を連携させるWebアプリケーション連携システムにおけるWebアプリケーション連携方法であって、

前記Webアプリケーション連携システムは、

前記クライアント端末からの要求に従って、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて、結果データが妥当である旨の結果の場合に実行するコードと、妥当でない旨の結果である場合に実行するコードとを含む連携処理実行プログラムにより、前記第2のWebアプリケーションへ処理を連携するための操作指示とを可能とする入力画面を当該クライアント端末のWebブラウザに表示可能な画面データであって、当該連携処理実行プログラムを含む画面データを、当該クライアントへ送信する画面データ送信ステップと、

40

前記クライアント端末から送信された前記第1のWebアプリケーションに対する入力データを受信する第一の入力データ受信ステップと、

前記第一の入力データ受信ステップにより受信された入力データに対する前記第1のWebアプリケーションによる検証の結果得られる結果データを、前記クライアント端末へ

50

送信する結果データ送信ステップと、

前記クライアント端末から送信された前記第2のWebアプリケーションに対する入力データを受信する第二の入力データ受信ステップとを有し、

前記クライアント端末は、

前記画面データ送信ステップにより送信された入力画面の画面データを受信する画面データ受信ステップと、

前記画面データ受信ステップにより受信された画面データを用いてWebブラウザに表示された入力画面を介して、ユーザにより、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて前記第2のWebアプリケーションへ処理を連携するための操作指示とがなされた場合、前記連携処理実行プログラムにより、当該入力データを前記第1のWebアプリケーションへ送信する第一の入力データ送信ステップと、

10

前記結果データ送信ステップにより送信された結果データを受信する結果データ受信ステップと、

前記結果データ受信ステップにより受信した結果データを用いて、前記連携処理実行プログラムにより、当該結果データが妥当である旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当である旨の結果の場合に実行するコードを実行することで、ユーザによる入力を介することなく、前記第2のWebアプリケーションに対して、前記入力データを送信する第二の入力データ送信ステップと

を有することを特徴とするWebアプリケーション連携方法。

20

【請求項6】

クライアント端末を含み、当該クライアント端末から入力された入力データの検証を第1のWebアプリケーションが行って、当該検証の結果得られる結果データに応じて第2のWebアプリケーションへ処理を連携させるWebアプリケーション連携システムにおいて実行可能なプログラムであって、

前記Webアプリケーション連携システムを、

前記クライアント端末からの要求に従って、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて、結果データが妥当である旨の結果の場合に実行するコードと、妥当でない旨の結果である場合に実行するコードとを含む連携処理実行プログラムにより、前記第2のWebアプリケーションへ処理を連携するための操作指示とを可能とする入力画面を当該クライアント端末のWebブラウザに表示可能な画面データであって、当該連携処理実行プログラムを含む画面データを、当該クライアントへ送信する画面データ送信手段と、

30

前記クライアント端末から送信された前記第1のWebアプリケーションに対する入力データを受信する第一の入力データ受信手段と、

前記第一の入力データ受信手段により受信された入力データに対する前記第1のWebアプリケーションによる検証の結果得られる結果データを、前記クライアント端末へ送信する結果データ送信手段と、

前記クライアント端末から送信された前記第2のWebアプリケーションに対する入力データを受信する第二の入力データ受信手段として機能させ、

40

前記クライアント端末を、

前記画面データ送信手段により送信された入力画面の画面データを受信する画面データ受信手段と、

前記画面データ受信手段により受信された画面データを用いてWebブラウザに表示された入力画面を介して、ユーザにより、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて前記第2のWebアプリケーションへ処理を連携するための操作指示とがなされた場合、前記連携処理実行プログラムにより、当該入力データを前記第1のWebアプリケーションへ送信する第一の入力データ送信手段と、

前記結果データ送信手段により送信された結果データを受信する結果データ受信手段と

50

前記結果データ受信手段により受信した結果データを用いて、前記連携処理実行プログラムにより、当該結果データが妥当である旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当である旨の結果の場合に実行するコードを実行することで、ユーザによる入力を介することなく、前記第2のWebアプリケーションに対して、前記入力データを送信する第二の入力データ送信手段として機能させることを特徴とするプログラム。

【請求項7】

請求項6に記載のプログラムをコンピュータ読み取り可能に記憶した記録媒体。

【請求項8】

情報処理装置からの要求に従って、入力データの入力と、当該入力データに基づいて第1のWebアプリケーションから得られる結果データに応じて、結果データが妥当である旨の結果の場合に実行するコードと、妥当でない旨の結果である場合に実行するコードとを含む連携処理実行プログラムにより、第2のWebアプリケーションへ処理を連携するための操作指示とを可能とする入力画面を当該情報処理装置のWebブラウザに表示可能な画面データであって、当該連携処理実行プログラムを含む画面データを、当該情報処理装置へ送信する画面データ送信手段と、前記情報処理装置から送信された前記第1のWebアプリケーションに対する入力データを受信する第一の入力データ受信手段と、前記第一の入力データ受信手段により受信された入力データに対する前記第1のWebアプリケーションによる検証の結果得られる結果データを、前記情報処理装置へ送信する結果データ送信手段と、前記情報処理装置から送信された前記第2のWebアプリケーションに対する入力データを受信する第二の入力データ受信手段とを有するWebアプリケーション連携システムと、ネットワークを介して通信可能な情報処理装置であって、

前記画面データ送信手段により送信された入力画面の画面データを受信する画面データ受信手段と、

前記画面データ受信手段により受信された画面データを用いてWebブラウザに表示された入力画面を介して、ユーザにより、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて前記第2のWebアプリケーションへ処理を連携するための操作指示とがなされた場合、前記連携処理実行プログラムにより、当該入力データを前記第1のWebアプリケーションへ送信する第一の入力データ送信手段と、

前記結果データ送信手段により送信された結果データを受信する結果データ受信手段と、前記結果データ受信手段により受信した結果データを用いて、前記連携処理実行プログラムにより、当該結果データが妥当である旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当である旨の結果の場合に実行するコードを実行することで、ユーザによる入力を介することなく、前記第2のWebアプリケーションに対して、前記入力データを送信する第二の入力データ送信手段とを有することを特徴とする情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数のサーバ(Webアプリケーション)とクライアントとが通信可能なシステムにおいて、第1のサーバから第2のサーバに遷移する際の制御に関する。

【背景技術】

【0002】

Webアプリケーション連携に関する技術がある。

【0003】

特許文献1に記載される分散サーバ連携方法は、制御部を別途設け、「管轄下の」Webアプリケーションを連携させる方式を採用している。なお、データのやり取りはHTTPのプロトコルでパラメータを渡すことで行っている。

10

20

30

40

50

## 【0004】

一方、一般的なWebアプリケーション（やWebページ）の連携は、HTMLのリンク機能（アンカ）を使用し、ページ遷移を行うことで実現している。この方法においても、データのやり取りはHTTPのプロトコルでパラメータを渡すことで行っている。

【特許文献1】特開2001-60187号公報

【発明の開示】

【発明が解決しようとする課題】

## 【0005】

ここで、連携するWebアプリケーション間で、やり取りするデータ（＝パラメータ）の妥当性を確認したいというニーズがある。同一Webアプリケーション内でデータをやり取りする場合は、その遷移時にパラメータのチェックを行い、妥当性が確認されれば次のページへ、そうでなければ元のページに戻り、再入力を促すのが一般的である。

10

## 【0006】

しかし、Webアプリケーション間の連携の場合、以下のような問題点がある。これは、上記特許文献1においても、他の一般的な連携方法においても同様である。

## 【0007】

ユーザの連携指示アクションは、連携先Webアプリケーションへの遷移である。よって、ユーザの連携指示アクションを受けると、後は、直ちに連携先のWebアプリケーションに遷移してしまう。しかし、パラメータの妥当性を判定できるのが連携元のWebアプリケーション（ないしそこが管理しているデータ）である場合もあり、その場合の妥当性の確認は連携元のWebアプリケーションで行わなければならない。

20

## 【0008】

つまり、データのフォーマットチェック等簡単な妥当性は、連携先ないし連携元のクライアントサイドのスクリプトでも検証できるが、データのアプリケーション的な意味においての妥当性は、そのデータの管轄アプリケーションである連携元のサーバでしか妥当性を判断できない場合がある、ということである。

## 【0009】

例えば、商品管理アプリケーションが帳票出力アプリケーションにデータを渡す場合、連携元の商品管理アプリケーションにおいて、入力された商品の在庫数を確認してから帳票出力したい場合などがある。

30

## 【0010】

この対策として、ユーザの連携指示アクション時にクライアントサイドのスクリプト言語を使用してローカルのみで入力値の妥当性確認を行うことが考えられる。しかしそれでは、入力値そのもののチェックはよいが、入力値の計算（合計等）や関連（大小関係）などを加味した複雑なチェックは難しい。また、それらの計算処理や、妥当性確認の処理などサーバ側のロジックを利用できない（クライアントとサーバで二重に定義することになる）。さらに、データベースに格納されている値と照合したチェックは非常に困難である。

## 【0011】

よって現状は、妥当性確認をある程度見切りとして諦めて安全性を落とすか、遷移元のWebアプリケーションに妥当性確認用の画面を用意し、一旦その画面による妥当性の確認をユーザに指示させて妥当性のチェックを行った後、改めてユーザに連携指示を行わせ、連携先Webアプリケーションに遷移するといった方法が取られている。しかし、このような指示操作をユーザに強いることは、ユーザの円滑な操作を阻害する要因になっている。

40

## 【0012】

本発明は、上記の問題点を解決するためになされたもので、本発明の目的は、ユーザに意識させることなく、第一のWebアプリケーションで入力データの妥当性の確認を行い、入力データが妥当である場合、第二のWebアプリケーションへ処理を連携させることができる仕組みを提供することである。

50

## 【課題を解決するための手段】

## 【0013】

本発明は、クライアント端末を含み、当該クライアント端末から入力された入力データの検証を第1のWebアプリケーションが行って、当該検証の結果得られる結果データに応じて第2のWebアプリケーションへ処理を連携させるWebアプリケーション連携システムであって、前記Webアプリケーション連携システムは、前記クライアント端末からの要求に従って、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて、結果データが妥当である旨の結果の場合に実行するコードと、妥当でない旨の結果である場合に実行するコードとを含む連携処理実行プログラムにより、前記第2のWebアプリケーションへ処理を連携するための操作指示とを可能とする入力画面を当該クライアント端末のWebブラウザに表示可能な画面データであって、当該連携処理実行プログラムを含む画面データを、当該クライアントへ送信する画面データ送信手段と、前記クライアント端末から送信された前記第1のWebアプリケーションに対する入力データを受信する第一の入力データ受信手段と、前記第一の入力データ受信手段により受信された入力データに対する前記第1のWebアプリケーションによる検証の結果得られる結果データを、前記クライアント端末へ送信する結果データ送信手段と、前記クライアント端末から送信された前記第2のWebアプリケーションに対する入力データを受信する第二の入力データ受信手段とを有し、前記クライアント端末は、前記画面データ送信手段により送信された入力画面の画面データを受信する画面データ受信手段と、前記画面データ受信手段により受信された画面データを用いてWebブラウザに表示された入力画面を介して、ユーザにより、入力データの入力と、当該入力データに基づいて前記第1のWebアプリケーションから得られる結果データに応じて前記第2のWebアプリケーションへ処理を連携するための操作指示とがなされた場合、前記連携処理実行プログラムにより、当該入力データを前記第1のWebアプリケーションへ送信する第一の入力データ送信手段と、前記結果データ送信手段により送信された結果データを受信する結果データ受信手段と、前記結果データ受信手段により受信した結果データを用いて、前記連携処理実行プログラムにより、当該結果データが妥当である旨の判定がされた場合、当該連携処理実行プログラムは、結果データが妥当である旨の結果の場合に実行するコードを実行することで、ユーザによる入力を介することなく、前記第2のWebアプリケーションに対して、前記入力データを送信する第二の入力データ送信手段とを有することを特徴とする。

10

20

30

## 【発明の効果】

## 【0014】

本発明によれば、ユーザが円滑にかつ安全に複数サーバの機能を連携して使用することができる。

## 【発明を実施するための最良の形態】

## 【0015】

図1は、本発明のサーバ連携システムの前提となるネットワークシステムの構成の一例を示す図である。

## 【0016】

図1において、10はインターネット等のネットワークである。このネットワーク10には、複数のWebサーバ1(1a, ..., 1b)や複数のクライアント2(2a, ..., 2b)が接続されている。

40

## 【0017】

本発明のWebアプリケーション連携システムでは、1又は複数のWebサーバ1上で動作する複数のWebアプリケーションを連携するものである。

## 【0018】

次に、図2を参照して、図1に示したWebサーバ1、クライアント2を実現可能な情報処理装置のハードウェア構成について説明する。

## 【0019】

50

図2は、図1に示したWebサーバ1、クライアント2を実現可能な情報処理装置のハードウェア構成の一例を示すブロック図である。

【0020】

Webサーバ1、クライアント2を実現可能な情報処理装置100は、CPU101、RAM102、ROM103、外部メモリ104、入力装置105、表示装置106、ネットワークインタフェースカード(NIC)107等を備え、これらがシステムバス108を介して互いに接続されている構成となっている。なお、実際には101~107とシステムバス108との間にはインタフェース回路やチップセット等が存在するが、図では省略されている。

【0021】

CPU101は、ROM103又は外部メモリ104等に記憶保存されたプログラムを読み出して実行することにより、各種制御処理を実行する。RAM102は、CPU101の作業領域として使用される。

【0022】

NIC107は、図1に示したネットワーク10等に有線又は無線で接続され、ネットワークを介して他の装置と通信可能である。

【0023】

外部メモリ104は、ハードディスク、フレキシブルディスク、CD-ROM、DVD、フラッシュメモリ等で構成される。

【0024】

次に、図3、図4を参照して、本発明の特徴を従来技術と比較して説明する。

【0025】

図3は、従来技術における具体的なアプリケーション連携の実行例を示した模式図である。

【0026】

図4は、本発明を実施した場合の具体的なアプリケーション連携の実行例を示した模式図である。

【0027】

図3において、201は、連携元のWebアプリケーション(この例では、商品管理アプリケーション)210からクライアントに送信されて、クライアント上で表示される入力画面である。

【0028】

図3に示す従来技術における実行例では、遷移直前に透過的に連携元で入力画面201の入力データのチェックを行うことができないため、ユーザは「確認画面へ」ボタン202をクリックして、妥当性検証だけの為の確認画面203へ遷移させる必要がある。ユーザが「確認画面へ」ボタン202をクリックすると、入力データ(ここでは"ABC123")が連携元のWebアプリケーション210に送信される。そして、このWebアプリケーション210で妥当性の確認が行われ、確認結果がクライアント側に返送され、クライアントは確認画面203へ画面遷移する。

【0029】

その後、ユーザは「発注書印刷」ボタン204をクリックして、連携先のWebアプリケーション(ここでは、帳票印刷アプリケーション)220への遷移をクライアント側から指示する必要がある。ユーザが「発注書印刷」ボタン204をクリックすると、入力画面201で入力したデータ(ここでは"ABC123")が連携先のWebアプリケーション202に送信される。そして、この連携先Webアプリケーション220から処理結果がクライアント側に返送され、クライアントは画面205へ画面遷移する。

【0030】

これに対して、図4に示す本発明を実施した場合の実行例では透過的に連携先に遷移することができるため、図3に示した従来の技術のようにユーザが確認画面へ画面遷移をさせる必要がない。ユーザは単に連携先のWebアプリケーション(帳票印刷アプリケーシ

10

20

30

40

50

ョン)への遷移をクライアント側から指示するだけでよい。以下、具体的に示す。

【0031】

図4において、301は、連携元のWebアプリケーション(この例では、商品管理アプリケーション)310からクライアントに送信されて(1)、クライアント上で表示される入力画面である。

【0032】

ユーザ「発注書印刷」ボタン302をクリックすると(2)、入力データ(ここでは"ABC123")が連携元のWebアプリケーション310に非同期通信(図中破線で示す)により送信される(3)。そして、このWebアプリケーション310で妥当性の確認が行われ、確認結果がクライアント側に非同期通信により返送される(4)。

10

【0033】

この確認結果を受信したクライアントは、上記確認結果を判断してOKの場合には、自動で入力データを連携先のWebアプリケーション(ここでは、帳票印刷アプリケーション)320へ送信する。そして、この連携先Webアプリケーション320から処理結果がクライアント側に返送され、クライアントは画面303へ画面遷移する。

【0034】

なお、クライアントは、確認結果がNG(エラー)の場合には、連携先のアプリケーションに遷移することなく、304に示すようにエラーである旨を表示して、再入力を促す。

【0035】

図5は、本発明のWebアプリケーション連携システムの構成及び動作の流れを示した図であり、図1と同一のものには同一の符号を付してある。また、図5に示す(1)~(6)は図4に示した(1)~(6)に対応するものとする。

20

【0036】

本発明のWebアプリケーション連携システムは構成として、連携元Webアプリケーション401と、それがWebブラウザ400上に表示する遷移元画面403があり、この遷移元画面403内にクライアントサイドスクリプト405が搭載される。

【0037】

また、連携先Webアプリケーション402と遷移先画面404がある。特に、連携先Webアプリケーション402は、連携元Webアプリケーション401と異なるWebサーバ上で起動していることを想定しており、束ねる管理アプリケーションもない疎結合な状態である。即ち、連携元Webアプリケーション401は、連携先Webアプリケーション402のアドレスと必要なパラメータのみを知っていればよい。

30

【0038】

なお、連携先Webアプリケーション402と連携元Webアプリケーション401とが同じWebサーバ上で起動している場合にも適用可能なこというまでもない。

【0039】

本実施形態では、連携元Webアプリケーション401は、図1に示したWebサーバ1aのCPUが外部メモリに格納されるプログラムをRAM上に読み出して実行することにより実現されるものとする。

40

【0040】

また、連携元Webアプリケーション401は、図1に示したWebサーバ1aのCPUが外部メモリに格納されるプログラムをRAM上に読み出して実行することにより実現されるものとする。

【0041】

さらに、Webブラウザは、図1に示したクライアント2aのCPUが外部メモリに格納されるプログラムをRAM上に読み出して実行することにより実現されるものとする。

【0042】

次に、動作の流れを時系列で説明する。

【0043】

50



まず、連携元Webアプリケーション401から遷移元画面403の画面データがWebブラウザ400に送信されると、これを受信したWebブラウザ400が遷移元画面403を表示する(1)。

【0044】

遷移元画面403は、連携先Webアプリケーション402を起動するリンク又はボタン(例えば図4の302)を備えている。

【0045】

ユーザは、連携先Webアプリケーション402に必要な事項を、遷移元画面403上でキーボード等の入力装置を用いて入力し、連携先Webアプリケーション402を起動するリンクをマウス等の入力装置を用いてクリックする(2)。

10

【0046】

すると、Webブラウザ400は、遷移元画面403内に記述されているクライアントサイドスクリプト405が起動する。そして、クライアントサイドスクリプト405は、連携元Webアプリケーション401に遷移元画面403で入力された値(入力データ)を送付し、妥当性検証要求を行う(3)。なお、この要求は非同期通信で行われる為、Webブラウザ上で画面の遷移は行われない。

【0047】

連携元Webアプリケーション401は、この妥当性検証要求を受け取り、値の妥当性検証を行い、妥当性検証結果をクライアントサイドスクリプト405に送付する(4)。

【0048】

20

クライアントサイドスクリプト405は、この妥当性検証結果を受け取り判断する。そして、クライアントサイドスクリプト405は、遷移元画面403を表示するWebブラウザ400は、値の妥当性が確認された場合(検証結果がOKの場合)、その値をパラメータとして連携先Webアプリケーション402に送信し、遷移要求する(5)。

【0049】

連携先Webアプリケーション402は、このパラメータを受け取り、該受け取ったパラメータに沿ってサービスを行い、遷移先画面404の画面データをWebブラウザ400に返信して、Webブラウザ400に遷移先画面404を表示させる(6)。

【0050】

図6は、本発明における制御処理動作の一例を示すフローチャートであり、図5に示した(3)~(6)の処理を詳細に示したものに対応する。なお、図中、S501, S502, S505, S506, S508, S510は、図5に示したWebブラウザ400上に表示される遷移元画面403内のクライアントサイドスクリプト405が実行するステップに対応する。また、S503, S504は、図5に示した連携元Webアプリケーション401が実行するステップに対応する。さらに、S509は、図5に示した連携先Webアプリケーション402が実行するステップに対応する。

30

【0051】

ユーザからの指示により遷移要求(連携先Webアプリケーション402への連携指示)を受けたWebブラウザ400は、遷移元画面403内のクライアントサイドスクリプト405を起動する。クライアントサイドスクリプト405は、ユーザが遷移元画面403内で入力した画面項目値(入力データ)を収集し、サーバに送付する形式に編集する(S501)。

40

【0052】

続いて、クライアントサイドスクリプト405は、連携元Webアプリケーション401に妥当性検証要求を送信する(S502)。このとき、クライアントサイドスクリプト405は、S501で収集・編集した画面項目値をパラメータとして併せて連携元Webアプリケーション401に送付する。なお、この要求は、非同期通信を用いて行う。

【0053】

そして、この要求を受理した連携元Webアプリケーション401は、送付された画面項目値を受け取り、これを元に必要な「計算」やデータの「取得」等の処理を行う(S5

50

03)。これは、画面入力項目より導出される項目（例えば、商品在庫等）がチェック対象となる場合があるため、且つ、連携先Webアプリケーション402に渡すパラメータとして使用される場合があるためである。例えば「ある商品の発注数の合計が、在庫数を超えていないか」という検証があった場合を想定する。まず、その商品の合計を「計算」する。そして、ある商品のID等をキーにデータベースよりその商品の在庫数データを「取得」する。

【0054】

そして、連携元Webアプリケーション401は、必要なデータが揃うと、そのデータを使用して妥当性の検証を行う（S504）。

【0055】

妥当性検証を終えると、連携元Webアプリケーション401は、その妥当性検証結果を遷移元画面403（クライアントサイドスクリプト405）に返送する（S505）。その際、必要に応じてS503で導出したデータ（計算結果）も併せて送付する。これにより、クライアントサイドスクリプト405は、ユーザに現状把握や次の処理の補助となる有益な情報を遷移元画面403上に表示することが可能となる。なお、上記連携元Webアプリケーション401から遷移元画面403に送信するデータの例を後述する図7に示してある。

【0056】

クライアントサイドスクリプト405は、妥当性検証結果を受け取ると、まず、必要に応じて受信したデータ（S503で導出されたデータ）を遷移元画面403上に設定する（S506）。

【0057】

次に、クライアントサイドスクリプト405は、連携元Webアプリケーション401から送付された情報のうち、妥当性検証結果をエラー判定する（妥当性検証結果がOKかNGか判定する）（S507）。

【0058】

そして、ステップS507において、エラー判定結果がOKだった場合（即ち、遷移元画面403の入力データを連携先Webアプリケーション402へ送信するのに妥当であると判断した場合）は、クライアントサイドスクリプト405は、遷移元画面403の画面項目から必要な値を取得し、パラメータとして編集した後、それを以って連携先Webアプリケーション402に遷移要求を行う（S508）。

【0059】

そして、連携先Webアプリケーション402は、クライアントサイドスクリプト405からの遷移要求（パラメータ）を受け取ると、このパラメータを用いて所定のサービスを行う。なお、このパラメータは、S504において連携元Webアプリケーション401によって検証済な妥当な値であることが保証されているため、連携先Webアプリケーション402ではパラメータエラーとなることなくサービスを実行可能となる。

【0060】

そして、上記所定のサービスが終了すると、連携先Webアプリケーション402は、遷移先画面404の画面データを上記遷移要求の送信元であるWebブラウザ400に送信する。この画面データを受信したWebブラウザ400は、この画面データに基づいて遷移先画面404を表示する（S509）。

【0061】

一方、ステップS507において、エラー判定結果がNGだった場合（即ち、遷移元画面403の入力データを連携先Webアプリケーション402へ送信するのに妥当でないと判断した場合）は、クライアントサイドスクリプト405は、遷移要求は行わず、遷移元画面403上にエラーの表示を行い、ユーザに再入力を促す（S510）。

【0062】

そして、ユーザは、上記ステップS505で表示された画面項目情報を参考にしながら、必要な部分のみ修正して、再度、連携先Webアプリケーション402への連携指示を

10

20

30

40

50

行うことができる。これにより、Webブラウザ400は、遷移元画面403内のクライアントサイドスクリプト405を起動し、再度、ステップS501から処理を行う。

【0063】

以下、図7のデータ例、図8の擬似コードを用いて、クライアントサイドスクリプト405の処理について説明する。

【0064】

図7は、図6のステップS505に示した妥当性検証終了の際、連携元Webアプリケーション401から遷移元画面403（クライアントサイドスクリプト405）に渡しているデータ例を示す図である。

【0065】

図7に示すように、図6のステップS505において、連携元Webアプリケーション401から遷移元画面403（クライアントサイドスクリプト405）に送信されるデータは、妥当性検証結果として、妥当性検証結果601（この例では"NG"）、妥当性検証の結果601が"NG"となった原因箇所602、エラーメッセージ603等を含む。なお、妥当性検証結果が"OK"の場合は、原因箇所602、エラーメッセージ603は含まれない。

【0066】

また、上記連携元Webアプリケーション401から遷移元画面403（クライアントサイドスクリプト405）に送信されるデータは、計算結果604も含んでいる。

【0067】

図8は、図6のステップS506、S507（S508、S510）の処理に対応するクライアントサイドスクリプト405の擬似コードの一例を示す図である。

【0068】

図8に示すように、まず、コード701が実行され、妥当性検証の結果如何にかかわらず、計算の結果（図7の604）を画面項目に表示する。

【0069】

次に、コード702が実行され、妥当性検証の結果（図7の601）を見て、それにより処理を分岐する。妥当性検証の結果が"OK"だった場合、コード703を実行して、必要なパラメータを画面より収集・編集して、そのパラメータを渡しながら連携先アプリケーションを起動する。

【0070】

一方、妥当性検証の結果が"NG"だった場合、コード707を実行して、妥当性検証の結果と同送されたエラーメッセージ（図7の603）を画面の所定の位置に表示し、かつ原因となった画面項目（図7の602）にエラー表示（背景を赤くするなど）を行う。

【0071】

以上説明したように、本実施形態のWebアプリケーション連携システムは、連携元Webアプリケーション401が、前記クライアント上でデータ入力と該入力データの連携先Webアプリケーション402への送信指示（連携指示）を可能とする入力画面（遷移元画面403）を表示させるための画面データと、前記クライアント上で実行されるプログラム（クライアントサイドスクリプト405）とを前記クライアントに送信する機能と、前記プログラムにより前記クライアントから返信されたデータを受信し、該受信データをデータ処理し、該データ処理結果を前記クライアントに返信する機能とを有するものであり、また、前記プログラム（クライアントサイドスクリプト405）は、前記入力画面からの入力データを前記連携先サーバへ送信させる指示がなされると、前記入力データを前記連携元Webアプリケーション401に送信して、さらに該連携元Webアプリケーション401から返信されるデータ処理結果を受信して、該受信したデータ処理結果に基づいて前記入力データを前記連携先Webアプリケーション402へ送信するべきか判断し、前記入力データを前記連携先Webアプリケーション402へ送信するべきであると判断した場合には、前記入力データを前記連携先Webアプリケーション402へ送信する機能をクライアントに実現させるためのプログラムである、構成を特徴とする。

10

20

30

40

50

## 【 0 0 7 2 】

即ち、本実施形態のWebアプリケーション連携は、一般的なWebアプリケーション（やWebページ）の連携に、機能を付加することで実現する。連携元Webアプリケーション401の表示した、遷移元画面のHTMLのリンク機能（アンカ）を押下された際、直接連携先Webアプリケーションに遷移せず、まずクライアントサイドスクリプト405の処理を起動する。その処理の中で、連携元Webアプリケーションに非同期通信で妥当性確認要求を行う。非同期通信を使用するため、画面遷移はない。そして、連携元Webアプリケーション401でチェックを行い、遷移元画面403で妥当性確認の結果を受け取り、妥当性が確認されれば、連携先Webアプリケーション402にパラメータを渡して遷移し、妥当でなければそのまま遷移元画面403にその旨エラーメッセージを表示、再入力を促すように構成する。

10

## 【 0 0 7 3 】

このような構成により、ユーザが円滑にかつ安全に複数Webアプリケーションの機能を連携して使用することができる。

## 【 0 0 7 4 】

また、一般的な疎結合なWebアプリケーションの連携時であっても、余分な遷移をすることなく、妥当性確認後のパラメータを連携先Webアプリケーションに渡すことができる。

## 【 0 0 7 5 】

さらに、連携先Webアプリケーションを選ばない。連携元Webアプリケーションは、連携先のURL（ロケーション情報）と必要なパラメータを知っていれば良いだけで、疎結合で柔軟な連携が可能である。

20

## 【 0 0 7 6 】

また、連携元Webアプリケーションで、妥当性確認を行う為、クライアント側のスクリプト等で妥当確認を行う場合や、連携先Webアプリケーションにおいて妥当性確認を行う場合等より、複雑できめ細かく妥当性を確認できる。

## 【 0 0 7 7 】

さらに、パラメータにエラーがあった場合も、そのエラーを連携元Webアプリケーションで検知できるため、より自然に再入力を促すことができる。もし連携先Webアプリケーションでパラメータエラーが発見された場合は、ユーザ操作により画面を戻して、再入力を行わなければならないが、このような事態を未然に防止することができる。

30

## 【 0 0 7 8 】

また、パラメータの妥当性結果の判断及び、該判断結果に基づく処理分岐をクライアント側で行っている為、より柔軟なサービスを行うことができる。例えば、受け取ったデータ見て判断し、内容や量によって、より情報を見易いように新規ウィンドウを起動して情報を表示し、該表示される情報を見ながらユーザが入力できるようになる等の柔軟なサービスが考えられる。

## 【 0 0 7 9 】

このように、ユーザに意識させることなく、連携先Webアプリケーションに遷移する前に、連携元Webアプリケーションで遷移元画面での入力値の妥当性を確認する仕組みを提供することができる。

40

## 【 0 0 8 0 】

なお、図7に示したデータや図8に示した擬似コードは、その内容はこれに限定されるものではなく、用途や目的に応じて、様々な構成や内容で構成されることは言うまでもない。

## 【 0 0 8 1 】

以上、一実施形態について示したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記録媒体等としての実施態様をとることが可能であり、具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

50

## 【 0 0 8 2 】

以下、図 9 に示すメモリマップを参照して本発明に係る Web アプリケーション連携システムを構成する Web サーバ 1 , クライアント 2 で読み取り可能なデータ処理プログラムの構成について説明する。

## 【 0 0 8 3 】

図 9 は、本発明に係る Web アプリケーション連携システムを構成する Web サーバ 1 , クライアント 2 で読み取り (読み出し) 可能な各種データ処理プログラムを格納する記録媒体 (記憶媒体) のメモリマップを説明する図である。

## 【 0 0 8 4 】

なお、特に図示しないが、記録媒体に記憶されるプログラム群を管理する情報、例えばバージョン情報、作成者等も記憶され、かつ、プログラム読み出し側の OS 等に依存する情報、例えばプログラムを識別表示するアイコン等も記憶される場合もある。

## 【 0 0 8 5 】

さらに、各種プログラムに従属するデータも上記ディレクトリに管理されている。また、インストールするプログラムやデータが圧縮されている場合に、解凍するプログラム等も記憶される場合もある。

## 【 0 0 8 6 】

本実施形態における図 5 , 図 6 , 図 8 に示す機能が外部からインストールされるプログラムによって、ホストコンピュータにより遂行されていてもよい。そして、その場合、CD-ROM やフラッシュメモリや FD 等の記録媒体により、あるいはネットワークを介して外部の記録媒体から、プログラムを含む情報群を出力装置に供給される場合でも本発明は適用されるものである。

## 【 0 0 8 7 】

以上のように、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ (または CPU や MPU) が記録媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成されることは言うまでもない。

## 【 0 0 8 8 】

この場合、記録媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記録媒体は本発明を構成することになる。

## 【 0 0 8 9 】

プログラムコードを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、DVD-ROM、磁気テープ、不揮発性のメモリカード、ROM、EEPROM、シリコンディスク等を用いることができる。

## 【 0 0 9 0 】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働している OS (オペレーティングシステム) 等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

## 【 0 0 9 1 】

さらに、記録媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わる CPU 等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

## 【 0 0 9 2 】

また、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器からな

10

20

30

40

50

る装置に適用してもよい。また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適応できることは言うまでもない。この場合、本発明を達成するためのソフトウェアによって表されるプログラムを格納した記録媒体を該システムあるいは装置に読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。

【0093】

さらに、本発明を達成するためのソフトウェアによって表されるプログラムをネットワーク上のサーバ、データベース等から通信プログラムによりダウンロードして読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。

【0094】

なお、上記実施形態では、遷移元画面403（クライアントサイドスクリプト405を含む）が、連携元Webアプリケーション401から送信されて、Webブラウザ400上に表示される構成について説明した。しかし、これは本発明における遷移元画面403の取得先を限定するものではない。例えば、他のサーバプログラムから遷移元画面403が送信されてWebブラウザ400上に表示される構成であってもよいし、HD、FD、CD-ROM等の記録媒体に遷移元画面403を格納しておき、該記録媒体から遷移元画面403が読み出されてWebブラウザ400上に表示される構成も本発明に含まれるものである。

【0095】

また、上記実施形態では、例えばHTMLにより記述される遷移元画面403内に含まれるクライアントサイドスクリプト405により図6のS501、S502、S506、S507（S508、S510）が実行される構成について説明した。しかし、クライアントで表示される遷移元画面403やクライアントで実行されるプログラムの形式はこれに限定されるものではない。例えば、アプレット形式のプログラムをサーバからクライアントにダウンロードし、該プログラムを実行することにより、遷移元画面403の表示や図6のS501、S502、S506、S507（S508、S510）に示した機能を実現するように構成してもよい。また、遷移元画面403の表示や図6のS501、S502、S506、S507（S508、S510）に示した機能を実現するプログラムをクライアントに予めインストールしておく構成であってもよい。

【0096】

なお、上述した実施形態およびその変形例を組み合わせた構成も全て本発明に含まれるものである。

【図面の簡単な説明】

【0097】

【図1】本発明のサーバ連携システムの前提となるネットワークシステムの構成の一例を示す図である。

【図2】図1に示したWebサーバ1、クライアント2を実現可能な情報処理装置のハードウェア構成の一例を示すブロック図である。

【図3】従来技術における具体的なアプリケーション連携の実行例を示した模式図である。

【図4】本発明を実施した場合の具体的なアプリケーション連携の実行例を示した模式図である。

【図5】本発明のWebアプリケーション連携システムの構成及び動作の流れを示した図である。

【図6】本発明における制御処理動作の一例を示すフローチャートである。

【図7】図6のステップS505に示した妥当性検証終了の際、連携元Webアプリケーション401から遷移元画面403（クライアントサイドスクリプト405）に渡しているデータ例を示す図である。

【図8】図6のステップS506、S507（S508、S510）の処理に対応するクライアントサイドスクリプト405の擬似コードの一例を示す図である。

10

20

30

40

50

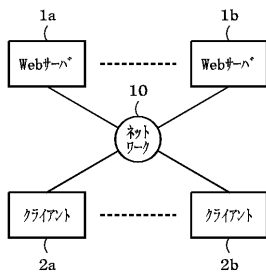
【図9】本発明に係るWebアプリケーション連携システムを構成するWebサーバ1，クライアント2で読み取り（読み出し）可能な各種データ処理プログラムを格納する記録媒体（記憶媒体）のメモリマップを説明する図である。

【符号の説明】

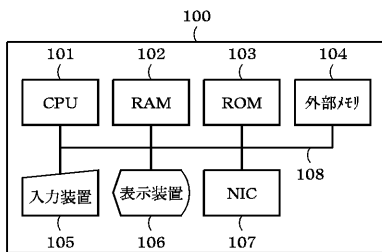
【0098】

- 1 a Webサーバ
- 1 b Webサーバ
- 1 0 ネットワーク
- 2 a クライアント
- 2 b クライアント
- 4 0 0 Webブラウザ
- 4 0 1 連携元Webアプリケーション
- 4 0 2 連携先Webアプリケーション
- 4 0 3 連携元画面
- 4 0 4 連携先画面
- 4 0 5 クライアントサイドスクリプト

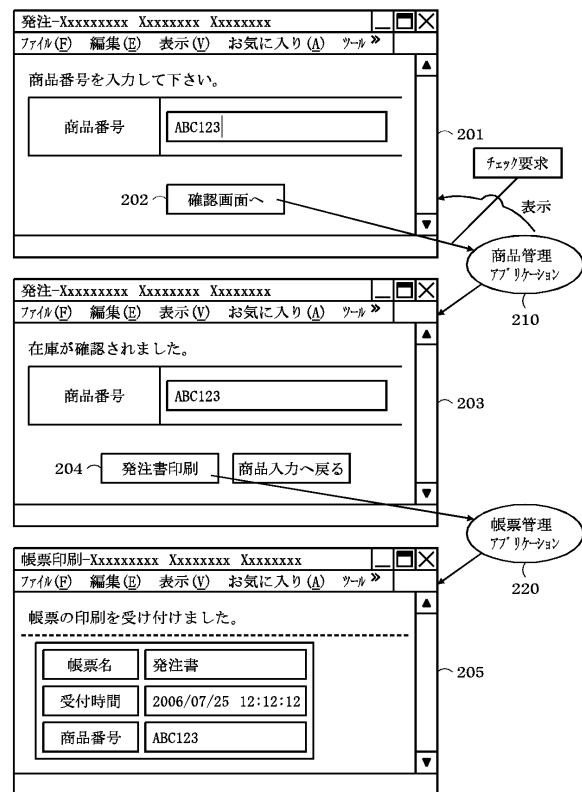
【図1】



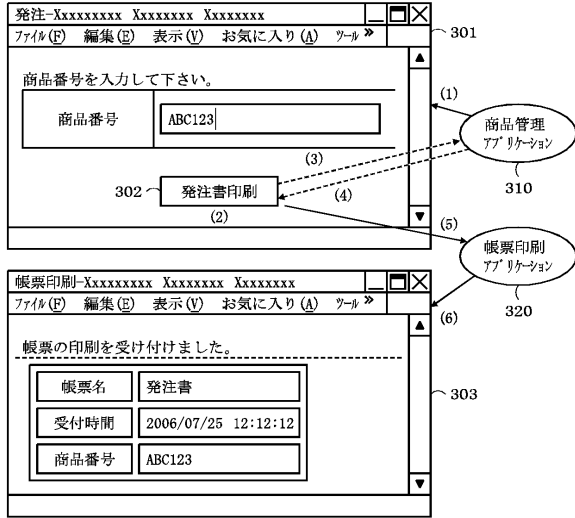
【図2】



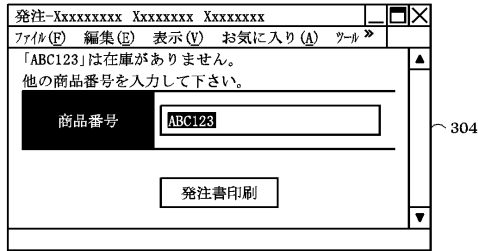
【図3】



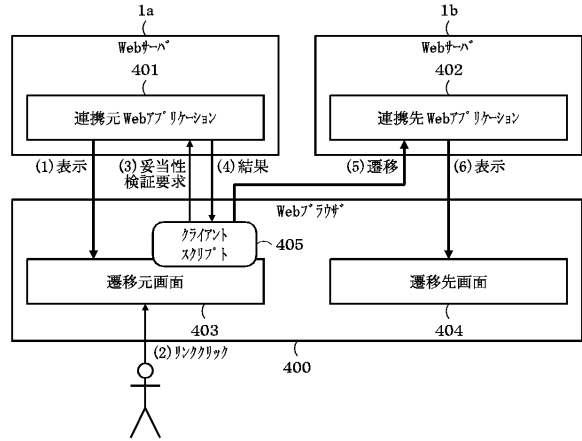
【図4】



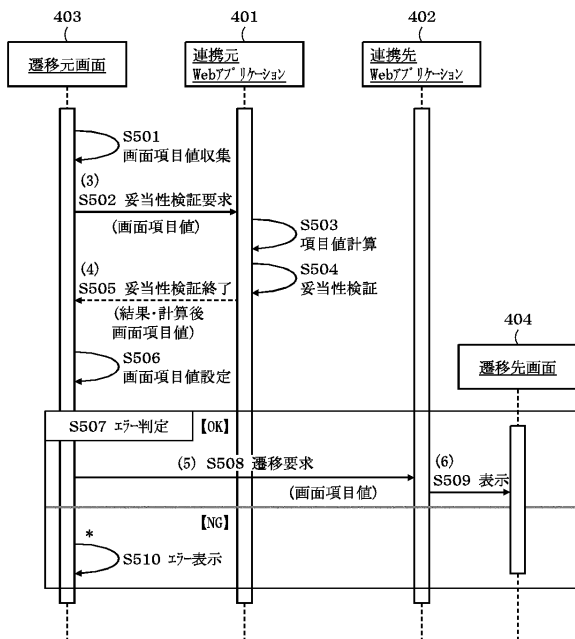
(補足)チェックの結果エラーだった場合は、連携元のアプリケーションでその旨表示し、再入力・再実行を行うことができる。



【図5】



【図6】



【図8】

```

// 画面項目値の設定 (S506)
ps= getElementById ("price_sum")
ps.value = data.items.price_sum;

if ( data.validation.result == "OK" ) { // エラーの判定 (S507)
  // 遷移要求 (S508)
  param = "?*+id="+getElementById ("product_id").value;
  window.open ("http://another_app"+param);
} else {
  // エラー表示 (S510)
  getElementById ("error_msg").value = data.validation.message;
  getElementById (data.validation.wrong).class = "ERROR";
}
}

```

【図7】

```

data={
  validation={
    result: "NG", // 601
    wrong: "price_sum", // 602
    message: "合計が所定の金額を超えています。", // 603
  },
  items={
    price_sum: "¥13,000" // 604
  }
}

```



【 図 9 】

FD/CD-ROM等の記憶媒体

ディレクトリ情報
第1の処理プログラム 図5の(1), 図6に示すフローチャートのS503,S504,S505 (図5の(4))に対応するプログラムコード群
遷移元画面 (クライアントスクリプトを含む) 図5の遷移元画面403本体,クライアントスクリプト405 (図6 に示すフローチャートのS501,S502,S506,S507 (S508, S510)に対応するプログラムコード群により構成される)

記憶媒体のメモリマップ

---

フロントページの続き

(58)調査した分野(Int.Cl. , DB名)

G 0 6 F 1 5 / 0 0

G 0 6 F 9 / 5 0