

(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2019-32883

(P2019-32883A)

(43) 公開日 平成31年2月28日(2019.2.28)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 13/00 (2006.01)</b>	G06F 13/00	5B084
<b>G06F 9/445 (2018.01)</b>	G06F 9/445	5B376

審査請求 有 請求項の数 11 O L (全 18 頁)

(21) 出願番号 特願2018-193268 (P2018-193268)  
 (22) 出願日 平成30年10月12日 (2018.10.12)  
 (62) 分割の表示 特願2014-264833 (P2014-264833) の分割  
 原出願日 平成26年12月26日 (2014.12.26)  
 (31) 優先権主張番号 201410075211.1  
 (32) 優先日 平成26年3月3日 (2014.3.3)  
 (33) 優先権主張国 中国 (CN)

(71) 出願人 512015127  
 バイドゥ オンライン ネットワーク テクノロジー (ペキン) カンパニー リミテッド  
 中華人民共和国 ペキン ハイディエン ディストリクト シャンディー テンス ストリート ナンバー 10 バイドゥ キャンパス 3エフ  
 (74) 代理人 100073184  
 弁理士 柳田 征史  
 (72) 発明者 ワン ジーグウ  
 中華人民共和国 ペキン ハイディエン ディストリクト シャンディー テンス ストリート ナンバー 10 バイドゥ キャンパス 3エフ

最終頁に続く

(54) 【発明の名称】 ウェブアプリケーションを実施する方法、装置、サーバ、及びシステム

(57) 【要約】 (修正有)

【課題】ウェブアプリケーションを実施する方法、装置、サーバ、及びシステムを提供する。

【解決手段】ウェブアプリケーションを実施する際に、現在のウェブアプリケーションの動作環境を検出し、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードする。

【効果】ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

【選択図】 図 1 a

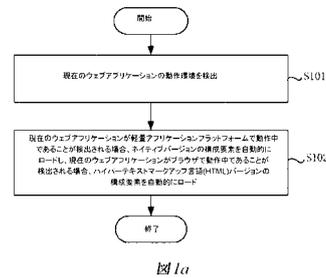


図 1a

**【特許請求の範囲】****【請求項 1】**

ブラウザと、ブラウザではない軽量アプリケーションプラットフォームとの、いずれを介しても動作することができるウェブアプリケーションを、実施する方法であって、

現在のウェブアプリケーションの動作環境を検出するステップと、

前記現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードするステップと、

前記現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語 (HTML) バージョンの構成要素を自動的にロードするステップと、

を含み、

前記ネイティブバージョンの構成要素を自動的にロードするステップが、

前記現在のウェブアプリケーションが必要とするネイティブバージョンの構成要素の中に、前のウェブアプリケーションで既に出現しローカルサイトに存在する特定の構成要素があるか否かを判断し、前記ローカルサイトに存在するネイティブバージョンの前記特定の構成要素を該ローカルサイトから自動的にロードし、前記ローカルサイトに存在しない他のネイティブバージョンの構成要素をサーバから自動的にロードするステップ、

を含み、

前記ローカルサイトから自動的にロードされたネイティブバージョンの前記特定の構成要素を用いて、軽量アプリケーションプラットフォームで動作中の前記現在のウェブアプリケーションの機能の一部を実施する、

ことを特徴とする、方法。

**【請求項 2】**

前記 HTML バージョンの構成要素を自動的にロードするステップが、

HTML バージョンの前記構成要素が前記ローカルサイトに存在するか否かを判断し、存在する場合、前記ローカルサイトから HTML バージョンの前記構成要素を自動的にロードし、存在しない場合、サーバから HTML バージョンの前記構成要素を自動的にロードするステップ、

を含むことを特徴とする、請求項 1 に記載の方法。

**【請求項 3】**

前記サーバからネイティブバージョンの前記構成要素又は HTML バージョンの前記構成要素をロードするステップに先立ち、前記方法が、

前記サーバが、ユーザによって提出されたネイティブバージョンの前記構成要素及び HTML バージョンの前記構成要素を受信して保存するステップ

を更に含むことを特徴とする、請求項 2 に記載の方法。

**【請求項 4】**

ネイティブバージョンの前記構成要素が、HTML 拡張タグを介してパラメータ及び属性を定義することを特徴とする、請求項 1 から 3 いずれか 1 項に記載の方法。

**【請求項 5】**

前記ローカルサイト又はサーバからネイティブバージョンの前記構成要素を自動的にロードするステップが、

前記ローカルサイト又はサーバからスクリプトを介して、ネイティブバージョンの前記構成要素を自動的にロードするステップ

を含むことを特徴とする、請求項 4 に記載の方法。

**【請求項 6】**

ブラウザと、ブラウザではない軽量アプリケーションプラットフォームとの、いずれを介しても動作することができるウェブアプリケーションを、実施する装置であって、

現在のウェブアプリケーションの動作環境を検出する検出モジュールと、

前記検出モジュールにより、前記現在のウェブアプリケーションが軽量アプリケーション

10

20

30

40

50

ンプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、前記検出モジュールにより、前記現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素を自動的にロードするロードモジュールと、  
を備え、

前記ロードモジュールが特に、

前記現在のウェブアプリケーションが必要とするネイティブバージョンの構成要素の中に、前のウェブアプリケーションで既に出現しローカルサイトに存在する特定の構成要素があるか否かを判断し、前記ローカルサイトに存在するネイティブバージョンの前記特定の構成要素を該ローカルサイトから自動的にロードし、前記ローカルサイトに存在しない  
10 10

他のネイティブバージョンの構成要素をサーバから自動的にロードすることに使用され、  
当該装置は、前記ローカルサイトから自動的にロードされたネイティブバージョンの前記特定の構成要素を用いて、軽量アプリケーションプラットフォームで動作中の前記現在のウェブアプリケーションの機能の一部を実施する、  
ことを特徴とする、装置。

【請求項 7】

前記ロードモジュールが特に、

HTMLバージョンの前記構成要素が前記ローカルサイトに存在するか否かを判断し、存在する場合、前記ローカルサイトからHTMLバージョンの前記構成要素を自動的にロードし、存在しない場合、サーバからHTMLバージョンの前記構成要素を自動的にロード  
20 20

に使用されることを特徴とする、請求項 6 に記載の装置。

【請求項 8】

前記サーバが、ユーザによって提出されるネイティブバージョンの前記構成要素及びHTMLバージョンの前記構成要素を記憶することを特徴とする、請求項 7 に記載の装置。

【請求項 9】

ネイティブバージョンの前記構成要素が、HTML 拡張タグを介してパラメータ及び属性を定義することを特徴とする、請求項 6 から 8 いずれか 1 項に記載の装置。

【請求項 10】

前記ロードモジュールが特に、

前記ローカルサイト又はサーバからスクリプトを介して、ネイティブバージョンの前記構成要素を自動的にロードすること  
30 30

に使用されることを特徴とする、請求項 9 に記載の装置。

【請求項 11】

請求項 6 から 10 のいずれか一項に記載のウェブアプリケーションを実施する装置が配置されるクライアントと、前記サーバと、を備えることを特徴とする、ウェブアプリケーションを実施するシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、モバイルインターネットの技術分野に関し、特に、ウェブアプリケーションを実施する方法、装置、サーバ、及びシステムに関する。  
40 40

【背景技術】

【0002】

モバイル端末がますますインテリジェントになるにつれて、様々なモバイルアプリケーションがますます豊富になってきている。現在、アプリケーション（アプリ）は開発手法に従って分類され、その開発手法は主に：

1. ネイティブアプリケーション（ネイティブアプリ）、
  2. ウェブアプリケーション（ウェブアプリ）、
  3. ハイブリッドアプリケーション（ハイブリッドアプリ）、
- 50 50

4. 軽量アプリケーション（軽量アプリ）、を含む。

【0003】

ネイティブアプリケーション（ネイティブアプリ）は主に、ネイティブプログラムで書かれ、ユーザによりモバイル端末にダウンロードされたあとにインストールされ動作し、異なる経験を異なるモバイルオペレーティングシステムに提供することができる。ウェブアプリケーション（ウェブアプリ）は、特定の機能を実施するウェブ技術に基づいて開発されるアプリケーションであり、アプリには、ユーザがダウンロードする必要なく、ウェブページを介してアクセスすることができる。ハイブリッドアプリケーション（ハイブリッドアプリ）は、シェルプログラムがページで実施することができない機能を完了することができるように、ウェブアプリケーションに基づいてページをそのシェルプログラムに埋め込むため、ユーザがシェルプログラムインストールパッケージをダウンロードする必要がある。軽量アプリケーション（軽量アプリ）とは、ダウンロードする必要なく検索時に使用されるアプリであり、ウェブアプリケーションのようにダウンロードの必要がなく、同時にネイティブアプリケーションの経験と同じ経験も与えることができ、したがって、軽量構成要素を使用してインターフェース開発を実施することができる。

10

【0004】

しかし、ネイティブアプリケーションは高い開発費用、長い保守時間期間、及び劣ったクロスプラットフォーム互換性を有する。ウェブアプリケーションは、動作中のモバイル端末のハードウェア性能によって制限され、例えば、アニメーション効果が十分にスムーズに再生されない等である。時には、アプリケーションの実際の機能を実施することができないことさえもあり、開発チームは、ウェブページ内の要素を利用し、機能を実施するために、大量のスクリプトプログラムコードを維持する必要があり、したがって、開発チームの作業負荷が増大する。ハイブリッドアプリケーションは、開発中にウェブページデザインの利便性を利用して、開発費用を低減するが、それでもなお、動作のためには、ユーザがモバイル端末にダウンロードしてインストールする必要があり、ユーザの視点から、ネイティブアプリケーションと比較して経験は改善されない。軽量アプリケーションは、ネイティブ機能を拡張するが、アクセスに使用するブラウザによっては、軽量構成要素開発インターフェースにアクセスする場合に劣った互換性を有し、時には、インターフェースがないこともあり、更には基本機能が影響を受けることもある。

20

30

【発明の概要】

【発明が解決しようとする課題】

【0005】

本発明は、関連分野での技術的問題のうちの1つを少なくともある程度解決することを目的とする。このために、本発明の目的は、ウェブアプリケーションを実施する方法を提案することである。この方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善することができ、HTMLタグ開発構成要素の採用により、ウェブアプリケーションの開発費用を低減するとともに、ウェブアプリケーションのロード速度を高速化することができる。

【0006】

本発明の第2の目的は、ウェブアプリケーションを実施する装置を提案することである。

40

【0007】

本発明の第3の目的は、ウェブアプリケーションを実施する方法を提案することである。

【0008】

本発明の第4の目的はサーバを提案することである。

【0009】

本発明の第5の目的は、ウェブアプリケーションを実施するシステムを提案することである。

50

## 【課題を解決するための手段】

## 【0010】

上記目的を達成するために、本発明の第1の態様の実施形態のウェブアプリケーションを実施する方法は、現在のウェブアプリケーションの動作環境を検出するステップと、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードするステップと、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素を自動的にロードするステップとを含む。

## 【0011】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

## 【0012】

上記目的を達成するために、本発明の第2の態様の実施形態のウェブアプリケーションを実施する装置は、現在のウェブアプリケーションの動作環境を検出する検出モジュールと、検出モジュールにより、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、検出モジュールにより、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素を自動的にロードするロードモジュールとを備える。

## 【0013】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施する装置は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

## 【0014】

上記目的を達成するために、本発明の第3の態様の実施形態のウェブアプリケーションを実施する方法は、クライアントによって送信されるロード要求を受信し、ロード要求に従って対応するバージョンの構成要素を取得するステップと、対応するバージョンの構成要素をクライアントに送信するステップであって、それにより、クライアントが対応するバージョンの構成要素をロードできるようにする、送信するステップとを含む。

## 【0015】

対応するバージョンの構成要素をクライアントに送信し、それにより、クライアントが対応するバージョンの構成要素をロードできるようにすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

## 【0016】

上記目的を達成するために、本発明の第4の態様の実施形態のサーバは、クライアント

10

20

30

40

50

によって送信されるロード要求を受信し、ロード要求に従って対応するバージョンの構成要素を取得する取得モジュールと、対応するバージョンの構成要素をクライアントに送信して、クライアントが対応するバージョンの構成要素をロードできるようにする送信モジュールとを備える。

【0017】

対応するバージョンの構成要素をクライアントに送信し、それにより、クライアントが対応するバージョンの構成要素をロードできるようにすることにより、本発明の実施形態のサーバは、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

10

【0018】

上記目的を達成するために、本発明の第5の態様の実施形態のウェブアプリケーションを実施するシステムは、ウェブアプリケーションを実施する装置が配置されるクライアントと、サーバとを備える。

【0019】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施するシステムは、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

20

【図面の簡単な説明】

【0020】

【図1a】本発明の一実施形態によるウェブアプリケーションを実施する方法のフローチャート

【図1b】本発明の一実施形態により軽量アプリケーションプラットフォームで動作中のアプリの概略図

30

【図2】ウェブアプリケーションがブラウザで動作中であることを示す効果図である。

【図3】ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることを示す効果図

【図4】本発明の別の実施形態による、ウェブアプリケーションを実施する方法のフローチャート

【図5】サーバが、ユーザによって提出される、ネイティブバージョンの構成要素とHTMLバージョンの構成要素とを受信することを示す効果概略図

【図6】軽量構成要素が、ローカルサイトからネイティブバージョンの構成要素を自動的にロードして、音声入力機能を実施することを示す概略図

【図7】本発明の一実施形態による、ウェブアプリケーションを実施する装置の構造概略図

40

【図8】本発明の更に別の実施形態による、ウェブアプリケーションを実施する方法のフローチャート

【図9】本発明の更に別の実施形態による、ウェブアプリケーションを実施する方法のフローチャート

【図10】本発明の一実施形態によるサーバの構造概略図

【図11】本発明の別の実施形態によるサーバの構造概略図

【図12】本発明の一実施形態による、ウェブアプリケーションを実施するシステムの構造概略図

【発明を実施するための形態】

50

## 【 0 0 2 1 】

本発明の実施形態について以下に詳細に説明し、実施形態の例を添付図面に示し、図面中、全体を通して、同じ又は同様の参照符号は同じ若しくは同様の要素、又は同じ若しくは同様の機能を有する要素を表す。添付図面を参照して以下に説明される実施形態は、例示であり、本発明への限定として理解されるのではなく、むしろ、単に本発明の説明に使用されることが意図される。

## 【 0 0 2 2 】

本発明の実施形態においてウェブアプリケーションを実施する方法、装置、サーバ、及びシステムについて、添付図面を参照して以下に説明する。

## 【 0 0 2 3 】

図 1 a は、本発明の一実施形態によるウェブアプリケーションを実施する方法のフローチャートである。

## 【 0 0 2 4 】

図 1 a に示されるように、ウェブアプリケーションを実施する方法は以下を含む：

S 1 0 1 : 現在のウェブアプリケーションの動作環境が検出される。

## 【 0 0 2 5 】

本発明の実施形態では、ウェブアプリケーション（ウェブアプリ）は、軽量アプリケーションプラットフォームを介して動作することができ、また、ブラウザを介して動作することもできる。軽量アプリケーションプラットフォームはブラウザではないが、ウェブページをブラウズできるアプリケーションであり、図 1 b が、軽量アプリケーションプラットフォームで動作中のアプリ（アプリケーション）を示すことに留意されたい。

## 【 0 0 2 6 】

現在のウェブアプリに配置される軽量構成要素により、現在のウェブアプリが軽量アプリケーションプラットフォームを介して動作していることが検出される場合、ネイティブバージョンの構成要素をロードする必要があり、現在のウェブアプリがブラウザを介して動作していることが検出される場合、HTML（ハイパーテキストマークアップ言語）バージョン構成要素をロードする必要がある。したがって、軽量構成要素は、ロードすべき構成要素のバージョンを決定するために、現在のウェブアプリケーションの動作環境を検出する必要がある。

## 【 0 0 2 7 】

S 1 0 2 : 現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素が自動的にロードされ、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素が自動的にロードされる。

## 【 0 0 2 8 】

本発明の実施形態では、軽量構成要素により、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を、スクリプトを介して自動的にロードし得る。特に、ネイティブバージョンの対応する構成要素がローカルに存在する場合、ネイティブバージョンの構成要素をローカルサイトから自動的にロードすることができ、例えば、ネイティブバージョンの特定の構成要素が前のウェブアプリに出現していた場合、ネイティブバージョンの構成要素をローカルサイトから直接ロードすることができ、ネイティブバージョンの構成要素がローカルに存在しない場合、ネイティブバージョンの構成要素をサーバから自動的にロードすることができる。同じ理屈で、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、軽量構成要素は、スクリプトを介してハイパーテキストマークアップ言語（HTML）バージョンの構成要素を自動的にロードし得る。特に、HTMLバージョンの構成要素がローカルに存在する場合、HTMLバージョンの構成要素がローカルサイトから自動的にロードされ、HTMLバージョンの構成要素がローカルに存在しない場合、HTMLバージョンの構成要素がサーバから自動的にロードされる。

10

20

30

40

50

## 【 0 0 2 9 】

図 2 は、ウェブアプリケーションがブラウザで動作中である場合を示す効果図である。図 2 に示されるように、ブラウザで動作中のウェブアプリケーションは単に、基本的なアプリケーション機能を完了し、表示される効果は比較的単純である。しかし、図 3 は、ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中である場合を示す効果図である。図 3 に示されるように、軽量アプリケーションプラットフォームで動作中のウェブアプリケーションによりピクチャを表示する効果は、ネイティブアプリケーションの効果達成し、したがって、ユーザエクスペリエンスが改善する。

## 【 0 0 3 0 】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

10

## 【 0 0 3 1 】

図 4 は、本発明の別の実施形態による、ウェブアプリケーションを実施する方法のフローチャートである。

20

## 【 0 0 3 2 】

図 4 に示されるように、ウェブアプリケーションを実施する方法は以下を含む：

S 4 0 1 : サーバは、ユーザによって提出されるネイティブバージョンの構成要素及び HTMLバージョンの構成要素を、受信して保存する。

## 【 0 0 3 3 】

本発明の実施形態では、図 5 に示されるように、ユーザは、ネイティブバージョンの構成要素と、HTMLバージョンの構成要素とを開発し、テスト後、軽量アプリケーションプラットフォームを介してサーバに提出することができ、サーバは、ユーザによって提出されるネイティブバージョンの構成要素及びHTMLバージョンの構成要素を受信して保存し得る。すなわち、ユーザは、要件及び嗜好に従って様々な構成要素を開発し、共有のためにサーバにアップロードし得る。そのような構成要素共有様式では、これらの構成要素は、ユーザ自身によって使用されるのみならず、他のユーザによって使用されることもできる。

30

## 【 0 0 3 4 】

更に、この手段により、ウェブアプリに関する開発労働力分担をより明確且つ明白にし得る。ユーザは、ウェブアプリを開発する際、ウェブアプリのサービス論理にフォーカスするだけでよく、ネイティブバージョンの構成要素及びHTMLバージョンの構成要素を開発する際、インタラクション経験及び性能最適化にフォーカスするだけでよい。

## 【 0 0 3 5 】

S 4 0 2 : 現在のウェブアプリケーションの動作環境が検出される。

40

## 【 0 0 3 6 】

本発明の実施形態では、ウェブアプリは、軽量アプリケーションプラットフォームを介して動作することができ、またブラウザを介して動作することもできる。現在のウェブアプリに配置される軽量構成要素により、現在のウェブアプリが軽量アプリケーションプラットフォームを介して動作することが検出される場合、ネイティブバージョンの構成要素をロードする必要がある。現在のウェブアプリがブラウザを介して動作することが検出される場合、HTMLバージョンの構成要素をロードする必要がある。したがって、軽量構成要素は、ロードすべき構成要素のバージョンを決定するために、現在のウェブアプリケーションの動作環境を検出する必要がある。

## 【 0 0 3 7 】

50

S 4 0 3 : 現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素が自動的にロードされ、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素が自動的にロードされる。

【 0 0 3 8 】

本発明の実施形態では、軽量構成要素により、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を、スクリプトを介して自動的にロードし得る。特に、ネイティブバージョンの対応する構成要素がローカルに存在する場合、ネイティブバージョンの構成要素をローカルサイトから自動的にロードすることができ、例えば、ネイティブバージョンの特定の構成要素が前のウェブアプリに出現していた場合、ネイティブバージョンの構成要素をローカルサイトから直接ロードすることができ、ネイティブバージョンの構成要素がローカルに存在しない場合、ネイティブバージョンの構成要素をサーバから自動的にロードすることができる。同じ理屈で、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、軽量構成要素は、スクリプトを介してHTMLバージョンの構成要素を自動的にロードし得る。特に、HTMLバージョンの構成要素がローカルに存在する場合、HTMLバージョンの構成要素がローカルサイトから自動的にロードされ、HTMLバージョンの構成要素がローカルに存在しない場合、HTMLバージョンの構成要素がサーバから自動的にロードされる。

10

【 0 0 3 9 】

例えば、図 6 に示されるように、ユーザはクリックして、掲示板 ( p o s t b a r ) アプリケーションに入り、掲示板アプリケーションのインターフェースに入った後、軽量構成要素は、ネイティブバージョンの構成要素をローカルサイトから自動的にロードし、したがって、図 6 の音声入力機能を実施することができる。

20

【 0 0 4 0 】

ネイティブバージョンの構成要素は、HTML 拡張タグを介してパラメータ及び属性を定義し得る。HTML 拡張タグを介するパラメータ及び属性を定義する手法は、開発を簡易化し、異なるプラットフォーム間の違いを低減し、開発効率を大幅に改善し得る。

【 0 0 4 1 】

例えば、HTML 拡張タグは以下の形態であり得る。

30

【 0 0 4 2 】

【 数 1 】

```
<html>
  <component type="album">
    <item>1.jpg</item>
    <item>2.jpg</item>
    <item>3.jpg</item>
  </component>
  <script src="WebappEx.js"> </script>
</html>
```

40

【 0 0 4 3 】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成

50

要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。加えて、軽量アプリケーションプラットフォームを介してHTMLバージョンの構成要素を共有することにより、多くのウェブアプリケーションのスタイルを一貫した状態に保ち、それにより、保守の作業負荷を低減し、アップグレード中及び最適化中のユーザエクスペリエンスを改善することができる。

【0044】

図7は、本発明の一実施形態による、ウェブアプリケーションを実施する装置の構造概略図である。

10

【0045】

図7に示されるように、ウェブアプリケーションを実施する装置は、検出モジュール110と、ロードモジュール120とを備える。

【0046】

検出モジュール110は、現在のウェブアプリケーションの動作環境の検出に使用される。

【0047】

本発明の実施形態では、ウェブアプリは、軽量アプリケーションプラットフォームを介して動作することができ、また、ブラウザを介して動作することもできる。現在のウェブアプリに配置される軽量構成要素により、現在のウェブアプリが軽量アプリケーションプラットフォームを介して動作することが検出される場合、ネイティブバージョンの構成要素をロードする必要がある。現在のウェブアプリがブラウザを介して動作することが検出される場合、HTMLバージョンの構成要素をロードする必要がある。したがって、検出モジュール110は、ロードすべき構成要素のバージョンを決定するために、現在のウェブアプリケーションの動作環境を検出する必要がある。

20

【0048】

ロードモジュール120は、検出モジュール110により、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、検出モジュール110により、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素を自動的にロードするのに使用される。

30

【0049】

本発明の実施形態では、検出モジュール110により、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ロードモジュール120は、スクリプトを介してネイティブバージョンの構成要素を自動的にロードし得る。特に、ネイティブバージョンの対応する構成要素がローカルに存在する場合、ネイティブバージョンの構成要素はローカルサイトから自動的にロードすることができる。例えば、ネイティブバージョンの特定の構成要素が前のウェブアプリに出現していた場合、ネイティブバージョンの構成要素をローカルサイトから直接ロードすることができる。ネイティブバージョンの構成要素がローカルに存在しない場合、ネイティブバージョンの構成要素をサーバから自動的にロードすることができる。同じ理屈で、検出モジュール110により、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合、ロードモジュール120は、スクリプトを介してHTMLバージョンの構成要素を自動的にロードし得る。特に、HTMLバージョンの構成要素がローカルに存在する場合、HTMLバージョンの構成要素がローカルサイトから自動的にロードされ、HTMLバージョンの構成要素がローカルに存在しない場合、HTMLバージョンの構成要素がサーバから自動的にロードされる。

40

【0050】

50

例えば、図 6 に示されるように、ユーザはクリックして、掲示板アプリケーションに入り、掲示板アプリケーションのインターフェースに入った後、軽量構成要素は、ネイティブバージョンの構成要素をローカルサイトから自動的にロードし、したがって、図 6 の音声入力機能を実施することができる。

【 0 0 5 1 】

ネイティブバージョンの構成要素は、HTML 拡張タグを介してパラメータ及び属性を定義し得る。

【 0 0 5 2 】

例えば、HTML 拡張タグは以下の形態であり得る。

【 0 0 5 3 】

【 数 2 】

```

<html>
  <component type="album">
    <item>1.jpg</item>
    <item>2.jpg</item>
    <item>3.jpg</item>

  </component>

  <script src="WebappEx.js"> </script>
</html>

```

10

20

【 0 0 5 4 】

加えて、軽量構成要素がサーバからロードされる前に、サーバは、ユーザによって提出される軽量構成要素を受信して保存し得る。

【 0 0 5 5 】

本発明の実施形態では、ユーザは、ネイティブバージョンの構成要素と、HTMLバージョンの構成要素とを開発し、テスト後、軽量アプリケーションプラットフォームを介してサーバに提出することができ、サーバは、ユーザによって提出されるネイティブバージョンの構成要素及びHTMLバージョンの構成要素を受信して保存し得る。すなわち、ユーザは、要件及び嗜好に従って様々な構成要素を開発し、共有のためにサーバにアップロードし得る。そのような構成要素共有様式では、これらの構成要素は、ユーザ自身によって使用されるのみならず、他のユーザによって使用されることもできる。

30

【 0 0 5 6 】

更に、この手段により、ウェブアプリに関する開発労働力分担をより明確且つ明白にし得る。ユーザは、ウェブアプリを開発する際、ウェブアプリのサービス論理にフォーカスするだけでよく、ネイティブバージョンの構成要素及びHTMLバージョンの構成要素を開発する際、インタラクション経験及び性能最適化にフォーカスするだけでよい。

40

【 0 0 5 7 】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態においてウェブアプリケーションを実施する装置は、ウェブアプリケーションが様々な環境でその機能を完了し得るよう、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションの口

50

ード速度が高速化する。加えて、軽量アプリケーションプラットフォームを介してHTMLバージョンの構成要素を共有することにより、多くのウェブアプリケーションのスタイルを一貫した状態に保ち、それにより、保守の作業負荷を低減し、アップグレード中及び最適化中のユーザエクスペリエンスを改善する。

【0058】

図8は、本発明の更に別の実施形態による、ウェブアプリケーションを実施する方法のフローチャートである。

【0059】

図8に示されるように、ウェブアプリケーションを実施する方法は以下を含む：

S801：クライアントによって送信されるロード要求が受信され、対応するバージョンの構成要素が、ロード要求に従って取得される。

10

【0060】

本発明の実施形態では、現在のウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合であって、ネイティブバージョンの構成要素がクライアントに存在しないとき、サーバは、クライアントによって送信されるロード要求を受信し、ロード要求に従ってネイティブバージョンの構成要素を取得する。同じ理屈で、現在のウェブアプリケーションがブラウザで動作中であることが検出される場合であって、HTMLバージョンの構成要素がクライアントに存在しないとき、サーバは、クライアントによって送信されるロード要求を受信し、ロード要求に従ってHTMLバージョンの構成要素を取得する。

20

【0061】

S802：対応するバージョンの構成要素がクライアントに送信され、クライアントが対応するバージョンの構成要素をロードできるようにする。

【0062】

本発明の実施形態では、サーバは、対応するバージョンの構成要素を取得した後、対応するバージョンの構成要素をクライアントに送信して、クライアントが対応するバージョンの構成要素をロードできるようにし得る。

【0063】

対応するバージョンの構成要素をクライアントに送信し、クライアントが対応するバージョンの構成要素をロードできるようにすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。

30

【0064】

図9は、本発明の更に別の実施形態による、ウェブアプリケーションを実施する方法のフローチャートである。

【0065】

図9に示されるように、ウェブアプリケーションを実施する方法は以下を含む：

S901：ネイティブバージョンの構成要素及びHTMLバージョンの構成要素が受信され保存される。

40

【0066】

本発明の実施形態では、図5に示されるように、ユーザは、ネイティブバージョンの構成要素と、HTMLバージョンの構成要素とを開発し、テスト後、軽量アプリケーションプラットフォームを介してサーバに提出することができ、サーバは、ユーザによって提出されるネイティブバージョンの構成要素及びHTMLバージョンの構成要素を受信して保存し得る。すなわち、ユーザは、要件及び嗜好に従って様々な構成要素を開発し、共有のためにサーバにアップロードし得る。そのような構成要素共有様式では、これらの構成要素は、ユーザ自身によって使用されるのみならず、他のユーザによって使用されることもできる。

50

## 【0067】

更に、この手段により、ウェブアプリに関する開発労働力分担をより明確且つ明白にし得る。ユーザは、ウェブアプリを開発する際、ウェブアプリのサービス論理にフォーカスするだけでよく、ネイティブバージョンの構成要素及びHTMLバージョンの構成要素を開発する際、インタラクション経験及び性能最適化にフォーカスするだけでよい。

## 【0068】

S902：クライアントによって送信されるロード要求が受信され、対応するバージョンの構成要素が、ロード要求に従って取得される。

## 【0069】

本発明の実施形態では、図5に示されるように、ユーザは、ネイティブバージョンの構成要素と、HTMLバージョンの構成要素とを開発し、テスト後、軽量アプリケーションプラットフォームを介してサーバに提出することができ、サーバは、ユーザによって提出されるネイティブバージョンの構成要素及びHTMLバージョンの構成要素を受信して保存し得る。すなわち、ユーザは、要件及び嗜好に従って様々な構成要素を開発し、共有のためにサーバにアップロードし得る。そのような構成要素共有様式では、これらの構成要素は、ユーザ自身によって使用されるのみならず、他のユーザによって使用されることもできる。

10

## 【0070】

S903：対応するバージョンの構成要素がクライアントに送信され、クライアントが対応するバージョンの構成要素をロードできるようにする。

20

## 【0071】

本発明の実施形態では、サーバは、対応するバージョンの構成要素を取得した後、対応するバージョンの構成要素をクライアントに送信して、クライアントが対応するバージョンの構成要素をロードできるようにし得る。

## 【0072】

対応するバージョンの構成要素をクライアントに送信し、クライアントが対応するバージョンの構成要素をロードできるようにすることにより、本発明の実施形態においてウェブアプリケーションを実施する方法は、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。加えて、対応するバージョンの構成要素をサーバに記憶することにより、多くのウェブアプリケーションのスタイルを一貫した状態に保ち、それにより、保守の作業負荷を低減し、アップグレード中及び最適化中のユーザエクスペリエンスを改善する。

30

## 【0073】

図10は、本発明の一実施形態によるサーバの構造概略図である。

## 【0074】

図10に示されるように、サーバは、取得モジュール210と、送信モジュール220とを備える。

## 【0075】

特に、取得モジュール210は、クライアントによって送信されるロード要求を受信し、ロード要求に従って対応するバージョンの構成要素を取得するのに使用される。

40

## 【0076】

送信モジュール220は、対応するバージョンの構成要素をクライアントに送信し、クライアントが対応するバージョンの構成要素をロードできるようにするのに使用される。

## 【0077】

図11は、本発明の別の実施形態によるサーバの構造概略図である。

## 【0078】

図11に示されるように、サーバは、取得モジュール210と、送信モジュール220と、保存モジュール230とを備える。

50

## 【0079】

取得モジュール210及び送信モジュール220は全体的に、図10に示される実施形態において説明されるものと一貫し、冗長になるため、ここで繰り返さない。

## 【0080】

保存モジュール230は、取得モジュールが、クライアントによって送信されるロード要求を受信する前に、ネイティブバージョンの構成要素と、ハイパーテキストマークアップ言語（HTML）バージョンの構成要素とを受信して保存するのに使用される。

## 【0081】

対応するバージョンの構成要素をクライアントに送信して、クライアントが対応するバージョンの構成要素をロードできるようにすることにより、本発明の実施形態のサーバは、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。更に、対応するバージョンの構成要素をサーバに記憶することにより、多くのウェブアプリケーションのスタイルを一貫した状態に保ち、それにより、保守の作業負荷を低減し、アップグレード中及び最適化中のユーザエクスペリエンスを改善する。

10

## 【0082】

図12は、本発明の一実施形態による、ウェブアプリケーションを実施するシステムの構造概略図である。

## 【0083】

図12に示されるように、ウェブアプリケーションを実施するシステムは、クライアント10と、サーバ20とを備える。

20

## 【0084】

クライアント10は、図7に記載される実施形態におけるウェブアプリケーションを実施する装置であり、冗長になるため、ここで繰り返さない。

## 【0085】

サーバ20は、図11に記載される実施形態におけるサーバであり、冗長になるため、ここで繰り返さない。

## 【0086】

ウェブアプリケーションが軽量アプリケーションプラットフォームで動作中であることが検出される場合、ネイティブバージョンの構成要素を自動的にロードし、ウェブアプリケーションがブラウザで動作中であることが検出される場合、HTMLバージョンの構成要素を自動的にロードすることにより、本発明の実施形態のサーバは、ウェブアプリケーションが様々な環境でその機能を完了し得るように、ウェブアプリケーションの互換性を改善し、HTMLタグ開発構成要素の利用により、ウェブアプリケーションの開発費用が低減するとともに、ウェブアプリケーションのロード速度が高速化する。加えて、対応するバージョンの構成要素をサーバに記憶することにより、多くのウェブアプリケーションのスタイルを一貫した状態に保ち、それにより、保守の作業負荷を低減し、アップグレード中及び最適化中のユーザエクスペリエンスを改善する。

30

## 【0087】

更に、「第1」及び「第2」という用語は、相対的な重要性を示す若しくは暗示するか、又は示される技術的特徴の数を暗黙的に指定すると理解されるものではなく、単に説明を目的とする。したがって、「第1」及び「第2」によって定義される特徴は、明示的又は暗黙的に1つ又は複数のその特徴を含み得る。本発明の説明では、明確且つ特別に別段のことが指定される場合を除き、「複数の」の意味は2つ又は3つ以上である。

40

## 【0088】

本明細書の説明では、「一実施形態」、「幾つかの実施形態」、「例」、「特定の例」、又は「幾つかの例」等の参照用語の記述は、その実施形態又は例と関連して説明される特定の特徵、構造、材料、又は特徴が、本発明の少なくとも1つの実施形態又は例に包含されることを意味することを意図される。本明細書では、上記用語の例示的表現は必ずし

50

も、同じ実施形態又は例を指すわけではない。更に、説明される特定の特徵、構造、材料、又は特徴は、任意の1つ又は複数の実施形態又は例で適宜組み合わせ得る。更に、相互に矛盾しない場合、当業者は、本説明において説明される異なる実施形態又は例及び異なる実施形態又は例の特徵を組み込み、組み合わせ得る。

【0089】

本発明の実施形態を上にし説明したが、上記実施形態が例示であり、本発明への限定として理解されるべきではないことを理解されたい。当業者は、本発明の範囲内で上記実施形態に変更、改変、置換、及び変形を行い得る。

【符号の説明】

【0090】

- 10 クライアント
- 20 サーバ
- 110 検出モジュール
- 120 ロードモジュール
- 210 取得モジュール
- 220 送信モジュール
- 230 保存モジュール

【図1a】

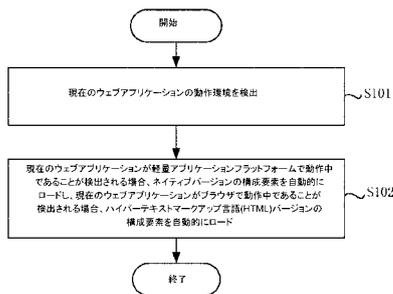


図1a

【図2】

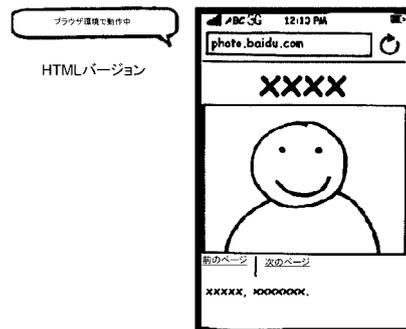


図2

【図1b】



図1b

【図3】

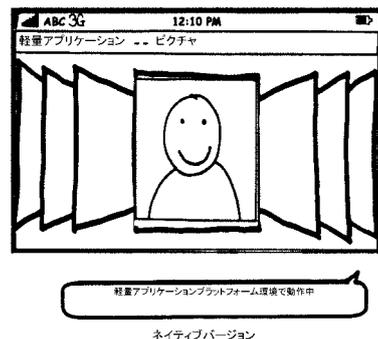


図3

【 図 4 】

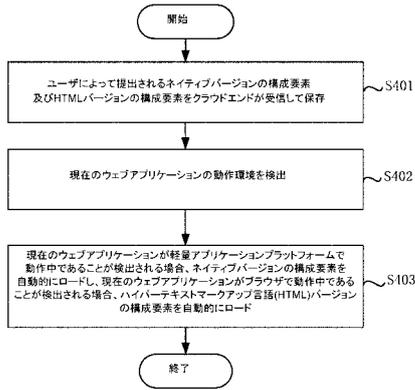


図4

【 図 6 】

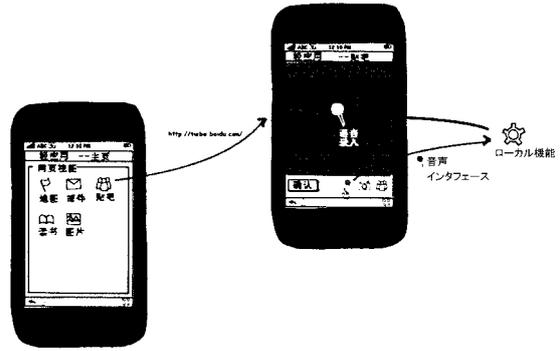


図6

【 図 5 】

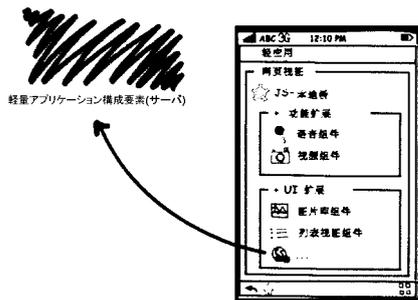


図5

【 図 7 】

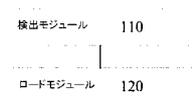


図7

【 図 8 】

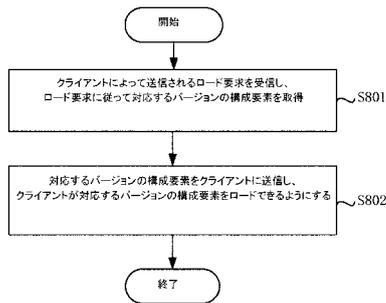


図8

【 図 10 】

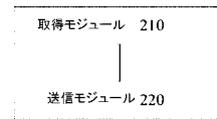


図10

【 図 9 】

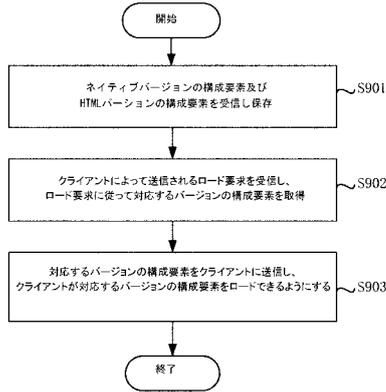


図9

【 図 11 】

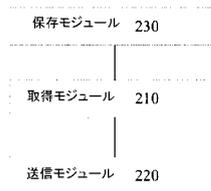


図11

【図 12】

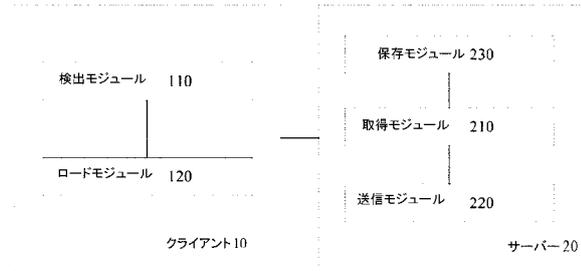


図12

---

フロントページの続き

(72)発明者 チェン ミンリーアン

中華人民共和国 ペキン ハイディエン ディストリクト シャンディー テンス ストリート  
ナンバー 10 バイドゥ キャンパス 3エフ

(72)発明者 リー ポンチャン

中華人民共和国 ペキン ハイディエン ディストリクト シャンディー テンス ストリート  
ナンバー 10 バイドゥ キャンパス 3エフ

(72)発明者 チャン ジアチェン

中華人民共和国 ペキン ハイディエン ディストリクト シャンディー テンス ストリート  
ナンバー 10 バイドゥ キャンパス 3エフ

Fターム(参考) 5B084 AA02 AA12 AB04 AB16 AB31 BA09 BB17 CE06 CE12 DB05

DC02

5B376 AC13 FA13